Apex Reference Guide

Version 63.0, Spring '25



Last updated: May 2, 2025

© Copyright 2000–2025 Salesforce, Inc. All rights reserved. Salesforce is a registered trademark of Salesforce, Inc., as are other names and marks. Other marks appearing herein may be trademarks of their respective owners.

CONTENTS

Apex Reference Guide 1
Release Notes 4
Apex DML Operations 5
Apex DML Statements 5
ApexPages Namespace 9
Action Class 9
Component Class 11
IdeaStandardController Class 13
IdeaStandardSetController Class 15
KnowledgeArticleVersionStandardController Class 19
Message Class 22
StandardController Class 26
StandardSetController Class 31
AppLauncher Namespace 42
AppMenu Class 43
ChangePasswordController Class 44
CommunityLogoController Class 45
EmployeeLoginLinkController Class 45
ForgotPasswordController Class 45
IdentityHeaderController Class 45
LoginFormController Class 45
SelfRegisterController Class 45
SocialLoginController Class 46
Approval Namespace 46
LockResult Class 46
ProcessRequest Class 48
ProcessResult Class 50
ProcessSubmitRequest Class 52
ProcessWorkitemRequest Class 56
UnlockResult Class 58
Auth Namespace 61
AuthConfiguration Class 64
AuthProviderCallbackState Class 75
AuthProviderPlugin Interface 77
AuthProviderPluginClass Class 80
AuthProviderTokenResponse Class 90
AuthToken Class 92
CommunitiesUtil Class 98
ConfigurableSelfRegHandler Interface 99

ConfirmUserRegistrationHandler Interface . . . 104
ConnectedAppPlugin Class . . . 107
CustomOneTimePasswordDeliveryHandler Interface . . . 114
CustomOneTimePasswordDeliveryResult Enum . . . 116
ExternalClientAppOauthHandler Class . . . 116
HeadlessSelfRegistrationHandler Interface . . . 119
HeadlessUserDiscoveryHandler Interface . . . 122
HeadlessUserDiscoveryResponse Class . . . 127
HttpCalloutMockUtil Class . . . 128
IntegratingAppType Enum . . . 129
InvocationContext Enum . . . 130
JWS Class . . . 131
JWT Class . . . 133
JWTBearerTokenExchange Class . . . 139
JWTUtil Class . . . 144
LightningLoginEligibility Enum . . . 147
LoginDiscoveryHandler Interface . . . 148
LoginDiscoveryMethod Enum . . . 155
MyDomainLoginDiscoveryHandler Interface . . . 156
Oauth2TokenExchangeHandler Class . . . 160
OAuth2TokenExchangeType Enum . . . 162
OAuthRefreshResult Class . . . 162
OauthToken Class . . . 165
OauthTokenType Enum . . . 166
RegistrationHandler Interface . . . 167
SamlJitHandler Interface . . . 171
SessionManagement Class . . . 176
SessionLevel Enum . . . 188
TokenValidationResult Class . . . 188
UserData Class . . . 195
VerificationAction Enum . . . 200
VerificationMethod Enum . . . 200
VerificationPolicy Enum . . . 201
VerificationResult Class . . . 201
Auth Exceptions . . . 204
Cache Namespace . . . 206
CacheBuilder Interface . . . 207
Org Class . . . 208
OrgPartition Class . . . 226
Partition Class . . . 229
Session Class . . . 244
SessionPartition Class . . . 261
Cache Exceptions . . . 264
Visibility Enum . . . 265

Canvas Namespace 265
ApplicationContext Interface 266
CanvasLifecycleHandler Interface 269
ContextTypeEnum Enum 272
EnvironmentContext Interface 272
RenderContext Interface 278
Test Class 280
Canvas Exceptions 284
ChatterAnswers Namespace 285
AccountCreator Interface 285
CommerceExtension Namespace 287
ExtensionInfo Class 288
Resolution Class 290
ResolutionException Class 292
ResolutionStates Enum 295
ResolutionStrategy Interface 295
CommerceOrders Namespace 297
CommercePayments Namespace 297
AbstractResponse Class 300
AbstractTransactionResponse Class 304
AddressRequest Class 308
AlternativePaymentMethodRequest Class 312
AlternativePaymentMethodResponse Class 316
AuditParamsRequest 319
AuthApiPaymentMethodRequest Class 321
AuthorizationRequest Class 323
AuthorizationResponse Class 328
AuthorizationReversalRequest Class 334
AuthorizationReversalResponse Class 338
BaseApiPaymentMethodRequest Class 343
BaseNotification Class 347
BasePaymentMethodRequest Class 351
BaseRequest Class 353
CaptureNotification Class 354
CaptureRequest Class 359
CaptureResponse Class 361
CardCategory Enum 365
CardPaymentMethodRequest Class 366
CardPaymentMethodResponse Class 372
CardType Enum 380
CustomMetadataTypeInfo Class 381
GatewayErrorResponse Class 382
GatewayNotificationResponse Class 383
GatewayResponse Interface 386

NotificationClient Class . . . 387
NotificationSaveResult Class . . . 388
NotificationStatus Enum . . . 389
PaymentGatewayAdapter Interface . . . 390
PaymentGatewayAsyncAdapter Interface . . . 391
PaymentGatewayContext Class . . . 394
PaymentGatewayNotificationContext Class . . . 396
PaymentGatewayNotificationRequest Class . . . 397
PaymentMethodDetailsResponse Class . . . 398
PaymentMethodTokenizationRequest Class . . . 400
PaymentMethodTokenizationResponse Class . . . 404
PaymentsHttp Class . . . 411
PostAuthApiPaymentMethodRequest Class . . . 412
PostAuthorizationRequest Class . . . 414
PostAuthorizationResponse Class . . . 417
ReferencedRefundNotification Class . . . 421
ReferencedRefundRequest . . . 425
ReferencedRefundResponse Class . . . 427
RefundRequest Class . . . 432
RequestType Enum . . . 433
SaleApiPaymentMethodRequest Class . . . 433
SaleRequest Class . . . 436
SaleResponse Class . . . 440
SalesforceResultCode Enum . . . 445
SalesforceResultCodeInfo . . . 446
CommerceTax Namespace . . . 447
AbstractTransactionResponse Class . . . 449
AddressesResponse Class . . . 455
AddressResponse Class . . . 457
AmountDetailsResponse Class . . . 458
CalculateTaxRequest Class . . . 461
CalculateTaxResponse Class . . . 464
CalculateTaxType Enum . . . 471
ErrorResponse Class . . . 472
HeaderTaxAddressesRequest Class . . . 474
ImpositionResponse Class . . . 478
JurisdictionResponse Class . . . 480
LineItemResponse Class . . . 484
LineTaxAddressesRequest Class . . . 489
RequestType Enum . . . 493
ResultCode Enum . . . 493
RuleDetailsResponse Class . . . 494
TaxAddressesRequest Class . . . 496
TaxAddressRequest Class . . . 500

TaxApiException Class . . . 505
TaxCustomerDetailsRequest Class . . . 507
TaxDetailsResponse Class . . . 511
TaxEngineAdapter Interface . . . 517
TaxEngineContext Class . . . 538
TaxLineItemRequest Class . . . 541
TaxSellerDetailsRequest Class . . . 546
TaxTransactionRequest Class . . . 549
TaxTransactionStatus Enum . . . 555
TaxTransactionType Enum . . . 555
Compression Namespace . . . 556
Level Enum . . . 556
Method Enum . . . 557
ZipEntry Class . . . 557
ZipReader Class . . . 562
ZipWriter Class . . . 565
Compression Exceptions . . . 572
ConnectApi Namespace . . . 572
ActionLinks Class . . . 577
Announcements Class . . . 585
BotVersionActivation Class . . . 591
CdpCalculatedInsight Class . . . 593
CdpIdentityResolution Class . . . 598
CdpQuery Class . . . 602
CdpSegment Class . . . 619
Chatter Class . . . 626
ChatterFavorites Class . . . 631
ChatterFeeds Class . . . 653
ChatterGroups Class . . . 1056
ChatterMessages Class . . . 1101
ChatterUsers Class . . . 1125
Clm Class . . . 1161
CommerceBuyerExperience Class . . . 1162
CommerceCart Class . . . 1220
CommerceCatalog Class . . . 1267
CommerceCatalogManagement Class . . . 1284
CommercePromotions Class . . . 1287
CommerceSearch Class . . . 1289
CommerceSearchConnectFamily Class . . . 1293
CommerceSearchSettings Class . . . 1295
CommerceStorePricing Class . . . 1298
CommerceWishlist Class . . . 1301
Communities Class . . . 1319
CommunityModeration Class . . . 1321

ContentHub Class 1350
ConversationApplicationDefinition Class 1425
Datacloud Class 1426
EinsteinLLM Class 1431
EmailMergeFieldService Class 1435
EmployeeProfiles Class 1435
Exchanges Class 1444
ExtendedCommerceDelivery Class 1446
ExternalEmailServices Class 1447
ExternalManagedAccount Class 1448
FieldService Class 1451
FulfillmentOrder Class 1453
Knowledge Class 1458
LightningScheduler Class 1463
ManagedContent Class 1469
ManagedContentChannels Class 1498
ManagedContentDelivery Class 1502
ManagedContentSpaces Class 1526
ManagedTopics Class 1528
MarketingIntegration Class 1543
Mentions Class 1545
Missions Class 1551
NamedCredentials Class 1556
NavigationMenu Class 1571
NextBestAction Class 1575
OmnichannelInventoryService Class 1582
OMSAnalytics Class 1590
Orchestration Class 1595
OrderPaymentSummary Class 1596
OrderSummary Class 1598
OrderSummaryCreation Class 1610
Organization Class 1611
PardotBusinessUnitContext Class 1612
Payments Class 1613
Personalization Class 1619
PickTicket Class 1631
QuestionAndAnswers Class 1632
Recommendations Class 1636
Records Class 1696
RegisterGuestBuyer Class 1698
Repricing Class 1699
ReturnOrder Class 1702
Routing Class 1704
SalesforceInbox Class 1709

Search Class 1710
Sites Class 1720
SmartDataDiscovery Class 1722
SocialEngagement Class 1722
Surveys Class 1733
TaxPlatform Class 1734
Topics Class 1735
UserProfiles Class 1772
Zones Class 1782
ConnectApi Input Classes 1792
ConnectApi Output Classes 1954
ConnectApi Enums 2346
ConnectApi Exceptions 2382
ConnectApi Utilities 2382
ConnectApi Release Notes 2383
Context Namespace 2383
Database Namespace 2384
Batchable Interface 2385
BatchableContext Interface 2387
Cursor Class (Beta) 2388
DeletedRecord Class 2390
DeleteFilter Enum 2391
DeleteResult Class 2391
DMLOptions Class 2393
DmlOptions.AssignmentRuleHeader Class 2396
DMLOptions.DuplicateRuleHeader Class 2398
DmlOptions.EmailHeader Class 2400
DuplicateError Class 2402
EmptyRecycleBinResult Class 2404
Error Class 2406
GetDeletedResult Class 2407
GetUpdatedResult Class 2409
LeadConvert Class 2410
LeadConvertResult Class 2421
MergeResult Class 2423
QueryLocator Class 2425
QueryLocatorIterator Class 2426
SaveResult Class 2428
UndeleteResult Class 2430
UpsertResult Class 2431
Datacloud Namespace 2433
AdditionalInformationMap Class 2434
DuplicateResult Class 2435
FieldDiff Class 2440

FindDuplicates Class 2441
FindDuplicatesByIds Class 2443
FindDuplicatesResult Class 2444
MatchRecord Class 2447
MatchResult Class 2449
DataRetrieval Namespace 2451
DataSource Namespace 2451
AsyncDeleteCallback Class 2454
AsyncSaveCallback Class 2455
AuthenticationCapability Enum 2455
AuthenticationProtocol Enum 2456
Capability Enum 2456
Column Class 2458
ColumnSelection Class 2481
Connection Class 2483
ConnectionParams Class 2488
DataSourceUtil Class 2492
DataType Enum 2493
DeleteContext Class 2494
DeleteResult Class 2495
Filter Class 2498
FilterType Enum 2500
IdentityType Enum 2501
Order Class 2501
OrderDirection Enum 2504
Provider Class 2504
QueryAggregation Enum 2506
QueryContext Class 2506
QueryUtils Class 2508
ReadContext Class 2511
SearchContext Class 2512
SearchUtils Class 2514
Table Class 2515
TableResult Class 2519
TableSelection Class 2525
UpsertContext Class 2526
UpsertResult Class 2527
DataSource Exceptions 2530
DataWeave Namespace 2531
Result Class 2531
Script Class 2532
Dom Namespace 2535
Document Class 2535
XmlNode Class 2538

XmlNodeType Enum . . . 2548
EventBus Namespace . . . 2549
ChangeEventHeader Class . . . 2550
EventPublishFailureCallback Interface . . . 2555
EventPublishSuccessCallback Interface . . . 2556
FailureResult Interface . . . 2557
SuccessResult Interface . . . 2558
TestBroker Class . . . 2559
TriggerContext Class . . . 2560
ExternalService Namespace . . . 2563
Flow Namespace . . . 2563
Interview Class . . . 2564
FormulaEval Namespace . . . 2568
FormulaBuilder Class . . . 2569
FormulaGlobal Enum . . . 2572
FormulaInstance Class . . . 2573
FormulaReturnType Enum . . . 2575
fsccashflow Namespace . . . 2576
FSCCashFlowUtil Class . . . 2576
Functions Namespace . . . 2586
Function Class . . . 2587
FunctionCallback Interface . . . 2592
FunctionErrorType Enum . . . 2593
FunctionInvocation Interface . . . 2593
FunctionInvocationError Interface . . . 2596
FunctionInvocationStatus Enum . . . 2597
FunctionInvokeMock Interface . . . 2598
MockFunctionInvocationFactory Class . . . 2601
industriesNlpSvc . . . 2602
NlpResponse Class . . . 2603
NlpSummarizationResult Class . . . 2604
IndustriesDigitalLending Namespace . . . 2604
Invocable Namespace . . . 2605
Action Class . . . 2605
Action.Error Class . . . 2610
Action.Result Class . . . 2612
IsvPartners Namespace . . . 2614
AppAnalytics Class . . . 2614
KbManagement Namespace . . . 2616
PublishingService Class . . . 2616
LxScheduler Namespace . . . 2628
GetAppointmentCandidatesInput Class . . . 2629
GetAppointmentCandidatesInputBuilder Class . . . 2631
GetAppointmentSlotsInput Class . . . 2638

GetAppointmentSlotsInputBuilder Class 2641
SchedulerResources Class 2648
SkillRequirement Class 2653
SkillRequirementBuilder Class 2653
WorkType Class 2654
WorkTypeBuilder Class 2655
ServiceResourceScheduleHandler Interface 2659
ServiceAppointmentRequestInfo Class 2662
ServiceResourceInfo Class 2666
ServiceResourceSchedule Class 2669
UnavailableTimeslot Class 2670
Messaging Namespace 2672
AttachmentRetrievalOption Enum 2673
Email Class (Base Email Methods) 2674
EmailFileAttachment Class 2677
InboundEmail Class 2679
InboundEmail.AuthenticationResult Class 2686
InboundEmail.AuthenticationResultField Class 2687
InboundEmail.BinaryAttachment Class 2689
InboundEmail.TextAttachment Class 2691
InboundEmailResult Class 2693
InboundEnvelope Class 2694
MassEmailMessage Class 2695
InboundEmail.Header Class 2698
PushNotification Class 2699
PushNotificationPayload Class 2702
CustomNotification Class 2704
RenderEmailTemplateBodyResult Class 2711
RenderEmailTemplateError Class 2713
SendEmailError Class 2714
SendEmailResult Class 2716
SingleEmailMessage Class 2717
Metadata Namespace 2733
AnalyticsCloudComponentLayoutItem Class 2736
ConsoleComponent Class 2740
Container Class 2742
CustomConsoleComponents Class 2745
CustomMetadata Class 2747
CustomMetadataValue Class 2749
DeployCallback Interface 2751
DeployCallbackContext Class 2753
DeployContainer Class 2754
DeployDetails Class 2757
DeployMessage Class 2758

DeployProblemType Enum 2763
DeployResult Class 2764
DeployStatus Enum 2772
FeedItemTypeEnum Enum 2772
FeedLayout Class 2774
FeedLayoutComponent Class 2777
FeedLayoutComponentType Enum 2779
FeedLayoutFilter Class 2780
FeedLayoutFilterPosition Enum 2782
FeedLayoutFilterType Enum 2782
Layout Class 2782
LayoutColumn Class 2790
LayoutHeader Enum 2792
LayoutItem Class 2792
LayoutSection Class 2797
LayoutSectionStyle Enum 2800
Metadata Class 2800
MetadataType Enum 2801
MetadataValue Class 2802
MiniLayout Class 2802
Operations Class 2804
PlatformActionList Class 2808
PlatformActionListContextEnum Enum 2810
PlatformActionListItem Class 2811
PlatformActionTypeEnum Enum 2813
PrimaryTabComponents Class 2813
QuickActionList Class 2815
QuickActionListItem Class 2816
RelatedContent Class 2817
RelatedContentItem Class 2818
RelatedList Class 2820
RelatedListItem Class 2821
ReportChartComponentLayoutItem Class 2824
ReportChartComponentSize Enum 2828
SidebarComponent Class 2828
SortOrder Enum 2832
StatusCode Enum 2833
SubtabComponents Class 2833
SummaryLayoutStyleEnum Enum 2835
SummaryLayout Class 2835
SummaryLayoutItem Class 2838
UiBehavior Enum 2840
PlaceQuote Namespace 2841
Pref_center Namespace 2841

LoadFormData Class 2841
LoadParameters Class 2848
PreferenceCenterApexHandler Interface 2849
SubmitFormData Class 2850
SubmitParameters Class 2854
TokenType Enum 2855
TokenUtility Class 2855
ValidationResult Class 2858
Process Namespace 2858
Plugin Interface 2859
PluginDescribeResult Class 2861
PluginDescribeResult.InputParameter Class 2863
PluginDescribeResult.OutputParameter Class 2867
PluginRequest Class 2869
PluginResult Class 2870
QuickAction Namespace 2871
DescribeAvailableQuickActionResult Class 2872
DescribeLayoutComponent Class 2874
DescribeLayoutItem Class 2875
DescribeLayoutRow Class 2877
DescribeLayoutSection Class 2878
DescribeQuickActionDefaultValue Class 2882
DescribeQuickActionParameter Class 2883
DescribeQuickActionResult Class 2885
QuickActionDefaults Class 2903
QuickActionDefaultsHandler Interface 2905
QuickActionRequest Class 2910
QuickActionResult Class 2913
SendEmailQuickActionDefaults Class 2915
Reports Namespace 2917
AggregateColumn Class 2921
BucketField Class 2922
BucketFieldValue Class 2929
BucketType Enum 2933
ColumnDataType Enum 2933
ColumnSortOrder Enum 2934
CrossFilter Class 2935
CsfGroupType Enum 2940
DateGranularity Enum 2940
DetailColumn Class 2941
Dimension Class 2942
EvaluatedCondition Class 2943
EvaluatedConditionOperator Enum 2946
FilterOperator Class 2947

FilterValue Class 2948
FormulaType Enum 2949
GroupingColumn Class 2949
GroupingInfo Class 2951
GroupingValue Class 2952
NotificationAction Interface 2954
NotificationActionContext Class 2956
ReportCsf Class 2957
ReportCurrency Class 2966
ReportDataCell Class 2967
ReportDescribeResult Class 2968
ReportDetailRow Class 2969
ReportDivisionInfo Class 2970
ReportExtendedMetadata Class 2971
ReportFact Class 2972
ReportFactWithDetails Class 2973
ReportFactWithSummaries Class 2974
ReportFilter Class 2976
ReportFormat Enum 2982
ReportFilterType Enum 2982
ReportInstance Class 2982
ReportManager Class 2985
ReportMetadata Class 2991
ReportResults Class 3010
ReportScopeInfo Class 3012
ReportScopeValue Class 3013
ReportType Class 3015
ReportTypeColumn Class 3016
ReportTypeColumnCategory Class 3017
ReportTypeMetadata Class 3019
SortColumn Class 3021
StandardDateFilter Class 3022
StandardDateFilterDuration Class 3026
StandardDateFilterDurationGroup Class 3027
StandardFilter Class 3028
StandardFilterInfo Class 3030
StandardFilterInfoPicklist Class 3031
StandardFilterType Enum 3033
SummaryValue Class 3033
ThresholdInformation Class 3034
TopRows Class 3035
Reports Exceptions 3038
RevSalesTrxn Namespace 3039
RichMessaging Namespace 3039

AbstractTiming Class . . . 3041
AddressableContact Class . . . 3041
AuthRequestHandler Interface . . . 3044
AuthRequestResponse Class . . . 3047
AuthRequestResult Class . . . 3049
AuthRequestResultStatus Enum . . . 3051
DeferredTiming Class . . . 3051
MessageDefinitionInputParameter Class . . . 3053
PaymentItemStatus Enum . . . 3057
PaymentLineItem Class . . . 3057
PaymentMethod Class . . . 3063
PostalAddress Class . . . 3064
ProcessPaymentHandler Interface . . . 3068
ProcessPaymentRequest Class . . . 3069
ProcessPaymentResult Class . . . 3073
ProcessPaymentResultStatus Enum . . . 3075
RecurringTiming Class . . . 3075
ShippingMethod Class . . . 3079
TimeSlotOption Class . . . 3083
TimingIntervalUnit Enum . . . 3086
TimingType Enum . . . 3087
Salesforce_Backup Namespace . . . 3087
SalesforceBackupIntegrationUtil Class . . . 3087
Schema Namespace . . . 3087
ChildRelationship Class . . . 3089
DataCategory Class . . . 3091
DataCategoryGroupSobjectTypePair Class . . . 3092
DescribeColorResult Class . . . 3094
DescribeDataCategoryGroupResult Class . . . 3096
DescribeDataCategoryGroupStructureResult Class . . . 3098
DescribeFieldResult Class . . . 3101
DescribeIconResult Class . . . 3118
DescribeSObjectResult Class . . . 3121
DescribeTabResult Class . . . 3143
DescribeTabSetResult Class . . . 3146
DisplayType Enum . . . 3149
FieldDescribeOptions Enum . . . 3150
FieldSet Class . . . 3151
FieldSetMember Class . . . 3155
PicklistEntry Class . . . 3157
RecordTypeInfo Class . . . 3159
SOAPType Enum . . . 3162
SObjectDescribeOptions Enum . . . 3162
SObjectField Class . . . 3163

SObjectType Class . . . 3164
Search Namespace . . . 3168
KnowledgeSuggestionFilter Class . . . 3168
QuestionSuggestionFilter Class . . . 3173
SearchResult Class . . . 3177
SearchResults Class . . . 3178
SuggestionOption Class . . . 3179
SuggestionResult Class . . . 3181
SuggestionResults Class . . . 3182
Sfc Namespace . . . 3183
ContentDownloadContext Enum . . . 3183
ContentDownloadHandler Class . . . 3184
ContentDownloadHandlerFactory Interface . . . 3185
Sfdc_Checkout Namespace . . . 3187
AsyncCartProcessor Interface . . . 3187
B2BCheckoutController Class . . . 3188
IntegrationInfo Class . . . 3189
IntegrationStatus Class . . . 3191
IntegrationStatus.Status Enum . . . 3191
Sfdc_Enablement Namespace . . . 3192
LearningEvaluation Class . . . 3192
LearningEvaluationResult Class . . . 3194
LearningItemEvaluationHandler Class . . . 3196
LearningItemProgressStatus Enum . . . 3198
LearningItemSerializeDeserializer Class . . . 3199
sfdc_surveys Namespace . . . 3202
SurveyInvitationLinkShortener Interface . . . 3202
Example Implementation to Associate SurveySubjects with SurveyInvitation and SurveyResponses . . . 3204
Site Namespace . . . 3205
UrlRewriter Interface . . . 3206
Site Exceptions . . . 3207
Slack Namespace . . . 3207
Support Namespace . . . 3209
EmailTemplateSelector Interface . . . 3209
MilestoneTriggerTimeCalculator Interface . . . 3211
System Namespace . . . 3213
AccessLevel Class . . . 3220
AccessType Enum . . . 3223
Address Class . . . 3224
Answers Class . . . 3229
ApexPages Class . . . 3231
Approval Class . . . 3234
Assert Class . . . 3245

AsyncInfo Class 3258
AsyncOptions Class 3260
Blob Class 3261
Boolean Class 3264
BusinessHours Class 3265
CallbackStatus Enum 3269
Callable Interface 3269
Cases Class 3272
Collator Class 3275
Comparable Interface 3277
Comparator Interface 3280
Continuation Class 3282
Cookie Class 3286
Crypto Class 3294
Custom Metadata Type Methods 3317
Custom Settings Methods 3320
Database Class 3331
Date Class 3426
Datetime Class 3436
Decimal Class 3460
Domain Class 3473
DomainCreator Class 3476
DomainParser Class 3480
DomainType Enum 3482
Double Class 3483
EmailMessages Class 3486
EncodingUtil Class 3490
Enum Methods 3494
EventBus Class 3494
Exception Class and Built-In Exceptions 3499
ExternalServiceTest Class 3504
FlexQueue Class 3505
FeatureManagement Class 3508
Formula Class 3514
FormulaRecalcFieldError Class 3515
FormulaRecalcResult Class 3516
Http Class 3518
HttpCalloutMock Interface 3519
HttpRequest Class 3519
HttpResponse Class 3529
Id Class 3536
Ideas Class 3542
InstallHandler Interface 3547
Integer Class 3550

JSON Class 3553
JSONGenerator Class 3559
JSONParser Class 3573
JSONToken Enum 3585
Label Class 3586
Limits Class 3589
List Class 3603
Location Class 3618
LoggingLevel Enum 3621
Long Class 3622
Map Class 3624
Matcher Class 3636
Math Class 3649
Messaging Class 3674
MultiStaticResourceCalloutMock Class 3682
Network Class 3685
Object Class 3691
OrgLimit Class 3693
OrgLimits Class 3695
PageReference Class 3696
Packaging Class 3709
Pattern Class 3710
Queueable Interface 3713
QueueableContext Interface 3716
QueueableDuplicateSignature Class 3716
QuickAction Class 3721
Quiddity Enum 3725
RemoteObjectController 3726
Request Class 3730
ResetPasswordResult Class 3731
RestContext Class 3732
RestRequest Class 3733
RestResponse Class 3739
SandboxPostCopy Interface 3743
Schedulable Interface 3745
SchedulableContext Interface 3746
Schema Class 3747
Search Class 3752
Security Class 3757
SelectOption Class 3761
Set Class 3767
Site Class 3778
SObject Class 3800
SObjectAccessDecision Class 3825

SoqlStubProvider Class . . . 3828
StaticResourceCalloutMock Class . . . 3831
String Class . . . 3833
StubProvider Interface . . . 3909
System Class . . . 3911
Test Class . . . 3937
Time Class . . . 3958
TimeZone Class . . . 3963
Trigger Class . . . 3966
TriggerOperation Enum . . . 3969
Type Class . . . 3969
UninstallHandler Interface . . . 3977
URL Class . . . 3980
UserInfo Class . . . 3990
UserManagement Class . . . 3999
UUID Class . . . 4018
Version Class . . . 4021
WebServiceCallout Class . . . 4024
WebServiceMock Interface . . . 4026
XmlStreamReader Class . . . 4028
XmlStreamWriter Class . . . 4041
TerritoryMgmt Namespace . . . 4048
OpportunityTerritory2AssignmentFilter Global Interface . . . 4049
TxnSecurity Namespace . . . 4052
Event Class . . . 4052
EventCondition Interface . . . 4057
AsyncCondition Interface . . . 4058
PolicyCondition Interface . . . 4060
UserProvisioning Namespace . . . 4061
ConnectorTestUtil Class . . . 4061
UserProvisioningLog Class . . . 4063
UserProvisioningPlugin Class . . . 4065
VisualEditor Namespace . . . 4070
DataRow Class . . . 4070
DesignTimePageContext Class . . . 4073
DynamicPickList Class . . . 4075
DynamicPickListRows Class . . . 4078
Wave Namespace . . . 4083
QueryBuilder Class . . . 4084
QueryNode Class . . . 4087
ProjectionNode Class . . . 4092
Templates Class . . . 4094
TemplatesSearchOptions Class . . . 4098
Appendices . . . 4100

Shipping Invoice Example 4100
Reserved Keywords 4111
Documentation Typographical Conventions 4113

APEX REFERENCE GUIDE

Apex is a strongly typed, object-oriented programming language that allows developers to execute flow and transaction control statements on the Salesforce Platform server, in conjunction with calls to the API. This reference guide includes built-in Apex classes, interfaces, enums, and exceptions, grouped by namespace. It also includes Apex DML statements to insert, update, merge, delete, and restore data in Salesforce.

For information on the Apex development process, see Apex Developer Guide.

Note: In API version 51.0 and earlier, Apex Reference information was included in the Apex Developer Guide in the **Apex Language Reference** section.

IN THIS SECTION:

Apex Release Notes

Use the Salesforce Release Notes to learn about the most recent updates and changes to Apex.

Apex DML Operations

You can perform DML operations using the Apex DML statements or the methods of the `Database` class. For lead conversion, use the `convertLead` method of the `Database` class. There is no DML counterpart for it.

ApexPages Namespace

The `ApexPages` namespace provides classes used in Visualforce controllers.

AppLauncher Namespace

The `AppLauncher` namespace provides methods for managing the appearance of apps in the App Launcher, including their visibility and sort order.

Approval Namespace

The `Approval` namespace provides classes and methods for approval processes.

Auth Namespace

The `Auth` namespace provides an interface and classes for single sign-on into Salesforce and session security management.

Cache Namespace

The `Cache` namespace contains methods for managing the platform cache.

Canvas Namespace

The `Canvas` namespace provides an interface and classes for canvas apps in Salesforce.

ChatterAnswers Namespace

The `ChatterAnswers` namespace provides an interface for creating Account records.

CommerceExtension Namespace

Use the `CommerceExtension` namespace to define resolution strategies for registered Commerce extensions.

CommerceOrders Namespace

The `CommerceOrders` namespace provides classes and methods to place orders with integrated pricing, configuration, and validation.

CommercePayments Namespace

Use the `CommercePayments` namespace to provide a safe and customizable platform for managing customer payments and refunds.

CommerceTax Namespace

Manage the communication between Salesforce and an external tax engine.

Compression Namespace

The Compression namespace provides classes and methods to create and extract zip files.

ConnectApi Namespace

The `ConnectApi` namespace (also called Connect in Apex) provides classes for accessing the same data available in Connect REST API. Use Connect in Apex to create custom experiences in Salesforce.

Context Namespace

The `Context` namespace provides classes and methods to manage the sharing and consumption of business application data by using Context Service.

Database Namespace

The `Database` namespace provides classes used with DML operations.

Datacloud Namespace

The `Datacloud` namespace provides classes and methods for retrieving information about duplicate rules. Duplicate rules let you control whether and when users can save duplicate records within Salesforce.

DataRetrieval Namespace

The `DataRetrieval` namespace provides classes and methods to record details of customer-agent engagements, as well as transcripts of their conversations.

DataSource Namespace

The `DataSource` namespace provides the classes for the Apex Connector Framework. Use the Apex Connector Framework to develop a custom adapter for Salesforce Connect. Then connect your Salesforce organization to any data anywhere via the Salesforce Connect custom adapter.

DataWeave Namespace

The DataWeave namespace provides classes and methods to support the invocation of DataWeave scripts from Apex.

Dom Namespace

The `Dom` namespace provides classes and methods for parsing and creating XML content.

EventBus Namespace

The `EventBus` namespace provides classes and methods for platform events and Change Data Capture events.

ExternalService Namespace

The `ExternalService` namespace provides dynamically generated Apex service interfaces and Apex classes for complex object data types.

Flow Namespace

The `Flow` namespace provides a class for advanced access to flows from Apex such as from Visualforce controllers and asynchronous Apex.

FormulaEval Namespace

The FormulaEval namespace provides classes and methods to evaluate dynamic formulas for SObjects and Apex objects. Use the methods to avoid unnecessary DML statements to recalculate formula field values or evaluate dynamic formula expressions.

fsccashflow Namespace

The `fsccashflow` namespace provides classes used in the FSCCashFlow Flexcards and its child Flexcards.

Functions Namespace

The Functions namespace provides classes and methods used to invoke and manage Salesforce Functions.

industriesNlpSvc

Stores the objects used in Industries Einstein Natural Language Processing (NLP) services.

IndustriesDigitalLending Namespace

The `industriesDigitalLending` namespace provides classes used in the Digital Lending OmniScripts and Integration Procedures.

Invocable Namespace

The `Invocable` namespace provides classes for calling invocable actions from Apex.

IsvPartners Namespace

The `IsvPartners` namespace provides a class associated with Salesforce ISV partner use cases, such as optimizing code, providing great customer trial experiences, and driving feature adoption.

KbManagement Namespace

The `KbManagement` namespace provides a class for managing knowledge articles.

LxScheduler Namespace

The `LxScheduler` namespace provides an interface and classes for integrating Salesforce Scheduler with external calendars.

Messaging Namespace

The `Messaging` namespace provides classes and methods for Salesforce outbound and inbound email functionality.

Metadata Namespace

The `Metadata` namespace provides classes and methods for working with custom metadata in Salesforce

PlaceQuote Namespace

The `PlaceQuote` namespace provides classes and methods to create or update quotes with pricing preferences and configuration options.

Pref_center Namespace

The Pref_center namespace provides an interface, classes, and methods to create and retrieve data in forms in Preference Manager. Preference Manager, previously called Preference Center, is a feature within the Privacy Center app.

Process Namespace

The `Process` namespace provides an interface and classes for passing data between your organization and a flow.

QuickAction Namespace

The `QuickAction` namespace provides classes and methods for quick actions.

Reports Namespace

The `Reports` namespace provides classes for accessing the same data as is available in the Salesforce Reports and Dashboards REST API.

RevSalesTrxn Namespace

The `RevSalesTrxn` namespace provides classes and methods to create a sales transaction, such as a quote or an order, with integrated pricing and configuration.

RichMessaging Namespace

Provides objects and methods for handling content in enhanced Messaging channels.

Salesforce_Backup Namespace

The `Salesforce_Backup` namespace provides classes and methods for Salesforce Backup.

Schema Namespace

The `Schema` namespace provides classes and methods for schema metadata information.

Search Namespace

The `Search` namespace provides classes for getting search results and suggestion results.

Sfc Namespace

The Sfc namespace contains classes used in Salesforce Files.

Sfdc_Checkout Namespace

The Sfdc_Checkout namespace provides an interface and classes for B2B Commerce apps in Salesforce.

Sfdc_Enablement Namespace

The `sfdc_enablement` namespace provides classes for creating custom learning items to implement custom exercise types in Enablement programs. Lightning web components are used to render the custom exercises on Program Builder.

sfdc_surveys Namespace

The `sfdc_surveys` namespace provides an interface for shortening survey invitations.

Site Namespace

The `Site` namespace provides an interface for rewriting Sites URLs.

Slack Namespace

The `Slack` Namespace provides tools designed to accelerate and ease the process of developing Slack apps on the Salesforce platform.

Support Namespace

The `Support` namespace provides an interface used for Case Feed.

System Namespace

The `System` namespace provides classes and methods for core Apex functionality.

TerritoryMgmt Namespace

The `TerritoryMgmt` namespace provides an interface used for territory management.

TxnSecurity Namespace

The `TxnSecurity` namespace provides an interface used for transaction security.

UserProvisioning Namespace

The `UserProvisioning` namespace provides methods for monitoring outbound user provisioning requests.

VisualEditor Namespace

The `VisualEditor` namespace provides classes and methods for interacting with the Lightning App Builder. The classes and methods in this namespace operate on Lightning components, which include Lightning web components and Aura components.

Wave Namespace

The classes in the `Wave` namespace are part of the CRM Analytics Analytics SDK, designed to facilitate querying CRM Analytics data from Apex code.

Appendices

Apex Release Notes

Use the Salesforce Release Notes to learn about the most recent updates and changes to Apex.

For Apex updates and changes that impact the Salesforce Platform, see the Apex Release Notes.

For new and changed Apex classes, methods, exceptions and interfaces, see Apex: New and Changed Items in the Salesforce Release Notes.

Apex DML Operations

You can perform DML operations using the Apex DML statements or the methods of the `Database` class. For lead conversion, use the `convertLead` method of the `Database` class. There is no DML counterpart for it.

SEE ALSO:

Apex Developer Guide: Working with Data in Apex

Database Class

Apex DML Statements

Use Data Manipulation Language (DML) statements to insert, update, merge, delete, and restore data in Salesforce.

The following Apex DML statements are available:

Insert Statement

The `insert` DML operation adds one or more sObjects, such as individual accounts or contacts, to your organization's data. `insert` is analogous to the INSERT statement in SQL.

Syntax

```
insert sObject
insert sObject[]
```

Example

The following example inserts an account named 'Acme':

```
Account newAcct = new Account(name = 'Acme');
try {
   insert newAcct;
} catch (DmlException e) {
// Process exception here
}
```

Note: For more information on processing `DmlExceptions`, see Bulk DML Exception Handling.

Update Statement

The `update` DML operation modifies one or more existing sObject records, such as individual accounts or contacts, in your organization's data. `update` is analogous to the UPDATE statement in SQL.

Syntax

```
update sObject
update sObject[]
```

Example

The following example updates the `BillingCity` field on a single account named 'Acme':

```
Account a = new Account(Name='Acme2');
insert(a);

Account myAcct = [SELECT Id, Name, BillingCity FROM Account WHERE Id = :a.Id];
myAcct.BillingCity = 'San Francisco';

try {
    update myAcct;
} catch (DmlException e) {
    // Process exception here
}
```

Note: For more information on processing `DmlExceptions`, see Bulk DML Exception Handling.

Upsert Statement

The `upsert` DML operation creates new records and updates sObject records within a single statement, using a specified field to determine the presence of existing objects, or the ID field if no field is specified.

Syntax

`upsert` ***`sObject`*** `[`***`opt_field`***`]`

`upsert` ***`sObject[]`*** `[`***`opt_field`***`]`

The `upsert` statement matches the sObjects with existing records by comparing values of one field. If you don't specify a field when calling this statement, the `upsert` statement uses the sObject's ID to match the sObject with existing records in Salesforce. Alternatively, you can specify a field to use for matching. For custom objects, specify a custom field marked as external ID. For standard objects, you can specify any field that has the `idLookup` attribute set to true. For example, the Email field of Contact or User has the `idLookup` attribute set. To check a field's attribute, see the Object Reference for Salesforce.

Also, you can use foreign keys to upsert sObject records if they have been set as reference fields. For more information, see Field Types in the *Object Reference for Salesforce*.

The optional field parameter, *`opt_field`*, is a field token (of type `Schema.SObjectField`). For example, to specify the MyExternalID custom field, the statement is:

```
upsert sObjectList Account.Fields.MyExternalId__c;
```

If the field used for matching doesn't have the `Unique` attribute set, the context user must have the "View All Records" object-level permission for the target object or the "View All Data" permission so that `upsert` doesn't accidentally insert a duplicate record.

Note: Custom field matching is case-insensitive only if the custom field has the **Unique** and **Treat "ABC" and "abc" as duplicate values (case insensitive)** attributes selected as part of the field definition. If so, "ABC123" is matched with "abc123." For more information, see "Create Custom Fields" in the Salesforce online help.

How Upsert Chooses to Insert or Update

Upsert uses the sObject record's primary key (the ID), an idLookup field, or an external ID field to determine whether it should create a record or update an existing one:

- If the key isn't matched, a new object record is created.
- If the key is matched once, the existing object record is updated.
- If the key is matched multiple times, an error is generated and the object record isn't inserted or updated.

Example

This example performs an upsert of a list of accounts.

```
List<Account> acctList = new List<Account>();
// Fill the accounts list with some accounts

try {
    upsert acctList;
} catch (DmlException e) {

}
```

This next example performs an upsert of a list of accounts using a foreign key for matching existing records, if any.

```
List<Account> acctList = new List<Account>();
// Fill the accounts list with some accounts

try {
    // Upsert using an external ID field
    upsert acctList myExtIDField__c;
} catch (DmlException e) {

}
```

Delete Statement

The `delete` DML operation deletes one or more existing sObject records, such as individual accounts or contacts, from your organization's data. `delete` is analogous to the `delete()` statement in the SOAP API.

Syntax

```
delete sObject
delete sObject[]
```

Example

The following example deletes all accounts that are named 'DotCom':

```
Account[] doomedAccts = [SELECT Id, Name FROM Account
                         WHERE Name = 'DotCom'];
try {
    delete doomedAccts;
} catch (DmlException e) {
    // Process exception here
}
```

Note: For more information on processing `DmlExceptions`, see Bulk DML Exception Handling.

Undelete Statement

The `undelete` DML operation restores one or more existing sObject records, such as individual accounts or contacts, from your organization's Recycle Bin. `undelete` is analogous to the UNDELETE statement in SQL.

Syntax

`undelete` ***sObject*** | ***ID***

`undelete` ***sObject[]*** | ***ID[]***

Example

The following example undeletes an account named 'Universal Containers'. The `ALL ROWS` keyword queries all rows for both top level and aggregate relationships, including deleted records and archived activities.

```
Account[] savedAccts = [SELECT Id, Name FROM Account WHERE Name = 'Universal Containers'
ALL ROWS];
try {
    undelete savedAccts;
} catch (DmlException e) {
    // Process exception here
}
```

Note: For more information on processing `DmlExceptions`, see Bulk DML Exception Handling.

Merge Statement

The `merge` statement merges up to three records of the same sObject type into one of the records, deleting the others, and re-parenting any related records.

Note: This DML operation does not have a matching Database system method.

Syntax

`merge` ***sObject sObject***

`merge` ***sObject sObject[]***

`merge` ***sObject ID***

`merge` ***sObject ID[]***

The first parameter represents the master record into which the other records are to be merged. The second parameter represents the one or two other records that should be merged and then deleted. You can pass these other records into the `merge` statement as a single sObject record or ID, or as a list of two sObject records or IDs.

Example

The following example merges two accounts named 'Acme Inc.' and 'Acme' into a single record:

```
List<Account> ls = new List<Account>{new Account(name='Acme Inc.'),new Account(name='Acme')};
insert ls;
Account masterAcct = [SELECT Id, Name FROM Account WHERE Name = 'Acme Inc.' LIMIT 1];
```

```
Account mergeAcct = [SELECT Id, Name FROM Account WHERE Name = 'Acme' LIMIT 1];
try {
    merge masterAcct mergeAcct;
} catch (DmlException e) {
    // Process exception here
}
```

Note: For more information on processing `DmlExceptions`, see Bulk DML Exception Handling.

ApexPages Namespace

The `ApexPages` namespace provides classes used in Visualforce controllers.

The following are the classes in the `ApexPages` namespace.

IN THIS SECTION:

Action Class

You can use `ApexPages.Action` to create an action method that you can use in a Visualforce custom controller or controller extension.

Component Class

Represents a dynamic Visualforce component in Apex.

IdeaStandardController Class

`IdeaStandardController` objects offer Ideas-specific functionality in addition to what is provided by the `StandardController`.

IdeaStandardSetController Class

`IdeaStandardSetController` objects offer Ideas-specific functionality in addition to what is provided by the `StandardSetController`.

KnowledgeArticleVersionStandardController Class

`KnowledgeArticleVersionStandardController` objects offer article-specific functionality in addition to what is provided by the `StandardController`.

Message Class

Contains validation errors that occur when the user saves the page that uses a standard controller.

StandardController Class

Use a StandardController when defining an extension for a standard controller.

StandardSetController Class

`StandardSetController` objects allow you to create list controllers similar to, or as extensions of, the pre-built Visualforce list controllers provided by Salesforce.

Action Class

You can use `ApexPages.Action` to create an action method that you can use in a Visualforce custom controller or controller extension.

Namespace

ApexPages

Usage

For example, you could create a `saveOver` method on a controller extension that performs a custom save.

Instantiation

The following code snippet illustrates how to instantiate a new `ApexPages.Action` object that uses the save action:

```
ApexPages.Action saveAction = new ApexPages.Action('{!save}');
```

IN THIS SECTION:

Action Constructors

Action Methods

Action Constructors

The following are constructors for `Action`.

IN THIS SECTION:

Action(action)

Creates a new instance of the `ApexPages.Action` class using the specified action.

`Action(action)`

Creates a new instance of the `ApexPages.Action` class using the specified action.

Signature

```
public Action(String action)
```

Parameters

`action`

Type: String

The action.

Action Methods

The following are methods for `Action`. All are instance methods.

IN THIS SECTION:

getExpression()
Returns the expression that is evaluated when the action is invoked.

invoke()
Invokes the action.

`getExpression()`

Returns the expression that is evaluated when the action is invoked.

Signature

```
public String getExpression()
```

Return Value

Type: String

`invoke()`

Invokes the action.

Signature

```
public System.PageReference invoke()
```

Return Value

Type: System.PageReference

Component Class

Represents a dynamic Visualforce component in Apex.

Namespace

ApexPages

Dynamic Component Properties

The following are properties for `Component`.

IN THIS SECTION:

childComponents
Returns a reference to the child components for the component.

expressions

Sets the content of an attribute using the expression language notation. The notation for this is `expressions.name_of_attribute`.

facets

Sets the content of a facet to a dynamic component. The notation is `facet.name_of_facet`.

childComponents

Returns a reference to the child components for the component.

Signature

```
public List <ApexPages.Component> childComponents {get; set;}
```

Property Value

Type: List<ApexPages.Component>

Example

```
Component.Apex.PageBlock pageBlk = new Component.Apex.PageBlock();

Component.Apex.PageBlockSection pageBlkSection = new
Component.Apex.PageBlockSection(title='dummy header');

pageBlk.childComponents.add(pageBlkSection);
```

expressions

Sets the content of an attribute using the expression language notation. The notation for this is `expressions.name_of_attribute`.

Signature

```
public String expressions {get; set;}
```

Property Value

Type: String

Example

```
Component.Apex.InputField inpFld = new
Component.Apex.InputField();
inpField.expressions.value = '{!Account.Name}';
inpField.expressions.id = '{!$User.FirstName}';
```

facets

Sets the content of a facet to a dynamic component. The notation is `facet.name_of_facet`.

Signature

```
public String facets {get; set;}
```

Property Value

Type: String

Usage

Note: This property is only accessible by components that support facets.

Example

```
Component.Apex.DataTable myDT = new
Component.Apex.DataTable();
Component.Apex.OutputText footer = new
Component.Apex.OutputText(value='Footer Copyright');
myDT.facets.footer = footer;
```

IdeaStandardController Class

`IdeaStandardController` objects offer Ideas-specific functionality in addition to what is provided by the `StandardController`.

Namespace

ApexPages

Usage

A method in the IdeaStandardController object is called by and operated on a particular instance of an IdeaStandardController.

Note: The `IdeaStandardSetController` and `IdeaStandardController` classes are currently available through a limited release program. For information on enabling these classes for your organization, contact your Salesforce representative.

In addition to the methods listed in this class, the `IdeaStandardController` class inherits all the methods associated with the `StandardController` class.

Instantiation

An IdeaStandardController object cannot be instantiated. An instance can be obtained through a constructor of a custom extension controller when using the standard ideas controller.

Example

The following example shows how an IdeaStandardController object can be used in the constructor for a custom list controller. This example provides the framework for manipulating the comment list data before displaying it on a Visualforce page.

```
public class MyIdeaExtension {

    private final ApexPages.IdeaStandardController ideaController;

    public MyIdeaExtension(ApexPages.IdeaStandardController controller) {
        ideaController = (ApexPages.IdeaStandardController)controller;
    }

    public List<IdeaComment> getModifiedComments() {
        IdeaComment[] comments = ideaController.getCommentList();
        // modify comments here
        return comments;
    }

}
```

The following Visualforce markup shows how the IdeaStandardController example shown above can be used in a page. This page must be named *detailPage* for this example to work.

Note: For the Visualforce page to display the idea and its comments, in the following example you need to specify the ID of a specific idea (for example, `/apex/detailPage?id=<ideaID>`) whose comments you want to view.

```
<!-- page named detailPage -->
<apex:page standardController="Idea" extensions="MyIdeaExtension">
    <apex:pageBlock title="Idea Section">
        <ideas:detailOutputLink page="detailPage" ideaId="{!idea.id}">{!idea.title}
        </ideas:detailOutputLink>
        <br/><br/>
        <apex:outputText >{!idea.body}</apex:outputText>
    </apex:pageBlock>
    <apex:pageBlock title="Comments Section">
        <apex:dataList var="a" value="{!modifiedComments}" id="list">
            {!a.commentBody}
        </apex:dataList>
        <ideas:detailOutputLink page="detailPage" ideaId="{!idea.id}"
               pageOffset="-1">Prev</ideas:detailOutputLink>
        |
        <ideas:detailOutputLink page="detailPage" ideaId="{!idea.id}"
               pageOffset="1">Next</ideas:detailOutputLink>
    </apex:pageBlock>
</apex:page>
```

SEE ALSO:

StandardController Class

IdeaStandardController Methods

The following are instance methods for `IdeaStandardController`.

IN THIS SECTION:

getCommentList()
Returns the list of read-only comments from the current page.

`getCommentList()`

Returns the list of read-only comments from the current page.

Signature

```
public IdeaComment[] getCommentList()
```

Return Value

Type: IdeaComment[]

This method returns the following comment properties:

- `id`
- `commentBody`
- `createdDate`
- `createdBy.Id`
- `createdBy.communityNickname`

IdeaStandardSetController Class

`IdeaStandardSetController` objects offer Ideas-specific functionality in addition to what is provided by the `StandardSetController`.

Namespace

ApexPages

Usage

Note: The `IdeaStandardSetController` and `IdeaStandardController` classes are currently available through a limited release program. For information on enabling these classes for your organization, contact your Salesforce representative.

In addition to the method listed above, the `IdeaStandardSetController` class inherits the methods associated with the `StandardSetController`.

Note: The methods inherited from the `StandardSetController` cannot be used to affect the list of ideas returned by the `getIdeaList` method.

Instantiation

An IdeaStandardSetController object cannot be instantiated. An instance can be obtained through a constructor of a custom extension controller when using the standard list controller for ideas.

Example: Displaying a Profile Page

The following example shows how an IdeaStandardSetController object can be used in the constructor for a custom list controller:

```
public class MyIdeaProfileExtension {
    private final ApexPages.IdeaStandardSetController ideaSetController;

    public MyIdeaProfileExtension(ApexPages.IdeaStandardSetController controller) {
        ideaSetController = (ApexPages.IdeaStandardSetController)controller;
    }

    public List<Idea> getModifiedIdeas() {
        Idea[] ideas = ideaSetController.getIdeaList();
        // modify ideas here
        return ideas;
    }

}
```

The following Visualforce markup shows how the IdeaStandardSetController example shown above and the `<ideas:profileListOutputLink>` component can display a profile page that lists the recent replies, submitted ideas, and votes associated with a user. Because this example does not identify a specific user ID, the page automatically shows the profile page for the current logged in user. This page must be named *`profilePage`* in order for this example to work:

```
<!-- page named profilePage -->
<apex:page standardController="Idea" extensions="MyIdeaProfileExtension"
recordSetVar="ideaSetVar">
    <apex:pageBlock >
        <ideas:profileListOutputLink sort="recentReplies" page="profilePage">
          Recent Replies</ideas:profileListOutputLink>
        |
        <ideas:profileListOutputLink sort="ideas" page="profilePage">Ideas Submitted
        </ideas:profileListOutputLink>
        |
        <ideas:profileListOutputLink sort="votes" page="profilePage">Ideas Voted
        </ideas:profileListOutputLink>
    </apex:pageBlock>
    <apex:pageBlock >
        <apex:dataList value="{!modifiedIdeas}" var="ideadata">
            <ideas:detailoutputlink ideaId="{!ideadata.id}" page="viewPage">
             {!ideadata.title}</ideas:detailoutputlink>
        </apex:dataList>
    </apex:pageBlock>
</apex:page>
```

In the previous example, the `<ideas:detailoutputlink>` component links to the following Visualforce markup that displays the detail page for a specific idea. This page must be named *`viewPage`* in order for this example to work:

```
<!-- page named viewPage -->
<apex:page standardController="Idea">
    <apex:pageBlock title="Idea Section">
        <ideas:detailOutputLink page="viewPage" ideaId="{!idea.id}">{!idea.title}
        </ideas:detailOutputLink>
        <br/><br/>
```

```
        <apex:outputText>{!idea.body}</apex:outputText>
    </apex:pageBlock>
</apex:page>
```

Example: Displaying a List of Top, Recent, and Most Popular Ideas and Comments

The following example shows how an IdeaStandardSetController object can be used in the constructor for a custom list controller:

Note: You must have created at least one idea for this example to return any ideas.

```
public class MyIdeaListExtension {
    private final ApexPages.IdeaStandardSetController ideaSetController;

    public MyIdeaListExtension (ApexPages.IdeaStandardSetController controller) {
        ideaSetController = (ApexPages.IdeaStandardSetController)controller;
    }

    public List<Idea> getModifiedIdeas() {
        Idea[] ideas = ideaSetController.getIdeaList();
        // modify ideas here
        return ideas;
    }
}
```

The following Visualforce markup shows how the IdeaStandardSetController example shown above can be used with the `<ideas:listOutputLink>` component to display a list of recent, top, and most popular ideas and comments. This page must be named *`listPage`* in order for this example to work:

```
<!-- page named listPage -->
<apex:page standardController="Idea" extensions="MyIdeaListExtension"
recordSetVar="ideaSetVar">
    <apex:pageBlock >
        <ideas:listOutputLink sort="recent" page="listPage">Recent Ideas
        </ideas:listOutputLink>
        |
        <ideas:listOutputLink sort="top" page="listPage">Top Ideas
        </ideas:listOutputLink>
        |
        <ideas:listOutputLink sort="popular" page="listPage">Popular Ideas
        </ideas:listOutputLink>
        |
        <ideas:listOutputLink sort="comments" page="listPage">Recent Comments
        </ideas:listOutputLink>
    </apex:pageBlock>
    <apex:pageBlock >
        <apex:dataList value="{!modifiedIdeas}" var="ideadata">
            <ideas:detailoutputlink ideaId="{!ideadata.id}" page="viewPage">
             {!ideadata.title}</ideas:detailoutputlink>
        </apex:dataList>
    </apex:pageBlock>
</apex:page>
```

In the previous example, the <ideas:detailoutputlink> component links to the following Visualforce markup that displays the detail page for a specific idea. This page must be named *viewPage*.

```
<!-- page named viewPage -->
<apex:page standardController="Idea">
    <apex:pageBlock title="Idea Section">
        <ideas:detailOutputLink page="viewPage" ideaId="{!idea.id}">{!idea.title}
        </ideas:detailOutputLink>
        <br/><br/>
        <apex:outputText>{!idea.body}</apex:outputText>
    </apex:pageBlock>
</apex:page>
```

SEE ALSO:

StandardSetController Class

IdeaStandardSetController Methods

The following are instance methods for `IdeaStandardSetController`.

IN THIS SECTION:

getIdeaList()
Returns the list of read-only ideas in the current page set.

`getIdeaList()`

Returns the list of read-only ideas in the current page set.

Signature

```
public Idea[] getIdeaList()
```

Return Value

Type: Idea[]

Usage

You can use the <ideas:listOutputLink>, <ideas:profileListOutputLink>, and <ideas:detailOutputLink> components to display profile pages as well as idea list and detail pages (see the examples below). The following is a list of properties returned by this method:

- `Body`
- `Categories`
- `Category`
- `CreatedBy.CommunityNickname`
- `CreatedBy.Id`
- `CreatedDate`

- `Id`
- `LastCommentDate`
- `LastComment.Id`
- `LastComment.CommentBody`
- `LastComment.CreatedBy.CommunityNickname`
- `LastComment.CreatedBy.Id`
- `NumComments`
- `Status`
- `Title`
- `VoteTotal`

KnowledgeArticleVersionStandardController Class

`KnowledgeArticleVersionStandardController` objects offer article-specific functionality in addition to what is provided by the `StandardController`.

Namespace

ApexPages

Usage

In addition to the method listed above, the `KnowledgeArticleVersionStandardController` class inherits all the methods associated with `StandardController`.

Note: Though inherited, the `edit`, `delete`, and `save` methods don't serve a function when used with the `KnowledgeArticleVersionStandardController` class.

Example

The following example shows how a `KnowledgeArticleVersionStandardController` object can be used to create a custom extension controller. In this example, you create a class named `AgentContributionArticleController` that allows customer-support agents to see pre-populated fields on the draft articles they create while closing cases.

Prerequisites:

1. Create an article type called *`FAQ`*. For instructions, see "Create Article Types" in the Salesforce online help.
2. Create a text custom field called `Details`. For instructions, see "Add Custom Fields to Article Types" in the Salesforce online help.
3. Create a category group called *`Geography`* and assign it to a category called *`USA`*. For instructions, see "Create and Modify Category Groups" and "Add Data Categories to Category Groups" in the Salesforce online help.
4. Create a category group called *`Topics`* and assign it a category called *`Maintenance`*.

```
/** Custom extension controller for the simplified article edit page that
    appears when an article is created on the close-case page.
*/
public class AgentContributionArticleController {
    // The constructor must take a ApexPages.KnowledgeArticleVersionStandardController as
 an argument
```

```
    public AgentContributionArticleController(
        ApexPages.KnowledgeArticleVersionStandardController ctl) {
        // This is the SObject for the new article.
        //It can optionally be cast to the proper article type.
        // For example, FAQ__kav article = (FAQ__kav) ctl.getRecord();
        SObject article = ctl.getRecord();
        // This returns the ID of the case that was closed.
        String sourceId = ctl.getSourceId();
        Case c = [SELECT Subject, Description FROM Case WHERE Id=:sourceId];

        // This overrides the default behavior of pre-filling the
        // title of the article with the subject of the closed case.
        article.put('title', 'From Case: '+c.subject);
        article.put('details__c',c.description);

        // Only one category per category group can be specified.
        ctl.selectDataCategory('Geography','USA');
        ctl.selectDataCategory('Topics','Maintenance');
    }
}
```

```
/** Test class for the custom extension controller.
*/
@isTest
private class AgentContributionArticleControllerTest {
    static testMethod void testAgentContributionArticleController() {
         String caseSubject = 'my test';
         String caseDesc = 'my test description';

         Case c = new Case();
         c.subject= caseSubject;
         c.description = caseDesc;
         insert c;
         String caseId = c.id;
         System.debug('Created Case: ' + caseId);

         ApexPages.currentPage().getParameters().put('sourceId', caseId);
         ApexPages.currentPage().getParameters().put('sfdc.override', '1');

         ApexPages.KnowledgeArticleVersionStandardController ctl =
            new ApexPages.KnowledgeArticleVersionStandardController(new FAQ__kav());

         new AgentContributionArticleController(ctl);

         System.assertEquals(caseId, ctl.getSourceId());
         System.assertEquals('From Case: '+caseSubject, ctl.getRecord().get('title'));
         System.assertEquals(caseDesc, ctl.getRecord().get('details__c'));
    }
}
```

If you created the custom extension controller for the purpose described in the previous example (that is, to modify submitted-via-case articles), complete the following steps after creating the class:

1. Log into your Salesforce organization and from Setup, enter *Knowledge Settings* in the Quick Find box, then select **Knowledge Settings**.

2. Click **Edit**.
3. Assign the class to the `Use Apex customization` field. This associates the article type specified in the new class with the article type assigned to closed cases.
4. Click **Save**.

IN THIS SECTION:

KnowledgeArticleVersionStandardController Constructors

KnowledgeArticleVersionStandardController Methods

SEE ALSO:

StandardController Class

KnowledgeArticleVersionStandardController Constructors

The following are constructors for `KnowledgeArticleVersionStandardController`.

IN THIS SECTION:

KnowledgeArticleVersionStandardController(article)
Creates a new instance of the `ApexPages.KnowledgeArticleVersionStandardController` class using the specified knowledge article.

`KnowledgeArticleVersionStandardController(article)`

Creates a new instance of the `ApexPages.KnowledgeArticleVersionStandardController` class using the specified knowledge article.

Signature

```
public KnowledgeArticleVersionStandardController(SObject article)
```

Parameters

`article`
Type: SObject

The knowledge article, such as `FAQ_kav`.

KnowledgeArticleVersionStandardController Methods

The following are instance methods for `KnowledgeArticleVersionStandardController`.

IN THIS SECTION:

getSourceId()
Returns the ID for the source object record when creating a new article from another object.

setDataCategory(categoryGroup, category)
Specifies a default data category for the specified data category group when creating a new article.

`getSourceId()`

Returns the ID for the source object record when creating a new article from another object.

Signature

```
public String getSourceId()
```

Return Value

Type: String

`setDataCategory(categoryGroup, category)`

Specifies a default data category for the specified data category group when creating a new article.

Signature

```
public Void setDataCategory(String categoryGroup, String category)
```

Parameters

`categoryGroup`
Type: String

`category`
Type: String

Return Value

Type: Void

Message Class

Contains validation errors that occur when the user saves the page that uses a standard controller.

Namespace

ApexPages

Usage

When using a standard controller, all validation errors, both custom and standard, that occur when the user saves the page are automatically added to the page error collections. If an `inputField` component is bound to the field with an error, the message is added to the component's error collection. All messages are added to the page's error collection. For more information, see Validation Rules and Standard Controllers in the *Visualforce Developer's Guide*.

If your application uses a custom controller or extension, you must use the `message` class for collecting errors.

Instantiation

In a custom controller or controller extension, you can instantiate a Message in one of these ways:

- ```
  ApexPages.Message myMsg = new ApexPages.Message(ApexPages.severity, summary);
  ```

  where `ApexPages.severity` is the enum that determines how severe a message is, and `summary` is the String used to summarize the message. For example:

  ```
  ApexPages.Message myMsg = new ApexPages.Message(ApexPages.Severity.FATAL, 'my error msg');
  ```

- ```
  ApexPages.Message myMsg = new ApexPages.Message(ApexPages.severity, summary, detail);
  ```

  where `ApexPages. severity` is the enum that determines how severe a message is, `summary` is the String used to summarize the message, and `detail` is the String used to provide more detailed information about the error.

ApexPages.Severity Enum

To specify the severity of the message, use the `ApexPages.Severity` enum values. The following are the valid values:

- `CONFIRM`
- `ERROR`
- `FATAL`
- `INFO`
- `WARNING`

All enums have access to standard methods, such as `name` and `value`.

IN THIS SECTION:

Message Constructors

Message Methods

Message Constructors

The following are constructors for `Message`.

IN THIS SECTION:

Message(severity, summary)

Creates a new instance of the `ApexPages.Message` class using the specified message severity and summary.

Message(severity, summary, detail)

Creates a new instance of the `ApexPages.Message` class using the specified message severity, summary, and message detail.

Message(severity, summary, detail, id)

Creates a new instance of the `ApexPages.Message` class using the specified severity, summary, detail, and component ID.

`Message(severity, summary)`

Creates a new instance of the `ApexPages.Message` class using the specified message severity and summary.

Signature

```
public Message(ApexPages.Severity severity, String summary)
```

Parameters

severity
Type: ApexPages.Severity

The severity of a Visualforce message.

summary
Type: String

The summary Visualforce message.

Message(severity, summary, detail)

Creates a new instance of the `ApexPages.Message` class using the specified message severity, summary, and message detail.

Signature

```
public Message(ApexPages.Severity severity, String summary, String detail)
```

Parameters

severity
Type: ApexPages.Severity

The severity of a Visualforce message.

summary
Type: String

The summary Visualforce message.

detail
Type: String

The detailed Visualforce message.

Message(severity, summary, detail, id)

Creates a new instance of the `ApexPages.Message` class using the specified severity, summary, detail, and component ID.

Signature

```
public Message(ApexPages.Severity severity, String summary, String detail, String id)
```

Parameters

severity
Type: ApexPages.Severity

The severity of a Visualforce message.

summary
Type: String

The summary Visualforce message.

detail
Type: String

The detailed Visualforce message.

id
Type: String

The ID of the Visualforce component to associate with the message, for example, a form field with an error.

Message Methods

The following are methods for `Message`. All are instance methods.

IN THIS SECTION:

getComponentLabel()
Returns the label of the associated `inputField` component. If no label is defined, this method returns `null`.

getDetail()
Returns the value of the detail parameter used to create the message. If no detail String was specified, this method returns `null`.

getSeverity()
Returns the severity enum used to create the message.

getSummary()
Returns the summary String used to create the message.

`getComponentLabel()`

Returns the label of the associated `inputField` component. If no label is defined, this method returns `null`.

Signature

```
public String getComponentLabel()
```

Return Value

Type: String

`getDetail()`

Returns the value of the detail parameter used to create the message. If no detail String was specified, this method returns `null`.

Signature

```
public String getDetail()
```

Return Value

Type: String

`getSeverity()`

Returns the severity enum used to create the message.

Signature

```
public ApexPages.Severity getSeverity()
```

Return Value

Type: ApexPages.Severity

`getSummary()`

Returns the summary String used to create the message.

Signature

```
public String getSummary()
```

Return Value

Type: String

StandardController Class

Use a StandardController when defining an extension for a standard controller.

Namespace

ApexPages

Usage

StandardController objects reference the pre-built Visualforce controllers provided by Salesforce. The only time it is necessary to refer to a StandardController object is when defining an extension for a standard controller. StandardController is the data type of the single argument in the extension class constructor.

Instantiation

You can instantiate a StandardController in the following way:

```
ApexPages.StandardController sc = new ApexPages.StandardController(sObject);
```

Example

The following example shows how a StandardController object can be used in the constructor for a standard controller extension:

```
public class myControllerExtension {

    private final Account acct;

    // The extension constructor initializes the private member
    // variable acct by using the getRecord method from the standard
    // controller.
    public myControllerExtension(ApexPages.StandardController stdController) {
        this.acct = (Account)stdController.getRecord();
    }

    public String getGreeting() {
        return 'Hello ' + acct.name + ' (' + acct.id + ')';
    }
}
```

The following Visualforce markup shows how the controller extension from above can be used in a page:

```
<apex:page standardController="Account" extensions="myControllerExtension">
    {!greeting} <p/>
    <apex:form>
        <apex:inputField value="{!account.name}"/> <p/>
        <apex:commandButton value="Save" action="{!save}"/>
    </apex:form>
</apex:page>
```

IN THIS SECTION:

StandardController Constructors

StandardController Methods

StandardController Constructors

The following are constructors for `StandardController`.

IN THIS SECTION:

StandardController(controllerSObject)

Creates a new instance of the `ApexPages.StandardController` class for the specified standard or custom object.

`StandardController(controllerSObject)`

Creates a new instance of the `ApexPages.StandardController` class for the specified standard or custom object.

Signature

```
public StandardController(SObject controllerSObject)
```

Parameters

controllerSObject
Type: SObject

A standard or custom object.

StandardController Methods

The following are methods for `StandardController`. All are instance methods.

IN THIS SECTION:

addFields(fieldNames)
When a Visualforce page is loaded, the fields accessible to the page are based on the fields referenced in the Visualforce markup. This method adds a reference to each field specified in `fieldNames` so that the controller can explicitly access those fields as well.

cancel()
Returns the PageReference of the cancel page.

delete()
Deletes record and returns the PageReference of the delete page.

edit()
Returns the PageReference of the standard edit page.

getId()
Returns the ID of the record that is currently in context, based on the value of the `id` query string parameter in the Visualforce page URL.

getRecord()
Returns the record that is currently in context, based on the value of the `id` query string parameter in the Visualforce page URL.

reset()
Forces the controller to reacquire access to newly referenced fields. Any changes made to the record prior to this method call are discarded.

save()
Saves changes and returns the updated PageReference.

view()
Returns the PageReference object of the standard detail page.

`addFields(fieldNames)`

When a Visualforce page is loaded, the fields accessible to the page are based on the fields referenced in the Visualforce markup. This method adds a reference to each field specified in `fieldNames` so that the controller can explicitly access those fields as well.

Signature

```
public Void addFields(List<String> fieldNames)
```

Parameters

fieldNames
Type: List<String>

Return Value

Type: Void

Usage

This method should be called before a record has been loaded—typically, it's called by the controller's constructor. If this method is called outside of the constructor, you must use the `reset()` method before calling `addFields()`.

The strings in `fieldNames` can either be the API name of a field, such as AccountId, or they can be explicit relationships to fields, such as `something__r.myField__c`.

This method is only for controllers used by dynamicVisualforce bindings.

cancel()

Returns the PageReference of the cancel page.

Signature

```
public System.PageReference cancel()
```

Return Value

Type: System.PageReference

delete()

Deletes record and returns the PageReference of the delete page.

Signature

```
public System.PageReference delete()
```

Return Value

Type: System.PageReference

edit()

Returns the PageReference of the standard edit page.

Signature

```
public System.PageReference edit()
```

Return Value

Type: System.PageReference

`getId()`

Returns the ID of the record that is currently in context, based on the value of the `id` query string parameter in the Visualforce page URL.

Signature

```
public String getId()
```

Return Value

Type: String

`getRecord()`

Returns the record that is currently in context, based on the value of the `id` query string parameter in the Visualforce page URL.

Signature

```
public SObject getRecord()
```

Return Value

Type: sObject

Usage

Note that only the fields that are referenced in the associated Visualforce markup are available for querying on this SObject. All other fields, including fields from any related objects, must be queried using a SOQL expression.

Tip: You can work around this restriction by including a hidden component that references any additional fields that you want to query. Hide the component from display by setting the component's `rendered` attribute to `false`.

Example

```
<apex:outputText
value="{!account.billingcity}
{!account.contacts}"
rendered="false"/>
```

`reset()`

Forces the controller to reacquire access to newly referenced fields. Any changes made to the record prior to this method call are discarded.

Signature

```
public Void reset()
```

Return Value

Type: Void

Usage

This method is only used if `addFields` is called outside the constructor, and it must be called directly before `addFields`.

This method is only for controllers used by dynamicVisualforce bindings.

`save()`

Saves changes and returns the updated PageReference.

Signature

```
public System.PageReference save()
```

Return Value

Type: System.PageReference

`view()`

Returns the PageReference object of the standard detail page.

Signature

```
public System.PageReference view()
```

Return Value

Type: System.PageReference

StandardSetController Class

`StandardSetController` objects allow you to create list controllers similar to, or as extensions of, the pre-built Visualforce list controllers provided by Salesforce.

Namespace

ApexPages

Usage

The `StandardSetController` class also contains a *prototype object*. This is a single `sObject` contained within the Visualforce StandardSetController class. If the prototype object's fields are set, those values are used during the save action, meaning that the values

are applied to every record in the set controller's collection. This is useful for writing pages that perform mass updates (applying identical changes to fields within a collection of objects).

Note: Fields that are required in other Salesforce objects will keep the same requiredness when used by the prototype object.

Instantiation

You can instantiate a StandardSetController in either of the following ways:

- From a list of sObjects:

```
List<account> accountList = [SELECT Name FROM Account LIMIT 20];
ApexPages.StandardSetController ssc = new ApexPages.StandardSetController(accountList);
```

- From a query locator:

```
ApexPages.StandardSetController ssc =
new ApexPages.StandardSetController(Database.getQueryLocator([SELECT Name,CloseDate FROM
 Opportunity]));
```

Note: The maximum record limit for StandardSetController is 10,000 records. Instantiating StandardSetController using a query locator returning more than 10,000 records causes a LimitException to be thrown. However, instantiating StandardSetController with a list of more than 10,000 records doesn't throw an exception, and instead truncates the records to the limit.

Example

The following example shows how a StandardSetController object can be used in the constructor for a custom list controller:

```
public class opportunityList2Con {
    // ApexPages.StandardSetController must be instantiated
    // for standard list controllers
    public ApexPages.StandardSetController setCon {
        get {
            if(setCon == null) {
                setCon = new ApexPages.StandardSetController(Database.getQueryLocator(
                    [SELECT Name, CloseDate FROM Opportunity]));
            }
            return setCon;
        }
        set;
    }

    // Initialize setCon and return a list of records
    public List<Opportunity> getOpportunities() {
        return (List<Opportunity>) setCon.getRecords();
    }
}
```

The following Visualforce markup shows how the controller above can be used in a page:

```
<apex:page controller="opportunityList2Con">
    <apex:pageBlock>
        <apex:pageBlockTable value="{!opportunities}" var="o">
            <apex:column value="{!o.Name}"/>
```

```
            <apex:column value="{!o.CloseDate}"/>
        </apex:pageBlockTable>
    </apex:pageBlock>
</apex:page>
```

IN THIS SECTION:

StandardSetController Constructors

StandardSetController Methods

StandardSetController Constructors

The following are constructors for `StandardSetController`.

IN THIS SECTION:

StandardSetController(queryLocator)

Creates an instance of the `ApexPages.StandardSetController` class for the list of objects returned by the query locator.

StandardSetController(controllerSObjects)

Creates an instance of the `ApexPages.StandardSetController` class for the specified list of standard or custom objects.

`StandardSetController(queryLocator)`

Creates an instance of the `ApexPages.StandardSetController` class for the list of objects returned by the query locator.

Signature

```
public StandardSetController(Database.QueryLocator queryLocator)
```

Parameters

`queryLocator`

Type: Database.QueryLocator

A query locator representing a list of sObjects.

`StandardSetController(controllerSObjects)`

Creates an instance of the `ApexPages.StandardSetController` class for the specified list of standard or custom objects.

Signature

```
public StandardSetController(List<sObject> controllerSObjects)
```

Parameters

`controllerSObjects`

Type: List on page 3603<sObject on page 3800>

A List of standard or custom objects.

Example

```
List<account> accountList = [SELECT Name FROM Account LIMIT 20];
ApexPages.StandardSetController ssc = new ApexPages.StandardSetController(accountList);
```

StandardSetController Methods

The following are methods for `StandardSetController`. All are instance methods.

IN THIS SECTION:

cancel()

Returns the PageReference of the original page, if known, or the home page.

first()

Changes the set of records that the controller returns to the first page of records.

getCompleteResult()

Indicates whether there are more records in the set than the maximum record limit. If this is false, there are more records than you can process using the list controller. The maximum record limit is 10,000 records.

getFilterId()

Returns the ID of the filter that is currently in context.

getHasNext()

Indicates whether there are more records after the current page set.

getHasPrevious()

Indicates whether there are more records before the current page set.

getListViewOptions()

Returns a list of the listviews available to the current user.

getPageNumber()

Returns the page number of the current page set. Note that the first page returns 1.

getPageSize()

Returns the number of records included in each page set.

getRecord()

Returns the sObject that represents the changes to the selected records. This retrieves the prototype object contained within the class, and is used for performing mass updates.

getRecords()

Returns the list of sObjects in the current page set. This list is immutable, i.e. you can't call `clear()` on it.

getResultSize()

Returns the number of records in the set.

getSelected()

Returns the list of sObjects that have been selected.

last()

Changes the set of records that the controller returns to the last page of records.

next()

Changes the set of records that the controller returns to the next page of records.

previous()

Changes the set of records that the controller returns to the previous page of records.

save()

Inserts new records or updates existing records that have been changed. After this operation is finished, it returns a PageReference to the original page, if known, or the home page.

setFilterID(filterId)

Sets the filter ID of the controller.

setpageNumber(pageNumber)

Sets the page number.

setPageSize(pageSize)

Sets the number of records in each page set.

setSelected(selectedRecords)

Set the selected records to the records specified in the *selectedRecords* argument.

cancel()

Returns the PageReference of the original page, if known, or the home page.

Signature

```
public System.PageReference cancel()
```

Return Value

Type: System.PageReference

SEE ALSO:

Visualforce Developer Guide: Standard List Controller Actions

first()

Changes the set of records that the controller returns to the first page of records.

Signature

```
public Void first()
```

Return Value

Type: Void

SEE ALSO:

Visualforce Developer Guide: Standard List Controller Actions

getCompleteResult()

Indicates whether there are more records in the set than the maximum record limit. If this is false, there are more records than you can process using the list controller. The maximum record limit is 10,000 records.

Signature

```
public Boolean getCompleteResult()
```

Return Value

Type: Boolean

getFilterId()

Returns the ID of the filter that is currently in context.

Note: The `getFilterID()` method doesn't support list views without filter IDs, such as the Recently Viewed list view. In these cases, the method returns the first filter ID of the object's available list views. If called within an `<apex:enhancedList>` component, the method returns the filter ID of the last used list view.

Signature

```
public String getFilterId()
```

Return Value

Type: String

SEE ALSO:

Visualforce Developer Guide: Standard List Controller Actions

Visualforce Developer Guide: List Views with Standard List Controllers

getHasNext()

Indicates whether there are more records after the current page set.

Signature

```
public Boolean getHasNext()
```

Return Value

Type: Boolean

getHasPrevious()

Indicates whether there are more records before the current page set.

Signature

```
public Boolean getHasPrevious()
```

Return Value

Type: Boolean

`getListViewOptions()`

Returns a list of the listviews available to the current user.

Signature

```
public System.SelectOption getListViewOptions()
```

Return Value

Type: System.SelectOption[]

SEE ALSO:

Visualforce Developer Guide: Standard List Controller Actions

Visualforce Developer Guide: List Views with Standard List Controllers

`getPageNumber()`

Returns the page number of the current page set. Note that the first page returns 1.

Signature

```
public Integer getPageNumber()
```

Return Value

Type: Integer

`getPageSize()`

Returns the number of records included in each page set.

Signature

```
public Integer getPageSize()
```

Return Value

Type: Integer

getRecord()

Returns the sObject that represents the changes to the selected records. This retrieves the prototype object contained within the class, and is used for performing mass updates.

Signature

```
public sObject getRecord()
```

Return Value

Type: sObject

SEE ALSO:

Visualforce Developer Guide: Building a Custom List Controller

getRecords()

Returns the list of sObjects in the current page set. This list is immutable, i.e. you can't call `clear()` on it.

Signature

```
public sObject[] getRecords()
```

Return Value

Type: sObject[]

SEE ALSO:

Visualforce Developer Guide: Building a Custom List Controller

getResultSize()

Returns the number of records in the set.

Signature

```
public Integer getResultSize()
```

Return Value

Type: Integer

getSelected()

Returns the list of sObjects that have been selected.

Signature

```
public sObject[] getSelected()
```

Return Value

Type: sObject[]

`last()`

Changes the set of records that the controller returns to the last page of records.

Signature

```
public Void last()
```

Return Value

Type: Void

SEE ALSO:

Visualforce Developer Guide: Standard List Controller Actions

`next()`

Changes the set of records that the controller returns to the next page of records.

Signature

```
public Void next()
```

Return Value

Type: Void

SEE ALSO:

Visualforce Developer Guide: Standard List Controller Actions

`previous()`

Changes the set of records that the controller returns to the previous page of records.

Signature

```
public Void previous()
```

Return Value

Type: Void

SEE ALSO:

Visualforce Developer Guide: Standard List Controller Actions

save()

Inserts new records or updates existing records that have been changed. After this operation is finished, it returns a PageReference to the original page, if known, or the home page.

Signature

```
public System.PageReference save()
```

Return Value

Type: System.PageReference

SEE ALSO:

Visualforce Developer Guide: Standard List Controller Actions

setFilterID(filterId)

Sets the filter ID of the controller.

Signature

```
public Void setFilterID(String filterId)
```

Parameters

filterId
Type: String

Return Value

Type: Void

setpageNumber(pageNumber)

Sets the page number.

Signature

```
public Void setpageNumber(Integer pageNumber)
```

Parameters

pageNumber
Type: Integer

Return Value

Type: Void

setPageSize(pageSize)

Sets the number of records in each page set.

Signature

```
public Void setPageSize(Integer pageSize)
```

Parameters

pageSize
Type: Integer

Return Value

Type: Void

setSelected(selectedRecords)

Set the selected records to the records specified in the *selectedRecords* argument.

Signature

```
public Void setSelected(sObject[] selectedRecords)
```

Parameters

selectedRecords
Type: sObject[]

Return Value

Type: Void

Usage

Use the `setSelected()` method in your Apex controller or controller extension to manually set the records displayed on a Visualforce page. The `setSelected()` method overwrites any previously selected records with the records specified in the *selectedRecords* argument.

Example

`AccountNamePage` shows a table of account names. `MyControllerExtension`'s constructor contains a SOQL query that returns a list of accounts. This list is passed into `setSelected()` so that the account records in the list are selected and displayed in the table.

```
<!-- AccountNamePage.page -->
<apex:page standardController="Account" recordSetVar="accounts"
extensions="MyControllerExtension">
    <apex:pageBlock>
        <apex:pageBlockTable value="{!accounts}" var="acc">
            <apex:column value="{!acc.name}"/>
```

```
        </apex:pageBlockTable>
    </apex:pageBlock>
</apex:page>


// MyControllerExtension.cls
public with sharing class MyControllerExtension {
    private ApexPages.StandardSetController setController;

    public MyControllerExtension(ApexPages.StandardSetController setController) {
        this.setController = setController;

        Account [] records = [SELECT Id, Name FROM Account LIMIT 30];
        setController.setSelected(records);
    }
}
```

SEE ALSO:

Visualforce Developer Guide: Accessing Data with List Controllers

AppLauncher Namespace

The `AppLauncher` namespace provides methods for managing the appearance of apps in the App Launcher, including their visibility and sort order.

The following class is in the `AppLauncher` namespace.

IN THIS SECTION:

AppMenu Class

Contains methods to set the appearance of apps in the App Launcher.

ChangePasswordController Class

This class and its methods are for internal use only.

CommunityLogoController Class

This class and its methods are for internal use only.

EmployeeLoginLinkController Class

This class and its methods are for internal use only.

ForgotPasswordController Class

This class and its methods are for internal use only.

IdentityHeaderController Class

This class and its methods are for internal use only.

LoginFormController Class

This class and its methods are for internal use only.

SelfRegisterController Class

This class and its methods are for internal use only.

SocialLoginController Class
This class and its methods are for internal use only.

AppMenu Class

Contains methods to set the appearance of apps in the App Launcher.

Namespace

AppLauncher

IN THIS SECTION:

AppMenu Methods

AppMenu Methods

The following are methods for `AppMenu`.

IN THIS SECTION:

setAppVisibility(appMenuItemId, isVisible)
Shows or hides specific apps in the App Launcher.

setOrgSortOrder(appIds)
Sets the organization-wide default sort order for the App Launcher based on a List of app menu item IDs in the desired order.

setUserSortOrder(appIds)
Sets an individual user's default sort order for the App Launcher based on a List of app menu item IDs in the desired order.

`setAppVisibility(appMenuItemId, isVisible)`

Shows or hides specific apps in the App Launcher.

Signature

```
public static void setAppVisibility(Id appMenuItemId, Boolean isVisible)
```

Parameters

`appMenuItemId`
Type: Id

The 15-character application ID value for an app. For more information, see the `ApplicationId` field for AppMenuItem or the `AppMenuItemId` field for UserAppMenuItem in the *Salesforce Object Reference*

`isVisible`
Type: Boolean

If `true`, the app is visible.

Return Value

Type: void

`setOrgSortOrder(appIds)`

Sets the organization-wide default sort order for the App Launcher based on a List of app menu item IDs in the desired order.

Signature

```
public static void setOrgSortOrder(List<Id> appIds)
```

Parameters

appIds
Type: List<Id>

A list of application ID values. For more information, see the `ApplicationId` field for AppMenuItem in the *Salesforce Object Reference*.

Return Value

Type: void

`setUserSortOrder(appIds)`

Sets an individual user's default sort order for the App Launcher based on a List of app menu item IDs in the desired order.

Signature

```
public static void setUserSortOrder(List<Id> appIds)
```

Parameters

appIds
Type: List<Id>

A list of application ID values. For more information, see the `AppMenuItemId` field for UserAppMenuItem in the *Salesforce Object Reference*.

Return Value

Type: void

ChangePasswordController Class

This class and its methods are for internal use only.

Namespace

AppLauncher

CommunityLogoController Class

This class and its methods are for internal use only.

Namespace

AppLauncher

EmployeeLoginLinkController Class

This class and its methods are for internal use only.

Namespace

AppLauncher

ForgotPasswordController Class

This class and its methods are for internal use only.

Namespace

AppLauncher

IdentityHeaderController Class

This class and its methods are for internal use only.

Namespace

AppLauncher

LoginFormController Class

This class and its methods are for internal use only.

Namespace

AppLauncher

SelfRegisterController Class

This class and its methods are for internal use only.

Namespace

AppLauncher

SocialLoginController Class

This class and its methods are for internal use only.

Namespace

AppLauncher

Approval Namespace

The `Approval` namespace provides classes and methods for approval processes.

The following are the classes in the `Approval` namespace.

IN THIS SECTION:

LockResult Class
The result of a record lock returned by a `System.Approval.lock()` method.

ProcessRequest Class
The `ProcessRequest` class is the parent class for the `ProcessSubmitRequest` and `ProcessWorkitemRequest` classes. Use the `ProcessRequest` class to write generic Apex that can process objects from either class.

ProcessResult Class
After you submit a record for approval, use the `ProcessResult` class to process the results of an approval process.

ProcessSubmitRequest Class
Use the `ProcessSubmitRequest` class to submit a record for approval.

ProcessWorkitemRequest Class
Use the `ProcessWorkitemRequest` class for processing an approval request after it is submitted.

UnlockResult Class
The result of a record unlock, returned by a `System.Approval.unlock()` method.

LockResult Class

The result of a record lock returned by a `System.Approval.lock()` method.

Namespace

Approval

Usage

The `System.Approval.lock()` methods return Approval.LockResult objects. Each element in a LockResult array corresponds to an element in the ID or sObject array passed as a parameter to a `lock` method. The first element in the LockResult array corresponds to the first element in the ID or sObject array, the second element corresponds to the second element, and so on. If only one ID or sObject is passed in, the LockResult array contains a single element.

Example

The following example obtains and iterates through the returned Approval.LockResult objects. It locks some queried accounts using `Approval.lock` with a `false` second parameter to allow partial processing of records on failure. Next, it iterates through the results to determine whether the operation was successful for each record. It writes the ID of every record that was processed successfully to the debug log, or writes error messages and failed fields of the failed records.

```
// Query the accounts to lock
Account[] accts = [SELECT Id from Account WHERE Name LIKE 'Acme%'];
// Lock the accounts
Approval.LockResult[] lrList = Approval.lock(accts, false);

// Iterate through each returned result
for(Approval.LockResult lr : lrList) {
    if (lr.isSuccess()) {
        // Operation was successful, so get the ID of the record that was processed
        System.debug('Successfully locked account with ID: ' + lr.getId());
    }
    else {
        // Operation failed, so get all errors
        for(Database.Error err : lr.getErrors()) {
            System.debug('The following error has occurred.');
            System.debug(err.getStatusCode() + ': ' + err.getMessage());
            System.debug('Account fields that affected this error: ' + err.getFields());
        }
    }
}
```

IN THIS SECTION:

LockResult Methods

SEE ALSO:

Approval Class

LockResult Methods

The following are methods for `LockResult`.

IN THIS SECTION:

getErrors()

If an error occurred, returns an array of one or more database error objects, providing the error code and description.

getId()

Returns the ID of the sObject you are trying to lock.

isSuccess()

A Boolean value that is set to `true` if the lock operation is successful for this object, or `false` otherwise.

`getErrors()`

If an error occurred, returns an array of one or more database error objects, providing the error code and description.

Signature

```
public List<Database.Error> getErrors()
```

Return Value

Type: List<Database.Error>

`getId()`

Returns the ID of the sObject you are trying to lock.

Signature

```
public Id getId()
```

Return Value

Type: Id

Usage

If the field contains a value, the object was locked. If the field is empty, the operation was not successful.

`isSuccess()`

A Boolean value that is set to `true` if the lock operation is successful for this object, or `false` otherwise.

Signature

```
public Boolean isSuccess()
```

Return Value

Type: Boolean

ProcessRequest Class

The `ProcessRequest` class is the parent class for the `ProcessSubmitRequest` and `ProcessWorkitemRequest` classes. Use the `ProcessRequest` class to write generic Apex that can process objects from either class.

Namespace

Approval

Usage

The request must be instantiated via the child classes, `ProcessSubmitRequest` and `ProcessWorkItemRequest`.

ProcessRequest Methods

The following are methods for `ProcessRequest`. All are instance methods.

IN THIS SECTION:

getComments()
Returns the comments that have been added previously to the approval request.

getNextApproverIds()
Returns the list of user IDs of user specified as approvers.

setComments(comments)
Sets the comments to be added to the approval request.

setNextApproverIds(nextApproverIds)
If the next step in your approval process is another Apex approval process, you specify exactly one user ID as the next approver. If not, you cannot specify a user ID and this method must be `null`. This method sets the ActorId field of the associated ProcessInstanceWorkItem.

`getComments()`

Returns the comments that have been added previously to the approval request.

Signature

```
public String getComments()
```

Return Value

Type: String

`getNextApproverIds()`

Returns the list of user IDs of user specified as approvers.

Signature

```
public ID[] getNextApproverIds()
```

Return Value

Type: ID[]

`setComments(comments)`

Sets the comments to be added to the approval request.

Signature

```
public Void setComments(String comments)
```

Parameters

comments
Type: String

Return Value

Type: Void

setNextApproverIds(nextApproverIds)

If the next step in your approval process is another Apex approval process, you specify exactly one user ID as the next approver. If not, you cannot specify a user ID and this method must be `null`. This method sets the ActorId field of the associated ProcessInstanceWorkItem.

Signature

```
public Void setNextApproverIds(ID[] nextApproverIds)
```

Parameters

nextApproverIds
Type: ID[]

Must be a single-entry list.

Return Value

Type: Void

ProcessResult Class

After you submit a record for approval, use the `ProcessResult` class to process the results of an approval process.

Namespace

Approval

Usage

A ProcessResult object is returned by the `process` method. You must specify the Approval namespace when creating an instance of this class. For example:

```
Approval.ProcessResult result = Approval.process(req1);
```

ProcessResult Methods

The following are methods for `ProcessResult`. All are instance methods.

IN THIS SECTION:

getEntityId()
The ID of the record being processed.

getErrors()
If an error occurred, returns an array of one or more database error objects including the error code and description.

getInstanceId()
The ID of the approval process that has been submitted for approval.

getInstanceStatus()
The status of the current approval process. Valid values are: Approved, Rejected, Removed or Pending.

getNewWorkitemIds()
The IDs of the new items submitted to the approval process. There can be 0 or 1 approval processes.

isSuccess()
A Boolean value that is set to `true` if the approval process completed successfully; otherwise, it is set to `false`.

`getEntityId()`

The ID of the record being processed.

Signature

```
public String getEntityId()
```

Return Value

Type: String

`getErrors()`

If an error occurred, returns an array of one or more database error objects including the error code and description.

Signature

```
public Database.Error[] getErrors()
```

Return Value

Type: Database.Error[]

`getInstanceId()`

The ID of the approval process that has been submitted for approval.

Signature

```
public String getInstanceId()
```

Return Value

Type: String

`getInstanceStatus()`

The status of the current approval process. Valid values are: Approved, Rejected, Removed or Pending.

Signature

```
public String getInstanceStatus()
```

Return Value

Type: String

`getNewWorkitemIds()`

The IDs of the new items submitted to the approval process. There can be 0 or 1 approval processes.

Signature

```
public ID[] getNewWorkitemIds()
```

Return Value

Type: ID[]

`isSuccess()`

A Boolean value that is set to `true` if the approval process completed successfully; otherwise, it is set to `false`.

Signature

```
public Boolean isSuccess()
```

Return Value

Type: Boolean

ProcessSubmitRequest Class

Use the `ProcessSubmitRequest` class to submit a record for approval.

Namespace

Approval

Usage

You must specify the Approval namespace when creating an instance of this class. The constructor for this class takes no arguments. For example:

```
Approval.ProcessSubmitRequest psr = new Approval.ProcessSubmitRequest();
```

Inherited Methods

In addition to the methods listed, the `ProcessSubmitRequest` class has access to all the methods in its parent class, ProcessRequest Class.

- getComments()
- getNextApproverIds()
- setComments(comments)
- setNextApproverIds(nextApproverIds)

Example

To view sample code, refer to Approval Processing Example.

ProcessSubmitRequest Methods

The following are methods for `ProcessSubmitRequest`. All are instance methods.

IN THIS SECTION:

getObjectId()

Returns the ID of the record that has been submitted for approval. For example, it can return an account, contact, or custom object record.

getProcessDefinitionNameOrId()

Returns the developer name or ID of the process definition.

getSkipEntryCriteria()

If `getProcessDefinitionNameOrId()` returns a value other than `null`, `getSkipEntryCriteria()` determines whether to evaluate the entry criteria for the process (`true`) or not (`false`).

getSubmitterId()

Returns the user ID of the submitter requesting the approval record. The user must be one of the allowed submitters in the process definition setup.

setObjectId(recordId)

Sets the ID of the record to be submitted for approval. For example, it can specify an account, contact, or custom object record.

setProcessDefinitionNameOrId(nameOrId)

Sets the developer name or ID of the process definition to be evaluated.

setSkipEntryCriteria(skipEntryCriteria)

If the process definition name or ID is not null, `setSkipEntryCriteria()` determines whether to evaluate the entry criteria for the process (`true`) or not (`false`).

setSubmitterId(userID)

Sets the user ID of the submitter requesting the approval record. The user must be one of the allowed submitters in the process definition setup. If you don't set a submitter ID, the process uses the current user as the submitter.

getObjectId()

Returns the ID of the record that has been submitted for approval. For example, it can return an account, contact, or custom object record.

Signature

```
public String getObjectId()
```

Return Value

Type: String

getProcessDefinitionNameOrId()

Returns the developer name or ID of the process definition.

Signature

```
public String getProcessDefinitionNameOrId()
```

Return Value

Type: String

Usage

The default is null. If the return value is `null`, when a user submits a record for approval Salesforce evaluates the entry criteria for all processes applicable to the user.

getSkipEntryCriteria()

If `getProcessDefinitionNameOrId()` returns a value other than `null`, `getSkipEntryCriteria()` determines whether to evaluate the entry criteria for the process (`true`) or not (`false`).

Signature

```
public Boolean getSkipEntryCriteria()
```

Return Value

Type: Boolean

getSubmitterId()

Returns the user ID of the submitter requesting the approval record. The user must be one of the allowed submitters in the process definition setup.

Signature

```
public String getSubmitterId()
```

Return Value

Type: String

setObjectId(recordId)

Sets the ID of the record to be submitted for approval. For example, it can specify an account, contact, or custom object record.

Signature

```
public Void setObjectId(String recordId)
```

Parameters

recordId
Type: String

Return Value

Type: Void

setProcessDefinitionNameOrId(nameOrId)

Sets the developer name or ID of the process definition to be evaluated.

Signature

```
public Void setProcessDefinitionNameOrId(String nameOrId)
```

Parameters

nameOrId
Type: String

The process definition developer name or process definition ID. The record is submitted to this specific process. If set to `null`, submission of a record approval follows standard evaluation; that is, every entry criteria of the process definition in the process order is evaluated and the one that satisfies is picked and submitted.

Return Value

Type: Void

Usage

If the process definition name or ID is not set via this method, then by default it is null. If it is null, the submission of a record for approval evaluates entry criteria for all processes applicable to the submitter. The order of evaluation is based on the process order of the setup.

`setSkipEntryCriteria(skipEntryCriteria)`

If the process definition name or ID is not null, `setSkipEntryCriteria()` determines whether to evaluate the entry criteria for the process (`true`) or not (`false`).

Signature

```
public Void setSkipEntryCriteria(Boolean skipEntryCriteria)
```

Parameters

`skipEntryCriteria`
Type: Boolean

If set to `true`, request submission skips the evaluation of entry criteria for the process set in setProcessDefinitionNameOrId(nameOrId). If the process definition name or ID is not specified, this parameter is ignored and standard evaluation is followed based on process order. If set to `false`, or if this method isn't called, the entry criteria is not skipped.

Return Value

Type: Void

`setSubmitterId(userID)`

Sets the user ID of the submitter requesting the approval record. The user must be one of the allowed submitters in the process definition setup. If you don't set a submitter ID, the process uses the current user as the submitter.

Signature

```
public Void setSubmitterId(String userID)
```

Parameters

`userID`
Type: String

The user ID on behalf of which the record is submitted. If set to `null`, the current user is the submitter. If the submitter is not set with this method, the default submitter is null (the current user).

Return Value

Type: Void

ProcessWorkitemRequest Class

Use the `ProcessWorkitemRequest` class for processing an approval request after it is submitted.

Namespace

Approval

Usage

You must specify the Approval namespace when creating an instance of this class. The constructor for this class takes no arguments. For example:

```
Approval.ProcessWorkitemRequest pwr = new Approval.ProcessWorkitemRequest();
```

Inherited Methods

In addition to the methods listed, the `ProcessWorkitemRequest` class has access to all the methods in its parent class, ProcessRequest Class:

- getComments()
- getNextApproverIds()
- setComments(comments)
- setNextApproverIds(nextApproverIds)

ProcessWorkitemRequest Methods

The following are methods for `ProcessWorkitemRequest`. All are instance methods.

IN THIS SECTION:

getAction()
Returns the type of action already associated with the approval request. Valid values are: Approve, Reject, or Removed.

getWorkitemId()
Returns the ID of the approval request that is in the process of being approved, rejected, or removed.

setAction(actionType)
Sets the type of action to take for processing an approval request.

setWorkitemId(id)
Sets the ID of the approval request that is being approved, rejected, or removed.

`getAction()`

Returns the type of action already associated with the approval request. Valid values are: Approve, Reject, or Removed.

Signature

```
public String getAction()
```

Return Value

Type: String

`getWorkitemId()`

Returns the ID of the approval request that is in the process of being approved, rejected, or removed.

Signature

```
public String getWorkitemId()
```

Return Value

Type: String

`setAction(actionType)`

Sets the type of action to take for processing an approval request.

Signature

```
public Void setAction(String actionType)
```

Parameters

`actionType`
Type: String

Valid values are: Approve, Reject, or Removed. Only system administrators can specify Removed.

Return Value

Type: Void

`setWorkitemId(id)`

Sets the ID of the approval request that is being approved, rejected, or removed.

Signature

```
public Void setWorkitemId(String id)
```

Parameters

`id`
Type: String

Return Value

Type: Void

UnlockResult Class

The result of a record unlock, returned by a `System.Approval.unlock()` method.

Namespace

Approval

Usage

The `System.Approval.unlock()` methods return Approval.UnlockResult objects. Each element in an UnlockResult array corresponds to an element in the ID or sObject array passed as a parameter to an `unlock` method. The first element in the UnlockResult array corresponds to the first element in the ID or sObject array, the second element corresponds to the second element, and so on. If only one ID or sObject is passed in, the UnlockResult array contains a single element.

Example

The following example shows how to obtain and iterate through the returned Approval.UnlockResult objects. It locks some queried accounts using `Approval.unlock` with a `false` second parameter to allow partial processing of records on failure. Next, it iterates through the results to determine whether the operation was successful for each record. It writes the ID of every record that was processed successfully to the debug log, or writes error messages and failed fields of the failed records.

```
// Query the accounts to unlock
Account[] accts = [SELECT Id from Account WHERE Name LIKE 'Acme%'];

for(Account acct:accts) {
   // Create an approval request for the account
   Approval.ProcessSubmitRequest req1 =
           new Approval.ProcessSubmitRequest();
   req1.setComments('Submitting request for approval.');
   req1.setObjectId(acct.id);

   // Submit the record to specific process and skip the criteria evaluation
   req1.setProcessDefinitionNameOrId('PTO_Request_Process');
   req1.setSkipEntryCriteria(true);

   // Submit the approval request for the account
   Approval.ProcessResult result = Approval.process(req1);

   // Verify the result
   System.assert(result.isSuccess());
}

// Unlock the accounts
Approval.UnlockResult[] urList = Approval.unlock(accts, false);

// Iterate through each returned result
for(Approval.UnlockResult ur : urList) {
    if (ur.isSuccess()) {
        // Operation was successful, so get the ID of the record that was processed
        System.debug('Successfully unlocked account with ID: ' + ur.getId());
    }
    else {
        // Operation failed, so get all errors
        for(Database.Error err : ur.getErrors()) {
            System.debug('The following error has occurred.');
            System.debug(err.getStatusCode() + ': ' + err.getMessage());
```

```
            System.debug('Account fields that affected this error: ' + err.getFields());
        }
    }
}
```

IN THIS SECTION:

UnlockResult Methods

SEE ALSO:

Approval Class

UnlockResult Methods

The following are methods for `UnlockResult`.

IN THIS SECTION:

getErrors()
If an error occurred, returns an array of one or more database error objects, providing the error code and description.

getId()
Returns the ID of the sObject you are trying to unlock.

isSuccess()
A Boolean value that is set to `true` if the unlock operation is successful for this object, or `false` otherwise.

`getErrors()`

If an error occurred, returns an array of one or more database error objects, providing the error code and description.

Signature

```
public List<Database.Error> getErrors()
```

Return Value

Type: List<Database.Error>

`getId()`

Returns the ID of the sObject you are trying to unlock.

Signature

```
public Id getId()
```

Return Value

Type: Id

Usage

If the field contains a value, the object was unlocked. If the field is empty, the operation was not successfult.

`isSuccess()`

A Boolean value that is set to `true` if the unlock operation is successful for this object, or `false` otherwise.

Signature

```
public Boolean isSuccess()
```

Return Value

Type: Boolean

Auth Namespace

The `Auth` namespace provides an interface and classes for single sign-on into Salesforce and session security management.

The following is the interface in the `Auth` namespace.

IN THIS SECTION:

AuthConfiguration Class

Contains methods for configuring settings for users to log in to a Salesforce org using their authentication provider credentials instead of their Salesforce credentials. The authentication provider can be any authentication provider that supports the OpenID Connect protocol, such as Google, Facebook, or Twitter. Users log in to either an Experience Cloud site (`https://`*`MyDomainName`*`.my.site.com`) or your My Domain login URL (`https://`*`MyDomainName`*`.my.salesforce.com`).

AuthProviderCallbackState Class

Provides request HTTP headers, body, and query parameters to the `AuthProviderPlugin.handleCallback` method for user authentication. This class allows you to group the information passed in rather than passing headers, body, and query parameters individually.

AuthProviderPlugin Interface

This interface is deprecated. For new development, use the abstract class `Auth.AuthProviderPluginClass` to create a custom OAuth-based authentication provider plug-in for single sign-on in to Salesforce.

AuthProviderPluginClass Class

Contains methods to create a custom OAuth-based authentication provider plug-in for single sign-on in to Salesforce. Use this class to create a custom authentication provider plug-in if you can't use one of the authentication providers that Salesforce provides.

AuthProviderTokenResponse Class

Stores the response from the `AuthProviderPlugin.handleCallback` method.

AuthToken Class

Contains methods for getting and revoking access and refresh tokens that are issued when a user logs in via a single sign-on (SSO) flow that uses an authentication provider, such as Facebook.

CommunitiesUtil Class

Contains methods for getting information about an Experience Cloud user.

ConfigurableSelfRegHandler Interface

Gives you more control over how customers or partners self-register for your Experience Cloud site by creating a class that implements `Auth.ConfigurableSelfRegHandler`. You choose the user information to collect, and how users identify themselves—with their email address, phone number, or another identifier. When verified, you create a customer or partner user and log in the user to your Experience Cloud site.

ConfirmUserRegistrationHandler Interface

Manages single sign-on (SSO) user mappings between Salesforce and a third-party identity provider. Use this interface to confirm user mappings before updating them.

ConnectedAppPlugin Class

Contains methods for extending the behavior of a connected app, for example, customizing how a connected app is invoked depending on the protocol used. This class gives you more control over the interaction between Salesforce and your connected app.

CustomOneTimePasswordDeliveryHandler Interface

To use a custom SMS provider to send one-time passwords (OTPs) for Experience Cloud identity verification, create a class that implements the `Auth.CustomOneTimePasswordDeliveryHandler` interface.

CustomOneTimePasswordDeliveryResult Enum

Indicates the status of an attempt to send a one-time password (OTP) to an external user via a custom messaging provider.

ExternalClientAppOauthHandler Class

Contains methods for extending the behavior of an external client app. For example, customize how an external client app is invoked depending on the protocol used. This class gives you more control over the interaction between Salesforce and your external client app.

HeadlessSelfRegistrationHandler Interface

Creates customer and partner users during the Headless Registration Flow.

HeadlessUserDiscoveryHandler Interface

Use this interface to create a headless user discovery handler that you implement during headless login, passwordless login, and forgot password flows.

HeadlessUserDiscoveryResponse Class

Contains methods to describe the result of headless user discovery using a handler that implements the `Auth.HeadlessUserDiscoveryHandler` interface during headless login, passwordless login, and forgot password flows.

HttpCalloutMockUtil Class

Contains a method to send fake HTTP callouts for classes in the `Auth` namespace.

IntegratingAppType Enum

Specifies whether you're integrating your app as a connected app or as an external client app in methods used in your customized Apex token exchange handler, which extends the `Auth.Oauth2TokenExchangeHandler` class.

InvocationContext Enum

The context in which the connected app is invoked, such as the protocol flow used and the token type issued, if any. Developers can use the context information to write code that is unique to the type of invocation.

JWS Class

Contains methods that apply a digital signature to a JSON Web Token (JWT), using a JSON Web Signature (JWS) data structure. This class creates the signed JWT bearer token, which can be used to request an OAuth access token in the OAuth 2.0 JWT bearer token flow.

JWT Class

Generates the JSON Claims Set in a JSON Web Token (JWT). The resulting Base64-encoded payload can be passed as an argument to create an instance of the `Auth.JWS` class.

JWTBearerTokenExchange Class

Contains methods that POST the signed JWT bearer token to a token endpoint to request an access token, in the OAuth 2.0 JWT bearer token flow.

JWTUtil Class

Contains methods for validating a JSON Web Token (JWT) from an external identity provider as part of the OAuth 2.0 token exchange flow. Use these methods as part of the `validateIncomingToken` method in the `Auth.Oauth2TokenExchangeHandler` class.

LightningLoginEligibility Enum

Contains a Lightning Login eligibility value used by the `Auth.SessionManagement.getLightningLoginEligibility` method.

LoginDiscoveryHandler Interface

Salesforce gives you the ability to log in users based on other verification methods than username and password. For example, it can prompt users to log in with their email, phone number, or another identifier like a Federation ID or device identifier. Login Discovery is available to these licenses: Customer Community, Customer Community Plus, External Identity, Partner Community, and Partner Community Plus.

LoginDiscoveryMethod Enum

Contains methods used to verify the user's identity when the My Domain login process uses Login Discovery.

MyDomainLoginDiscoveryHandler Interface

The handler used to implement the My Domain Login Discovery page, which is an interview-based (two-step) login process. First the user is prompted for a unique identifier such as an email address or phone number. Then the handler determines (discovers) how to authenticate the user. Either the user enters a password or is directed to an identity provider's login page.

Oauth2TokenExchangeHandler Class

Use this class to create a token exchange handler that validates tokens from an external identity provider and maps the token's subject to a Salesforce user during the OAuth 2.0 token exchange flow. The handler can also be used to create users by setting up a new User object and returning it to Salesforce for automatic insertion.

OAuth2TokenExchangeType Enum

Used during the OAuth 2.0 token exchange flow to specify the type of token that's being exchanged for a Salesforce token.

OAuthRefreshResult Class

Stores the result of an `AuthProviderPluginClass` refresh method. OAuth authentication flow provides a refresh token that can be used to get a new access token. Access tokens have a limited lifetime as specified by the session timeout value. When an access token expires, use a refresh token to get a new access token.

OauthToken Class

Contains a method to revoke OAuth access tokens and refresh tokens. This method supports opaque tokens and JSON Web Token (JWT)-based access tokens, including guest and named user JWT-based access tokens.

OauthTokenType Enum

Specifies the type of Salesforce-issued OAuth 2.0 token being revoked in the `OauthToken.revokeToken` method.

RegistrationHandler Interface

Salesforce provides the ability to use an authentication provider, such as Facebook© or Janrain©, for single sign-on into Salesforce.

SamlJitHandler Interface

Use this interface to control and customize Just-in-Time user provisioning logic during SAML single sign-on.

SessionManagement Class

Contains methods for verifying users' identity, creating custom login flows, customizing security levels, and defining trusted IP ranges for a current session.

SessionLevel Enum

An `Auth.SessionLevel` enum value is used by the `SessionManagement.setSessionLevel` method.

TokenValidationResult Class

Contains methods that describe the result of the token validation performed by a token exchange handler using the `validateIncomingToken` method in the `Auth.Oauth2TokenExchangeHandler` class during the OAuth 2.0 token exchange flow.

UserData Class

Stores user information for `Auth.RegistrationHandler`.

VerificationAction Enum

Indicates the method that you use to send a one-time password (OTP) to a user during the headless passwordless login flow.

VerificationMethod Enum

Contains the different ways users can identify themselves when logging in. You can use it to implement mobile-centric passwordless login pages and to self-register (and deregister) verification methods.

VerificationPolicy Enum

The `Auth.VerificationPolicy` enum contains an identity verification policy value used by the `SessionManagement.generateVerificationUrl` method.

VerificationResult Class

Contains the result of a verification challenge that you invoke when you create your own Verify page. The challenge can be initiated by either the `System.UserManagement.verifyPasswordlessLogin` or `System.UserManagement.verifySelfRegistration` method.

Auth Exceptions

The `Auth` namespace contains some exception classes.

AuthConfiguration Class

Contains methods for configuring settings for users to log in to a Salesforce org using their authentication provider credentials instead of their Salesforce credentials. The authentication provider can be any authentication provider that supports the OpenID Connect protocol, such as Google, Facebook, or Twitter. Users log in to either an Experience Cloud site (`https://`*`MyDomainName`*`.my.site.com`) or your My Domain login URL (`https://`*`MyDomainName`*`.my.salesforce.com`).

Namespace

Auth

Example

This example shows how to call some methods on the `Auth.AuthConfiguration` class. Before you can run this sample, you must provide valid values for the URLs and developer name.

```
String communityUrl = 'MyDomainName.my.site.com';
String startUrl = '<Add URL>';
Auth.AuthConfiguration authConfig = new Auth.AuthConfiguration(communityUrl,startUrl);
```

```
List<AuthProvider> authPrvs = authConfig.getAuthProviders();
String bColor = authConfig.getBackgroundColor();
String fText = authConfig.getFooterText();

String sso = Auth.AuthConfiguration.getAuthProviderSsoUrl(communityUrl, startUrl,
'developerName');
```

AuthConfiguration Constructors

The following are constructors for `AuthConfiguration`.

Note: The `AuthConfiguration (networkId, startUrl)` constructor is deprecated in API version 56.0 and later.

`AuthConfiguration(communityOrCustomUrl, startUrl)`

Creates an instance of the `AuthConfiguration` class using the specified URL for an Experience Cloud site or a My Domain subdomain and the start URL for authenticated users.

Signature

```
public AuthConfiguration(String communityOrCustomUrl, String startUrl)
```

Parameters

`communityOrCustomUrl`
Type: String

The URL for the domain, which can be a Salesforce subdomain created with My Domain (`my.salesforce.com`) or a subdomain of an Experience Cloud site (`force.com`).

`startUrl`
Type: String

The page users see after successfully logging in to the Experience Cloud site or My Domain subdomain.

AuthConfiguration Methods

The following are methods for `AuthConfiguration`. Use these methods to manage and customize authentication for a Salesforce community.

IN THIS SECTION:

getAllowInternalUserLoginEnabled()
Indicates whether the Experience Cloud site allows internal users to log in using the Experience Cloud site login page. To enable, admins configure the setting **Allow internal users to log in directly to the experience** on the Login & Registration page in Experience Workspaces. It's disabled by default.

getAuthConfig()
Returns the AuthConfig sObject, which represents the authentication options for an Experience Cloud site or Salesforce My Domain subdomain.

getAuthConfigProviders()
Returns the list of authentication providers configured for an Experience Cloud site or Salesforce My Domain subdomain.

getAuthProviders()

Returns the list of authentication providers available for an Experience Cloud site or Salesforce My Domain subdomain.

getAuthProviderSsoDomainUrl(communityUrl, startUrl, developerName)

Returns the single sign-on URL for an Experience Cloud site subdomain.

getAuthProviderSsoUrl(communityUrl, startUrl, developerName)

Returns the single sign-on URL for an Experience Cloud site or Salesforce My Domain subdomain.

getBackgroundColor()

Returns the color for the background of the login page for a community.

getCertificateLoginEnabled(domainUrl)

Returns true if certificate-based authentication is enabled for the My Domain URL.

getCertificateLoginUrl(domainUrl, startUrl)

Returns the certificate-based authentication endpoint for the My Domain URL if the org has certificate-based authentication enabled.

getDefaultProfileForRegistration()

Returns the profile ID assigned to new community users.

getFooterText()

Returns the text at the bottom of the login page for a community.

getForgotPasswordUrl()

Returns the URL for the standard or custom Forgot Password page that is specified for an Experience Cloud site or portal by the administrator.

getHeadlessForgotPasswordEnabled()

Returns `true` if the Headless Forgot Password Flow is enabled.

getHeadlessFrgtPswEnabled()

This method will be deprecated in a future release. Use the `getHeadlessForgotPasswordEnabled()` method in this class instead.

getHeadlessPasswordlessLoginEnabled()

Determines if headless passwordless login is enabled.

getHeadlessRegistrationEnabled()

Determines if the Headless Registration Flow is enabled.

getLogoUrl()

Returns the location of the icon image at the bottom of the login page for a community.

getRightFrameUrl()

Returns the URL for the right-frame content to display on the right side of the Experience Cloud site login page. The admin supplies the URL.

getSamlProviders()

Returns the list of SAML-based authentication providers available for an Experience Cloud site or Salesforce My Domain subdomain.

getSamlSsoUrl(communityUrl, startURL, samlId)

Returns the single sign-on URL for an Experience Cloud site or Salesforce My Domain subdomain.

getSelfRegistrationEnabled()

Indicates whether the current community allows new users to create their own account by filling out a registration form.

getSelfRegistrationUrl()

Returns the location of the self-registration page for new users to sign up for an account with a community.

getStartUrl()

Returns the start page of an Experience Cloud site or Salesforce My Domain subdomain. This URL is the first page that users see when they log in.

getUsernamePasswordEnabled()

Indicates whether the current community is set to display a login form asking for a username and password. You can configure the community not to request a username and password if it is for unauthenticated users or users logging in with a third-party authentication provider.

isCommunityUsingSiteAsContainer()

Returns `true` if the Experience Cloud site uses Site.com pages; otherwise, returns `false`.

`getAllowInternalUserLoginEnabled()`

Indicates whether the Experience Cloud site allows internal users to log in using the Experience Cloud site login page. To enable, admins configure the setting **Allow internal users to log in directly to the experience** on the Login & Registration page in Experience Workspaces. It's disabled by default.

Signature

```
public Boolean getAllowInternalUserLoginEnabled()
```

Return Value

Type: Boolean

Usage

If true, internal users log in to an Experience Cloud site from the site's login page with their internal credentials. If they navigate to their internal org from the Experience Cloud site, they don't have to log in again.

`getAuthConfig()`

Returns the AuthConfig sObject, which represents the authentication options for an Experience Cloud site or Salesforce My Domain subdomain.

Signature

```
public AuthConfig getAuthConfig()
```

Return Value

Type: AuthConfig

The AuthConfig sObject for the Experience Cloud site or Salesforce My Domain subdomain.

`getAuthConfigProviders()`

Returns the list of authentication providers configured for an Experience Cloud site or Salesforce My Domain subdomain.

Signature

```
public List<AuthConfigProviders> getAuthConfigProviders()
```

Return Value

Type: List<AuthConfigProviders>

A list of authentication providers (AuthConfigProviders sObjects), which are children of the AuthProvider sObject.

`getAuthProviders()`

Returns the list of authentication providers available for an Experience Cloud site or Salesforce My Domain subdomain.

Signature

```
public List<AuthProvider> getAuthProviders()
```

Return Value

Type: List<AuthProvider>

A list of authentication providers (AuthProvider sObjects) for the Experience Cloud site or My Domain subdomain.

`getAuthProviderSsoDomainUrl(communityUrl, startUrl, developerName)`

Returns the single sign-on URL for an Experience Cloud site subdomain.

> **Note:** For better performance, we recommend using this method instead of `getAuthProviderSsoUrl`. If the authentication provider has `User Subdomain for Callback` enabled, changing the single sign-on URL also changes the callback URL to use the Experience Cloud site subdomain. Before switching to this method, update the callback URL in your third-party applications to avoid getting an invalid callback URL error during single sign-on.

Signature

```
public static String getAuthProviderSsoDomainUrl(String communityUrl, String startUrl,
String developerName)
```

Parameters

`communityUrl`
Type: String

The URL for the Experience Cloud site subdomain. If null or specified as an empty string, you get the single sign-on URL for the org's My Domain.

`startUrl`
Type: String

The page that users see after logging in to the Experience Cloud site subdomain.

`developerName`
Type: String

The unique name of the authentication provider.

Return Value

Type: String

The Single Sign-On Initialization URL for the Experience Cloud site subdomain.

getAuthProviderSsoUrl(communityUrl, startUrl, developerName)

Returns the single sign-on URL for an Experience Cloud site or Salesforce My Domain subdomain.

Signature

```
public static String getAuthProviderSsoUrl(String communityUrl, String startUrl, String developerName)
```

Parameters

communityUrl
Type: String

The URL for the Experience Cloud site or My Domain subdomain. If not null and not specified as an empty string, you get the URL for the Experience Cloud site. If null or specified as an empty string, you get the URL for a custom domain.

startUrl
Type: String

The page that users see after logging in to the Experience Cloud site or My Domain subdomain.

developerName
Type: String

The unique name of the authentication provider.

Return Value

Type: String

The Single Sign-On Initialization URL for the Experience Cloud site or Salesforce My Domain subdomain.

getBackgroundColor()

Returns the color for the background of the login page for a community.

Signature

```
public String getBackgroundColor()
```

Return Value

Type: String

getCertificateLoginEnabled(domainUrl)

Returns true if certificate-based authentication is enabled for the My Domain URL.

Signature

```
public Boolean getCertificateLoginEnabled(String domainUrl)
```

Parameters

domainUrl
Type: String

The My Domain URL that is being checked for certificate-based authentication.

Return Value

Type: Boolean

getCertificateLoginUrl(domainUrl, startUrl)

Returns the certificate-based authentication endpoint for the My Domain URL if the org has certificate-based authentication enabled.

Signature

```
public static String getCertificateLoginUrl(String domainUrl, String startUrl)
```

Parameters

domainUrl
Type: String

The My Domain URL being checked for its certificate-based authentication endpoint .

startUrl
Type: String

The page that the user is directed to after logging in to the My Domain with certificate-based authentication.

Return Value

Type: String

The certificate-based authentication endpoint for the My Domain URL:
mydomainURL`:8443/services/certauth?startURL=`***startURLParam***

getDefaultProfileForRegistration()

Returns the profile ID assigned to new community users.

Signature

```
public String getDefaultProfileForRegistration()
```

Return Value

Type: String

The profile ID.

`getFooterText()`

Returns the text at the bottom of the login page for a community.

Signature

```
public String getFooterText()
```

Return Value

Type: String

The text string displayed at the bottom of the login page, for example "Log in with an existing account."

`getForgotPasswordUrl()`

Returns the URL for the standard or custom Forgot Password page that is specified for an Experience Cloud site or portal by the administrator.

Signature

```
public String getForgotPasswordUrl()
```

Return Value

Type: String

URL for the standard or custom Forgot Password page.

`getHeadlessForgotPasswordEnabled()`

Returns `true` if the Headless Forgot Password Flow is enabled.

Signature

```
public Boolean getHeadlessForgotPasswordEnabled()
```

Return Value

Type: Boolean

`getHeadlessFrgtPswEnabled()`

This method will be deprecated in a future release. Use the `getHeadlessForgotPasswordEnabled()` method in this class instead.

Signature

```
public Boolean getHeadlessFrgtPswEnabled()
```

Return Value

Type: Boolean

getHeadlessPasswordlessLoginEnabled()

Determines if headless passwordless login is enabled.

Signature

```
public Boolean getHeadlessPasswordlessLoginEnabled()
```

Return Value

Type: Boolean

Returns `true` if headless passwordless login is enabled.

getHeadlessRegistrationEnabled()

Determines if the Headless Registration Flow is enabled.

Signature

```
public Boolean getHeadlessRegistrationEnabled()
```

Return Value

Type: Boolean

Returns `true` if headless registration is enabled.

getLogoUrl()

Returns the location of the icon image at the bottom of the login page for a community.

Signature

```
public String getLogoUrl()
```

Return Value

Type: String

The path to the icon image.

getRightFrameUrl()

Returns the URL for the right-frame content to display on the right side of the Experience Cloud site login page. The admin supplies the URL.

Signature

```
public String getLoginRightFrameUrl()
```

Return Value

Type: String

URL for the right-frame content of the Experience Cloud site login page. Salesforce creates an inline (iframe) on the right side of the login page to display the contents specified by the URL.

getSamlProviders()

Returns the list of SAML-based authentication providers available for an Experience Cloud site or Salesforce My Domain subdomain.

Signature

```
public List<SamlSsoConfig> getSamlProviders()
```

Return Value

Type: List<SamlSsoConfig>

A list of SAML-based authentication providers, which are SamlSsoConfig sObjects.

getSamlSsoUrl(communityUrl, startURL, samlId)

Returns the single sign-on URL for an Experience Cloud site or Salesforce My Domain subdomain.

Signature

```
public static String getSamlSsoUrl(String communityUrl, String startURL, String samlId)
```

Parameters

communityUrl
Type: String

The URL for the Experience Cloud site or My Domain subdomain. If not `null` and not specified as an empty string, you get the URL for the Experience Cloud site. If `null` or specified as an empty string, you get the URL for a My Domain subdomain.

startUrl
Type: String

The page users see after successfully logging in to the Experience Cloud site or My Domain subdomain

samlId
Type: String

The unique identifier of the SamlSsoConfig standard object for the Experience Cloud site or My Domain subdomain

Return Value

Type: String

The Single Sign-On Initialization URL for the Experience Cloud site or Salesforce My Domain subdomain.

getSelfRegistrationEnabled()

Indicates whether the current community allows new users to create their own account by filling out a registration form.

Signature

```
public Boolean getSelfRegistrationEnabled()
```

Return Value

Type: Boolean

`getSelfRegistrationUrl()`

Returns the location of the self-registration page for new users to sign up for an account with a community.

Signature

```
public String getSelfRegistrationUrl()
```

Return Value

Type: String

The location of the self-registration page.

`getStartUrl()`

Returns the start page of an Experience Cloud site or Salesforce My Domain subdomain. This URL is the first page that users see when they log in.

Signature

```
public String getStartUrl()
```

Return Value

Type: String

The location of the start page for the Experience Cloud site or My Domain subdomain.

`getUsernamePasswordEnabled()`

Indicates whether the current community is set to display a login form asking for a username and password. You can configure the community not to request a username and password if it is for unauthenticated users or users logging in with a third-party authentication provider.

Signature

```
public Boolean getUsernamePasswordEnabled()
```

Return Value

Type: Boolean

`isCommunityUsingSiteAsContainer()`

Returns `true` if the Experience Cloud site uses Site.com pages; otherwise, returns `false`.

Signature

```
public Boolean isCommunityUsingSiteAsContainer()
```

Return Value

Type: Boolean

AuthProviderCallbackState Class

Provides request HTTP headers, body, and query parameters to the `AuthProviderPlugin.handleCallback` method for user authentication. This class allows you to group the information passed in rather than passing headers, body, and query parameters individually.

Namespace

Auth

IN THIS SECTION:

AuthProviderCallbackState Constructors

AuthProviderCallbackState Properties

SEE ALSO:

handleCallback(authProviderConfiguration, callbackState)

AuthProviderCallbackState Constructors

The following are constructors for `AuthProviderCallbackState`.

IN THIS SECTION:

AuthProviderCallbackState(headers, body, queryParameters)

Creates an instance of the `AuthProviderCallbackState` class using the specified HTTP headers, body, and query parameters of the authentication request.

`AuthProviderCallbackState(headers, body, queryParameters)`

Creates an instance of the `AuthProviderCallbackState` class using the specified HTTP headers, body, and query parameters of the authentication request.

Signature

```
public AuthProviderCallbackState(Map<String,String> headers, String body,
Map<String,String> queryParameters)
```

Parameters

headers
Type: Map<String,String>

The HTTP headers of the authentication request.

body
Type: String

The HTTP body of the authentication request.

queryParameters
Type: Map<String,String>

The HTTP query parameters of the authentication request.

AuthProviderCallbackState Properties

The following are properties for `AuthProviderCallbackState`.

IN THIS SECTION:

body
The HTTP body of the authentication request.

headers
The HTTP headers of the authentication request.

queryParameters
The HTTP query parameters of the authentication request.

body

The HTTP body of the authentication request.

Signature

```
public String body {get; set;}
```

Property Value

Type: String

headers

The HTTP headers of the authentication request.

Signature

```
public Map<String,String> headers {get; set;}
```

Property Value

Type: Map<String,String>

`queryParameters`

The HTTP query parameters of the authentication request.

Signature

```
public Map<String,String> queryParameters {get; set;}
```

Property Value

Type: Map<String,String>

AuthProviderPlugin Interface

This interface is deprecated. For new development, use the abstract class `Auth.AuthProviderPluginClass` to create a custom OAuth-based authentication provider plug-in for single sign-on in to Salesforce.

Namespace

Auth

Usage

Deprecated. Existing implementations that use `Auth.AuthProviderPlugin` still work. For new development, use `Auth.AuthProviderPluginClass`.

IN THIS SECTION:

AuthProviderPlugin Methods

AuthProviderPlugin Example Implementation

AuthProviderPlugin Methods

The following methods are for `AuthProviderPlugin`, which, as of API version 39.0, is deprecated. Use themethods in `AuthProviderPluginClass` instead.

IN THIS SECTION:

getCustomMetadataType()
Deprecated as of API version 39.0. Use the corresponding method in `Auth.AuthProviderPluginClass`.

getUserInfo(authProviderConfiguration, response)
Deprecated as of API version 39.0. Use the corresponding method in `Auth.AuthProviderPluginClass`.

handleCallback(authProviderConfiguration, callbackState)
Deprecated as of API version 39.0. Use the corresponding method in `Auth.AuthProviderPluginClass`.

initiate(authProviderConfiguration, stateToPropagate)

Deprecated as of API version 39.0. Use the corresponding method in `Auth.AuthProviderPluginClass`.

SEE ALSO:

Salesforce Help: Create a Custom External Authentication Provider

`getCustomMetadataType()`

Deprecated as of API version 39.0. Use the corresponding method in `Auth.AuthProviderPluginClass`.

Signature

```
public String getCustomMetadataType()
```

Return Value

Type: String

The custom metadata type API name for the authentication provider.

Usage

Returns the custom metadata type API name for a custom OAuth-based authentication provider for single sign-on to Salesforce. The `getCustomMetatadaType()` method returns only custom metadata type names. It does not return custom metadata record names.

`getUserInfo(authProviderConfiguration, response)`

Deprecated as of API version 39.0. Use the corresponding method in `Auth.AuthProviderPluginClass`.

Signature

```
public Auth.UserData getUserInfo(Map<String,String> authProviderConfiguration,
Auth.AuthProviderTokenResponse response)
```

Parameters

`authProviderConfiguration`

Type: Map<String,String>

The configuration for the custom authentication provider. When you create a custom metadata type in Salesforce, the configuration populates with the custom metadata type default values. Or you can set the configuration with values you enter when you create the custom provider in Auth. Providers in Setup.

`response`

Type: Auth.AuthProviderTokenResponse

The OAuth access token, OAuth secret or refresh token, and state provided by the authentication provider to authenticate the current user.

Return Value

Type: Auth.UserData

Creates a new instance of the `Auth.UserData` class.

Usage

Returns information from the custom authentication provider about the current user. The registration handler and other authentication provider flows use this information.

handleCallback(authProviderConfiguration, callbackState)

Deprecated as of API version 39.0. Use the corresponding method in `Auth.AuthProviderPluginClass`.

Signature

```
public Auth.AuthProviderTokenResponse handleCallback(Map<String,String>
authProviderConfiguration, Auth.AuthProviderCallbackState callbackState)
```

Parameters

authProviderConfiguration
Type: Map<StringString>

The configuration for the custom authentication provider. When you create a custom metadata type in Salesforce, the configuration populates with the custom metadata type default values. Or you can set the configuration with values you enter when you create the custom provider in Auth. Providers in Setup.

callbackState
Type: Auth.AuthProviderCallbackState

The class that contains the HTTP headers, body, and queryParams of the authentication request.

Return Value

Type: Auth.AuthProviderTokenResponse

Creates an instance of the `AuthProviderTokenResponse` class.

Usage

Uses the authentication provider's supported authentication protocol to return an OAuth access token, OAuth secret or refresh token, and the state passed in when the request for the current user was initiated.

initiate(authProviderConfiguration, stateToPropagate)

Deprecated as of API version 39.0. Use the corresponding method in `Auth.AuthProviderPluginClass`.

Signature

```
public System.PageReference initiate(Map<String,String> authProviderConfiguration,
String stateToPropagate)
```

Parameters

authProviderConfiguration
Type: Map<StringString>

The configuration for the custom authentication provider. When you create a custom metadata type in Salesforce, the configuration populates with the custom metadata type default values. Or you can set the configuration with values you enter when you create the custom provider in Auth. Providers in Setup.

stateToPropagate
Type: String

The state passed in to initiate the authentication request for the user.

Return Value

Type: System.PageReference

The URL of the page where the user is redirected for authentication.

Usage

Returns the URL where the user is redirected for authentication.

AuthProviderPlugin Example Implementation

We've removed the example implementation for the `Auth.AuthProviderPlugin` interface because we've deprecated the interface and replaced it with an abstract class. See AuthProviderPluginClass Class.

AuthProviderPluginClass Class

Contains methods to create a custom OAuth-based authentication provider plug-in for single sign-on in to Salesforce. Use this class to create a custom authentication provider plug-in if you can't use one of the authentication providers that Salesforce provides.

Namespace

Auth

Usage

To create a custom authentication provider for single sign-on, create a class that extends `Auth.AuthProviderPluginClass`. This class allows you to store the custom configuration for your authentication provider and handle authentication protocols when users log in to Salesforce with their login credentials for an external service provider. In Salesforce, the class that implements this interface appears in the `Provider Type` drop-down list in Auth. Providers in Setup. Make sure that the user you specify to run the class has "Customize Application" and "Manage Auth. Providers" permissions.

As of API version 39.0, use the abstract class `AuthProviderPluginClass` to create a custom external authentication provider. This class replaces the `AuthProviderPlugin` interface. If you've already implemented a custom authentication provider plug-in using the interface, it still works. However, use `AuthProviderPluginClass` to extend your plug-in. If you haven't created an interface, create a custom authentication provider plug-in by extending this abstract class. For more information, see AuthProviderPluginClass Code Example.

IN THIS SECTION:

AuthProviderPluginClass Methods

AuthProviderPluginClass Code Example

AuthProviderPluginClass Methods

The `AuthProviderPluginClass` methods don't support DML options.

IN THIS SECTION:

getCustomMetadataType()

Returns the custom metadata type API name for a custom OAuth-based authentication provider for single sign-on to Salesforce.

getUserInfo(authProviderConfiguration, response)

Returns information from the custom authentication provider about the current user. This information is used by the registration handler and in other authentication provider flows.

handleCallback(authProviderConfiguration, callbackState)

Uses the authentication provider's supported authentication protocol to return an OAuth access token, OAuth secret or refresh token, and the state passed in when the request for the current user was initiated.

initiate(authProviderConfiguration, stateToPropagate)

Returns the URL where the user is redirected for authentication.

refresh(authProviderConfiguration, refreshToken)

Returns a new access token, which is used to update an expired access token.

`getCustomMetadataType()`

Returns the custom metadata type API name for a custom OAuth-based authentication provider for single sign-on to Salesforce.

Signature

```
public String getCustomMetadataType()
```

Return Value

Type: String

The custom metadata type API name for the authentication provider.

Usage

The `getCustomMetadataType()` method returns only custom metadata type names. It does not return custom metadata record names. As of API version 39.0, use this method when extending `Auth.AuthProviderPluginClass` to create a custom external authentication provider.

`getUserInfo(authProviderConfiguration, response)`

Returns information from the custom authentication provider about the current user. This information is used by the registration handler and in other authentication provider flows.

Signature

```
public Auth.UserData getUserInfo(Map<String,String> authProviderConfiguration,
Auth.AuthProviderTokenResponse response)
```

Parameters

authProviderConfiguration

Type: Map<String,String>

The configuration for the custom authentication provider. When you create a custom metadata type in Salesforce, the configuration populates it with the custom metadata type default values. Or you can set the configuration with values that you enter when you create the custom provider in Auth. Providers in Setup.

response

Type: Auth.AuthProviderTokenResponse

The OAuth access token, OAuth secret or refresh token, and state provided by the authentication provider to authenticate the current user.

Return Value

Type: Auth.UserData

Creates a new instance of the `Auth.UserData` class.

Usage

As of API version 39.0, use this method when extending `Auth.AuthProviderPluginClass` to create a custom authentication provider.

Note: You might choose to get user information in the response from the `handleCallback` method or by another method. However, you must still call `getUserInfo` in the custom authentication handler to avoid getting errors about mixing objects. For example, if you don't call `getUserInfo`, and then try to insert a contact in the `Auth.RegistrationHandler.createUser` method, you get the error, "You cannot mix EntityObjects with different UddInfos within one transaction."

To avoid this error, call `getUserInfo` with dummy user information as follows.

```
HttpRequest req = new HttpRequest();
String url = 'https://login.salesforce.com/';
req.setEndpoint(url);
req.setMethod('GET');
Http http = new Http();
HTTPResponse res = http.send(req);
```

handleCallback(authProviderConfiguration, callbackState)

Uses the authentication provider's supported authentication protocol to return an OAuth access token, OAuth secret or refresh token, and the state passed in when the request for the current user was initiated.

Signature

```
public Auth.AuthProviderTokenResponse handleCallback(Map<String,String>
authProviderConfiguration, Auth.AuthProviderCallbackState callbackState)
```

Parameters

authProviderConfiguration
Type: Map<StringString>

The configuration for the custom authentication provider. When you create a custom metadata type in Salesforce, the configuration populates with the custom metadata type default values. Or you can set the configuration with values you enter when you create the custom provider in Auth. Providers in Setup.

callbackState
Type: Auth.AuthProviderCallbackState

The class that contains the HTTP headers, body, and queryParams of the authentication request.

Return Value

Type: Auth.AuthProviderTokenResponse

Creates an instance of the `AuthProviderTokenResponse` class.

Usage

As of API version 39.0, use this method when extending `Auth.AuthProviderPluginClass` to create a custom authentication provider.

`initiate(authProviderConfiguration, stateToPropagate)`

Returns the URL where the user is redirected for authentication.

Signature

```
public System.PageReference initiate(Map<String,String> authProviderConfiguration,
String stateToPropagate)
```

Parameters

authProviderConfiguration
Type: Map<StringString>

The configuration for the custom authentication provider. When you create a custom metadata type in Salesforce, the configuration populates with the custom metadata type default values. Or you can set the configuration with values you enter when you create the custom provider in Auth. Providers in Setup.

stateToPropagate
Type: String

The state passed in to initiate the authentication request for the user.

Return Value

Type: System.PageReference

The URL of the page where the user is redirected for authentication.

Usage

As of API version 39.0, use this method when extending `Auth.AuthProviderPluginClass` to create a custom authentication provider.

`refresh(authProviderConfiguration, refreshToken)`

Returns a new access token, which is used to update an expired access token.

Signature

```
public Auth.OAuthRefreshResult refresh(Map<String,String> authProviderConfiguration,
String refreshToken)
```

Parameters

`authProviderConfiguration`
Type: Map<String,String>

The configuration for the custom authentication provider. When you create a custom metadata type in Salesforce, the configuration populates with the custom metadata type default values. Or you can set the configuration with values you enter when you create the custom provider in Auth. Providers in Setup.

`refreshToken`
Type: String

The refresh token for the user who is logged in.

Return Value

Type: Auth.OAuthRefreshResult

Returns the new access token, or an error message if an error occurs.

Usage

A successful request returns a `Auth.OAuthRefreshResult` with the access token and refresh token in the response. If you receive an error, make sure that you set the error string to the error message. A `NULL` error string indicates no error.

The refresh method works only with named credentials; it doesn't respect the standard OAuth refresh flow. The refresh method with named credentials works only if the earlier request returns a 401.

AuthProviderPluginClass Code Example

The following example demonstrates how to implement a custom Auth. provider plug-in using the abstract class, `Auth.AuthProviderPluginClass`.

```
        global class Concur extends Auth.AuthProviderPluginClass {

        // Use this URL for the endpoint that the
        // authentication provider calls back to for configuration.
        public String redirectUrl;
        private String key;
        private String secret;
```

```
        // Application redirection to the Concur website for
        // authentication and authorization.
        private String authUrl;

        // URI to get the new access token from concur using the GET verb.
        private String accessTokenUrl;

        // Api name for the custom metadata type created for this auth provider.
        private String customMetadataTypeApiName;

        // Api URL to access the user in Concur
        private String userAPIUrl;

        // Version of the user api URL to access data from Concur
        private String userAPIVersionUrl;

        global String getCustomMetadataType() {
            return customMetadataTypeApiName;
        }

        global PageReference initiate(Map<string,string>
          authProviderConfiguration, String stateToPropagate)
          {
              authUrl = authProviderConfiguration.get('Auth_Url__c');
              key = authProviderConfiguration.get('Key__c');

              // Here the developer can build up a request of some sort.
              // Ultimately, they return a URL where we will redirect the user.
              String url = authUrl + '?client_id='+ key
+'&scope=USER,EXPRPT,LIST&redirect_uri='+ redirectUrl + '&state=' + stateToPropagate;
              return new PageReference(url);
          }

          global Auth.AuthProviderTokenResponse handleCallback(Map<string,string>
          authProviderConfiguration, Auth.AuthProviderCallbackState state )
          {
              // Here, the developer will get the callback with actual protocol.
              // Their responsibility is to return a new object called
              // AuthProviderTokenResponse.
              // This will contain an optional accessToken and refreshToken
              key = authProviderConfiguration.get('Key__c');
              secret = authProviderConfiguration.get('Secret__c');
              accessTokenUrl = authProviderConfiguration.get('Access_Token_Url__c');

              Map<String,String> queryParams = state.queryParameters;
              String code = queryParams.get('code');
              String sfdcState = queryParams.get('state');

              HttpRequest req = new HttpRequest();
              String url = accessTokenUrl+'?code=' + code + '&client_id=' + key +
              '&client_secret=' + secret;
              req.setEndpoint(url);
              req.setHeader('Content-Type','application/xml');
```

```
        req.setMethod('GET');

        Http http = new Http();
        HTTPResponse res = http.send(req);
        String responseBody = res.getBody();
        String token = getTokenValueFromResponse(responseBody, 'Token', null);

        return new Auth.AuthProviderTokenResponse('Concur', token,
        'refreshToken', sfdcState);
    }

    global Auth.UserData getUserInfo(Map<string,string>
    authProviderConfiguration,
    Auth.AuthProviderTokenResponse response)
    {
        //Here the developer is responsible for constructing an
        //Auth.UserData object
        String token = response.oauthToken;
        HttpRequest req = new HttpRequest();
        userAPIUrl = authProviderConfiguration.get('API_User_Url__c');
        userAPIVersionUrl = authProviderConfiguration.get
        ('API_User_Version_Url__c');
        req.setHeader('Authorization', 'OAuth ' + token);
        req.setEndpoint(userAPIUrl);
        req.setHeader('Content-Type','application/xml');
        req.setMethod('GET');

        Http http = new Http();
        HTTPResponse res = http.send(req);
        String responseBody = res.getBody();
        String id = getTokenValueFromResponse(responseBody,
        'LoginId',userAPIVersionUrl);
        String fname = getTokenValueFromResponse(responseBody,
        'FirstName', userAPIVersionUrl);
        String lname = getTokenValueFromResponse(responseBody,
        'LastName', userAPIVersionUrl);
        String flname = fname + ' ' + lname;
        String uname = getTokenValueFromResponse(responseBody,
        'EmailAddress', userAPIVersionUrl);
        String locale = getTokenValueFromResponse(responseBody,
        'LocaleName', userAPIVersionUrl);
        Map<String,String> provMap = new Map<String,String>();
        provMap.put('what1', 'noidea1');
        provMap.put('what2', 'noidea2');
        return new Auth.UserData(id, fname, lname, flname,
        uname, 'what', locale, null, 'Concur', null, provMap);
    }

    private String getTokenValueFromResponse(String response,
    String token, String ns)
    {
        Dom.Document docx = new Dom.Document();
        docx.load(response);
        String ret = null;
```

```
            dom.XmlNode xroot = docx.getrootelement() ;
            if(xroot != null){ ret = xroot.getChildElement(token, ns).getText();
            }
        return ret;
        }

    }
```

Sample Test Classes

The following example contains test classes for the Concur class.

```
        @IsTest
        public class ConcurTestClass {

            private static final String OAUTH_TOKEN = 'testToken';
            private static final String STATE = 'mocktestState';
            private static final String REFRESH_TOKEN = 'refreshToken';
            private static final String LOGIN_ID = 'testLoginId';
            private static final String USERNAME = 'testUsername';
            private static final String FIRST_NAME = 'testFirstName';
            private static final String LAST_NAME = 'testLastName';
            private static final String EMAIL_ADDRESS = 'testEmailAddress';
            private static final String LOCALE_NAME = 'testLocalName';
            private static final String FULL_NAME = FIRST_NAME + ' ' + LAST_NAME;
            private static final String PROVIDER = 'Concur';
            private static final String REDIRECT_URL =
            'http://localhost/services/authcallback/orgId/Concur';
            private static final String KEY = 'testKey';
            private static final String SECRET = 'testSecret';
            private static final String STATE_TO_PROPOGATE = 'testState';
            private static final String ACCESS_TOKEN_URL =
            'http://www.dummyhost.com/accessTokenUri';
            private static final String API_USER_VERSION_URL =
            'http://www.dummyhost.com/user/20/1';
            private static final String AUTH_URL =
            'http://www.dummy.com/authurl';
            private static final String API_USER_URL =
            'www.concursolutions.com/user/api';

        // In the real world scenario, the key and value would be read
        // from the (custom fields in) custom metadata type record.
        private static Map<String,String> setupAuthProviderConfig ()
        {
            Map<String,String> authProviderConfiguration = new Map<String,String>();

            authProviderConfiguration.put('Key__c', KEY);
            authProviderConfiguration.put('Auth_Url__c', AUTH_URL);
            authProviderConfiguration.put('Secret__c', SECRET);
            authProviderConfiguration.put('Access_Token_Url__c', ACCESS_TOKEN_URL);
            authProviderConfiguration.put('API_User_Url__c',API_USER_URL);
            authProviderConfiguration.put('API_User_Version_Url__c',
```

```
                API_USER_VERSION_URL);
                authProviderConfiguration.put('Redirect_Url__c',REDIRECT_URL);
                return authProviderConfiguration;

            }

            static testMethod void testInitiateMethod()
            {
                String stateToPropogate = 'mocktestState';
                Map<String,String> authProviderConfiguration = setupAuthProviderConfig();

                Concur concurCls = new Concur();
                concurCls.redirectUrl = authProviderConfiguration.get('Redirect_Url__c');

            PageReference expectedUrl = new
PageReference(authProviderConfiguration.get('Auth_Url__c') + '?client_id='+
          authProviderConfiguration.get('Key__c') +'&scope=USER,EXPRPT,LIST&redirect_uri='+

            authProviderConfiguration.get('Redirect_Url__c') + '&state=' +
            STATE_TO_PROPOGATE);
                PageReference actualUrl = concurCls.initiate(authProviderConfiguration,
STATE_TO_PROPOGATE);
                System.assertEquals(expectedUrl.getUrl(), actualUrl.getUrl());
            }

            static testMethod void testHandleCallback()
            {
                Map<String,String> authProviderConfiguration =
                setupAuthProviderConfig();
                Concur concurCls = new Concur();
                concurCls.redirectUrl = authProviderConfiguration.get
                ('Redirect_Url_c');

                Test.setMock(HttpCalloutMock.class, new
                ConcurMockHttpResponseGenerator());

                Map<String,String> queryParams = new Map<String,String>();
                queryParams.put('code','code');
                queryParams.put('state',authProviderConfiguration.get('State_c'));
                Auth.AuthProviderCallbackState cbState =
                new Auth.AuthProviderCallbackState(null,null,queryParams);
                Auth.AuthProviderTokenResponse actualAuthProvResponse =
                concurCls.handleCallback(authProviderConfiguration, cbState);
                Auth.AuthProviderTokenResponse expectedAuthProvResponse =
                new Auth.AuthProviderTokenResponse(
                'Concur', OAUTH_TOKEN, REFRESH_TOKEN, null);

                System.assertEquals(expectedAuthProvResponse.provider,
                actualAuthProvResponse.provider);
                System.assertEquals(expectedAuthProvResponse.oauthToken,
                actualAuthProvResponse.oauthToken);
                System.assertEquals(expectedAuthProvResponse.oauthSecretOrRefreshToken,
                actualAuthProvResponse.oauthSecretOrRefreshToken);
                System.assertEquals(expectedAuthProvResponse.state,
```

```
        actualAuthProvResponse.state);

    }

    static testMethod void testGetUserInfo()
    {
        Map<String,String> authProviderConfiguration =
        setupAuthProviderConfig();
        Concur concurCls = new Concur();

        Test.setMock(HttpCalloutMock.class, new
        ConcurMockHttpResponseGenerator());

        Auth.AuthProviderTokenResponse response =
        new Auth.AuthProviderTokenResponse(
        PROVIDER, OAUTH_TOKEN ,'sampleOauthSecret', STATE);
        Auth.UserData actualUserData = concurCls.getUserInfo(
        authProviderConfiguration, response) ;

        Map<String,String> provMap = new Map<String,String>();
        provMap.put('key1', 'value1');
        provMap.put('key2', 'value2');

        Auth.UserData expectedUserData = new Auth.UserData(LOGIN_ID,
        FIRST_NAME, LAST_NAME, FULL_NAME, EMAIL_ADDRESS,
        null, LOCALE_NAME, null, PROVIDER, null, provMap);

        System.assertNotEquals(expectedUserData,null);
        System.assertEquals(expectedUserData.firstName,
        actualUserData.firstName);
        System.assertEquals(expectedUserData.lastName,
        actualUserData.lastName);
        System.assertEquals(expectedUserData.fullName,
        actualUserData.fullName);
        System.assertEquals(expectedUserData.email,
        actualUserData.email);
        System.assertEquals(expectedUserData.username,
        actualUserData.username);
        System.assertEquals(expectedUserData.locale,
        actualUserData.locale);
        System.assertEquals(expectedUserData.provider,
        actualUserData.provider);
        System.assertEquals(expectedUserData.siteLoginUrl,
        actualUserData.siteLoginUrl);
    }


    // Implement a mock http response generator for Concur.
    public class ConcurMockHttpResponseGenerator implements HttpCalloutMock
    {
        public HTTPResponse respond(HTTPRequest req)
        {
            String namespace = API_USER_VERSION_URL;
            String prefix = 'mockPrefix';
```

```
                Dom.Document doc = new Dom.Document();
                Dom.XmlNode xmlNode = doc.createRootElement(
                'mockRootNodeName', namespace, prefix);
                xmlNode.addChildElement('LoginId', namespace, prefix)
                .addTextNode(LOGIN_ID);
                xmlNode.addChildElement('FirstName', namespace, prefix)
                .addTextNode(FIRST_NAME);
                xmlNode.addChildElement('LastName', namespace, prefix)
                .addTextNode(LAST_NAME);
                xmlNode.addChildElement('EmailAddress', namespace, prefix)
                .addTextNode(EMAIL_ADDRESS);
                xmlNode.addChildElement('LocaleName', namespace, prefix)
                .addTextNode(LOCALE_NAME);
                xmlNode.addChildElement('Token', null, null)
                .addTextNode(OAUTH_TOKEN);
                System.debug(doc.toXmlString());
                // Create a fake response
                HttpResponse res = new HttpResponse();
                res.setHeader('Content-Type', 'application/xml');
                res.setBody(doc.toXmlString());
                res.setStatusCode(200);
                return res;
            }

        }
    }
```

AuthProviderTokenResponse Class

Stores the response from the `AuthProviderPlugin.handleCallback` method.

Namespace

Auth

IN THIS SECTION:

AuthProviderTokenResponse Constructors

AuthProviderTokenResponse Properties

AuthProviderTokenResponse Constructors

The following are constructors for `AuthProviderTokenResponse`.

IN THIS SECTION:

AuthProviderTokenResponse(provider, oauthToken, oauthSecretOrRefreshToken, state)

Creates an instance of the `AuthProviderTokenResponse` class using the specified authentication provider, OAuth access token, OAuth secret or refresh token, and state for a custom authentication provider plug-in.

`AuthProviderTokenResponse(provider, oauthToken, oauthSecretOrRefreshToken, state)`

Creates an instance of the `AuthProviderTokenResponse` class using the specified authentication provider, OAuth access token, OAuth secret or refresh token, and state for a custom authentication provider plug-in.

Signature

```
public AuthProviderTokenResponse(String provider, String oauthToken, String
oauthSecretOrRefreshToken, String state)
```

Parameters

provider
Type: String

The custom authentication provider.

oauthToken
Type: String

The OAuth access token.

oauthSecretOrRefreshToken
Type: String

The OAuth secret or refresh token for the currently logged-in user.

state
Type: String

The state passed in to initiate the authentication request for the user.

AuthProviderTokenResponse Properties

The following are properties for `AuthProviderTokenResponse`.

IN THIS SECTION:

oauthSecretOrRefreshToken
The OAuth secret or refresh token for the currently logged-in user.

oauthToken
The OAuth access token.

provider
The authentication provider.

state
The state passed in to initiate the authentication request for the user.

`oauthSecretOrRefreshToken`

The OAuth secret or refresh token for the currently logged-in user.

Signature

```
public String oauthSecretOrRefreshToken {get; set;}
```

Property Value

Type: String

`oauthToken`

The OAuth access token.

Signature

```
public String oauthToken {get; set;}
```

Property Value

Type: String

`provider`

The authentication provider.

Signature

```
public String provider {get; set;}
```

Property Value

Type: String

`state`

The state passed in to initiate the authentication request for the user.

Signature

```
public String state {get; set;}
```

Property Value

Type: String

AuthToken Class

Contains methods for getting and revoking access and refresh tokens that are issued when a user logs in via a single sign-on (SSO) flow that uses an authentication provider, such as Facebook.

Namespace

Auth

Usage

To authenticate users via an authentication provider, you must create a class that implements the `Auth.RegistrationHandler` interface. When a user logs in to Salesforce via a provider such as Facebook, they're issued an access token and in some cases, a refresh token. To retrieve and revoke these tokens, use the methods in the `Auth.AuthToken` class.

AuthToken Methods

The following are methods for `AuthToken`. All methods are static.

IN THIS SECTION:

getAccessToken(authProviderId, providerName)
Returns an access token for the current user using the specified 18-character identifier of an AuthProvider definition in your org and the proper name of the provider, such as Salesforce or Facebook.

getAccessTokenMap(authProviderId, providerName)
Returns a map from the provider's identifier to the access token for the currently logged-in Salesforce user. The identifier value depends on the provider. For example, for Salesforce, it's the user ID, while for Facebook, it's the user number.

refreshAccessToken(authProviderId, providerName, oldAccessToken)
Returns a map from the third-party provider's identifier containing a refreshed access token for the currently logged-in Salesforce user.

revokeAccess(authProviderId, providerName, userId, remoteIdentifier)
Revokes the access token for a specified SSO user from a provider such as Facebook. You can use this method only if the `IsNotSsoUsable` field on the associated ThirdPartyAccountLink object is set to `false`.

`getAccessToken(authProviderId, providerName)`

Returns an access token for the current user using the specified 18-character identifier of an AuthProvider definition in your org and the proper name of the provider, such as Salesforce or Facebook.

Signature

```
public static String getAccessToken(String authProviderId, String providerName)
```

Parameters

`authProviderId`
Type: String

`providerName`
Type: String

The proper name of the provider. Here are valid values for each provider type.

- Apple—`Apple`

- Custom—For a custom authentication provider, use the value in the `FriendlyName` field on the AuthProvider object, such as `MyProvider`.
- Facebook—`Facebook`
- GitHub—`GitHub`
- Google—`Google`
- Janrain—Use the proper name of the third party, such as `Yahoo!`.
- LinkedIn—`LinkedIn`
- Microsoft—`Microsoft`
- Microsoft Access Control Service—`Microsoft Access Control Service`
- MuleSoft—`MuleSoft`
- Open ID Connect—`Open ID Connect`
- Salesforce—`Salesforce`
- Slack—`Slack`
- Twitter—This method doesn't support the Twitter authentication provider.

Note: The `providerName` value that you pass into this method can be different from the value that's returned if you query the `ProviderType` field on the AuthProvider object. For example, for Open ID Connect providers, `OpenIdConnect` is the `ProviderType` value for the AuthProvider object, but the expected `providerName` is `Open ID Connect`.

Return Value

Type: String

`getAccessTokenMap(authProviderId, providerName)`

Returns a map from the provider's identifier to the access token for the currently logged-in Salesforce user. The identifier value depends on the provider. For example, for Salesforce, it's the user ID, while for Facebook, it's the user number.

Signature

```
public static Map<String, String> getAccessTokenMap(String authProviderId, String
providerName)
```

Parameters

`authProviderId`
Type: String

`providerName`
Type: String

The proper name of the provider. Here are valid values for each provider type.

- Apple—`Apple`
- Custom—For a custom authentication provider, use the value in the `FriendlyName` field on the AuthProvider object, such as `MyProvider`.
- Facebook—`Facebook`
- GitHub—`GitHub`

- Google—`Google`
- Janrain—Use the proper name of the third party, such as `Yahoo!`.
- LinkedIn—`LinkedIn`
- Microsoft—`Microsoft`
- Microsoft Access Control Service—`Microsoft Access Control Service`
- MuleSoft—`MuleSoft`
- Open ID Connect—`Open ID Connect`
- Salesforce—`Salesforce`
- Slack—`Slack`
- Twitter—This method doesn't support the Twitter authentication provider.

Note: The `providerName` value that you pass into this method can be different from the value that's returned if you query the `ProviderType` field on the AuthProvider object. For example, for Open ID Connect providers, `OpenIdConnect` is the `ProviderType` value for the AuthProvider object, but the expected `providerName` is `Open ID Connect`.

Return Value

Type: Map<String, String>

`refreshAccessToken(authProviderId, providerName, oldAccessToken)`

Returns a map from the third-party provider's identifier containing a refreshed access token for the currently logged-in Salesforce user.

Signature

```
public static Map<String, String> refreshAccessToken(String authProviderId, String
providerName, String oldAccessToken)
```

Parameters

`authProviderId`
Type: String

`providerName`
Type: String

The proper name of the third party. Here are valid values for each provider type.

- Apple—`Apple`
- Custom—For a custom authentication provider, use the value in the `FriendlyName` field on the AuthProvider object, such as `MyProvider`.
- Facebook—`Facebook`
- GitHub—`GitHub`
- Google—`Google`
- Janrain—Use the proper name of the third party, such as `Yahoo!`.
- LinkedIn—`LinkedIn`
- Microsoft—`Microsoft`
- Microsoft Access Control Service—`Microsoft Access Control Service`

- MuleSoft—`MuleSoft`
- Open ID Connect—`Open ID Connect`
- Salesforce—`Salesforce`
- Slack—`Slack`
- Twitter—This method doesn't support the Twitter authentication provider.

Note: The `providerName` value that you pass into this method can be different from the value that's returned if you query the `ProviderType` field on the AuthProvider object. For example, for Open ID Connect providers, `OpenIdConnect` is the `ProviderType` value for the AuthProvider object, but the expected `providerName` is `Open ID Connect`.

`oldAccessToken`
Type: String

Return Value

Type: Map<String, String>

Usage

The returned map contains `AccessToken` and `RefreshError` keys. Evaluate the keys in the response to check if the request was successful. For a successful request, the `RefreshError` value is `null`, and `AccessToken` is a token value. For an unsuccessful request, the `RefreshError` value is an error message, and the `AccessToken` value is `null`.

When successful, this method updates the token stored in the database, which you can get using `Auth.AuthToken.getAccessToken()`.

If you're using an OpenID Connect authentication provider, an `id_token` isn't required in the response from the provider. If a **Token Issuer** is specified in the **Auth. Provider** settings and an `id_token` is provided anyway, Salesforce verifies it.

Example

```
String accessToken = Auth.AuthToken.getAccessToken('0SOD000000000DeOAI', 'Open ID Connect');
Map<String, String> responseMap = Auth.AuthToken.refreshAccessToken('0SOD000000000DeOAI',
 'Open ID Connect', accessToken);
```

A successful request includes the access token in the response.

```
 (RefreshError,null)(AccessToken,00DD00000007BhE!AQkAQFzj...)
```

`revokeAccess(authProviderId, providerName, userId, remoteIdentifier)`

Revokes the access token for a specified SSO user from a provider such as Facebook. You can use this method only if the `IsNotSsoUsable` field on the associated ThirdPartyAccountLink object is set to `false`.

Signature

```
public static Boolean revokeAccess(String authProviderId, String providerName, String
userId, String remoteIdentifier)
```

Parameters

authProviderId
Type: String

The ID of the authentication provider in Salesforce.

providerName
Type: String

The name of the third party. Here are valid `providerName` values for each provider type.

Important: The `providerName` value that you pass into this method must be lowercase.

- Apple—`apple`
- Custom—For a custom authentication provider, use a lowercase version of the value in the `FriendlyName` field on the AuthProvider object. For example, if the `FriendlyName` is `MyProvider`, use `myprovider`.
- Facebook—`facebook`
- GitHub—`github`
- Google—`google`
- Janrain—Use a lowercase version of the name of the third party, such as `yahoo!`.
- LinkedIn—`linkedin`
- Microsoft—`microsoft`
- Microsoft Access Control Service—`microsoft access control service`
- MuleSoft—`mulesoft`
- Open ID Connect—`open id connect`
- Salesforce—`salesforce`
- Slack—`slack`
- Twitter—`twitter`

Note: The `providerName` that you pass into this method is different from the value that you get if you query the `ProviderType` field on the AuthProvider object. For example, for Open ID Connect providers, the `providerType` value for the AuthProvider object is `OpenIdConnect`, but the `providerName` for the `revokeAccess` method is `open id connect`.

userId
Type: String

The 15-character ID for the user whose access is being revoked.

remoteIdentifier
Type: String

The unique ID for the user in the third-party system (this value is in the associated ThirdPartyAccountLink standard object).

Return Value

Type: Boolean

The return value is `true` if the `revokeAccess()` operation is successful; otherwise `false`.

Example

The following example revokes a Facebook user's access token.

```
Auth.AuthToken.revokeAccess('0SOxx00000#####', 'facebook', '005xx00000#####',
'ThirdPartyIdentifier_exist214176560#####');
```

CommunitiesUtil Class

Contains methods for getting information about an Experience Cloud user.

Namespace

Auth

Example

The following example directs a guest (unauthenticated) user to one page, and authenticated users of the Experience Cloud site's parent organization to another page.

```
if (Auth.CommunitiesUtil.isGuestUser())
    // Redirect to the login page if user is an unauthenticated user
    return new PageReference(LOGIN_URL);

if (Auth.CommunitiesUtil.isInternalUser())
    // Redirect to the home page if user is an internal user
    return new PageReference(HOME_URL);
```

CommunitiesUtil Methods

The following are methods for `CommunitiesUtil`. All methods are static.

IN THIS SECTION:

getLogoutUrl()
Returns the page to display after the current Experience Cloud user logs out.

getUserDisplayName()
Returns the current user's Experience Cloud display name.

isGuestUser()
Indicates whether the current user isn't logged in to the Experience Cloud site. Redirect the user to log in, if necessary.

isInternalUser()
Indicates whether the current user is logged in as a member of the parent Salesforce organization, such as an employee.

`getLogoutUrl()`

Returns the page to display after the current Experience Cloud user logs out.

Signature

```
public static String getLogoutUrl()
```

Return Value

Type: String

`getUserDisplayName()`

Returns the current user's Experience Cloud display name.

Signature

```
public static String getUserDisplayName()
```

Return Value

Type: String

`isGuestUser()`

Indicates whether the current user isn't logged in to the Experience Cloud site. Redirect the user to log in, if necessary.

Signature

```
public static Boolean isGuestUser()
```

Return Value

Type: Boolean

`isInternalUser()`

Indicates whether the current user is logged in as a member of the parent Salesforce organization, such as an employee.

Signature

```
public static Boolean isInternalUser()
```

Return Value

Type: Boolean

ConfigurableSelfRegHandler Interface

Gives you more control over how customers or partners self-register for your Experience Cloud site by creating a class that implements `Auth.ConfigurableSelfRegHandler`. You choose the user information to collect, and how users identify themselves—with their email address, phone number, or another identifier. When verified, you create a customer or partner user and log in the user to your Experience Cloud site.

Namespace

Auth

Usage

You set up site self-registration declaratively on the Login & Registration (L&R) page of the Administration workspace. When combined with a configurable self-registration setup, the handler class can programmatically fill in user fields, including custom fields, and determine how to create a user and log them in.

When you select the Configurable Self-Reg Page registration page, you choose the user fields to collect from the self-registration form, such as last name, first name, username, nickname, mobile, or email. You also determine the verification method that the user identifies themselves with, which can be email, mobile, or neither. Salesforce generates the `Auth.ConfigurableSelfRegHandler` handler, which contains logic on how to create an Experience Cloud site member. Modify the handler to change how users are created, and how collected user information is used.

You can add custom logic to ensure that the email or phone number is unique to the customer or partner who's registering. For example, you can add a custom unique field, and write a copy of the email or phone number to it. You can also change how the user is created. By default, the user is created as a contact associated with the account that you select on the L&R page.

The generated `ConfigurableSelfRegHandler` is located on the Setup Apex Classes page, and begins with `AutocreatedConfigSelfReg`, for example, `AutocreatedConfigSelfReg1532475901849`.

For an example, see ConfigurableSelfRegHandler Example Implementation. For more details, see Salesforce Customer Identity in *Salesforce Help*.

IN THIS SECTION:

ConfigurableSelfRegHandler Method

ConfigurableSelfRegHandler Example Implementation

This Apex code implements the `Auth.ConfigurableSelfRegHandler` interface. After the customer or partner fills out the sign-up page and submits it, the handler is invoked to create an Experience Cloud member with the supplied information. If the registration process requires email or phone verification, the verification process finishes before the `Auth.ConfigurableSelfRegHandler.createUser` is invoked. If verification isn't required, `createUser` is invoked when the customer or partner submits the page.

ConfigurableSelfRegHandler Method

The following is the method for `ConfigurableSelfRegHandler`.

IN THIS SECTION:

createUser(accountId, profileId, registrationAttributes, password)

Create a community member from the information that the visitor provided on your community's self-registration page.

`createUser(accountId, profileId, registrationAttributes, password)`

Create a community member from the information that the visitor provided on your community's self-registration page.

Signature

```
public Id createUser(Id accountId, Id profileId, Map<Schema.SObjectField,String>
registrationAttributes, String password)
```

Parameters

accountId
Type: Id

Default account with which the new user is associated. This value comes from the Account field setting on Login and Registration (L&R) page under Registration Page Configuration.

profileID
Type: Id

Profile to assign the new user. This value comes from the Profile field setting on the L&R page under Registration Page Configuration.

registrationAttributes
Type: Map<Schema.sObjectField,String>

A map of attributes that the registering user entered on the self-registration page. The fields that appear on the self-registration page come from the User Fields selected on the L&R page when the registration type is Configurable Self-Reg Page.

password
Type: String

The password entered by the user if "Include Password" is selected on the L&R page. (If a password isn't entered, the handler must generate one because a password is required to create a user.)

Return Value

Type: Id

Returns an identifier for the created User object. `Auth.ConfigurableSelfRegHandler` inserts a user and then returns the ID of that user.

ConfigurableSelfRegHandler Example Implementation

This Apex code implements the `Auth.ConfigurableSelfRegHandler` interface. After the customer or partner fills out the sign-up page and submits it, the handler is invoked to create an Experience Cloud member with the supplied information. If the registration process requires email or phone verification, the verification process finishes before the `Auth.ConfigurableSelfRegHandler.createUser` is invoked. If verification isn't required, `createUser` is invoked when the customer or partner submits the page.

Verification occurs by email if the admin chose Email as the verification method when setting up the Configurable Self-Reg handler on the Login & Registration (L&R) page. When a visitor clicks the sign-up link from the login page, Salesforce prompts for an email address and then sends a one-time password to the specified email address. If the visitor enters the verification code successfully on the verify page, the user is created and logged in. Likewise, if the admin chose Text Message as the verification method on the L&R page, the visitor is prompted to enter a phone number. Salesforce sends a challenge (verification code) via SMS to the user. If successful, the user is created and logged in. Requiring verification before creating a user reduces the number of dummy users cluttering your org.

The `Auth.ConfigurableSelfRegHandler` class contains logic for generating the user fields required to create a user in case the user doesn't supply them. The handler generates default values, ensuring that the values are unique by appending a timestamp. You can modify the handler to make sure that the email address and phone number of the customer or partner are also unique.

```
global class AutocreatedConfigSelfReg implements Auth.ConfigurableSelfRegHandler {

    private final Long CURRENT_TIME = Datetime.now().getTime();
    private final String[] UPPERCASE_CHARS = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'.split('');
    private final String[] LOWERCASE_CHARS = 'abcdefghijklmnopqrstuvwxyz'.split('');
    private final String[] NUMBER_CHARS = '1234567890'.split('');
    private final String[] SPECIAL_CHARS = '!#$%-_=+<>'.split('');

    // This method is called once after verification (if any was configured).
    // This method should create a user and insert it.
    // Password can be null.
    // Return null or throw an exception to fail creation.
    global Id createUser(Id accountId, Id profileId, Map<SObjectField, String>
registrationAttributes, String password) {
        User u = new User();
        u.ProfileId = profileId;
        for (SObjectField field : registrationAttributes.keySet()) {
            String value = registrationAttributes.get(field);
            u.put(field, value);
        }

        u = handleUnsetRequiredFields(u);
        generateContact(u, accountId);
        if (String.isBlank(password)) {
            password = generateRandomPassword();
        }
        Site.validatePassword(u, password, password);
        if (u.contactId == null) {
            return Site.createExternalUser(u, accountId, password);
        }
        u.languagelocalekey = UserInfo.getLocale();
        u.localesidkey = UserInfo.getLocale();
        u.emailEncodingKey = 'UTF-8';
        u.timeZoneSidKey = UserInfo.getTimezone().getID();
        insert u;
        System.setPassword(u.Id, password);
        return u.id;
    }
    // Method to autogenerate a password if one isn't passed in.
    // By setting a password for a user, we won't send a
    // welcome email to set the password.
    private String generateRandomPassword() {
        String[] characters = new List<String>(UPPERCASE_CHARS);
        characters.addAll(LOWERCASE_CHARS);
        characters.addAll(NUMBER_CHARS);
        characters.addAll(SPECIAL_CHARS);
        String newPassword = '';
        Boolean needsUpper = true, needsLower = true, needsNumber = true, needsSpecial =
true;
        while (newPassword.length() < 50) {
            Integer randomInt = generateRandomInt(characters.size());
```

```
            String c = characters[randomInt];
            if (needsUpper && c.isAllUpperCase()) {
                needsUpper = false;
            } else if (needsLower && c.isAllLowerCase()) {
                needsLower = false;
            } else if (needsNumber && c.isNumeric()) {
                needsNumber = false;
            } else if (needsSpecial && !c.isAlphanumeric()) {
                needsSpecial = false;
            }
            newPassword += c;
        }
        newPassword = addMissingPasswordRequirements(newPassword, needsLower, needsUpper,
 needsNumber, needsSpecial);
        return newPassword;
    }

    private String addMissingPasswordRequirements(String password, Boolean addLowerCase,
Boolean addUpperCase, Boolean addNumber, Boolean addSpecial) {
        if (addLowerCase) {
            password += LOWERCASE_CHARS[generateRandomInt(LOWERCASE_CHARS.size())];
        }
        if (addUpperCase) {
            password += UPPERCASE_CHARS[generateRandomInt(UPPERCASE_CHARS.size())];
        }
        if (addNumber) {
            password += NUMBER_CHARS[generateRandomInt(NUMBER_CHARS.size())];
        }
        if (addSpecial) {
            password += SPECIAL_CHARS[generateRandomInt(SPECIAL_CHARS.size())];
        }
        return password;
    }
   // Generates a random number from 0 up to, but not including, max.
    private Integer generateRandomInt(Integer max) {
        return Math.mod(Math.abs(Crypto.getRandomInteger()), max);
    }

    // Loops over required fields that were not passed in to
    // set to some default value.
    private User handleUnsetRequiredFields(User u) {
        if (String.isBlank(u.LastName)){
            u.LastName = generateLastName();
        }
        if (String.isBlank(u.Username)) {
            u.Username = generateUsername();
        }
        if (String.isBlank(u.Email)) {
            u.Email = generateEmail();
        }
        if (String.isBlank(u.Alias)) {
            u.Alias = generateAlias();
        }
        if (String.isBlank(u.CommunityNickname)) {
```

```
            u.CommunityNickname = generateCommunityNickname();
        }
        return u;
    }
    // Method to construct a contact for a user.
    private void generateContact(User u, Id accountId) {
        // Add logic here if you want to build your own
        // contact for the use.
    }
   // Default implementation to try to provide uniqueness.
    private String generateAlias() {
        String timeString = String.valueOf(CURRENT_TIME);
        return timeString.substring(timeString.length() - 8);
    }
    // Default implementation to try to provide uniqueness.
    private String generateLastName() {
        return 'ExternalUser' + CURRENT_TIME;
    }
    // Default implementation to try to provide uniqueness.
    private String generateUsername() {
        return 'externaluser' + CURRENT_TIME + '@company.com';
    }
    // Default implementation to try to provide uniqueness.
    private String generateEmail() {
        return 'externaluser' + CURRENT_TIME + '@company.com';
    }
    // Default implementation to try to provide uniqueness.
    private String generateCommunityNickname() {
        return 'ExternalUser' + CURRENT_TIME;
    }
}
```

ConfirmUserRegistrationHandler Interface

Manages single sign-on (SSO) user mappings between Salesforce and a third-party identity provider. Use this interface to confirm user mappings before updating them.

Namespace

Auth

Usage

When you set up SSO with a third-party identity provider, you create a class that implements a registration handler using the `Auth.RegistrationHandler` interface. This class manages the process of creating and updating users. For advanced use cases that require you to confirm user information during the update process, implement the `Auth.ConfirmUserRegistrationHandler` interface in your class. This interface must be implemented in addition to `Auth.RegistrationHandler`.

You can use the `Auth.ConfirmUserRegistrationHandler` interface to ensure that users are mapped correctly between Salesforce and the third party. When a user who has previously logged in with an authentication provider logs in again, you can confirm

that the incoming user data is consistent with the user's third-party identifier. If not, you can identify which user is supposed to be logged in.

You can also use the `Auth.ConfirmUserRegistrationHandler` interface to switch context for users with multiple records. For example, a user has two records—an admin user and a standard user. When the user logs in, the third-party identity provider confirms the account used to log in and sends the response to Salesforce via the UserInfo endpoint. You can then use this information to determine whether to log in the user as an admin or standard user.

IN THIS SECTION:

ConfirmUserRegistrationHandler Methods

ConfirmUserRegistrationHandler Example Implementation

ConfirmUserRegistrationHandler Methods

The following are methods for `ConfirmUserRegistrationHandler`.

IN THIS SECTION:

confirmUser(userId, tpalId, portalId, userdata)
Returns the ID of the user to be logged in based on their mapping to a third-party identifier. This method is called before calling the `updateUser()` method. It's called only if the incoming user has previously logged in and has a third-party account link to a Salesforce user.

confirmUser(userId, tpalId, portalId, userdata)

Returns the ID of the user to be logged in based on their mapping to a third-party identifier. This method is called before calling the `updateUser()` method. It's called only if the incoming user has previously logged in and has a third-party account link to a Salesforce user.

Signature

```
public Id confirmUser(Id userId, Id tpalId, Id portalId, Auth.UserData userdata)
```

Parameters

userId
Type: Id

The ID of the user who is mapped to the third-party identifier via a third-party account link.

tpalId
Type: Id

The third-party account link corresponding to the third-party identifier.

portalId
Type: Id

The portal ID the user is logging in to. If there's no portal configured, this value can be null.

userData
Type: Auth.UserData

Contains user information from the third-party identity provider.

Return Value

Type: Id

The Id of the user to be logged in. If null, login fails.

ConfirmUserRegistrationHandler Example Implementation

This example implements the `Auth.ConfirmUserRegistrationHandler` interface during the user update process to confirm that the correct user is logging in based on their email address and last name.

```
global class StandardUserRegistrationHandler implements Auth.RegistrationHandler,
Auth.ConfirmUserRegistrationHandler {
    global User createUser(Id portalId, Auth.UserData data){
        User u = new User();
        Profile p = [SELECT Id FROM profile WHERE name='Standard User'];
        u.username = data.username + '@salesforce.com';
        u.email = data.email;
        u.lastName = data.lastName;
        u.firstName = data.firstName;
        String alias = data.username;
        if(alias.length() > 8) {
            alias = alias.substring(0, 8);
        }
        u.alias = alias;
        u.languagelocalekey = data.attributeMap.get('language');
        u.localesidkey = data.locale;
        u.emailEncodingKey = 'UTF-8';
        u.timeZoneSidKey = 'America/Los_Angeles';
        u.profileId = p.Id;
        return u;
    }

    global void updateUser(Id userId, Id portalId, Auth.UserData data) {
        User u = new User(id=userId);
        u.username = data.username + '@salesforce.com';
        u.email = data.email;
        u.lastName = data.lastName;
        u.firstName = data.firstName;
        String alias = data.username;
        if(alias.length() > 8) {
            alias = alias.substring(0, 8);
        }
        u.alias = alias;
        u.languagelocalekey = data.attributeMap.get('language');
        u.localesidkey = data.locale;
        update(u);
    }

    global Id confirmUser(Id userId, Id tpalId, Id portalId, Auth.UserData data) {
        if (data.email.contains(data.lastName)) { // looks genuine
            return userId;
        } else { // find the right user
            User confirmedUser = [SELECT id FROM user WHERE email=:data.email];
            return confirmedUser.Id;
```

```
        }
    }
}
```

The following example tests the implementation:

```
@isTest
public class StandardUserRegistrationHandlerTest {
    static testMethod void testConfirmUser() {
        StandardUserRegistrationHandler handler = new StandardUserRegistrationHandler();
        Auth.UserData sampleData = new Auth.UserData('idA', 'firstName', 'A',
            'firstName A', 'userA@example.org', null, 'usernameA', 'en_US', 'facebook',
            null, new Map<String, String>{'language' => 'en_US'});
        User u = handler.createUser(null, sampleData);
        insert(u);
        String uid = u.id;

        sampleData = new Auth.UserData('idB', 'firstName', 'B',
            'firstName B', 'userA@example.org', null, 'usernameB', 'en_US', 'facebook',
            null, new Map<String, String>{}); // note that user B is using userA's email
       Id confirmedUserId = handler.confirmUser(uid, '060xx0000004Eh6', null, sampleData);

        System.assertEquals(uid, confirmedUserId); // we should see userA's id
    }
}
```

ConnectedAppPlugin Class

Contains methods for extending the behavior of a connected app, for example, customizing how a connected app is invoked depending on the protocol used. This class gives you more control over the interaction between Salesforce and your connected app.

Namespace

Auth

Usage

When you create a connected app, you specify general information about the app and settings for OAuth, web apps, mobile apps, and canvas apps. To customize how the app is invoked, create a connected app handler with this `ConnectedAppPlugin` Apex class. For example, use this class to support new authentication protocols or respond to user attributes in a way that benefits a business process.

When you create a connected app handler, you also configure the `ConnectedAppPlugin` class to run as an execution user. The execution user authorizes access for the connected app. For example, when you use the `authorize` method, the execution user authorizes the connected app to access data.

If you don't specify an execution user, the plug-in runs as an Automated Process User, which is a system user that executes tasks behind the scenes. Most `ConnectedAppPlugin` methods require that you specify an execution user, with the exception of the `customAttributes` method. For more information, see Create a Custom Connected App Handler.

Example

This example authorizes the connected app user to use the connected app if the context is SAML and the user has reached the quota tracked in a custom field. It returns the user's permission set assignments. The example uses `Auth.InvocationContext` to modify a SAML assertion before it's sent to the service provider.

```
global class ConnectedAppPluginExample extends Auth.ConnectedAppPlugin
{
    // Authorize the app if the user has achieved quota tracked in a custom field
   global override Boolean authorize(Id userId, Id connectedAppId, Boolean isAdminApproved,
 Auth.InvocationContext context)
    {
        // Create a custom boolean field HasAchievedQuota__c on the user record
        // and then uncomment the block below
        // User u = [select id, HasAchievedQuota__c from User where id =: userId].get(0);

        // return u.HasAchievedQuota__c;

        return isAdminApproved;
    }

    // Call a flow during refresh
    global override void refresh(Id userId, Id connectedAppId, Auth.InvocationContext
context)
    {
       try
       {
          Map<String, Object> inputVariables = new Map<String, Object>();
          inputVariables.put('userId', userId);
          inputVariables.put('connectedAppId', connectedAppId);

         // Create a custom trigger ready flow and uncomment the block below
         // Flow.Interview.MyCustomFlow interview = new
Flow.Interview.MyCustomFlow(inputVariables);
         // interview.start();
        } catch ( Exception e ) {
           System.debug('FLOW Exception:' + e);
        }
    }

    // Return a user's permission set assignments
    global override Map<String,String> customAttributes(Id userId, Id connectedAppId,
Map<String,String>
        formulaDefinedAttributes, Auth.InvocationContext context)
    {
        List<PermissionSetAssignment> psas = [SELECT id, PermissionSet.Name FROM
PermissionSetAssignment
        WHERE PermissionSet.IsOwnedByProfile = false AND (AssigneeId = :userId)];
        String permsets = '[';
        for (PermissionSetAssignment psa :psas)
        {
            permsets += psa.PermissionSet.Name + ';';
        }
        permsets += ']';
        formulaDefinedAttributes.put('PermissionSets', permsets);
```

```
        return formulaDefinedAttributes;
    }
}
```

IN THIS SECTION:

ConnectedAppPlugin Methods

ConnectedAppPlugin Methods

The following are methods for `ConnectedAppPlugin`.

IN THIS SECTION:

authorize(userId, connectedAppId, isAdminApproved)

Deprecated and available only in API versions 35.0 and 36.0. As of version 37.0, use `authorize(userId, connectedAppId, isAdminApproved, context)` instead.

authorize(userId, connectedAppId, isAdminApproved, context)

Authorizes the specified user to access the connected app. If the connected app is set for users to self-authorize, this method isn't invoked.

customAttributes(userId, connectedAppId, formulaDefinedAttributes)

Deprecated and available only in API versions 35.0 and 36.0. As of version 37.0, use `customAttributes(userId, connectedAppId, formulaDefinedAttributes, context)` instead.

customAttributes(userId, connectedAppId, formulaDefinedAttributes, context)

Sets new attributes for the specified user. When the connected app gets the user's attributes from the UserInfo endpoint or through a SAML assertion, use this method to update the attribute values.

modifySAMLResponse(authSession, connectedAppId, samlResponse)

Modifies the XML generated by the Salesforce SAML Identity Provider (IDP) before it's sent to the service provider.

refresh(userId, connectedAppId)

Deprecated and available only in API versions 35.0 and 36.0. As of version 37.0, use `refresh(userId, connectedAppId, context)` instead.

refresh(userId, connectedAppId, context)

Salesforce calls this method during a refresh token exchange.

`authorize(userId, connectedAppId, isAdminApproved)`

Deprecated and available only in API versions 35.0 and 36.0. As of version 37.0, use `authorize(userId, connectedAppId, isAdminApproved, context)` instead.

Signature

```
public Boolean authorize(Id userId, Id connectedAppId, Boolean isAdminApproved)
```

Parameters

`userId`
Type: Id

The 15-character ID of the user attempting to use the connected app.

connectedAppId
Type: String

The 15-character ID of the connected app.

isAdminApproved
Type: Boolean

The approval state of the specified user when the connected app requires approval.

Return Value

Type: Boolean

If the connected app requires admin approval, a returned value of `true` indicates that the current user is approved.

authorize(userId, connectedAppId, isAdminApproved, context)

Authorizes the specified user to access the connected app. If the connected app is set for users to self-authorize, this method isn't invoked.

Signature

```
public Boolean authorize(Id userId, Id connectedAppId, Boolean isAdminApproved,
Auth.InvocationContext context)
```

Parameters

userId
Type: Id

The 15-character ID of the user attempting to use the connected app.

connectedAppId
Type: Id

The 15-character ID of the connected app.

isAdminApproved
Type: Boolean

The approval state of the specified user when the connected app requires approval.

context
Type: InvocationContext

The context in which the connected app is invoked.

Return Value

Type: Boolean

If the connected app requires admin approval, a returned value of `true` indicates that the user is approved.

Usage

`ConnectedAppPlugin` runs on behalf of the current user. But the user must have permission to use the connected app for the plug-in to work. Use this method to authorize the user.

customAttributes(userId, connectedAppId, formulaDefinedAttributes)

Deprecated and available only in API versions 35.0 and 36.0. As of version 37.0, use `customAttributes(userId, connectedAppId, formulaDefinedAttributes, context)` instead.

Signature

```
public Map<String,String> customAttributes(Id userId, Id connectedAppId,
Map<String,String> formulaDefinedAttributes,)
```

Parameters

userId
Type: Id

The 15-character ID of the user attempting to use the connected app.

connectedAppId
Type: Id

The 15-character ID of the connected app.

formulaDefinedAttributes
Type: Map<String,String>

A map of the new set of attributes from the UserInfo endpoint (OAuth) or from a SAML assertion. For more information, see The UserInfo Endpoint in the online help.

Return Value

Type: Map<String,String>

A map of the updated set of attributes.

customAttributes(userId, connectedAppId, formulaDefinedAttributes, context)

Sets new attributes for the specified user. When the connected app gets the user's attributes from the UserInfo endpoint or through a SAML assertion, use this method to update the attribute values.

Signature

```
public Map<String,String> customAttributes(Id userId, Id connectedAppId,
Map<String,String> formulaDefinedAttributes, Auth.InvocationContext context)
```

Parameters

userId
Type: Id

The 15-character ID of the user attempting to use the connected app.

connectedAppId
Type: Id

The 15-character ID for the connected app.

formulaDefinedAttributes
Type: Map<String,String>

A map of the current set of attributes from the UserInfo endpoint (OAuth) or from a SAML assertion. For more information, see The UserInfo Endpoint in the online help.

context
Type: InvocationContext

The context in which the connected app is invoked.

Return Value

Type: Map<String,String>

A map of the updated set of attributes.

modifySAMLResponse(authSession, connectedAppId, samlResponse)

Modifies the XML generated by the Salesforce SAML Identity Provider (IDP) before it's sent to the service provider.

Signature

```
public dom.XmlNode modifySAMLResponse(Map<String,String> authSession, Id connectedAppId,
dom.XmlNode samlResponse)
```

Parameters

authSession
Type: Map<String,String>

The attributes for the authorized user's session. The map includes the 15-character ID of the authorized user who's accessing the connected app.

connectedAppId
Type: Id

The 15-character ID of the connected app.

samlResponse
Type: Dom.XmlNode

Contains the SAML XML response generated by the IDP.

Return Value

Type: Dom.XmlNode

Returns an instance of Dom.XmlNode containing the modified SAML XML response.

Usage

Use this method to modify the XML SAML response to perform an action based on the context of the SAML request before it's verified, signed, and sent to the target service provider. This method enables developers to extend the connected app plug-in to meet their specific needs.

The developer assumes full responsibility for changes made within the connected app plug-in. The plug-in must include validation and error handling. If the plug-in throws an exception, catch it, log it, and stop the process. Don't send anything to the target service provider.

`refresh(userId, connectedAppId)`

Deprecated and available only in API versions 35.0 and 36.0. As of version 37.0, use `refresh(userId, connectedAppId, context)` instead.

Signature

```
public void refresh(Id userId, Id connectedAppId)
```

Parameters

userId
Type: Id

The 15-character ID of the user requesting the refresh token.

connectedAppId
Type: Id

The 15-character ID of the connected app.

Return Value

Type: void

`refresh(userId, connectedAppId, context)`

Salesforce calls this method during a refresh token exchange.

Signature

```
public void refresh(Id userId, Id connectedAppId, Auth.InvocationContext context)
```

Parameters

userId
Type: Id

The 15-character ID of the user requesting the refresh token.

connectedAppId
Type: Id

The 15-character ID of the connected app.

context
Type: InvocationContext

The context in which the connected app is invoked.

Return Value

Type: void

CustomOneTimePasswordDeliveryHandler Interface

To use a custom SMS provider to send one-time passwords (OTPs) for Experience Cloud identity verification, create a class that implements the `Auth.CustomOneTimePasswordDeliveryHandler` interface.

Namespace

Auth

IN THIS SECTION:

CustomOneTimePasswordDeliveryHandler Methods

CustomOneTimePasswordDeliveryHandler Example Implementation

CustomOneTimePasswordDeliveryHandler Methods

The following are methods for `CustomOneTimePasswordDeliveryHandler`.

IN THIS SECTION:

sendOneTimePassword(userId, phoneNumber, oneTimePassword, networkId, defaultText, expId)
Calls out to an external SMS messaging provider to send a Salesforce one-time password to an external user for identity verification. Returns an `Auth.CustomOneTimePasswordDeliveryResult` indicating whether the provider sent the message.

`sendOneTimePassword(userId, phoneNumber, oneTimePassword, networkId, defaultText, expId)`

Calls out to an external SMS messaging provider to send a Salesforce one-time password to an external user for identity verification. Returns an `Auth.CustomOneTimePasswordDeliveryResult` indicating whether the provider sent the message.

Signature

```
public Auth.CustomOneTimePasswordDeliveryResult sendOneTimePassword(Id userId, String phoneNumber, String oneTimePassword, String defaultText, Id networkId, String experienceId)
```

Parameters

`userId`
Type: Id

ID of the external user.

`phoneNumber`
Type: String

The user's phone number. The phone number isn't necessarily verified by Salesforce.

`oneTimePassword`
Type: String

The OTP that the user receives.

networkId
Type: String

ID of the Experience Cloud site.

defaultText
Type: Id

The content of the default SMS message that the user receives. You can create custom messages instead of sending the default. For example, write code to send custom messages based on the Experience Cloud site ID.

expId
Type: String

A custom value that determines what the user experiences.

Return Value

Type: Auth.CustomOneTimePasswordDeliveryResult

CustomOneTimePasswordDeliveryHandler Example Implementation

This example implements the `Auth.CustomOneTimePasswordDeliveryHandler` interface. For a detailed explanation of this example, see Example: Custom One-Time Password Delivery Handler in Salesforce Help.

```
global class TelesignMessaging implements Auth.CustomOneTimePasswordDeliveryHandler{

  global Auth.CustomOneTimePasswordDeliveryResult sendOneTimePassword(Id userId, String
phoneNumber, String oneTimePassword,
  String defaultText, Id networkId, String experienceId){

    //Send the message from Telesign
    HttpRequest request = new HttpRequest();
    //The commented-out code on the next line isn't necessary if you use named credentials

    //request.setEndpoint('https://rest-ww.telesign.com/v1/messaging');
    request.setEndpoint('callout:Telesign_SMS_Named');
    request.setMethod('POST');
   String requestBody = 'is_primary=true&phone_number=' + phoneNumber + '&message='+'Custom
 OTP%20'+ oneTimePassword+';
'+defaultText+'&message_type=OTP';

    request.setHeader('accept', 'application/json');
    request.setHeader('content-type', 'application/x-www-form-urlencoded');
    //The commented-out code on the next line isn't necessary if you use named credentials

    //request.setHeader('authorization', 'Basic <Base64-encoded Telesign customer ID:API
key>');
    request.setBody(requestBody);

    HttpResponse response = new Http().send(request);
    // Handle the response as needed
    return Auth.CustomOneTimePasswordDeliveryResult.SUCCESS;
  }
}
```

CustomOneTimePasswordDeliveryResult Enum

Indicates the status of an attempt to send a one-time password (OTP) to an external user via a custom messaging provider.

Usage

To use this feature, contact Salesforce Customer Support.

This enum specifies the result of the `sendOneTimePassword` method in an implementation of the `Auth.CustomOneTimePasswordDeliveryHandler` interface.

Enum Values

The following are the values of the `Auth.CustomOneTimePasswordDeliveryResult` enum.

Value	Description
`COUNTRY_BLOCK`	Indicates that the user's phone number has a country code that Salesforce doesn't support.
`EXCEPTION`	Indicates that the handler threw an exception.
`INVALID_PHONE_NUMBER`	Indicates that the user's phone number isn't valid. For example, it's in the wrong format or contains characters that aren't numbers.
`MESSAGE_LIMIT_EXCEEDED`	Indicates that your Experience Cloud site reached the message limit allowed by your license.
`PROVIDER_ERROR`	Indicates an error with the custom OTP service.
`SUCCESS`	Indicates that the OTP message was successfully sent to the user.

ExternalClientAppOauthHandler Class

Contains methods for extending the behavior of an external client app. For example, customize how an external client app is invoked depending on the protocol used. This class gives you more control over the interaction between Salesforce and your external client app.

Namespace

Auth

Usage

When you create an external client app, you specify general information about the app and settings for OAuth. To customize how the app is invoked, create a external client app handler with the `ExternalClientAppOauthHandler` Apex class. For example, use this class to support new authentication protocols or respond to user attributes in a way that benefits the business process.

When you create an external client app handler, you also configure the `ExternalClientAppOauthHandler` class to run as an execution user. The execution user authorizes access for the external client app. For example, when you use the authorize method, the execution user authorizes the external client app to access data.

If you don't specify an execution user, the plug-in runs as an Automated Process User, which is a system user that executes tasks behind the scenes. Most `ExternalClientAppOauthHandler` methods require that you specify an execution user, with the exception of the `customAttributes` method.

IN THIS SECTION:

ExternalClientAppOauthHandler Methods

ExternalClientAppOauthHandler Methods

The following are methods for `ExternalClientAppOauthHandler`.

IN THIS SECTION:

authorize(userId, ecAppId, isAdminApproved, context)

Authorizes the specified user to access the external client app. If the external client app is set for users to self-authorize, this method isn't invoked.

customAttributes(userId, ecAppId, formulaDefinedAttributes, context)

Sets new attributes for the specified user. When the external client app gets the user's attributes from the UserInfo endpoint, use this method to update the attribute values.

refresh(userId, ecAppId, context)

Salesforce calls this method during a refresh token exchange.

`authorize(userId, ecAppId, isAdminApproved, context)`

Authorizes the specified user to access the external client app. If the external client app is set for users to self-authorize, this method isn't invoked.

Signature

```
public Boolean authorize(Id userId, Id ecAppId, Boolean isAdminApproved,
Auth.InvocationContext context)
```

Parameters

`userId`
Type: Id

The 15-character ID of the user attempting to use the external client app.

`ecAppId`
Type: Id

The 15-character ID of the external client app.

`isAdminApproved`
Type: Boolean

The approval state of the specified user when the external client app requires approval.

`context`
Type: Auth.InvocationContext on page 130

The context in which the external client app is invoked.

Return Value

Type: Boolean

A returned value of `true` indicates that the user is approved.

customAttributes(userId, ecAppId, formulaDefinedAttributes, context)

Sets new attributes for the specified user. When the external client app gets the user's attributes from the UserInfo endpoint, use this method to update the attribute values.

Signature

```
public Map<String,String> customAttributes(Id userId, Id ecAppId, Map<String,String>
formulaDefinedAttributes, Auth.InvocationContext context)
```

Parameters

userId
Type: Id

The 15-character ID of the user attempting to use the external client app.

ecAppId
Type: Id

The 15-character ID for the external client app.

formulaDefinedAttributes
Type: Map<String,String>

A map of the current set of attributes from the UserInfo endpoint (OAuth) or from a SAML assertion. For more information, see The UserInfo Endpoint in the online help.

context
Type: Auth.InvocationContext

The context in which the external client app is invoked.

Return Value

Type: Map<String,String>

A map of the updated set of attributes.

refresh(userId, ecAppId, context)

Salesforce calls this method during a refresh token exchange.

Signature

```
public void refresh(Id userId, Id ecAppId, Auth.InvocationContext context)
```

Parameters

userId
Type: Id

The 15-character ID of the user requesting the refresh token.

ecAppId
Type: Id

The 15-character ID of the external client app.

context
Type: Auth.InvocationContext

The context in which the external client app is invoked.

Return Value

Type: void

HeadlessSelfRegistrationHandler Interface

Creates customer and partner users during the Headless Registration Flow.

Namespace

Auth

Usage

The Headless Registration Flow allows you to control user registration experience in a third-party app while using Salesforce to authenticate users and manage their data access. When you set up this flow, add users in the class that is implementing the `Auth.HeadlessSelfRegistrationHandler` interface. This class runs after the user verifies their identity. For a detailed explanation of headless registration, see Headless Registration Flow for Private Clients or Headless Registration Flow for Public Clients, depending on your app type.

IN THIS SECTION:

HeadlessSelfRegistrationHandler Methods

The following are methods for `HeadlessSelfRegistrationHandler`.

HeadlessSelfRegistrationHandler Example Implementation

This example class implements the `Auth.HeadlessSelfRegistrationHandler` interface to create a user. It finds or creates an account to store the new user and creates a contact to associate with the account. It then creates the user based on information that your client sends to Headless Registration API.

HeadlessSelfRegistrationHandler Methods

The following are methods for `HeadlessSelfRegistrationHandler`.

IN THIS SECTION:

createUser(profileId, data, customUserDataMap, experienceId, password)
Returns a User object using information submitted by your off-platform app to Headless Registration API. The User object can be a new user that hasn't been inserted in your org's database, or it can represent an existing user record. If it's a new User object, Salesforce inserts the user record for you.

createUser(profileId, data, customUserDataMap, experienceId, password)

Returns a User object using information submitted by your off-platform app to Headless Registration API. The User object can be a new user that hasn't been inserted in your org's database, or it can represent an existing user record. If it's a new User object, Salesforce inserts the user record for you.

Signature

```
public User createUser(Id profileId, Auth.UserData data, String customUserDataMap,
String experienceId, String password)
```

Parameters

profileId
Type: Id

The ID of the profile that is assigned to new users.

data
Type: Auth.UserData

A class that stores information about the user, such as their name and locale.

customUserDataMap
Type: String

A string representation of a JSON object containing custom user information passed in during registration. We recommend that you deserialize this string into the equivalent Apex class structure. Determine what custom information to collect when you build your app's registration experience.

experienceId
Type: String

A custom value that determines what the end user experiences.

password
Type: String

The user password.

Return Value

Type: User

HeadlessSelfRegistrationHandler Example Implementation

This example class implements the `Auth.HeadlessSelfRegistrationHandler` interface to create a user. It finds or creates an account to store the new user and creates a contact to associate with the account. It then creates the user based on information that your client sends to Headless Registration API.

```
global class ExampleHeadlessReg implements Auth.HeadlessSelfRegistrationHandler {
    // TO DO: Update this constant with the actual value for your use case
    private static final String CUSTOMER_ACCOUNT = 'My Account';

    /*
    * Retrieve an existing account or create a new one if it doesn't exist
    *
    * @param accountName - The name of the Account to find or create
    * @return Account - The found or newly created Account record
    */
    private Account findOrCreateAccount(String accountName) {
        List<Account> existingAccounts = [SELECT Id FROM Account WHERE Name=:accountName
LIMIT 1];

        if (existingAccounts.isEmpty()) {
            Account newAccount = new Account(Name = accountName);
            insert(newAccount);
            return newAccount;
        }

        return existingAccounts[0];
    }

    /*
    * Create a contact and associate it with an account
    *
    * @param account - The Account object to associate the contact with
    * @param user - The User object containing the first and last name for the contact
    * @return Contact - The newly created contact record
    */
    private Contact createContact(Account account, User user) {
        Contact c = new Contact();
        c.accountId = account.Id;
        c.firstName = user.firstName;
        c.lastName = user.lastName;

        insert(c);

        return c;
    }

    //TO DO: Implement any additional password validation that you want in this method.
    // In this example, the password was already checked to ensure that it complies with
the org's password policy,
    // and the password, if present, is set automatically for the new user when they are
returned from the createUserMethod.
    private Boolean isPasswordValid(String password) {
        return true;
    }
```

```
    global User createUser(Id profileId, Auth.UserData data, String customUserDataMap,
String experienceId, String password){
        if (!isPasswordValid(password)) {
            return null;
        }

        User u = new User();
        u.Username = data.username;
        u.ProfileId = profileId;
        u.Email = data.email;
        u.LastName = data.lastName;
        u.FirstName = data.firstName;
        String alias = data.username;
        // Alias must be 8 characters or less
        if (alias.length() > 8) {
            alias = alias.substring(0, 8);
        }
        u.Alias = alias;
        Account a = findOrCreateAccount(CUSTOMER_ACCOUNT);
        Contact c = createContact(a, u);
        u.ContactId = c.Id;
        u.LanguageLocaleKey = UserInfo.getLocale();
        u.LocaleSidKey = UserInfo.getLocale();
        u.EmailEncodingKey = 'UTF-8';
        u.TimeZoneSidKey = UserInfo.getTimezone().getID();

        return u;
    }
}
```

HeadlessUserDiscoveryHandler Interface

Use this interface to create a headless user discovery handler that you implement during headless login, passwordless login, and forgot password flows.

Namespace

Auth

Usage

Develop headless authorization flows where users log in to an off-platform app with an identifier other than their username, such as an email address, phone number, or order number. When a user enters the identifier in your headless app, your app sends the identifying information to a Salesforce endpoint. Salesforce then passes the identifying information to your implementation of the `Auth.HeadlessUserDiscoveryHandler` interface. The handler finds the user's account and its associated email address or phone number.

Headless user discovery supports these use cases.

- Headless login with any identifier and a password. For example, a user goes to your headless app and enters their order number and password to log in.

- Headless login with any identifier and a one-time password (OTP). For example, a user goes to your app and enters just their order number. Your Apex handler finds the user's account based on the order number. Salesforce sends an OTP to the verified email address that's associated with the account. To log in, the user enters the OTP.
- Headless password reset with any identifier. For example a user goes to your app and enters their phone number. Your Apex handler finds the user account and Salesforce sends an OTP to the user's verified phone number. To verify their identity for password reset, the user enters the OTP and can then set a new password.

Headless user discovery is supported for Headless Identity API flows and OAuth 2.0 for First-Party Applications flows. For more information about supported flows and implementation details, see Headless Login Without a Username.

IN THIS SECTION:

HeadlessUserDiscoveryHandler Methods

HeadlessUserDiscoveryHandler Example Implementation

HeadlessUserDiscoveryHandler Methods

The following are methods for `HeadlessUserDiscoveryHandler`.

IN THIS SECTION:

discoverUserFromLoginHint(networkId, loginHint, verificationAction, customDataJson, requestAttributes)
Finds a user's Salesforce account based on user information, such as their email address, phone number, or other data, that's passed to a Salesforce endpoint during headless login, passwordless login, and forgot password flows.

`discoverUserFromLoginHint(networkId, loginHint, verificationAction, customDataJson, requestAttributes)`

Finds a user's Salesforce account based on user information, such as their email address, phone number, or other data, that's passed to a Salesforce endpoint during headless login, passwordless login, and forgot password flows.

Signature

```
public Auth.HeadlessUserDiscoveryResponse discoverUserFromLoginHint(Id networkId, String loginHint, Auth.VerificationAction verificationAction, String customDataJson, Map<String,String> requestAttributes)
```

Parameters

`networkId`
Type: Id

The ID of the Experience Cloud site where your headless app sends requests.

`loginHint`
Type: String

Information about the user that Salesforce can use to find their associated account, such as their email address or phone number.

`verificationAction`
Type: Auth.VerificationAction on page 200

The verification method that's used to log the user in, either email or SMS.

customDataJson
Type: String

Custom user data, such as first name, that you collect when the user logs in to your headless app.

requestAtttibutes
Type: Map<String,String>

Information about the login request that's based on the user's browser state when accessing the login page. `requestAttributes` passes in the CommunityUrl, IpAddress, UserAgent, Platform, Application, City, Country, and Subdivision values. The City, Country, and Subdivision values come from IP geolocation.

Return Value

Type: Auth.HeadlessUserDiscoveryResponse on page 127

If the handler finds a user, it returns a user ID. If not, it returns an error message.

HeadlessUserDiscoveryHandler Example Implementation

Here's an example implementation of the `Auth.HeadlessUserDiscoveryHandler` interface. This example supports login with email and login with SMS.

The `discoverUserFromLoginHint` method uses custom logic to search for a user account with a verified email address or phone number that matches the data passed in the login hint. As a security best practice, Salesforce always recommends writing code to determine if the user's email address or phone number is verified.

For users logging in with email, the custom logic first checks whether the email address passed in the login hint is in a valid format. Then, to look for a verified Salesforce email address that matches the email address passed in the login hint, it queries the TwoFactorMethodsInfo object. If successful, it returns an instance of `Auth.HeadlessUserDiscoveryResponse` with the user ID. If something goes wrong, it returns an instance of `Auth.HeadlessUserDiscoveryResponse` with a custom error message. In this example, it returns error messages when the email address format isn't valid, the email address isn't verified, there's no user with that email address, or there are multiple users with that email address.

For users logging in with SMS, the custom logic is similar. It checks whether the phone number passed in the login hint is in a valid format. Then, it looks for a verified Salesforce phone number that matches the phone number passed in the login hint. If successful, it returns an instance of `Auth.HeadlessUserDiscoveryResponse` with the user ID, and if not, it returns custom error messages.

```
/*
 * Headless User Discovery Handler
 */
global class MyHeadlessUserDiscoveryHandler implements Auth.HeadlessUserDiscoveryHandler
{


  /*
  * This method handles the logic to determine the user account based on the loginHint and
 verificationMethod
  */
  global Auth.HeadlessUserDiscoveryResponse discoverUserFromLoginHint(Id networkId, String
 loginHint,
  Auth.VerificationAction verificationAction, String customDataJson,
Map<String,String>requestAttributes) {
    if (verificationAction == Auth.VerificationAction.EMAIL) {
      return doLookupByVerifiedEmail(loginHint, verificationAction);
```

```
    } else if (verificationAction == Auth.VerificationAction.SMS) {
      return doLookupByVerifiedMobile(loginHint, verificationAction);
    } else {
      return new Auth.HeadlessUserDiscoveryResponse(null, 'Unsupported
Auth.VerificationAction');
    }
  }

  private Auth.HeadlessUserDiscoveryResponse doLookupByVerifiedEmail(String loginHint,
Auth.VerificationAction verificationAction) {
    if (String.isBlank(loginHint) || !isValidEmail(loginHint)) {
     return new Auth.HeadlessUserDiscoveryResponse(null, 'Invalid email sent as loginHint:
 ' + loginHint);
    }
    // Search for an user account by email
    List<User> users = [SELECT Id FROM User WHERE Email = :loginHint AND IsActive = TRUE];

    if (!users.isEmpty() && users.size() == 1) {
      Id userId = users[0].Id;
      // Check if the user has a verified email
      List<TwoFactorMethodsInfo> verifiedInfo = [SELECT HasUserVerifiedEmailAddress FROM
TwoFactorMethodsInfo WHERE UserId = :userId];
      if (!verifiedInfo.isEmpty() && verifiedInfo[0].HasUserVerifiedEmailAddress == true)
 {
        // Prepare and return HeadlessUserDiscoveryResponse with userId
        return new Auth.HeadlessUserDiscoveryResponse(new Set<Id>{userId}, null);
      } else {
        // Return HeadlessUserDiscoveryResponse with error message
        return new Auth.HeadlessUserDiscoveryResponse(null, 'Email ' + loginHint + ' not
verified for the given user account');
      }
    } else {
      if (users.isEmpty()) {
        return new Auth.HeadlessUserDiscoveryResponse(null, 'No user identified for the
email: ' + loginHint);
      } else {
       return new Auth.HeadlessUserDiscoveryResponse(null, 'Multiple users identified for
 the email: ' + loginHint);
      }
    }
  }

  private Auth.HeadlessUserDiscoveryResponse doLookupByVerifiedMobile(String loginHint,
Auth.VerificationAction verificationAction) {
    String formattedSms = !String.isBlank(loginHint) ? getFormattedSms(loginHint) : null;

    if (String.isBlank(formattedSms)) {
      return new Auth.HeadlessUserDiscoveryResponse(null, 'Invalid phone number sent as
loginHint: ' + loginHint);
    }
    // Search for an user account by phone
    List<User> users = [SELECT Id FROM User WHERE MobilePhone = :loginHint AND IsActive =
 TRUE];
    if (!users.isEmpty() && users.size() == 1) {
```

```
      Id userId = users[0].Id;
      // Check if the user has a verified phone
      List<TwoFactorMethodsInfo> verifiedInfo = [SELECT HasUserVerifiedMobileNumber FROM
TwoFactorMethodsInfo WHERE UserId = :userId];
      if (!verifiedInfo.isEmpty() && verifiedInfo[0].HasUserVerifiedMobileNumber == true)
 {
        // Prepare and return HeadlessUserDiscoveryResponse with userId
        return new Auth.HeadlessUserDiscoveryResponse(new Set<Id>{userId}, null);
      } else {
        // Return HeadlessUserDiscoveryResponse with error message
       return new Auth.HeadlessUserDiscoveryResponse(null, ' ' + loginHint + ' not verified
 for the given user account');
      }
    } else {
      if (users.isEmpty()) {
        return new Auth.HeadlessUserDiscoveryResponse(null, 'No user identified for the
phone number: ' + loginHint);
      } else {
       return new Auth.HeadlessUserDiscoveryResponse(null, 'Multiple users identified for
 the phone number: ' + loginHint);
      }
    }
  }

  private boolean isValidEmail(String identifier) {
    String emailRegex =
'^[a-zA-Z0-9._|\\\\%#~`=?&/$^*!}{+-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{2,4}$';
    // source: http://www.regular-expressions.info/email.html
    Pattern EmailPattern = Pattern.compile(emailRegex);
    Matcher EmailMatcher = EmailPattern.matcher(identifier);
    if (EmailMatcher.matches()) { return true; }
    else { return false; }
  }

  private String getFormattedSms(String identifier) {
    // Accept SMS input formats with 1 or 2 digits country code, 3 digits area code and 7
 digits number
    // You can customize the SMS regex to allow different formats
   String smsRegex = '^(\\+?\\d{1,2}?[\\s-])?(\\(?\\d{3}\\)?[\\s-]?\\d{3}[\\s-]?\\d{4})$';

    Pattern smsPattern = Pattern.compile(smsRegex);
    Matcher smsMatcher = SmsPattern.matcher(identifier);
    if (smsMatcher.matches()) {
      try {
       // Format user input into the verified SMS format '+xx xxxxxxxxxx' before DB lookup

        // Append US country code +1 by default if no country code is provided
        String countryCode = smsMatcher.group(1) == null ? '+1' : smsMatcher.group(1);
        return System.UserManagement.formatPhoneNumber(countryCode, smsMatcher.group(2));

      } catch(System.InvalidParameterValueException e) {
        return null;
      }
```

```
    } else { return null; }
  }
```

HeadlessUserDiscoveryResponse Class

Contains methods to describe the result of headless user discovery using a handler that implements the `Auth.HeadlessUserDiscoveryHandler` interface during headless login, passwordless login, and forgot password flows.

Namespace

Auth

Usage

Use this class to return a user ID if headless user discovery was successful, or return custom error messages if not.

IN THIS SECTION:

HeadlessUserDiscoveryResponse Constructors

HeadlessUserDiscoveryResponse Properties

HeadlessUserDiscoveryResponse Constructors

The following are constructors for `HeadlessUserDiscoveryResponse`.

IN THIS SECTION:

HeadlessUserDiscoveryResponse(userIds, customErrorMessage)
Creates an instance of the `Auth.HeadlessUserDiscoveryResponse` class to describe the result of headless user discovery based on data passed into the `login_hint` during headless login, passwordless login, and forgot password flows.

`HeadlessUserDiscoveryResponse(userIds, customErrorMessage)`

Creates an instance of the `Auth.HeadlessUserDiscoveryResponse` class to describe the result of headless user discovery based on data passed into the `login_hint` during headless login, passwordless login, and forgot password flows.

Signature

```
public HeadlessUserDiscoveryResponse(Set<Id> userIds, String customErrorMessage)
```

Parameters

`userIds`
Type: Set<Id>

The user ID that's associated with the data passed in the `login_hint` parameter. If there are multiple users associated with the data, it can return multiple IDs, but headless user discovery fails.

`customErrorMessage`
Type: String

A custom error message that's returned if headless user discovery fails.

HeadlessUserDiscoveryResponse Properties

The following are properties for `HeadlessUserDiscoveryResponse`.

IN THIS SECTION:

customErrorMessage
A custom error message that's returned if headless user discovery fails. For example, write custom logic in your headless user discovery handler to see if the user's email address is verified. Then return a custom error message for when it isn't verified.

userIds
The user ID for the external user that's associated with the data passed into the `login_hint` parameter. If there are multiple users associated with the data, it can return multiple IDs, but headless user discovery fails.

`customErrorMessage`

A custom error message that's returned if headless user discovery fails. For example, write custom logic in your headless user discovery handler to see if the user's email address is verified. Then return a custom error message for when it isn't verified.

Signature

```
public String customErrorMessage {get; set;}
```

Property Value

Type: String

`userIds`

The user ID for the external user that's associated with the data passed into the `login_hint` parameter. If there are multiple users associated with the data, it can return multiple IDs, but headless user discovery fails.

Signature

```
public Set<Id> userIds {get; set;}
```

Property Value

Type: Set<Id>

HttpCalloutMockUtil Class

Contains a method to send fake HTTP callouts for classes in the `Auth` namespace.

Namespace

Auth

Usage

Use the `setHttpMock` method in this class to test HTTP callouts when implementing the `Auth.JWTBearerTokenExchange` and `Auth.JWTUtil` classes.

For the `Auth.JWTBearerTokenExchange` class, mock callouts to the OAuth token endpoint when using the `JWTBearerTokenExchange` method.

For the `Auth.JWTUtil` class, mock callouts to the identity provider's JSON Web Key Set (JWKS) endpoint when using the `validateJWTWithKeysEndpoint` method.

For more information on mocking HTTP callouts, see Testing HTTP Callouts by Implementing the HttpCalloutMock Interface.

IN THIS SECTION:

HttpCalloutMockUtil Methods

HttpCalloutMockUtil Methods

The following are methods for `HttpCalloutMockUtil`.

IN THIS SECTION:

setHttpMock(mock)
Mocks an HTTP callout using an implementation of the `System.HttpCalloutMock` interface.

`setHttpMock(mock)`

Mocks an HTTP callout using an implementation of the `System.HttpCalloutMock` interface.

Signature

```
public static void setHttpMock(System.HttpCalloutMock mock)
```

Parameters

`mock`
Type: System.HttpCalloutMock

A class that implements the System.HttpCalloutMock interface to return a fake HTTP response for a given request to the OAuth token endpoint or a JWKS endpoint on an external identity provider, depending on your use case.

Return Value

Type: void

IntegratingAppType Enum

Specifies whether you're integrating your app as a connected app or as an external client app in methods used in your customized Apex token exchange handler, which extends the `Auth.Oauth2TokenExchangeHandler` class.

Usage

See Token Exchange Handler Validation and Subject Mapping.

Enum Values

The following are the values of the `Auth.IntegratingAppType` enum.

Value	Description
`CA`	Indicates a Salesforce connected app.
`ECA`	Indicates a Salesforce external client app.

InvocationContext Enum

The context in which the connected app is invoked, such as the protocol flow used and the token type issued, if any. Developers can use the context information to write code that is unique to the type of invocation.

Enum Values

The following are the values of the `Auth.InvocationContext` enum.

Value	Description
`ASSET_TOKEN`	Reserved for future use.
`OAUTH1`	Context used when authentication is through an OAuth 1.0A flow.
`OAUTH2_JWT_BEARER_TOKEN`	Context used when authentication is through a JSON-based Web Token (JWT) bearer token flow.
`OAUTH2_SAML_ASSERTION`	Context used when authentication is through an OAuth 2.0 SAML assertion flow.
`OAUTH2_SAML_BEARER_ASSERTION`	Context used when authentication is through an OAuth 2.0 SAML bearer assertion flow.
`OAUTH2_USERNAME_PASSWORD`	Context used when authentication is through an OAuth 2.0 username-password flow.
`OAUTH2_USER_AGENT_ID_TOKEN`	Context used when issuing an ID token through an OAuth 2.0 user-agent flow.
`OAUTH2_USER_AGENT_TOKEN`	Context used when authentication is through an OAuth 2.0 user agent flow.
`OAUTH2_WEB_SERVER`	Context used when authentication is through a web server authentication flow.
`OPENIDCONNECT`	Context used when authentication is through an OpenID Connect authentication flow.
`REFRESH_TOKEN`	Context used when renewing tokens issued by a web server or user-agent flow.
`SAML_ASSERTION`	Context used when authentication is through a SAML assertion flow.
`UNKNOWN`	Context is unknown.

Value	Description
`USERID_ENDPOINT`	Context used when issuing an access token through a UserInfo endpoint.

SEE ALSO:

Salesforce Help: Authenticating Apps with OAuth

JWS Class

Contains methods that apply a digital signature to a JSON Web Token (JWT), using a JSON Web Signature (JWS) data structure. This class creates the signed JWT bearer token, which can be used to request an OAuth access token in the OAuth 2.0 JWT bearer token flow.

Namespace

Auth

Usage

Use the methods in this class to sign the JWT bearer token with the X509 certificate.

IN THIS SECTION:

JWS Constructors

JWS Methods

JWS Constructors

The following are constructors for `JWS`.

IN THIS SECTION:

JWS(jwt, certDevName)
Creates an instance of the `JWS` class using the specified `Auth.JWT` payload and the certificate used for signing the JWT bearer token.

JWS(payload, certDevName)
Creates an instance of the `JWS` class using the specified payload and certificate used for signing the JWT bearer token.

`JWS(jwt, certDevName)`

Creates an instance of the `JWS` class using the specified `Auth.JWT` payload and the certificate used for signing the JWT bearer token.

Signature

```
public JWS(Auth.JWT jwt, String certDevName)
```

Parameters

jwt
Type: Auth.JWT

The Base64-encoded JSON Claims Set in the JWT bearer token generated by `Auth.JWT`.

certDevName
Type: String

The `Unique Name` for a certificate stored in the Salesforce org's Certificate and Key Management page to use for signing the JWT bearer token.

Usage

Calls the `toJSONString()` method in `Auth.JWT` and sets the resulting string as the payload of the JWT bearer token. Alternatively, you can specify the payload directly using `JWS(payload, certDevName)`.

`JWS(payload, certDevName)`

Creates an instance of the `JWS` class using the specified payload and certificate used for signing the JWT bearer token.

Signature

```
public JWS(String payload, String certDevName)
```

Parameters

payload
Type: String

The Base64-encoded JSON Claims Set in the JWT bearer token.

certDevName
Type: String

The `Unique Name` for a certificate stored in the Salesforce org's Certificate and Key Management page to use for signing the JWT bearer token.

Usage

Sets the *payload* string as the payload of the JWT bearer token. Alternatively, if you generate the payload using `Auth.JWT`, you can use `JWS(jwt, certDevName)` instead.

JWS Methods

The following are methods for `JWS`. All are instance methods.

IN THIS SECTION:

clone()
Makes a duplicate copy of the JWS object.

getCompactSerialization()

Returns the compact serialization representation of the JWS as a concatenated string, with the encoded JWS header, encoded JWS payload, and encoded JWS signature strings separated by period ('.') characters.

clone()

Makes a duplicate copy of the JWS object.

Signature

```
public Object clone()
```

Return Value

Type: JWS

getCompactSerialization()

Returns the compact serialization representation of the JWS as a concatenated string, with the encoded JWS header, encoded JWS payload, and encoded JWS signature strings separated by period ('.') characters.

Signature

```
public String getCompactSerialization()
```

Return Value

Type: String

JWT Class

Generates the JSON Claims Set in a JSON Web Token (JWT). The resulting Base64-encoded payload can be passed as an argument to create an instance of the `Auth.JWS` class.

Namespace

Auth

Usage

Use the methods in this class to generate the payload in a JWT bearer token for the OAuth 2.0 JWT bearer token flow. For more information and a full code sample, see JWTBearerTokenExchange Class.

IN THIS SECTION:

JWT Methods

JWT Methods

The following are methods for `JWT`. All are instance methods.

IN THIS SECTION:

clone()

Makes a duplicate copy of the JWT object.

getAdditionalClaims()

Returns a map of additional claims in the JWT, where the key string contains the name of the claim, and the value contains the value of the claim.

getAud()

Returns the audience (`aud`) claim that identifies the intended recipients of the JWT.

getIss()

Returns the issuer (`iss`) claim that identifies the issuer of the JWT.

getNbfClockSkew()

Returns the not before (`nbf`) claim that identifies the time before which the JWT must not be accepted for processing, while allowing some leeway for clock skew. This method returns a `NoAccess` exception for JWTs generated using methods in the `Auth.JWTUtil` class. To return the `nbf` claim for these JWTs, use the `getAdditionalClaims` method instead.

getSub()

Returns the subject (`sub`) claim that identifies the current user of the JWT.

getValidityLength()

Returns the length of time (in seconds) that the JWT is valid, which affects the expiration (`exp`) claim. This method returns a `NoAccess` exception for JWTs generated using methods in the `Auth.JWTUtil` class. To return the validity length for these JWTs, use the `getAdditionalClaims` method instead.

setAdditionalClaims(additionalClaims)

Sets the additional claims in the JWT. Returned by the `getAdditionalClaims` method.

setAud(aud)

Sets the audience (`aud`) claim in the JWT. Returned by the `getAud` method.

setIss(iss)

Sets the issuer (`iss`) claim in the JWT. Returned by the `getIss` method.

setNbfClockSkew(nbfClockSkew)

Sets the not before (`nbf`) claim in the JWT. Returned by the `getNbfClockSkew` method. This method returns a `NoAccess` exception for JWTs generated using methods in the `Auth.JWTUtil` class. For these JWTs, the incoming JWT determines the `nbf` claim.

setSub(sub)

Sets the subject (`sub`) claim in the JWT. Returned by the `getSub` method.

setValidityLength(validityLength)

Sets the length of time (in seconds) that the JWT is valid, which affects the expiration (`exp`) claim. Returned by the `getValidityLength` method. This method returns a `NoAccess` exception for JWTs generated using methods in the `Auth.JWTUtil` class. For these JWTs, the incoming JWT determines the validity length.

toJSONString()

Generates the JSON object representation of the Claims Set as an encoded JWT payload.

clone()

Makes a duplicate copy of the JWT object.

Signature

```
public Object clone()
```

Return Value

Type: JWT

getAdditionalClaims()

Returns a map of additional claims in the JWT, where the key string contains the name of the claim, and the value contains the value of the claim.

Signature

```
public Map<String,Object> getAdditionalClaims()
```

Return Value

Type: Map<String,Object>

The claims returned depend on how the JWT was generated.

If the JWT was generated using other methods in the `Auth.JWT` class, this method returns the claims that were set using the `setAdditionalClaims` method.

For JWTs generated using methods in the `Auth.JWTUtil` class, the `getAdditionalClaims` method returns all claims except for these three.

- `aud` (audience)—Use the `getAud` method instead.
- `iss` (issuer)—Use the `getIss` method instead.
- `sub` (subject)—Use the `getSub` method instead.

For these JWTs, when the incoming JWT has a claim that stores an inner JSON list, the claim value is returned as a string.

getAud()

Returns the audience (`aud`) claim that identifies the intended recipients of the JWT.

Signature

```
public String getAud()
```

Return Value

Type: String

getIss()

Returns the issuer (`iss`) claim that identifies the issuer of the JWT.

Signature

```
public String getIss()
```

Return Value

Type: String

`getNbfClockSkew()`

Returns the not before (`nbf`) claim that identifies the time before which the JWT must not be accepted for processing, while allowing some leeway for clock skew. This method returns a `NoAccess` exception for JWTs generated using methods in the `Auth.JWTUtil` class. To return the `nbf` claim for these JWTs, use the `getAdditionalClaims` method instead.

Signature

```
public Integer getNbfClockSkew()
```

Return Value

Type: Integer

`getSub()`

Returns the subject (`sub`) claim that identifies the current user of the JWT.

Signature

```
public String getSub()
```

Return Value

Type: String

`getValidityLength()`

Returns the length of time (in seconds) that the JWT is valid, which affects the expiration (`exp`) claim. This method returns a `NoAccess` exception for JWTs generated using methods in the `Auth.JWTUtil` class. To return the validity length for these JWTs, use the `getAdditionalClaims` method instead.

Signature

```
public Integer getValidityLength()
```

Return Value

Type: Integer

`setAdditionalClaims(additionalClaims)`

Sets the additional claims in the JWT. Returned by the `getAdditionalClaims` method.

Signature

```
public void setAdditionalClaims(Map<String,Object> additionalClaims)
```

Parameters

additionalClaims
Type: Map<String,Object>

Return Value

Type: void

Usage

Additional claims must not include any standard claims.

setAud(aud)

Sets the audience (`aud`) claim in the JWT. Returned by the `getAud` method.

Signature

```
public void setAud(String aud)
```

Parameters

aud
Type: String

Return Value

Type: void

setIss(iss)

Sets the issuer (`iss`) claim in the JWT. Returned by the `getIss` method.

Signature

```
public void setIss(String iss)
```

Parameters

iss
Type: String

Return Value

Type: void

setNbfClockSkew(nbfClockSkew)

Sets the not before (nbf) claim in the JWT. Returned by the getNbfClockSkew method. This method returns a NoAccess exception for JWTs generated using methods in the Auth.JWTUtil class. For these JWTs, the incoming JWT determines the nbf claim.

Signature

```
public void setNbfClockSkew(Integer nbfClockSkew)
```

Parameters

nbfClockSkew
Type: Integer

Return Value

Type: void

setSub(sub)

Sets the subject (sub) claim in the JWT. Returned by the getSub method.

Signature

```
public void setSub(String sub)
```

Parameters

sub
Type: String

Return Value

Type: void

setValidityLength(validityLength)

Sets the length of time (in seconds) that the JWT is valid, which affects the expiration (exp) claim. Returned by the getValidityLength method. This method returns a NoAccess exception for JWTs generated using methods in the Auth.JWTUtil class. For these JWTs, the incoming JWT determines the validity length.

Signature

```
public void setValidityLength(Integer validityLength)
```

Parameters

validityLength
Type: Integer

Return Value

Type: void

`toJSONString()`

Generates the JSON object representation of the Claims Set as an encoded JWT payload.

Signature

```
public String toJSONString()
```

Return Value

Type: String

JWTBearerTokenExchange Class

Contains methods that POST the signed JWT bearer token to a token endpoint to request an access token, in the OAuth 2.0 JWT bearer token flow.

Namespace

Auth

Usage

Use the methods in this class to post a signed JWT bearer token to the OAuth token endpoint, in exchange for an access token.

To test HTTP callouts to the token endpoint, use the `Auth.HttpCalloutMockUtil` class.

Example

In the following example application, the Apex controller:

1. Creates the JSON Claims Set.
2. Specifies the scope of the request with additional claims.
3. Creates the signed JWT.
4. Specifies the token endpoint and POSTs to it.
5. Gets the access token from the HTTP response.

```
public class MyController{


     public MyController() {
        Auth.JWT jwt = new Auth.JWT();
        jwt.setSub('user@salesforce.com');
        jwt.setAud('https://login.salesforce.com');
        jwt.setIss('3MVG99OxTyEMCQ3gNp2PjkqeZKxnmAiG1xV4oHh9AKL_rSK.BoSVPGZHQ
ukXnVjzRgSuQqGn75NL7yfkQcyy7');
```

```
        //Additional claims to set scope
        Map<String, Object> claims = new Map<String, Object>();
        claims.put('scope', 'scope name');

        jwt.setAdditionalClaims(claims);

        //Create the object that signs the JWT bearer token
        Auth.JWS jws = new Auth.JWS(jwt, 'CertFromCertKeyManagement');

        //Get the resulting JWS in case debugging is required
        String token = jws.getCompactSerialization();

        //Set the token endpoint that the JWT bearer token is posted to
        String tokenEndpoint = 'https://login.salesforce.com/services/oauth2/token';

        //POST the JWT bearer token
       Auth.JWTBearerTokenExchange bearer = new Auth.JWTBearerTokenExchange(tokenEndpoint,
 jws);

        //Get the access token
        String accessToken = bearer.getAccessToken();

    }
}
```

IN THIS SECTION:

JWTBearerTokenExchange Constructors

JWTBearerTokenExchange Methods

JWTBearerTokenExchange Constructors

The following are constructors for `JWTBearerTokenExchange`.

IN THIS SECTION:

JWTBearerTokenExchange(tokenEndpoint, jws)

Creates an instance of the `JWTBearerTokenExchange` class using the specified token endpoint and the signed JWT bearer token.

JWTBearerTokenExchange()

Creates an instance of the `Auth.JWTBearerTokenExchange` class.

`JWTBearerTokenExchange(tokenEndpoint, jws)`

Creates an instance of the `JWTBearerTokenExchange` class using the specified token endpoint and the signed JWT bearer token.

Signature

```
public JWTBearerTokenExchange(String tokenEndpoint, Auth.JWS jws)
```

Parameters

tokenEndpoint
Type: String

The token endpoint that the signed JWT bearer token is POSTed to.

jws
Type: Auth.JWS

The signed JWT bearer token.

JWTBearerTokenExchange()

Creates an instance of the `Auth.JWTBearerTokenExchange` class.

Signature

```
public JWTBearerTokenExchange()
```

JWTBearerTokenExchange Methods

The following are methods for `JWTBearerTokenExchange`. All are instance methods.

IN THIS SECTION:

clone()
Makes a duplicate copy of the JWTBearerTokenExchange object.

getAccessToken()
Returns the `access_token` in the token response to the JWT bearer token request.

getGrantType()
Returns the grant type specified in the JWT bearer token request. The grant type value defaults to `urn:ietf:params:oauth:grant-type:jwt-bearer`.

getHttpResponse()
Returns the full `System.HttpResponse` token response to the JWT bearer token request.

getJWS()
Returns the JWS specified in the JWT bearer token request.

getTokenEndpoint()
Returns the token endpoint that the JWT bearer token request is POSTed to.

setGrantType(grantType)
Sets the grant type in the JWT bearer token request. Returned by the `getGrantType()` method.

setJWS(jws)
Sets the JWS in the JWT bearer token request. Returned by the `getJWS()` method.

setTokenEndpoint(tokenEndpoint)
Sets the token endpoint that the JWT bearer token request is POSTed to. Returned by the `getTokenEndpoint()` method.

clone()

Makes a duplicate copy of the JWTBearerTokenExchange object.

Signature

```
public Object clone()
```

Return Value

Type: JWTBearerTokenExchange

getAccessToken()

Returns the `access_token` in the token response to the JWT bearer token request.

Signature

```
public String getAccessToken()
```

Return Value

Type: String

Usage

This method extracts the `access_token` from the token response. If the token response issues the access token in a different parameter, the request fails.

If you want the full HTTP token response returned, use `getHttpResponse` instead.

getGrantType()

Returns the grant type specified in the JWT bearer token request. The grant type value defaults to `urn:ietf:params:oauth:grant-type:jwt-bearer`.

Signature

```
public String getGrantType()
```

Return Value

Type: String

getHttpResponse()

Returns the full `System.HttpResponse` token response to the JWT bearer token request.

Signature

```
public System.HttpResponse getHttpResponse()
```

Return Value

Type: System.HttpResponse

Usage

You can get the access token from the full `System.HttpResponse`. If you want only the `access_token` from the token response, you can use `getAccessToken` instead.

`getJWS()`

Returns the JWS specified in the JWT bearer token request.

Signature

```
public Auth.JWS getJWS()
```

Return Value

Type: Auth.JWS

`getTokenEndpoint()`

Returns the token endpoint that the JWT bearer token request is POSTed to.

Signature

```
public String getTokenEndpoint()
```

Return Value

Type: String

`setGrantType(grantType)`

Sets the grant type in the JWT bearer token request. Returned by the `getGrantType()` method.

Signature

```
public void setGrantType(String grantType)
```

Parameters

grantType
Type: String

Return Value

Type: void

`setJWS(jws)`

Sets the JWS in the JWT bearer token request. Returned by the `getJWS()` method.

Signature

```
public void setJWS(Auth.JWS jws)
```

Parameters

`jws`
Type: Auth.JWS

Return Value

Type: void

`setTokenEndpoint(tokenEndpoint)`

Sets the token endpoint that the JWT bearer token request is POSTed to. Returned by the `getTokenEndpoint()` method.

Signature

```
public void setTokenEndpoint(String tokenEndpoint)
```

Parameters

`tokenEndpoint`
Type: String

Return Value

Type: void

JWTUtil Class

Contains methods for validating a JSON Web Token (JWT) from an external identity provider as part of the OAuth 2.0 token exchange flow. Use these methods as part of the `validateIncomingToken` method in the `Auth.Oauth2TokenExchangeHandler` class.

Namespace

Auth

Usage

See Token Exchange Handler Validation and Subject Mapping.

If the methods in this class fail, Salesforce returns an `Auth.JWTValidationException` exception.

IN THIS SECTION:

JWTUtil Methods

JWTUtil Methods

The following are methods for `JWTUtil`.

IN THIS SECTION:

parseJWTFromStringWithoutValidation(incomingJWT)

Parses a JWT from an encoded string into a header, payload, and signature. Use this method to decode the JWT without validating it.

validateJWTWithCert(incomingJWT, certDeveloperName)

Parses and validates the JWT using a certificate saved in Salesforce. The certificate can be self-signed or signed by a certificate authority.

validateJWTWithKey(incomingJWT, publicKey)

Parses and validates the JWT using a public key from the external identity provider.

validateJWTWithKeysEndpoint(incomingJWT, keysEndpoint, shouldUseCache)

Parses and validates the JWT using a remote JSON Web Key Set (JWKS) endpoint on your external identity provider.

`parseJWTFromStringWithoutValidation(incomingJWT)`

Parses a JWT from an encoded string into a header, payload, and signature. Use this method to decode the JWT without validating it.

Signature

```
public static Auth.JWT parseJWTFromStringWithoutValidation(String incomingJWT)
```

Parameters

`incomingJWT`
Type: String

The JWT from your identity provider.

Return Value

Type:Auth.JWT

`validateJWTWithCert(incomingJWT, certDeveloperName)`

Parses and validates the JWT using a certificate saved in Salesforce. The certificate can be self-signed or signed by a certificate authority.

Signature

```
public static Auth.JWT validateJWTWithCert(String incomingJWT, String certDeveloperName)
```

Parameters

incomingJWT
Type: String

The JWT from your identity provider.

certDeveloperName
Type: String

A certificate saved in the Certificate and Key Management page in Setup.

Return Value

Type: Auth.JWT

validateJWTWithKey(incomingJWT, publicKey)

Parses and validates the JWT using a public key from the external identity provider.

Signature

```
public static Auth.JWT validateJWTWithKey(String incomingJWT, String publicKey)
```

Parameters

incomingJWT
Type: String

The JWT from your identity provider.

publicKey
Type: String

The public key from your identity provider.

Return Value

Type: Auth.JWT

validateJWTWithKeysEndpoint(incomingJWT, keysEndpoint, shouldUseCache)

Parses and validates the JWT using a remote JSON Web Key Set (JWKS) endpoint on your external identity provider.

Signature

```
public static Auth.JWT validateJWTWithKeysEndpoint(String incomingJWT, String
keysEndpoint, Boolean shouldUseCache)
```

Parameters

incomingJWT
Type: String

The JWT from your identity provider.

keysEndpoint
Type: String

A URL pointing to a valid JWKS endpoint on your identity provider. The JWKS returned by the endpoint must conform to the specification defined in RFC 7517: JSON Web Key (JWK).

To test HTTP callouts to the JWKS endpoint, use the `Auth.HttpCalloutMockUtil` class.

shouldUseCache
Type: Boolean

Indicates whether the certificate from the keys endpoint is cached.

Return Value

Type: Auth.JWT

LightningLoginEligibility Enum

Contains a Lightning Login eligibility value used by the `Auth.SessionManagement.getLightningLoginEligibility` method.

Usage

If you use the Discovery page type, users can verify themselves with Lightning Login. Lightning Login lets internal users log in with Salesforce Authenticator instead of a password. Certain conditions must be met for Lightning Login to succeed.

Call `Auth.SessionManagement.getLightningLoginEligibility` before or after a login attempt to get the eligibility status. You can call after a login attempt to determine why the login attempt failed.

Enum Values

The following are the values of the `Auth.LightningLoginEligibility` enum.

Value	Description
`ELIGIBLE`	All eligibility conditions are met. The admin has enabled Salesforce Authenticator and Lightning Login, assigned the user Lightning Login user permission, and selected **Allow only for users with the Lightning Login User permission** from the Session Settings Setup page. The user has set up Salesforce Authenticator and enrolled in Lightning Login.
`ORG_AUTHENTICATOR_NOT_ENABLED`	The admin hasn't enabled Salesforce Authenticator.
`ORG_PREF_NOT_ENABLED`	The admin hasn't enabled Lightning Login. The Admin must select **Allow Lightning Login** from the Session Settings Setup page.
`USER_AUTHENTICATOR_NOT_CONNECTED`	The user hasn't set up Salesforce Authenticator.
`USER_NOT_ALLOWED`	The admin hasn't granted the user AllowLightningLogin user permission. Allowing Lightning Login to certain users requires the OnlyLLPermUserAllowed org preference. Admins must select **Allow only for users with the Lightning Login User permission** from the Session Settings Setup page.

Value	Description
`USER_NOT_ENROLLED`	The user hasn't enrolled in Lightning Login.
`USER_PERM_NOT_ENABLED`	The admin hasn't granted the user the Lightning Login Eligible user permission.

LoginDiscoveryHandler Interface

Salesforce gives you the ability to log in users based on other verification methods than username and password. For example, it can prompt users to log in with their email, phone number, or another identifier like a Federation ID or device identifier. Login Discovery is available to these licenses: Customer Community, Customer Community Plus, External Identity, Partner Community, and Partner Community Plus.

Namespace

Auth

Usage

Implement a `Auth.LoginDiscoveryHandler` for an interview-based log in. The handler looks up a user from the identifier entered, and can call `Site.passwordlessLogin` to determine which credential to use, such as email or SMS. Or the handler can redirect a user to a third-party identity provider for login. With this handler, the login page doesn't show a password field. However, you can use `Site.passwordlessLogin` to then prompt for a password.

From the user perspective, the user enters an identifier at the log in prompt. Then the user completes the login by entering a PIN or password. Or, if SSO-enabled, the user bypasses login.

For an example, see LoginDiscoveryHandler Example Implementation. For more details, see Salesforce Customer Identity in *Salesforce Help*.

IN THIS SECTION:

LoginDiscoveryHandler Method

LoginDiscoveryHandler Example Implementation

LoginDiscoveryHandler Method

Here's the method for `LoginDiscoveryHandler`.

IN THIS SECTION:

login(identifier, startUrl, requestAttributes)
Log in the customer or partner given the specified identifier, such as email or phone number. If successful, redirect the user to the Experience Cloud site page specified by the start URL.

`login(identifier, startUrl, requestAttributes)`

Log in the customer or partner given the specified identifier, such as email or phone number. If successful, redirect the user to the Experience Cloud site page specified by the start URL.

Signature

```
public System.PageReference login(String identifier, String startUrl,
Map<String,String>requestAttributes)
```

Parameters

identifier
Type: String

Identifier the customer or partner entered at the login prompt, for example, an email address or phone number.

startUrl
Type: String

Path to the Experience Cloud site page requested by the customer or partner. The user is redirected to this location after successful login.

requestAttributes
Type: Map<String,String>

Information about the login request based on the user's browser state when accessing the login page. `requestAttributes` passes in the CommunityUrl, IpAddress, UserAgent, Platform, Application, City, Country, and Subdivision values. The City, Country, and Subdivision values come from IP geolocation.

Return Value

Type: System.PageReference

The URL of the page where the user is redirected.

Example

Here's a sample `requestAttributes` response.

```
CommunityUrl=http://my-developer-edition.mycompany.com:5555/discover
IpAddress=55.555.0.0
UserAgent=Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_4) AppleWebKit/605.1.15 (KHTML,
like Gecko) Version/11.1 Safari/605.1.15
Platform=Mac OSX
Application=Browser
City=San Mateo
Country=United States
Subdivision=California
```

LoginDiscoveryHandler Example Implementation

This Apex code example implements the `Auth.LoginDiscoveryHandler` interface. It checks whether the user who is logging in has a verified email or phone number, depending on which identifier was supplied on the login page. If verified, with `Auth.VerificationMethod.EMAIL` or `Auth.VerificationMethod.SMS`, we send a challenge to the identifier, either the user's email address or mobile device. If the user enters the code correctly on the verify page, the user is redirected to the Experience Cloud site's page specified by the start URL. If the user isn't verified, the user must enter a password to log in. The handler also checks that the email and phone number are unique with this code: `users.size()==1`.

Note: Passwordless login works only with verified methods. You can check the verification status on the User object, for example, with User list view, a report, or the API. Make sure that your solution handles the case where the user doesn't have a verification method. This code example falls back to a password.

The default discoverable login handler checks whether the user entered a valid email address or phone number before redirecting the user to the verification page. If an invalid entry is made, the handler returns an error. Because this behavior is vulnerable to user enumeration attack, make sure that your solution prevents this attack. For example, you can create a dummy page similar to the verification page and redirect the user to the dummy page when invalid user identifier is entered. Also, use generic error messages to avoid providing additional information.

The `discoveryResult` function calls the `Site.passwordlessLogin` method to log the user in with the specified verification method. The `getSsoRedirect` function looks up whether the user logs in with SAML or an Auth Provider. Add the implementation-specific logic to handle the lookup.

```
global class AutocreatedDiscLoginHandler1535377170343 implements Auth.LoginDiscoveryHandler
 {

global PageReference login(String identifier, String startUrl, Map<String, String>
requestAttributes) {
    if (identifier != null && isValidEmail(identifier)) {
        // Search for user by email.
        List<User> users = [SELECT Id FROM User WHERE Email = :identifier AND IsActive =
TRUE];
        if (!users.isEmpty() && users.size() == 1) {
            // User must have a verified email before using this verification method.
            // We cannot send messages to unverified emails.
            // You can check if the user's email verified bit set and add the
            // password verification method as fallback.
            List<TwoFactorMethodsInfo> verifiedInfo = [SELECT HasUserVerifiedEmailAddress
 FROM TwoFactorMethodsInfo WHERE UserId = :users[0].Id];
            if (!verifiedInfo.isEmpty() && verifiedInfo[0].HasUserVerifiedEmailAddress ==
 true) {
                // Use email verification method if the user's email is verified.
                return discoveryResult(users[0], Auth.VerificationMethod.EMAIL, startUrl,
 requestAttributes);
            } else {
                // Use password verification method as fallback
                // if the user's email is unverified.
              return discoveryResult(users[0], Auth.VerificationMethod.PASSWORD, startUrl,
 requestAttributes);
            }
        } else {
            throw new Auth.LoginDiscoveryException('No unique user found. User count=' +
users.size());
        }
    }
    if (identifier != null) {
        String formattedSms = getFormattedSms(identifier);
        if (formattedSms != null) {
            // Search for user by SMS.
            List<User> users = [SELECT Id FROM User WHERE MobilePhone = :formattedSms AND
 IsActive = TRUE];
            if (!users.isEmpty() && users.size() == 1) {
                // User must have a verified SMS before using this verification method.
```

```
                // We cannot send messages to unverified mobile numbers.
                // You can check if the user's mobile verified bit is set or add
                // the password verification method as fallback.
              List<TwoFactorMethodsInfo> verifiedInfo = [SELECT HasUserVerifiedMobileNumber
 FROM TwoFactorMethodsInfo WHERE UserId = :users[0].Id];
               if (!verifiedInfo.isEmpty() && verifiedInfo[0].HasUserVerifiedMobileNumber
 == true) {
                   // Use SMS verification method if the user's mobile number is verified.

                  return discoveryResult(users[0], Auth.VerificationMethod.SMS, startUrl,
 requestAttributes);
                } else {
                    // Use password verification method as fallback if the user's
                    // mobile number is unverified.
                    return discoveryResult(users[0], Auth.VerificationMethod.PASSWORD,
startUrl, requestAttributes);
                }
            } else {
               throw new Auth.LoginDiscoveryException('No unique user found. User count='
 + users.size());
            }
        }
    }
    if (identifier != null) {
        // You can customize the code to find user via other attributes,
        // such as SSN or Federation ID.
    }
    throw new Auth.LoginDiscoveryException('Invalid Identifier');
}
private boolean isValidEmail(String identifier) {
    String emailRegex =
'^[a-zA-Z0-9._|\\\\%#~`=?&/$^*!}{+-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{2,4}$';
    // source: https://www.regular-expressions.info/email.html
    Pattern EmailPattern = Pattern.compile(emailRegex);
    Matcher EmailMatcher = EmailPattern.matcher(identifier);
    if (EmailMatcher.matches()) { return true; }
    else { return false; }
}
private String getFormattedSms(String identifier) {
    // Accept SMS input formats with 1- or 2-digit country code,
    // 3-digit area code, and 7-digit number.
    // You can customize the SMS regex to allow different formats.
   String smsRegex = '^(\\+?\\d{1,2}?[\\s-])?(\\(?\\d{3}\\)?[\\s-]?\\d{3}[\\s-]?\\d{4})$';

    Pattern smsPattern = Pattern.compile(smsRegex);
    Matcher smsMatcher = SmsPattern.matcher(identifier);
    if (smsMatcher.matches()) {
        try {
            // Format user input into the verified SMS format '+xx xxxxxxxxxx'
            // before DB lookup. If no country code is provided, append
            // US country code +1 for the default.
           String countryCode = smsMatcher.group(1) == null ? '+1' : smsMatcher.group(1);

          return System.UserManagement.formatPhoneNumber(countryCode, smsMatcher.group(2));
```

```
        } catch(System.InvalidParameterValueException e) {
            return null;
        }
    } else { return null; }
}
private PageReference getSsoRedirect(User user, String startUrl, Map<String, String>
requestAttributes) {
    // You can look up to check whether the user should log in with
    // SAML or an Auth Provider and return the URL to initialize SSO.
    return null;
}
private PageReference discoveryResult(User user, Auth.VerificationMethod method, String
startUrl, Map<String, String> requestAttributes) {
    // Only users with an External Identity or community license can log in
    // using Site.passwordlessLogin. Use getSsoRedirect to let your org employees
    // log in to an Experience Cloud site.
    PageReference ssoRedirect = getSsoRedirect(user, startUrl, requestAttributes);
    if (ssoRedirect != null) {
        return ssoRedirect;
    } else {
        if (method != null) {
            List<Auth.VerificationMethod> methods = new List<Auth.VerificationMethod>();
            methods.add(method);
            PageReference pwdlessRedirect = Site.passwordlessLogin(user.Id, methods,
startUrl);
            if (pwdlessRedirect != null) {
                return pwdlessRedirect;
            } else {
               throw new Auth.LoginDiscoveryException('No Passwordless Login redirect URL
 returned for verification method: ' + method);
            }
        } else {
            throw new Auth.LoginDiscoveryException('No method found');
        }
    }
}
}
```

Code Example: Filter Login Discovery Users by Profile

Your production org can have multiple users with the same verified email address and mobile number. But your customers must have unique ones. To address this problem, you can add a few lines of code that filters users by profile to ensure uniqueness. This code example handles users with the External Identity User profile, but can be adapted to support other use cases. For example, you can modify the first line of code to address users with other user licenses or criteria.

Login Discovery is available with the following user licenses: Customer Community, Customer Community Plus, External Identity, Partner Community, and Partner Community Plus. It depends on which profiles have access to your Experience Cloud site.

```
global class AutocreatedDiscLoginHandler1551301979709 implements Auth.LoginDiscoveryHandler
 {

global PageReference login(String identifier, String startUrl, Map<String, String>
requestAttributes) {
```

```
    if (identifier != null && isValidEmail(identifier)) {
        // Ensure uniqueness by profile
        Profile p = [SELECT id FROM profile WHERE name = 'External Identity User'];
        List<User> users = [SELECT Id FROM User WHERE Email = :identifier AND IsActive =
TRUE AND profileId=:p.id];
        if (!users.isEmpty() && users.size() == 1) {
            // User must have verified email before using this verification method. We
cannot send messages to unverified emails.
            // You can check if the user has email verified bit on and add the password
verification method as fallback.
            List<TwoFactorMethodsInfo> verifiedInfo = [SELECT HasUserVerifiedEmailAddress
 FROM TwoFactorMethodsInfo WHERE UserId = :users[0].Id];
            if (!verifiedInfo.isEmpty() && verifiedInfo[0].HasUserVerifiedEmailAddress ==
 true) {
                // Use email verification method if the user's email is verified.
                return discoveryResult(users[0], Auth.VerificationMethod.EMAIL, startUrl,
 requestAttributes);
            } else {
                // Use password verification method as fallback if the user's email is
unverified.
              return discoveryResult(users[0], Auth.VerificationMethod.PASSWORD, startUrl,
 requestAttributes);
            }
        } else {
            throw new Auth.LoginDiscoveryException('No unique user found. User count=' +
users.size());
        }
    }
    if (identifier != null) {
        String formattedSms = getFormattedSms(identifier);
        if (formattedSms != null) {
            // Ensure uniqueness by profile
            Profile p = [SELECT id FROM profile WHERE name = 'External Identity User'];
            List<User> users = [SELECT Id FROM User WHERE MobilePhone = :formattedSms AND
 IsActive = TRUE AND profileId=:p.id];
            if (!users.isEmpty() && users.size() == 1) {
                // User must have verified SMS before using this verification method. We
cannot send messages to unverified mobile numbers.
               // You can check if the user has mobile verified bit on or add the password
 verification method as fallback.
              List<TwoFactorMethodsInfo> verifiedInfo = [SELECT HasUserVerifiedMobileNumber
 FROM TwoFactorMethodsInfo WHERE UserId = :users[0].Id];
               if (!verifiedInfo.isEmpty() && verifiedInfo[0].HasUserVerifiedMobileNumber
 == true) {
                   // Use SMS verification method if the user's mobile number is verified.

                  return discoveryResult(users[0], Auth.VerificationMethod.SMS, startUrl,
 requestAttributes);
                } else {
                    // Use password verification method as fallback if the user's mobile
number is unverified.
                    return discoveryResult(users[0], Auth.VerificationMethod.PASSWORD,
startUrl, requestAttributes);
                }
```

```
            } else {
                throw new Auth.LoginDiscoveryException('No unique user found. User count='
 + users.size());
            }
        }
    }
    if (identifier != null) {
        // You can customize the code to find user via other attributes, such as SSN or
Federation ID
    }
    throw new Auth.LoginDiscoveryException('Invalid Identifier');
}

private boolean isValidEmail(String identifier) {
    String emailRegex =
'^[a-zA-Z0-9._|\\\\%#~`=?&/$^*!}{+-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{2,4}$';
    // source: https://www.regular-expressions.info/email.html
    Pattern EmailPattern = Pattern.compile(emailRegex);
    Matcher EmailMatcher = EmailPattern.matcher(identifier);
    if (EmailMatcher.matches()) { return true; }
    else { return false; }
}

private String getFormattedSms(String identifier) {
    // Accept SMS input formats with 1 or 2 digits country code, 3 digits area code and 7
 digits number
    // You can customize the SMS regex to allow different formats
   String smsRegex = '^(\\+?\\d{1,2}?[\\s-])?(\\(?\\d{3}\\)?[\\s-]?\\d{3}[\\s-]?\\d{4})$';

    Pattern smsPattern = Pattern.compile(smsRegex);
    Matcher smsMatcher = SmsPattern.matcher(identifier);
    if (smsMatcher.matches()) {
        try {
            // Format user input into the verified SMS format '+xx xxxxxxxxxx' before DB
lookup
            // Append US country code +1 by default if no country code is provided
           String countryCode = smsMatcher.group(1) == null ? '+1' : smsMatcher.group(1);

          return System.UserManagement.formatPhoneNumber(countryCode, smsMatcher.group(2));

        } catch(System.InvalidParameterValueException e) {
            return null;
        }
    } else { return null; }
}

private PageReference getSsoRedirect(User user, String startUrl, Map<String, String>
requestAttributes) {
    // You can look up if the user should log in with SAML or an Auth Provider and return
 the URL to initialize SSO.
    return null;
}

private PageReference discoveryResult(User user, Auth.VerificationMethod method, String
```

```
startUrl, Map<String, String> requestAttributes) {
    //Only users with an External Identity or community license can login using
Site.passwordlessLogin
    //Use getSsoRedirect to enable your org employees to log in to an Experience Cloud
site
    PageReference ssoRedirect = getSsoRedirect(user, startUrl, requestAttributes);
    if (ssoRedirect != null) {
        return ssoRedirect;
    } else {
        if (method != null) {
            List<Auth.VerificationMethod> methods = new List<Auth.VerificationMethod>();
            methods.add(method);
            PageReference pwdlessRedirect = Site.passwordlessLogin(user.Id, methods,
startUrl);
            if (pwdlessRedirect != null) {
                return pwdlessRedirect;
            } else {
               throw new Auth.LoginDiscoveryException('No Passwordless Login redirect URL
 returned for verification method: ' + method);
            }
        } else {
            throw new Auth.LoginDiscoveryException('No method found');
        }
    }
}
}
```

LoginDiscoveryMethod Enum

Contains methods used to verify the user's identity when the My Domain login process uses Login Discovery.

Usage

Specifies the verification method used to authenticate internal users when My Domain is set up for Login Discovery.

Enum Values

`Auth.LoginDiscoveryMethod` enum has the following values.

Value	Description
`LIGHTNING_LOGIN`	Verify identity by Lightning Login, which lets internal users log in with Salesforce Authenticator.
`PASSWORD`	Verify identity by entering a password.

MyDomainLoginDiscoveryHandler Interface

The handler used to implement the My Domain Login Discovery page, which is an interview-based (two-step) login process. First the user is prompted for a unique identifier such as an email address or phone number. Then the handler determines (discovers) how to authenticate the user. Either the user enters a password or is directed to an identity provider's login page.

Namespace

Auth

Usage

Implement `MyDomainLoginDiscoveryHandler` to let My Domain users log in with something other than their username and password. This handler contains the logic to look up the user based on the identifier value entered on the login page. The `Auth.MyDomainLoginDiscoveryHandler.login` method is invoked when the identifier page is submitted and finds the user that corresponds to the submitted identifier. The `Auth.SessionManagement.finishLoginDiscovery` method sends the user to the authentication mechanism and then logs in the user.

Register the handler from the My Domain Setup page. Under Authentication Configuration, select the **Discovery** Login Page Type. For Login Discovery Handler, select this handler from the list of Apex classes.

For an example, see MyDomainLoginDiscoveryHandler Example Implementation. For more details, search for My Domain Login Discovery in *Salesforce Help*.

IN THIS SECTION:

MyDomainLoginDiscoveryHandler Method

MyDomainLoginDiscoveryHandler Example Implementation

MyDomainLoginDiscoveryHandler Method

`MyDomainLoginDiscoveryHandler` has the following method.

IN THIS SECTION:

login(identifier, startUrl, requestAttributes)
Log in a Salesforce user given the specified identifier, such as email or phone number. If successful, redirect the user to the page specified by the start URL.

`login(identifier, startUrl, requestAttributes)`

Log in a Salesforce user given the specified identifier, such as email or phone number. If successful, redirect the user to the page specified by the start URL.

Signature

```
public System.PageReference login(String identifier, String startUrl, Map<String,String>
requestAttributes)
```

Parameters

identifier
Type: String

Identifier the Salesforce user entered at the login prompt, for example, an email address or phone number.

startUrl
Type: String

The page users see after successfully logging in to the My Domain subdomain.

requestAttributes
Type: Map <String, String>

Information about the login request based on the user's browser state when accessing the login page. `requestAttributes` passes in the MyDomainUrl, IpAddress, UserAgent, Platform, Application, City, Country, and Subdivision values. The City, Country, and Subdivision values come from IP address geolocation.

Return Value

Type: System.PageReference

The URL of the page where the user is redirected to complete authentication.

Example

Here's a sample `requestAttributes` response.

```
CommunityUrl=http://my-dev-ed.my.salesforce.com:5555/discover
IpAddress=55.255.0.0
UserAgent=Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_4) AppleWebKit/605.1.15 (KHTML,
like Gecko) Version/11.1 Safari/605.1.15
Platform=Mac OSX
Application=Browser
City=San Mateo
Country=United States
Subdivision=California
```

MyDomainLoginDiscoveryHandler Example Implementation

Here's an example of the `Auth.MyDomainLoginDiscoveryHandler` interface. This sample class contains the default logic for My Domain login discovery using password authentication. You can customize the code to ensure it meets your needs. The requestAttributes parameter provides additional information that you can use in the discovery logic. Attributes include MyDomainUrl, IpAddress, UserAgent, and location information (such as Country and City). Use `Auth.DiscoveryCustomErrorException` to throw custom errors to display on the login page.

To implement this interface, the My Domain login page type must be set to Discovery.

```
// This sample class contains the default logic for My Domain login discovery by password.
// You can customize the code to ensure it meets your needs. The requestAttributes parameter
// provides additional information you can use in the discovery logic. Attributes include MyDomainUrl,
// IpAddress, UserAgent, and location information (such as Country and City).
// Use Auth.DiscoveryCustomErrorException to throw custom errors which will be shown on login page.
 global class MyDomainDiscLoginDefaultHandler implements Auth.MyDomainLoginDiscoveryHandler {
 global PageReference login(String identifier, String startUrl, Map<String, String> requestAttributes)
{
```

```
   if (identifier != null) {
        // Search for user by email
       List<User> users = [SELECT Id FROM User WHERE Email = :identifier AND IsActive = TRUE];
        if (!users.isEmpty() && users.size() == 1) {
            return discoveryResult(users[0], startUrl, requestAttributes);
        } else {
         throw new Auth.LoginDiscoveryException('No unique user found. User count=' + users.size());
        }
    }
    throw new Auth.LoginDiscoveryException('Invalid Identifier');
  }

private PageReference getSsoRedirect(User user, String startUrl, Map<String, String> requestAttributes) {
   // You can look up if the user should log in with SAML or an Auth Provider and return the URL to initialize SSO. For example:
    // SamlSsoConfig SSO = [select Id from SamlSsoConfig where DeveloperName='SamlTest' limit 1];

    // To get the URL for a My Domain subdomain, you can pass null in the communityURL
parameter.
    // String ssoUrl = Auth.AuthConfiguration.getSamlSsoUrl(null, startUrl, SSO.Id);
    // return new PageReference(ssoUrl);
    return null;
  }

private PageReference discoveryResult(User user, String startUrl, Map<String, String> requestAttributes)

     {
     PageReference ssoRedirect = getSsoRedirect(user, startUrl, requestAttributes);
      if (ssoRedirect != null) {
        return ssoRedirect;
      }
      else {
      return Auth.SessionManagement.finishLoginDiscovery(Auth.LoginDiscoveryMethod.password, user.Id);
      }
   }
}
```

Test Class for MyDomainDiscLoginDefaultHandler Class

The following is the test class for MyDomainDiscoveryLoginHandler. For the test to work, your org must have the My Domain login page type set to Discovery.

```
// Test class for MyDomainDiscLoginDefaultHandler
@isTest
class MyDomainDiscLoginDefaultHandlerTest {
    /* Test Discoverable handler login.
       Create a user with specific email identifier and invoke login.
       Expected : User should be discovered and pagereference should be returned.
     */
    @isTest static void testLogin() {
        // Create user
        String identifierEmail = getUniqueName() + '@test.org';
        createTestUser(identifierEmail);
        Map<String, String> requestAttributes = new Map<String, String>();
        String startUrl = '';
```

```
  MyDomainDiscLoginDefaultHandler myDomainDiscLoginDefaultHandler = new MyDomainDiscLoginDefaultHandler();
    // Invoke login method from handler with the email of user created
 PageReference  pageReference = myDomainDiscLoginDefaultHandler.login(identifierEmail, startUrl, requestAttributes);
    // Asser page reference is returned
    System.assertNotEquals(null, pageReference, 'Page reference was not returned');
}
/* Test Discoverable handler login with invalid (non-existing) user.
    Expected : Auth.LoginDiscoveryException
 */
@isTest static void testLoginWithInvalidUser() {
    try {
        Map<String, String> requestAttributes = new Map<String, String>();
        String startUrl = '';
        String uniqueName = getUniqueName();
        String email = uniqueName + '@test.org';
    MyDomainDiscLoginDefaultHandler myDomainDiscLoginDefaultHandler = new MyDomainDiscLoginDefaultHandler();
        // Invoke login method from handler with non-existing user
        myDomainDiscLoginDefaultHandler.login(email, startUrl, requestAttributes);
    }catch (Auth.LoginDiscoveryException loginDiscoveryException) {
        // Assert exception message
   System.assert(loginDiscoveryException.getMessage().contains('No unique user found'), 'message=' + loginDiscoveryException.getMessage());
    }
}
/*
   Generate a random name
 */
private static String getUniqueName() {
    String orgId = UserInfo.getOrganizationId();
  String dateString = String.valueof(Datetime.now()).replace(' ','').replace(':','').replace('-','');
    Integer randomInt = Integer.valueOf(math.rint(math.random()*1000000));
    String uniqueName = orgId + dateString + randomInt;
    return uniqueName;
}
/*
  Create user with given email.
 */
private static void createTestUser(String identifierEmail)
{
    String uniqueName = getUniqueName();
    Profile pf = [SELECT Id FROM Profile WHERE Name='Standard User'];
    String profileID = pf.Id;
    String fName = 'fname';
    String lName = uniqueName + '-lname';
    User tuser = new User(  firstname = fName,
                            lastName = lName,
                            email = identifierEmail,
                            Username = uniqueName + '@test.org',
                            EmailEncodingKey = 'ISO-8859-1',
                            Alias = uniqueName.substring(18, 23),
                            TimeZoneSidKey = 'America/Los_Angeles',
                            LocaleSidKey = 'en_US',
                            LanguageLocaleKey = 'en_US',
                            ProfileId = profileID);
    insert tuser;
```

```
    }
}
```

Oauth2TokenExchangeHandler Class

Use this class to create a token exchange handler that validates tokens from an external identity provider and maps the token's subject to a Salesforce user during the OAuth 2.0 token exchange flow. The handler can also be used to create users by setting up a new User object and returning it to Salesforce for automatic insertion.

Namespace

Auth

Usage

See Token Exchange Handler Validation and Subject Mapping.

IN THIS SECTION:

Oauth2TokenExchangeHandler Methods

Oauth2TokenExchangeHandler Methods

The following are methods for `Oauth2TokenExchangeHandler`.

IN THIS SECTION:

getUserForTokenSubject(networkId, result, canCreateUser, appDeveloperName, appType)
Finds the subject defined in the external identity provider's token so that it can be mapped to a Salesforce subject.

validateIncomingToken(appDeveloperName, appType, incomingToken, tokenType)
Validates an access token, refresh token, ID token, SAML 2.0 assertion, or JWT passed from an external identity provider during the OAuth 2.0 token exchange flow.

`getUserForTokenSubject(networkId, result, canCreateUser, appDeveloperName, appType)`

Finds the subject defined in the external identity provider's token so that it can be mapped to a Salesforce subject.

Signature

```
public User getUserForTokenSubject(Id networkId, Auth.TokenValidationResult result,
Boolean canCreateUser, String appDeveloperName, Auth.IntegratingAppType appType)
```

Parameters

networkId
Type: Id

The identifier for the Salesforce user, if one exists.

result
Type: Auth.TokenValidationResult

The result of the token validation performed by the `validateIncomingToken` method in the Auth.Oauth2TokenExchangeHandler class.

canCreateUser
Type: Boolean

Specifies whether the handler can set up a User object if no user exists. Salesforce automatically inserts the user into this object.

appDeveloperName
Type: String

The developer name of the Salesforce connected app or external client app that's being used to integrate your app with Salesforce.

appType
Type: Auth.IntegratingAppType

Specifies whether your app is integrated with Salesforce as a connected app or as an external client app.

Return Value

Type: User

Returns a User object with the user information obtained from the token, from Salesforce, and from callouts to the identity provider, if applicable. The User object can be an existing user record or a new user that hasn't been inserted in the database. If it's a new user, Salesforce automatically inserts the user on behalf of the token exchange handler.

validateIncomingToken(appDeveloperName, appType, incomingToken, tokenType)

Validates an access token, refresh token, ID token, SAML 2.0 assertion, or JWT passed from an external identity provider during the OAuth 2.0 token exchange flow.

Signature

```
public Auth.TokenValidationResult validateIncomingToken(String appDeveloperName,
Auth.IntegratingAppType appType, String incomingToken, Auth.OAuth2TokenExchangeType
tokenType)
```

Parameters

appDeveloperName
Type: String

The developer name of the Salesforce connected app or external client app that's being used to integrate your app with Salesforce.

appType
Type: Auth.IntegratingAppType

Specifies whether your app is integrated with Salesforce as a connected app or as an external client app.

incomingToken
Type: String

The token from the external identity provider.

tokenType
Type: Auth.OAuth2TokenExchangeType

The type of token from the external identity provider. It can be an access token, a refresh token, an ID token, a SAML 2.0 assertion, or any token that's formatted as a JSON Web Token (JWT).

Return Value

Type: Auth.TokenValidationResult

Returns information about whether the token is valid, data extracted from the token, the token itself, and the token type. It can also return a custom error message if the validation failed.

OAuth2TokenExchangeType Enum

Used during the OAuth 2.0 token exchange flow to specify the type of token that's being exchanged for a Salesforce token.

Usage

During the token exchange flow, your app requests a token from Salesforce by sending a POST request with a token from an external identity provider. The request includes a `subject_token_type` parameter to specify the type of token. The values specified in this enum must correspond to the `subject_token_type` in the token request.

Enum Values

The following are the values of the `Auth.OAuth2TokenExchangeType` enum.

Value	Description
`ACCESS_TOKEN`	An access token from the identity provider. The corresponding `subject_token_type` is `urn:ietf:params:oauth:token-type:access_token`.
`ID_TOKEN`	An ID token from the identity provider. The corresponding `subject_token_type` is `urn:ietf:params:oauth:token-type:id_token`.
`JWT`	A token from the identity provider that's formatted as a JSON Web Token (JWT). The corresponding `subject_token_type` is `urn:ietf:params:oauth:token-type:JWT`.
`REFRESH_TOKEN`	A refresh token from the identity provider. The corresponding `subject_token_type` is `urn:ietf:params:oauth:token-type:refresh_token`.
`SAML_2`	A SAML 2.0 assertion from the identity provider. The corresponding `subject_token_type` is `urn:ietf:params:oauth:token-type:saml2`.

OAuthRefreshResult Class

Stores the result of an `AuthProviderPluginClass` refresh method. OAuth authentication flow provides a refresh token that can be used to get a new access token. Access tokens have a limited lifetime as specified by the session timeout value. When an access token expires, use a refresh token to get a new access token.

Namespace

Auth

Usage

The `OAuthRefreshResult` class contains the parameters, `accessToken`, `refreshToken`, and `error`, all of which are of type `string`. For a code example, see .

IN THIS SECTION:

OAuthRefreshResult Constructors

OAuthRefreshResult Properties

OAuthRefreshResult Constructors

The following are constructors for `OAuthRefreshResult`.

IN THIS SECTION:

OAuthRefreshResult(accessToken, refreshToken, error)

Creates an instance of the `OAuthRefreshResult` class using the specified access token, refresh token, and error for a custom authentication provider plug-in.

OAuthRefreshResult(accessToken, refreshToken)

Creates an instance of the `OAuthRefreshResult` class using the specified access token and refresh token for a custom authentication provider plug-in. Use this method when you know that the refresh was successful.

`OAuthRefreshResult(accessToken, refreshToken, error)`

Creates an instance of the `OAuthRefreshResult` class using the specified access token, refresh token, and error for a custom authentication provider plug-in.

Signature

```
public OAuthRefreshResult(String accessToken, String refreshToken, String error)
```

Parameters

`accessToken`
Type: String

OAuth access token for the user who is currently logged in.

`refreshToken`
Type: String

OAuth refresh token for the user who is currently logged in.

`error`
Type: String

Error that occurred when a user attempted to authenticate with the custom authentication provider.

`OAuthRefreshResult(accessToken, refreshToken)`

Creates an instance of the `OAuthRefreshResult` class using the specified access token and refresh token for a custom authentication provider plug-in. Use this method when you know that the refresh was successful.

Signature

```
public OAuthRefreshResult(String accessToken, String refreshToken)
```

Parameters

`accessToken`
Type: String

The OAuth access token for the user who is logged in.

`refreshToken`
Type: String

The OAuth refresh token for the user who is logged in.

OAuthRefreshResult Properties

The following are properties for `OAuthRefreshResult`.

IN THIS SECTION:

accessToken
The OAuth access token for the user who is currently logged in.

error
Error that occurs when a user unsuccessfully attempts to authenticate with the custom authentication provider.

refreshToken
The OAuth refresh token for the user who is currently logged in.

`accessToken`

The OAuth access token for the user who is currently logged in.

Signature

```
public String accessToken {get; set;}
```

Property Value

Type: String

`error`

Error that occurs when a user unsuccessfully attempts to authenticate with the custom authentication provider.

Signature

```
public String error {get; set;}
```

Property Value

Type: String

`refreshToken`

The OAuth refresh token for the user who is currently logged in.

Signature

```
public String refreshToken {get; set;}
```

Property Value

Type: String

OauthToken Class

Contains a method to revoke OAuth access tokens and refresh tokens. This method supports opaque tokens and JSON Web Token (JWT)-based access tokens, including guest and named user JWT-based access tokens.

Namespace

Auth

Usage

When a client completes an authorization flow and is authorized to access Salesforce data, they're issued an access token, which the client can use to make authenticated requests for protected Salesforce resources. The client can also use refresh tokens to get more access tokens. If you don't want the client to access Salesforce data anymore, revoke its Salesforce tokens.

This class is distinct from the `Auth.AuthToken` class, which contains a method to revoke tokens issued by a third-party provider instead of Salesforce tokens.

IN THIS SECTION:

OauthToken Methods

OauthToken Methods

The following are methods for `OauthToken`.

IN THIS SECTION:

revokeToken(type, authToken)
Revokes Salesforce-issued OAuth tokens.

`revokeToken(type, authToken)`

Revokes Salesforce-issued OAuth tokens.

Signature

```
public static Boolean revokeToken(Auth.OauthTokenType type, String authToken)
```

Parameters

`type`
Type: Auth.OauthTokenType

Specifies the type of token to be revoked. To revoke an opaque access token, use the `ACCESS_TOKEN` value. To revoke a refresh token and any associated access tokens, use the `REFRESH_TOKEN` value. To revoke a refresh token and associated access tokens, use the `DELETE_TOKEN` value. To revoke a JSON Web Token (JWT)-based access token, use the `ORG_JWT` value.

`authToken`
Type: String

The access token (opaque or JWT-based), refresh token, or delete token issued by Salesforce.

Return Value

Type: Boolean

The method returns `true` if successful, and `false` if not. For invalid or expired tokens, the method returns a `NoDataFoundException` exception.

OauthTokenType Enum

Specifies the type of Salesforce-issued OAuth 2.0 token being revoked in the `OauthToken.revokeToken` method.

Enum Values

The following are the values of the `Auth.OauthTokenType` enum.

Value	Description
`ACCESS_TOKEN`	An opaque access token, which Salesforce grants to a client when it successfully completes an authorization flow. Salesforce grants opaque access tokens by default.
`DELETE_TOKEN`	A delete token, which can be queried and used to revoke refresh tokens and associated access tokens.
`REFRESH_TOKEN`	A refresh token, which Salesforce grants to a client as a result of the refresh token flow. Refresh tokens are used to get more access tokens.
`ORG_JWT`	A JSON Web Token (JWT)-based access token, which Salesforce grants to a client when it successfully completes an authorization flow. Salesforce grants JWT-based access tokens if you enable them for a connected app or external client app.

RegistrationHandler Interface

Salesforce provides the ability to use an authentication provider, such as Facebook© or Janrain©, for single sign-on into Salesforce.

Namespace

Auth

Usage

To set up single sign-on, you must create a class that implements `Auth.RegistrationHandler`. Classes implementing the `Auth.RegistrationHandler` interface are specified as the `Registration Handler` in authentication provider definitions, and enable single sign-on into Salesforce portals and organizations from third-party services such as Facebook. Using information from the authentication providers, your class must perform the logic of creating and updating user data as appropriate, including any associated account and contact records.

Note: During the user update process, you can use the `confirmUser()` method to ensure that users are correctly mapped between Salesforce and the third party. For more information, see the ConfirmUserRegistrationHandler Interface.

IN THIS SECTION:

RegistrationHandler Methods

Storing User Information and Getting Access Tokens

Auth.RegistrationHandler Example Implementation

Auth.RegistrationHandler Error Example

This example implements the `Auth.RegistrationHandler` interface and shows how to use a custom exception to display an error message in the URL of the page. If you don't use a custom exception, the error code and description appear in the URL and the error description appears on the page.

RegistrationHandler Methods

The following are methods for `RegistrationHandler`.

IN THIS SECTION:

createUser(portalId, userData)

Returns a User object using the specified portal ID and user information from the third party, such as the username and email address. The User object corresponds to the third party's user information. It can be a new user that hasn't been inserted in your org's database, or it can represent an existing user record in the database. If it's a new User object, Salesforce inserts a user record for you.

updateUser(userId, portalId, userData)

Updates the specified user's information. This method is called if the user has logged in before with the authentication provider and then logs in again.

`createUser(portalId, userData)`

Returns a User object using the specified portal ID and user information from the third party, such as the username and email address. The User object corresponds to the third party's user information. It can be a new user that hasn't been inserted in your org's database, or it can represent an existing user record in the database. If it's a new User object, Salesforce inserts a user record for you.

Signature

```
public User createUser(ID portalId, Auth.UserData userData)
```

Parameters

portalId
Type: ID

userData
Type: Auth.UserData

Return Value

Type: User

Usage

The *portalID* value can be null or an empty key if there's no portal configured with this provider.

updateUser(userId, portalId, userData)

Updates the specified user's information. This method is called if the user has logged in before with the authentication provider and then logs in again.

Signature

```
public Void updateUser(ID userId, ID portalId, Auth.UserData userData)
```

Parameters

userId
Type: ID

portalId
Type: ID

userData
Type: Auth.UserData

Return Value

Type: Void

Usage

The *portalID* value can be null or an empty key if there's no portal configured with this provider.

Storing User Information and Getting Access Tokens

The `Auth.UserData` class is used to store user information for `Auth.RegistrationHandler`. The third-party authentication provider can send back a large collection of data about the user, including their username, email address, locale, and so on. Frequently used data is converted into a common format with the `Auth.UserData` class and sent to the registration handler.

If the registration handler wants to use the rest of the data, the `Auth.UserData` class has an `attributeMap` variable. The attribute map is a map of strings (`Map<String, String>`) for the raw values of all the data from the third party. Because the map is `<String, String>`, values that the third party returns that aren't strings (like an array of URLs or a map) are converted into an appropriate string representation. The map includes everything returned by the third-party authentication provider, including the items automatically converted into the common format.

The constructor for `Auth.UserData` has the following syntax:

```
Auth.UserData(String identifier,
              String firstName,
              String lastName,
              String fullName,
              String email,
              String link,
              String userName,
              String locale,
              String provider,
              String siteLoginUrl,
              Map<String, String> attributeMap)
```

To learn about `Auth.UserData` properties, see Auth.UserData Class.

Note: You can only perform DML operations on additional sObjects in the same transaction with User objects under certain circumstances. For more information, see sObjects That Cannot Be Used Together in DML Operations.

For all authentication providers except Janrain, after a user is authenticated using a provider, the access token associated with that provider for this user can be obtained in Apex using the `Auth.AuthToken` Apex class. `Auth.AuthToken` provides two methods to retrieve access tokens. One is `getAccessToken`, which obtains a single access token. Use this method if the user ID is mapped to a single third-party user. If the user ID is mapped to multiple third-party users, use `getAccessTokenMap`, which returns a map of access tokens for each third-party user. For more information about authentication providers, see Authentication Providers in *Salesforce Help*.

When using Janrain as an authentication provider, you must use the Janrain `accessCredentials` dictionary values to retrieve the access token or its equivalent. Only some providers supported by Janrain provide an access token, while other providers use other fields. The Janrain `accessCredentials` fields are returned in the `attributeMap` variable of the `Auth.UserData` class. See the Janrain `auth_info` documentation for more information on `accessCredentials`.

Note: Not all Janrain account types return `accessCredentials`. Sometimes you must change your account type to receive the information.

To learn about the `Auth.AuthToken` methods, see Auth.AuthToken Class.

Auth.RegistrationHandler Example Implementation

This example implements the `Auth.RegistrationHandler` interface that creates as well as updates a standard user based on data provided by the authentication provider. Error checking has been omitted to keep the example simple.

```
global class StandardUserRegistrationHandler implements Auth.RegistrationHandler{
    global User createUser(Id portalId, Auth.UserData data) {
        User u = new User();
        Profile p = [SELECT Id FROM profile WHERE name='Standard User'];
        u.Username = data.username + '@salesforce.com';
        u.Email = data.email;
        u.LastName = data.lastName;
        u.FirstName = data.firstName;
```

```
        String alias = data.username;
        if(alias.length() > 8) {
            alias = alias.substring(0, 8);
        }
        u.Alias = alias;
        u.LanguageLocaleKey = data.attributeMap.get('language');
        u.LocaleSidKey = data.locale;
        u.EmailEncodingKey = 'UTF-8';
        u.TimeZoneSidKey = 'America/Los_Angeles';
        u.ProfileId = p.Id;
        return u;
    }

    global void updateUser(Id userId, Id portalId, Auth.UserData data) {
        User u = new User(id=userId);
        u.Username = data.username + '@salesforce.com';
        u.Email = data.email;
        u.LastName = data.lastName;
        u.FirstName = data.firstName;
        String alias = data.username;
        if(alias.length() > 8) {
            alias = alias.substring(0, 8);
        }
        u.Alias = alias;
        u.LanguageLocaleKey = data.attributeMap.get('language');
        u.TimeZoneSidKey = data.locale;
        update(u);
    }
}
```

The following example tests the above code.

```
@isTest
private class StandardUserRegistrationHandlerTest {
static testMethod void testCreateAndUpdateUser() {
    StandardUserRegistrationHandler handler = new StandardUserRegistrationHandler();
    Auth.UserData sampleData = new Auth.UserData('testId', 'testFirst', 'testLast',
        'testFirst testLast', 'testuser@example.org', null, 'testuserlong', 'en_US',
'facebook',
        null, new Map<String, String>{'language' => 'en_US'});
    User u = handler.createUser(null, sampleData);
    System.assertEquals('testuserlong@salesforce.com', u.username);
    System.assertEquals('testuser@example.org', u.email);
    System.assertEquals('testLast', u.lastName);
    System.assertEquals('testFirst', u.firstName);
    System.assertEquals('testuser', u.alias);
    insert(u);
    String uid = u.id;

    sampleData = new Auth.UserData('testNewId', 'testNewFirst', 'testNewLast',
        'testNewFirst testNewLast', 'testnewuser@example.org', null, 'testnewuserlong',
'en_US', 'facebook',
        null, new Map<String, String>{});
    handler.updateUser(uid, null, sampleData);
```

```
    User updatedUser = [SELECT username, email, firstName, lastName, alias FROM user WHERE
 id=:uid];
    System.assertEquals('testnewuserlong@salesforce.com', updatedUser.username);
    System.assertEquals('testnewuser@example.org', updatedUser.email);
    System.assertEquals('testNewLast', updatedUser.lastName);
    System.assertEquals('testNewFirst', updatedUser.firstName);
    System.assertEquals('testnewu', updatedUser.alias);
}
}
```

Auth.RegistrationHandler Error Example

This example implements the `Auth.RegistrationHandler` interface and shows how to use a custom exception to display an error message in the URL of the page. If you don't use a custom exception, the error code and description appear in the URL and the error description appears on the page.

To limit this example to the custom exception, some code was omitted.

```
global class RegHandler implements Auth.RegistrationHandler {

    class RegHandlerException extends Exception {}

        global User createUser(Id portalId, Auth.UserData data){
            List<Profile> profiles = [SELECT Id, Name, UserType FROM Profile WHERE Name =
 'Power User'];
            Profile profile = profiles.isEmpty() ? null : profiles[0];
            if(profile==null)
               throw new RegHandlerException('Cannot find the profile. For help, contact
your administrator.');
...
        }

        global void updateUser(Id userId, Id portalId, Auth.UserData data){
            User u = new User(id=userId);
            u.lastName = data.lastName;
            u.firstName = data.firstName;
            update(u);
        }
}
```

SamlJitHandler Interface

Use this interface to control and customize Just-in-Time user provisioning logic during SAML single sign-on.

Namespace

Auth

Usage

To use custom logic for user provisioning during SAML single sign-on, you must create a class that implements `Auth.SamlJitHandler`. This allows you to incorporate organization-specific logic (such as populating custom fields) when users

log in to Salesforce with single sign-on. Keep in mind that your class must perform the logic of creating and updating user data as appropriate, including any associated account and contact records.

In Salesforce, you specify your class that implements this interface in the `SAML JIT Handler` field in SAML Single Sign-On Settings. Make sure that the user you specify to run the class has "Manage Users" permission.

IN THIS SECTION:

SamlJitHandler Methods

SamlJitHandler Example Implementation

SamlJitHandler Methods

The following are methods for `SamlJitHandler`.

IN THIS SECTION:

createUser(samlSsoProviderId, communityId, portalId, federationId, attributes, assertion)

Returns a User object using the specified Federation ID. The User object corresponds to the user information. This object can be a new user that hasn't been inserted in the database or an existing user record in the database.

updateUser(userId, samlSsoProviderId, communityId, portalId, federationId, attributes, assertion)

Updates the specified user's information. This method is called if the user has logged in before with SAML single sign-on and then logs in again, or if your application is using the `Existing User Linking URL`.

createUser(samlSsoProviderId, communityId, portalId, federationId, attributes, assertion)

Returns a User object using the specified Federation ID. The User object corresponds to the user information. This object can be a new user that hasn't been inserted in the database or an existing user record in the database.

Signature

```
public User createUser(Id samlSsoProviderId, Id communityId, Id portalId, String
federationId, Map<String,String> attributes, String assertion)
```

Parameters

samlSsoProviderId

Type: Id

The ID of the SamlSsoConfig standard object.

communityId

Type: Id

The ID of the Experience Cloud site. This parameter can be `null` if you're not creating an Experience Cloud user.

portalId

Type: Id

The ID of the portal. This parameter can be `null` if you're not creating a portal user.

federationId
Type: String

The ID Salesforce expects to be used for this user.

attributes
Type: Map<String,String>

All attributes in the SAML assertion that were added to the default assertion; for example, custom attributes. Attributes are case-sensitive.

If the assertion is encrypted, the attribute map contains a decrypted assertion stored as a value with the key `Sfdc.SamlAssertion`.

assertion
Type: String

The default SAML assertion, base-64 encoded.

If the assertion is encrypted, this parameter is also encrypted. To access the decrypted assertion, see the `Sfdc.SamlAssertion` key in the attribute map.

Return Value

Type: User

A User sObject.

Usage

The *communityId* and *portalId* parameter values can be `null` or the associated keys can be empty if there's no Experience Cloud site or portal configured with this organization.

updateUser(userId, samlSsoProviderId, communityId, portalId, federationId, attributes, assertion)

Updates the specified user's information. This method is called if the user has logged in before with SAML single sign-on and then logs in again, or if your application is using the `Existing User Linking URL`.

Signature

```
public void updateUser(Id userId, Id samlSsoProviderId, Id communityId, Id portalId,
String federationId, Map<String,String> attributes, String assertion)
```

Parameters

userId
Type: Id

The ID of the Salesforce user.

samlSsoProviderId
Type: Id

The ID of the SamlSsoConfig object.

communityId
Type: Id

The ID of the Experience Cloud site. This type can be `null` if you're not updating an Experience Cloud user.

portalId
Type: Id

The ID of the portal. This type can be `null` if you're not updating a portal user.

federationId
Type: String

The ID Salesforce expects to be used for this user.

attributes
Type: Map<String,String>

All attributes in the SAML assertion that were added to the default assertion; for example, custom attributes. Attributes are case-sensitive.

If the assertion is encrypted, the attribute map also contains a decrypted assertion stored as a value with the key `Sfdc.SamlAssertion`.

assertion
Type: String

The default SAML assertion, base-64 encoded.

If the assertion is encrypted, this parameter is also encrypted. To access the decrypted assertion, see the `Sfdc.SamlAssertion` key in the attribute map.

Return Value

Type: void

SamlJitHandler Example Implementation

This is an example implementation of the `Auth.SamlJitHandler` interface. This code uses private methods to handle accounts and contacts (handleContact() and handleAccount()), which aren't included in this example.

```
global class StandardUserHandler implements Auth.SamlJitHandler {
    private class JitException extends Exception{}
    private void handleUser(boolean create, User u, Map<String, String> attributes,
        String federationIdentifier, boolean isStandard) {
        if(create && attributes.containsKey('User.Username')) {
            u.Username = attributes.get('User.Username');
        }
        if(create) {
            if(attributes.containsKey('User.FederationIdentifier')) {
                u.FederationIdentifier = attributes.get('User.FederationIdentifier');
            } else {
                u.FederationIdentifier = federationIdentifier;
            }
        }
        if(attributes.containsKey('User.ProfileId')) {
            String profileId = attributes.get('User.ProfileId');
            Profile p = [SELECT Id FROM Profile WHERE Id=:profileId];
```

```
            u.ProfileId = p.Id;
        }
        if(attributes.containsKey('User.UserRoleId')) {
            String userRole = attributes.get('User.UserRoleId');
            UserRole r = [SELECT Id FROM UserRole WHERE Id=:userRole];
            u.UserRoleId = r.Id;
        }
        if(attributes.containsKey('User.Phone')) {
            u.Phone = attributes.get('User.Phone');
        }
        if(attributes.containsKey('User.Email')) {
            u.Email = attributes.get('User.Email');
        }

 //More attributes here - removed for length

        //Handle custom fields here

        if(!create) {
            update(u);
        }
    }

    private void handleJit(boolean create, User u, Id samlSsoProviderId, Id communityId,
Id portalId,
        String federationIdentifier, Map<String, String> attributes, String assertion) {
        if(communityId != null || portalId != null) {
            String account = handleAccount(create, u, attributes);
            handleContact(create, account, u, attributes);
            handleUser(create, u, attributes, federationIdentifier, false);
        } else {
            handleUser(create, u, attributes, federationIdentifier, true);
        }
    }

    global User createUser(Id samlSsoProviderId, Id communityId, Id portalId,
        String federationIdentifier, Map<String, String> attributes, String assertion) {
        User u = new User();
        handleJit(true, u, samlSsoProviderId, communityId, portalId,
            federationIdentifier, attributes, assertion);
        return u;
    }

    global void updateUser(Id userId, Id samlSsoProviderId, Id communityId, Id portalId,
        String federationIdentifier, Map<String, String> attributes, String assertion) {
        User u = [SELECT Id FROM User WHERE Id=:userId];
        handleJit(false, u, samlSsoProviderId, communityId, portalId,
            federationIdentifier, attributes, assertion);
    }
}
```

SessionManagement Class

Contains methods for verifying users' identity, creating custom login flows, customizing security levels, and defining trusted IP ranges for a current session.

Namespace

Auth

IN THIS SECTION:

SessionManagement Methods

SessionManagement Methods

The following are methods for `SessionManagement`. All methods are static. Use these methods to customize your user identity verification flows, manage the use of time-based one-time password (TOTP) apps like Google Authenticator, or create custom login flows. Other methods validate a user's incoming IP address against trusted IP range settings for an organization or profile.

IN THIS SECTION:

finishLoginDiscovery(method, userId)
Finishes the My Domain Login Discovery login process.

finishLoginFlow()
Finish the Visualforce Page login flow process, and redirect the user to the default home page.

finishLoginFlow(startUrl)
Finish the Visualforce Page login flow process, and redirect the user to the specified start URL.

generateVerificationUrl(policy, description, destinationUrl)
Initiates a user identity verification flow with the verification method that the user registered with, and returns a URL to the identity verification screen. For example, if you have a custom Visualforce page that displays sensitive account details, you can prompt the user to verify identity before viewing it.

getCurrentSession()
Returns a map of attributes for the current session.

getLightningLoginEligibility(userId)
Returns the eligibility status for a user who's logging in with Lightning Login when you set up your org with My Domain and use the Login Discovery page type. Use this method to redirect the user to a custom login flow. For example, use after a login attempt to redirect the user to password flow if the user is ineligible for Lightning Login.

getQrCode()
Returns a map containing a URL to a quick response (QR) code and a time-based one-time password (TOTP) shared secret to configure authenticator apps or devices for multi-factor authentication (MFA).

getRequiredSessionLevelForProfile(profileId)
Indicates the required login security session level for the given profile.

ignoreForConcurrentSessionLimit(sessions)
This method is reserved for internal Salesforce use.

inOrgNetworkRange(ipAddress)

Indicates whether the given IP address is within the organization's trusted IP range according to the organization's Network Access settings.

isIpAllowedForProfile(profileId, ipAddress)

Indicates whether the given IP address is within the trusted IP range for the given profile.

setSessionLevel(level)

Sets the user's current session security level.

validateTotpTokenForKey(sharedKey, totpCode)

Deprecated. Use `validateTotpTokenForKey(totpSharedKey, totpCode, description)` instead.

validateTotpTokenForKey(totpSharedKey, totpCode, description)

Indicates whether a time-based one-time password (TOTP) code (token) is valid for the given shared key.

validateTotpTokenForUser(totpCode)

Deprecated. Use `validateTotpTokenForUser(totpCode, description)` instead.

validateTotpTokenForUser(totpCode, description)

Indicates whether a time-based one-time password (TOTP) code (token) is valid for the current user.

verifyDeviceFlow(userCode, startUrl)

Verifies the user code entered during the device authentication flow and redirects users to the OAuth approval page. If users aren't logged in, they must log in. After successful login, users are prompted to allow the device to access Salesforce data.

`finishLoginDiscovery(method, userId)`

Finishes the My Domain Login Discovery login process.

Signature

```
public static System.PageReference finishLoginDiscovery(Auth.LoginDiscoveryMethod
method, Id userId)
```

Parameters

`method`

Type: Auth.LoginDiscoveryMethod LoginDiscoveryMethod Enum

Verification method used with My Domain Login Discovery.

`userId`

Type: Id

ID used to log in the user. The user must be active.

Return Value

Type: System.PageReference

Usage

Include this method when implementing the `MyDomainLoginDiscoveryHandler` interface to direct users to an authentication mechanism, and then log them in. If users enter a username in the login page, they are sent to the password page for authentication.

If users are enrolled in Lightning Login, they are directed to the Salesforce Authenticator to authenticate. If users are SSO-enabled, they are sent to the suitable identity provider (IdP) to authenticate.

The calling user requires Manage Users permission. If the user passed in is frozen or inactive, the method throws an exception.

After implementing the `MyDomainLoginDiscoveryHandler` interface, register the Login Discovery handler from the My Domain Setup page. Under Authentication Configuration, select this handler from the list of Apex classes.

`finishLoginFlow()`

Finish the Visualforce Page login flow process, and redirect the user to the default home page.

Signature

```
public static System.PageReference finishLoginFlow()
```

Return Value

Type: System.PageReference

Usage

Include this method in the Apex controller of the Visualforce Page login flow when creating login flows programmatically. This method indicates that the login flow is finished and redirects the user to the Experience Cloud site's default home page. The login process runs in a restricted session until users complete the process. Calling this method indicates that the login flow is complete, lifts the restriction, and gives users full access to the Experience Cloud site.

`finishLoginFlow(startUrl)`

Finish the Visualforce Page login flow process, and redirect the user to the specified start URL.

Signature

```
public static System.PageReference finishLoginFlow(String startUrl)
```

Parameters

`startUrl`
Type: String

Path to the page that users see when they access the Experience Cloud site.

Return Value

Type: System.PageReference

Usage

Include this method in the Apex controller of the Visualforce Page login flow when creating login flows programmatically. This method indicates that the login flow is finished and redirects the user to the specified location in the Experience Cloud site. The login process runs in a restricted session until users complete the process. Calling this method indicates that the login flow is complete, lifts the restriction, and gives users full access to the Experience Cloud site.

`generateVerificationUrl(policy, description, destinationUrl)`

Initiates a user identity verification flow with the verification method that the user registered with, and returns a URL to the identity verification screen. For example, if you have a custom Visualforce page that displays sensitive account details, you can prompt the user to verify identity before viewing it.

Signature

```
public static String generateVerificationUrl(Auth.VerificationPolicy policy, String
description, String destinationUrl)
```

Parameters

policy
Type: Auth.VerificationPolicy

The session security policy required to initiate identity verification for the user's session. For example, if the policy is set to High Assurance level of session security, and the user's current session has the standard level of session security, the user's session is raised to high assurance after successful verification of identity. In the Setup user interface, this value is shown in the Triggered By column of Identity Verification History.

description
Type: String

The custom description that describes the activity requiring identity verification; for example, "Complete purchase and check out". This text appears to users when they verify their identity in Salesforce and, if they use Salesforce Authenticator version 2 or later, in the Salesforce Authenticator mobile app. In addition, in the Setup user interface, this text is shown in the Activity Message column of Identity Verification History.

destinationUrl
Type: String

The relative or absolute Salesforce URL that you want to redirect the user to after identity verification—for example, `/apex/mypage`. The user is redirected to *destinationUrl* when the identity verification flow is complete, regardless of success. For example, if a user chooses to not respond to the identity challenge and cancels it, the user is still redirected to *destinationUrl*. As a best practice, ensure that your code for this page manually checks that the security policy was satisfied (and the user didn't just manually type the URL in the browser). For example, if the *policy* is High Assurance, the target page checks that the user's session is high assurance before allowing access.

Return Value

Type: String

The URL where the user is redirected to verify identity.

Usage

- If the user is already registered to confirm identity using a time-based one-time password (TOTP), then the user is redirected to the one-time password identity verification flow and asked to provide a code.
- If the user isn't registered with any verification method (such as one-time password or Salesforce Authenticator version 2 or later), the user is prompted to download and verify identity using Salesforce Authenticator. The user can also choose a different verification method.

getCurrentSession()

Returns a map of attributes for the current session.

Signature

```
public static Map<String, String> getCurrentSession()
```

Return Value

Type: Map<String, String>

Usage

The map includes a `ParentId` value, which is the 18-character ID for the parent session, if one exists (for example, if the current session is for a canvas app). If the current session doesn't have a parent, this value is null. The map also includes the `LogoutUrl` assigned to the current session.

If you create an Apex test method that calls this method, the test fails with an error such as, "Unexpected Exception: Current session unavailable." An error occurs because there isn't a session in the context through which the test is being run.

When a session is reused, Salesforce updates the `LoginHistoryId` with the value from the most recent login.

Example

The following example shows the name-value pairs in a map returned by `getCurrentSession()`. Note that `UsersId` includes an "s" in the name to match the name of the corresponding field in the AuthSession object.

```
{
SessionId=0Ak###############,
UserType=Standard,
ParentId=0Ak###############,
NumSecondsValid=7200,
LoginType=SAML Idp Initiated SSO,
LoginDomain=null,
LoginHistoryId=0Ya###############,
Username=user@domain.com,
CreatedDate=Wed Jul 30 19:09:29 GMT 2014,
SessionType=Visualforce,
LastModifiedDate=Wed Jul 30 19:09:16 GMT 2014,
LogoutUrl=https://google.com,
SessionSecurityLevel=STANDARD,
UsersId=005###############,
SourceIp=1.1.1.1
}
```

getLightningLoginEligibility(userId)

Returns the eligibility status for a user who's logging in with Lightning Login when you set up your org with My Domain and use the Login Discovery page type. Use this method to redirect the user to a custom login flow. For example, use after a login attempt to redirect the user to password flow if the user is ineligible for Lightning Login.

Signature

```
public static Auth.LightningLoginEligibility getLightningLoginEligibility(Id userId)
```

Parameters

userId
Type: Id

ID of the user who is logging in.

Return Value

Type: Auth.LightningLoginEligibility

Returns the current eligibility status.

Example

```
Auth.LightningLoginEligibility eligibility =
    Auth.SessionManagement.getLightningLoginEligibility(id);
if (eligibility == Auth.LightningLoginEligibility.ELIGIBLE) {
    // success
}
```

getQrCode()

Returns a map containing a URL to a quick response (QR) code and a time-based one-time password (TOTP) shared secret to configure authenticator apps or devices for multi-factor authentication (MFA).

Signature

```
public static Map<String, String> getQrCode()
```

Return Value

Type: Map<String, String>

Usage

The QR code encodes the returned secret as well as the current user's username. The keys are `qrCodeUrl` and `secret`. Calling this method does not change any state for the user, nor does it read any state from the user. This method returns a brand new secret every time it is called, does not save that secret anywhere, and does not validate the TOTP token. The admin must explicitly save the values for the user after verifying a TOTP token with the secret.

The `secret` is a base32-encoded string of a 20-byte shared key.

Example

The following is an example of how to request the QR code.

```
public String getGetQRCode() {
        return getQRCode();
```

```
    }
    public String getQRCode() {
       Map<String, String> codeResult = Auth.SessionManagement.getQrCode();
       String result = 'URL: '+codeResult.get('qrCodeUrl') + ' SECRET:  ' +
codeResult.get('secret');
       return result;
    }
```

The following is an example of a returned map.

```
{qrCodeUrl=https://www.salesforce.com/secur/qrCode?w=200&h=200&t=tf&u=user%0000000000.com&s=AAAAA7B5BBBB5AAAAAAA66BBBB,

      secret=AAAAA7B5AAAAAA5BBBBBBBBB66AAA}
```

getRequiredSessionLevelForProfile(profileId)

Indicates the required login security session level for the given profile.

Signature

```
public static Auth.SessionLevel getRequiredSessionLevelForProfile(String profileId)
```

Parameters

profileId
Type: String

The 15-character profile ID.

Return Value

Type: Auth.SessionLevel

The session security level required at login for the profile with the ID *profileId*. You can customize the assignment of each level in Session Settings. For example, you can set the High Assurance level to apply only to users who authenticated with multi-factor authentication (MFA) or through a specific identity provider.

ignoreForConcurrentSessionLimit(sessions)

This method is reserved for internal Salesforce use.

Signature

```
public static Map<String,String> ignoreForConcurrentSessionLimit(Object sessions)
```

Parameters

sessions
Type: Object

Return Value

Type: Map<String, String>

inOrgNetworkRange(ipAddress)

Indicates whether the given IP address is within the organization's trusted IP range according to the organization's Network Access settings.

Signature

```
public static Boolean inOrgNetworkRange(String ipAddress)
```

Parameters

ipAddress
Type: String

The IP address to validate.

Return Value

Type: Boolean

Usage

If a trusted IP range is not defined, this returns `false`, and throws an exception if the IP address is not valid.

Trusted IP Range Exists?	User is in the Trusted IP Range?	Return Value
Yes	Yes	`true`
Yes	No	`false`
No	N/A	`false`

isIpAllowedForProfile(profileId, ipAddress)

Indicates whether the given IP address is within the trusted IP range for the given profile.

Signature

```
public static Boolean isIpAllowedForProfile(String profileId, String ipAddress)
```

Parameters

profileId
Type: String

The 15-character alphanumeric string for the current user's profile ID.

ipAddress
Type: String

The IP address to validate.

Return Value

Type: Boolean

Usage

If a trusted IP range is not defined, this returns `true`, and throws an exception if the IP address is not valid or if the profile ID is not valid.

Trusted IP Range Exists?	User is in the Trusted IP Range?	Return Value
Yes	Yes	`true`
Yes	No	`false`
No	N/A	`true`

`setSessionLevel(level)`

Sets the user's current session security level.

Signature

```
public static Void setSessionLevel(Auth.SessionLevel level)
```

Parameters

`level`
Type: Auth.SessionLevel

The session security level to assign to the user. The meaning of each level can be customized in the Session Settings for each organization, such as setting the High Assurance level to apply only to users who authenticated with multi-factor authentication (MFA) or through a specific identity provider.

Return Value

Type: Void

Usage

This setting affects the session level of all sessions associated with the current session, such as Visualforce or UI access.

If you create an Apex test method that calls this method, the test fails with an error such as, "Unexpected Exception: Current session unavailable." An error occurs because there isn't a session in the context through which the test is being run.

Example

The following is an example class for setting the session level.

```
public class RaiseSessionLevel{
    public void setLevelHigh() {
        Auth.SessionManagement.setSessionLevel(Auth.SessionLevel.HIGH_ASSURANCE);
    }
    public void setLevelStandard() {
        Auth.SessionManagement.setSessionLevel(Auth.SessionLevel.STANDARD);
```

```
    }
}
```

validateTotpTokenForKey(sharedKey, totpCode)

Deprecated. Use `validateTotpTokenForKey(totpSharedKey, totpCode, description)` instead.

Signature

```
public static Boolean validateTotpTokenForKey(String sharedKey, String totpCode)
```

Parameters

sharedKey
Type: String

The shared (secret) key. The *sharedKey* must be a base32-encoded string of a 20-byte shared key.

totpCode
Type: String

The time-based one-time password (TOTP) code to validate.

Return Value

Type: Boolean

Usage

If the key is invalid or doesn't exist, this method throws an invalid parameter value exception or a no data found exception, respectively. If the current user exceeds the maximum of 10 token validation attempts, this method throws a security exception.

validateTotpTokenForKey(totpSharedKey, totpCode, description)

Indicates whether a time-based one-time password (TOTP) code (token) is valid for the given shared key.

Signature

```
public static Boolean validateTotpTokenForKey(String totpSharedKey, String totpCode,
String description)
```

Parameters

totpSharedKey
Type: String

The shared (secret) key. The *totpSharedKey* must be a base32-encoded string of a 20-byte shared key.

totpCode
Type: String

The time-based one-time password (TOTP) code to validate.

description
Type: String

The custom description that describes the activity requiring identity verification; for example, "Complete purchase and check out". In the Setup user interface, this text is shown in the Activity Message column of Identity Verification History. The *description* must be 128 characters or fewer. If you provide a value that's longer, it's truncated to 128 characters.

Return Value

Type: Boolean

Usage

If the key is invalid or doesn't exist, this method throws an invalid parameter value exception or a no data found exception, respectively. If the current user exceeds the maximum of 10 token validation attempts, this method throws a security exception.

validateTotpTokenForUser(totpCode)

Deprecated. Use `validateTotpTokenForUser(totpCode, description)` instead.

Signature

```
public static Boolean validateTotpTokenForUser(String totpCode)
```

Parameters

totpCode
Type: String

The time-based one-time password (TOTP) code to validate.

Return Value

Type: Boolean

Usage

If the current user does not have a TOTP code, this method throws an exception. If the current user has attempted too many validations, this method throws an exception.

validateTotpTokenForUser(totpCode, description)

Indicates whether a time-based one-time password (TOTP) code (token) is valid for the current user.

Signature

```
public static Boolean validateTotpTokenForUser(String totpCode, String description)
```

Parameters

totpCode
Type: String

The time-based one-time password (TOTP) code to validate.

description
Type: String

The custom description that describes the activity requiring identity verification; for example, "Complete purchase and check out". This text appears to users when they verify their identity in Salesforce and, if they use Salesforce Authenticator version 2 or later, in the Salesforce Authenticator mobile app. In addition, in the Setup user interface, this text is shown in the Activity Message column of Identity Verification History. The *description* must be 128 characters or fewer. If you provide a value that's longer, it's truncated to 128 characters.

Return Value

Type: Boolean

Usage

If the current user does not have a TOTP code, or if the current user has attempted too many validations, this method throws an exception.

verifyDeviceFlow(userCode, startUrl)

Verifies the user code entered during the device authentication flow and redirects users to the OAuth approval page. If users aren't logged in, they must log in. After successful login, users are prompted to allow the device to access Salesforce data.

Signature

```
public static System.PageReference verifyDeviceFlow(String userCode, String startUrl)
```

Parameters

userCode
Type: String

Human-readable user code provided to the user by Salesforce. The user must enter this code at the verification URL to approve device access to Salesforce data.

startURL
Type: String

The URL for the page that the user is redirected to after successful login and approval of the device to access Salesforce data. If you don't specify a start URL, the user is redirected to the Home page.

Return Value

Type:System.PageReference

Usage

Include this method in the Apex controller when creating a custom Visualforce User Code Verification page for the OAuth 2.0 device authentication flow. This method verifies the user code, prompts the user to log in as needed, and prompts the user to allow the device access to Salesforce data. Upon successful verification and authentication, the user is redirected to the page defined by the start URL.

SessionLevel Enum

An `Auth.SessionLevel` enum value is used by the `SessionManagement.setSessionLevel` method.

Namespace

Auth

Enum Values

Value	Description
`LOW`	The user's security level for the current session meets the lowest requirements. Note: This low level is not available, nor used, in the Salesforce UI. User sessions through the Salesforce UI are either standard or high assurance. You can set this level using the API, but users assigned this level will experience unpredictable and reduced functionality in their Salesforce organization.
`STANDARD`	The user's security level for the current session meets the Standard requirements set in the current organization Session Security Levels.
`HIGH_ASSURANCE`	The user's security level for the current session meets the High Assurance requirements set in the current organization Session Security Levels.

Usage

With session-level security, you control user access to features that support it, such as connected apps and reporting. For example, you can customize an organization's Session Settings to require users to log in with multi-factor authentication (MFA) to get a High Assurance session. Then, you can restrict access to a specific connected app by requiring a High Assurance session level in the settings for the connected app.

TokenValidationResult Class

Contains methods that describe the result of the token validation performed by a token exchange handler using the `validateIncomingToken` method in the `Auth.Oauth2TokenExchangeHandler` class during the OAuth 2.0 token exchange flow.

Namespace

Auth

Usage

For a full example implementation that shows how to get information from the `TokenValidationResult` class, see OAuth 2.0 Token Exchange Handler Examples.

Example

Here's is an example of the `Auth.TokenValidationResult` class.

```
global class TokenValidationResult {
    global TokenValidationResult(Boolean valid) { this.isValid = valid; }

    global TokenValidationResult(Boolean isValid, Object data, Auth.UserData userData,
        String token, Auth.OAuth2TokenExchangeType tokenType, String customErrorMsg) {

        this.isValid = isValid;
        this.data = data;
        this.userData = userData;
        this.token = token;
        this.tokenType = tokenType;
        this.customErrorMsg = customErrorMsg;

    }

    global Boolean isValid;
    global Object data;
    global Auth.UserData userData;
    global String token;
    global Auth.OAuth2TokenExchangeType tokenType; //Enum
    global String customErrorMsg; //Custom error message that's returned to the client if
 token validation fails

    global Boolean isValid(){
        return isValid;
    }
    global Object getData(){
        return data;
    }
    global Auth.UserData getUserData(){
        return userData;
    }
    global String getToken(){
        return token;
    }
    global OAuth2TokenExchangeType getTokenType(){
        return tokenType;
    }
    global String getCustomErrorMessage(){
        return customErrorMsg;
    }
}
```

IN THIS SECTION:

TokenValidationResult Constructors

TokenValidationResult Properties

TokenValidationResult Methods

TokenValidationResult Constructors

The following are constructors for `TokenValidationResult`.

IN THIS SECTION:

TokenValidationResult(isValid, data, userData, token, tokenType, customErrorMsg)

Creates an instance of the `Auth.TokenValidationResult` class to describe the result of token validation performed during the OAuth 2.0 token exchange flow.

TokenValidationResult(valid)

Creates an instance of the `Auth.TokenValidationResult` class to describe a valid token validation result during the OAuth 2.0 token exchange flow.

`TokenValidationResult(isValid, data, userData, token, tokenType, customErrorMsg)`

Creates an instance of the `Auth.TokenValidationResult` class to describe the result of token validation performed during the OAuth 2.0 token exchange flow.

Signature

```
public TokenValidationResult(Boolean isValid, Object data, Auth.UserData userData,
String token, Auth.OAuth2TokenExchangeType tokenType, String customErrorMsg)
```

Parameters

`isValid`
Type: Boolean

If `true`, the token is valid.

`data`
Type: Object

Stores custom data that isn't stored in `userData`.

`userData`
Type: Auth.UserData

Stores information about a Salesforce user.

`token`
Type: String

The token from the external identity provider.

`tokenType`
Type: Auth.OAuth2TokenExchangeType

The type of token from the external identity provider.

`customErrorMsg`
Type: String

A custom error message that's returned if the token validation fails.

`TokenValidationResult(valid)`

Creates an instance of the `Auth.TokenValidationResult` class to describe a valid token validation result during the OAuth 2.0 token exchange flow.

Signature

```
public TokenValidationResult(Boolean valid)
```

Parameters

`valid`
Type: Boolean

Indicates a valid token validation result.

TokenValidationResult Properties

The following are properties for `TokenValidationResult`.

IN THIS SECTION:

customErrorMsg
A custom error message that's returned if token validation fails.

data
Contains information about the user that isn't stored in the `Auth.UserData` class, such as information obtained via callouts to the external identity provider.

isValid
Indicates whether the token is valid or not, based on the custom validation logic in your token exchange handler.

token
The token from the external identity provider.

tokenType
The type of token from the external identity provider. It can be an access token, refresh token, ID token, SAML 2.0 assertion, or a JSON Web Token (JWT).

userData
Information about the user that's obtained from the identity provider's token.

`customErrorMsg`

A custom error message that's returned if token validation fails.

Signature

```
public String customErrorMsg {get; set;}
```

Property Value

Type: String

data

Contains information about the user that isn't stored in the `Auth.UserData` class, such as information obtained via callouts to the external identity provider.

Signature

```
public Object data {get; set;}
```

Property Value

Type: Object

isValid

Indicates whether the token is valid or not, based on the custom validation logic in your token exchange handler.

Signature

```
public Boolean isValid {get; set;}
```

Property Value

Type: Boolean

token

The token from the external identity provider.

Signature

```
public String token {get; set;}
```

Property Value

Type: String

tokenType

The type of token from the external identity provider. It can be an access token, refresh token, ID token, SAML 2.0 assertion, or a JSON Web Token (JWT).

Signature

```
public Auth.OAuth2TokenExchangeType tokenType {get; set;}
```

Property Value

Type: Auth.OAuth2TokenExchangeType

`userData`

Information about the user that's obtained from the identity provider's token.

Signature

```
public Auth.UserData userData {get; set;}
```

Property Value

Type: Auth.UserData

TokenValidationResult Methods

The following are methods for `TokenValidationResult`.

IN THIS SECTION:

getCustomErrorMessage()
Retrieves the `CustomErrorMsg` that's returned when token validation fails.

getData()
Retrieves data from the identity provider token. This data can include custom data that isn't stored in the `userData` property.

getToken()
Retrieves the token that was passed from the external identity provider.

getTokenType()
Retrieves the type of token that was passed from the external identity provider.

getUserData()
Retrieves information about the user. The user information can be obtained from the identity provider's token or from callouts to the identity provider, if applicable.

isValid
Indicates whether the token is valid or not, based on the custom validation logic in your token exchange handler.

`getCustomErrorMessage()`

Retrieves the `CustomErrorMsg` that's returned when token validation fails.

Signature

```
public String getCustomErrorMessage()
```

Return Value

Type: String

`getData()`

Retrieves data from the identity provider token. This data can include custom data that isn't stored in the `userData` property.

Signature

```
public Object getData()
```

Return Value

Type: Object

`getToken()`

Retrieves the token that was passed from the external identity provider.

Signature

```
public String getToken()
```

Return Value

Type: String

`getTokenType()`

Retrieves the type of token that was passed from the external identity provider.

Signature

```
public Auth.OAuth2TokenExchangeType getTokenType()
```

Return Value

Type: Auth.OAuth2TokenExchangeType

`getUserData()`

Retrieves information about the user. The user information can be obtained from the identity provider's token or from callouts to the identity provider, if applicable.

Signature

```
public Auth.UserData getUserData()
```

Return Value

Type: Auth.UserData

`isValid`

Indicates whether the token is valid or not, based on the custom validation logic in your token exchange handler.

Signature

```
public Boolean isValid {get; set;}
```

Property Value

Type: Boolean

UserData Class

Stores user information for `Auth.RegistrationHandler`.

Namespace

Auth

IN THIS SECTION:

UserData Constructors

UserData Properties

UserData Constructors

The following are constructors for `UserData`.

IN THIS SECTION:

UserData(identifier, firstName, lastName, fullName, email, link, userName, locale, provider, siteLoginUrl, attributeMap)
Creates a new instance of the `Auth.UserData` class using the specified arguments.

`UserData(identifier, firstName, lastName, fullName, email, link, userName, locale, provider, siteLoginUrl, attributeMap)`

Creates a new instance of the `Auth.UserData` class using the specified arguments.

Signature

```
public UserData(String identifier, String firstName, String lastName, String fullName,
String email, String link, String userName, String locale, String provider, String
siteLoginUrl, Map<String,String> attributeMap)
```

Parameters

identifier
Type: String

An identifier from the third party for the authenticated user, such as the Facebook user number or the Salesforce user ID.

firstName
Type: String

The first name of the authenticated user, according to the third party.

lastName
Type: String

The last name of the authenticated user, according to the third party.

fullName
Type: String

The full name of the authenticated user, according to the third party.

email
Type: String

The email address of the authenticated user, according to the third party.

link
Type: String

A stable link for the authenticated user such as `https://www.facebook.com/MyUsername`.

userName
Type: String

The username of the authenticated user in the third party.

locale
Type: String

The standard locale string for the authenticated user.

provider
Type: String

The service used to log in, such as Facebook or Janrain.

siteLoginUrl
Type: String

The site login page URL passed in if used with a site; `null` otherwise.

attributeMap
Type: Map<String, String>

A map of data from the third party, in case the handler has to access non-standard values. For example, when using Janrain as a provider, the fields Janrain returns in its `accessCredentials` dictionary are placed into the `attributeMap`. These fields vary by provider.

UserData Properties

The following are properties for `UserData`.

IN THIS SECTION:

identifier

An identifier from the third party for the authenticated user, such as the Facebook user number or the Salesforce user ID.

firstName

The first name of the authenticated user, according to the third party.

lastName
The last name of the authenticated user, according to the third party.

fullName
The full name of the authenticated user, according to the third party.

email
The email address of the authenticated user, according to the third party.

link
A stable link for the authenticated user such as `https://www.facebook.com/MyUsername`.

username
The username of the authenticated user in the third party.

locale
The standard locale string for the authenticated user.

provider
The service used to log in, such as Facebook or Janrain.

siteLoginUrl
The site login page URL passed in if used with a site; `null` otherwise.

attributeMap
A map of data from the third party, in case the handler has to access non-standard values. For example, when using Janrain as a provider, the fields Janrain returns in its `accessCredentials` dictionary are placed into the `attributeMap`. These fields vary by provider.

`identifier`

An identifier from the third party for the authenticated user, such as the Facebook user number or the Salesforce user ID.

Signature

```
public String identifier {get; set;}
```

Property Value

Type: String

`firstName`

The first name of the authenticated user, according to the third party.

Signature

```
public String firstName {get; set;}
```

Property Value

Type: String

lastName

The last name of the authenticated user, according to the third party.

Signature

```
public String lastName {get; set;}
```

Property Value

Type: String

fullName

The full name of the authenticated user, according to the third party.

Signature

```
public String fullName {get; set;}
```

Property Value

Type: String

email

The email address of the authenticated user, according to the third party.

Signature

```
public String email {get; set;}
```

Property Value

Type: String

link

A stable link for the authenticated user such as `https://www.facebook.com/MyUsername`.

Signature

```
public String link {get; set;}
```

Property Value

Type: String

username

The username of the authenticated user in the third party.

Signature

```
public String username {get; set;}
```

Property Value

Type: String

`locale`

The standard locale string for the authenticated user.

Signature

```
public String locale {get; set;}
```

Property Value

Type: String

`provider`

The service used to log in, such as Facebook or Janrain.

Signature

```
public String provider {get; set;}
```

Property Value

Type: String

`siteLoginUrl`

The site login page URL passed in if used with a site; `null` otherwise.

Signature

```
public String siteLoginUrl {get; set;}
```

Property Value

Type: String

`attributeMap`

A map of data from the third party, in case the handler has to access non-standard values. For example, when using Janrain as a provider, the fields Janrain returns in its `accessCredentials` dictionary are placed into the `attributeMap`. These fields vary by provider.

Signature

```
public Map<String, String> attributeMap {get; set;}
```

Property Value

Type: Map<String, String>

VerificationAction Enum

Indicates the method that you use to send a one-time password (OTP) to a user during the headless passwordless login flow.

Usage

Use this enum to specify the user's method of receiving a one-time password when you implement the `Auth.HeadlessUserDiscoveryHandler` interface.

Enum Values

The following are the values of the `Auth.VerificationAction` enum.

Value	Description
`EMAIL`	Indicates that the user is verifying their identity with email.
`SMS`	Indicates that the user is verifying their identity with SMS.

VerificationMethod Enum

Contains the different ways users can identify themselves when logging in. You can use it to implement mobile-centric passwordless login pages and to self-register (and deregister) verification methods.

Usage

The enum value is an argument in `System.Site.passwordlessLogin`, `System.UserManagement.registerVerificationMethod`, and `System.UserManagement.deregisterVerificationMethod` on page 4001 methods. The value indicates the method used to verify a user's identity.

Enum Values

The following are the values of the `Auth.VerificationMethod` enum.

Value	Description
`BUILT_IN_AUTHENTICATOR`	Identity verified with a built-in authenticator.
`EMAIL`	Identity verified with a verification code sent in an email message.
`PASSWORD`	Identity verified with a password.

Value	Description
SALESFORCE_AUTHENTICATOR	Identity verified by Salesforce Authenticator.
SECURITY_KEY	Identity verified by a WebAuthn-compatible physical security key. Includes all security keys registered or used after Summer '22.
SMS	Identity verified with a verification code sent via SMS message.
TOTP	Identity verified with a time-based one-time password (TOTP).
U2F	Identity verified by a U2F physical security key, such as a YubiKey. Note: For U2F security keys registered or used after Summer '22, use SECURITY_KEY instead.

VerificationPolicy Enum

The `Auth.VerificationPolicy` enum contains an identity verification policy value used by the `SessionManagement.generateVerificationUrl` method.

Usage

The enum value is an argument in the `SessionManagement.generateVerificationUrl` method. The value indicates the session security policy required to initiate identity verification for the user's session.

Enum Values

The `Auth.VerificationPolicy` enum has this value.

Value	Description
HIGH_ASSURANCE	The security level for the user's current session must be High Assurance.

VerificationResult Class

Contains the result of a verification challenge that you invoke when you create your own Verify page. The challenge can be initiated by either the `System.UserManagement.verifyPasswordlessLogin` or `System.UserManagement.verifySelfRegistration` method.

Namespace

Auth

Usage

When users sign up for or log in to your Experience Cloud site with an email address or phone number, Salesforce sends them a verification code. At the same time, Salesforce generates the Verify page for users to enter the code to verify their identity. You can replace the Salesforce-generated Verify page with one that you create with Visualforce. Then invoke the verification challenge and, if the verification

code is entered correctly, log in the user. For sign-up, you use the `System.UserManagement.verifySelfRegistration` method. For passwordless login, you use the System.UserManagement.verifyPasswordlessLogin method. The methods return the verification result, which contains the message displayed as a result of the challenge. This message also indicates whether the challenge is successful and where to direct the user when the verification code is entered correctly.

Example

This code contains the result of a verification challenge that registers a new user.

```
String id = System.UserManagement.initSelfRegistration
          (Auth.VerificationMethod.SMS, user);
        Auth.VerificationResult res = System.UserManagement.verifySelfRegistration
          (Auth.VerificationMethod.SMS, id, '123456', null);
        if(res.success == true){
     //redirect
}
```

IN THIS SECTION:

VerificationResult Constructor

VerificationResult Properties

VerificationResult Method

VerificationResult Constructor

`VerificationResult` has the following constructor.

IN THIS SECTION:

VerificationResult(redirect, success, message)

Creates an instance of the `VerificationResult` class that contains the verification result from `System.UserManagement.verifySelfRegistration`.

`VerificationResult(redirect, success, message)`

Creates an instance of the `VerificationResult` class that contains the verification result from `System.UserManagement.verifySelfRegistration`.

Signature

```
public VerificationResult(System.PageReference redirect, Boolean success, String message)
```

Parameters

redirect

Type: System.PageReferenceSystem.PageReference

Where user is directed upon successful verification.

success

Type: Boolean

Indicates whether verification succeeded.

message
Type: String

Message that displays as a result of a verification challenge.

VerificationResult Properties

The following are properties for `VerificationResult`.

IN THIS SECTION:

message
Message that displays as a result of a verification challenge. SUCCESS if the identity verification is successful. Other values are FAILURE, PENDING, RATE_LIMITED, or FAILURE_REPORT.

redirect
Where the user is directed after entering the verification code successfully, for example, the Experience Cloud site's home page or location specified by the start URL.

success
The verification challenge is successful.

message

Message that displays as a result of a verification challenge. SUCCESS if the identity verification is successful. Other values are FAILURE, PENDING, RATE_LIMITED, or FAILURE_REPORT.

Signature

```
public String message {get; set;}
```

Property Value

Type: String

redirect

Where the user is directed after entering the verification code successfully, for example, the Experience Cloud site's home page or location specified by the start URL.

Signature

```
public System.PageReference redirect {get; set;}
```

Property Value

Type: System.PageReferenceSystem.PageReference

success

The verification challenge is successful.

Signature

```
public Boolean success {get; set;}
```

Property Value

Type: Boolean

VerificationResult Method

`VerificationResult` has the following method.

IN THIS SECTION:

clone()
Duplicates the Auth.VerificationResult object.

clone()

Duplicates the Auth.VerificationResult object.

Signature

```
public Object clone()
```

Return Value

Type: VerificationResult

Auth Exceptions

The `Auth` namespace contains some exception classes.

All exception classes support built-in methods for returning the error message and exception type. See Exception Class and Built-In Exceptions.

The `Auth` namespace contains the following exception.

Exception	Description
`Auth.AuthProviderPluginException`	Throw this exception to indicate that an error occurred when using the auth provider plug-in. Use to display a custom error message to the user. To get the error message and write it to debug log, use the `String getMessage()`.
`Auth.ConnectedAppPluginException`	Throw this exception to indicate that an error occurred while running the custom behavior for a connected app. To get the error message and write it to debug log, use the `String getMessage()`.

Exception	Description
`Auth.DiscoveryCustomErrorException`	Throw this exception to customize error messages that appear on Discovery logins and Configurable Self-Registration pages. An error message can have up to 200 characters. Use custom error exceptions to localize error messages. Include this exception in: • `Auth.MyDomainLoginDiscoveryHandler` to show a custom error message on the My Domain login page • `Auth.LoginDiscoveryHandler` to show an error message on the Experience Cloud site login page • `Auth.ConfigurableSelfRegHandler` to show an error message on the Experience Cloud site self-registration Verify page The Verify page shows up if you configured self-registration with either an **Email** or **Text Message** verification method. If you didn't set up sign-up with a verification method, the error message appears on the self-registration page. To get the error message and write it to debug log, use the `String getMessage()`.
`Auth.JWTBearerTokenExchange.JWTBearerTokenExchangeException`	Throw this exception to indicate a problem with the response from the token endpoint in the JWTBearerTokenExchange class. This exception occurs during the OAuth 2.0 JWT bearer token flow when the HTTP response: • Fails to return an access token • Isn't in JSON format • Returns a response code other than a 200 "OK" success code To get the error message and write it to debug log, use the `String getMessage()`.
`Auth.JWTValidationException`	Throws this exception to indicate failure to validate a JWT using methods in the `JWTUtil` class. This exception occurs during the OAuth 2.0 token exchange flow in these scenarios. • Can't parse the JWT • Can't validate the JWT using a certificate, a public key, or the remote keys endpoint, depending on which method you use
`Auth.LoginDiscoveryException`	Throw this exception to indicate that an error occurred when executing the Login Discovery handler. For an example, see LoginDiscoveryHandler Example Implementation. To get the error message and write it to debug log, use the `String getMessage()`.
`Auth.VerificationException`	Throw this exception to trigger verification based on the passed-in policy. You can throw this exception in an Apex trigger or Visualforce controller. The system automatically sends you to the verification endpoint, if possible. Note: You can't catch this exception. The exception immediately triggers the verification.

Examples

This example uses `AuthProviderPluginException` to throw a custom exception in a custom authentication provider implementation. Use this exception if you want the end user to see a specific message, passing in the error message as a parameter. If you use another exception, users see a standard Salesforce error message.

```
global override Auth.OAuthRefreshResult refresh(Map<string,string>
authProviderConfiguration,String refreshToken){
            HttpRequest req = new HttpRequest();
            String accessToken = null;
            String error  = null;
            try {

            // DEVELOPER TODO: Make a refresh token flow using refreshToken passed
            // in as an argument to get the new access token
            // accessToken = ...
            } catch (System.CalloutException e) {
            error = e.getMessage();
            }
            catch(Exception e) {
            error = e.getMessage();
            throw new Auth.AuthProviderPluginException('My custom error');
            }

            return new Auth.OAuthRefreshResult(accessToken,refreshToken, error);

            }
```

This example uses `Auth.VerificationException` to trigger verification if a user attempts to create an account without a high assurance session.

```
trigger testTrigger on Account (before insert) {
    Map<String, String> sessionMap = auth.SessionManagement.getCurrentSession();
    if(!sessionMap.get('SessionSecurityLevel').equals('HIGH_ASSURANCE')) {
        throw new Auth.VerificationException(
            Auth.VerificationPolicy.HIGH_ASSURANCE, 'Insert Account');
    }
}
```

Cache Namespace

The `Cache` namespace contains methods for managing the platform cache.

The following are the classes in the `Cache` namespace.

IN THIS SECTION:

CacheBuilder Interface

An interface for safely retrieving and removing values from a session or org cache. Use the interface to generate a value that you want to store in the cache. The interface checks for cache misses, which means you no longer need to check for null cache values yourself.

Org Class

Use the `Cache.Org` class to add, retrieve, and manage values in the org cache. Unlike the session cache, the org cache is not tied to any session and is available to the organization across requests and to all users.

OrgPartition Class

Contains methods to manage cache values in the org cache of a specific partition. Unlike the session cache, the org cache is not tied to any session. It's available to the organization across requests and to all users.

Partition Class

Base class of `Cache.OrgPartition` and `Cache.SessionPartition`. Use the subclasses to manage the cache partition for org caches and session caches.

Session Class

Use the `Cache.Session` class to add, retrieve, and manage values in the session cache. The session cache is active as long as the user's Salesforce session is valid (the user is logged in, and the session is not expired).

SessionPartition Class

Contains methods to manage cache values in the session cache of a specific partition.

Cache Exceptions

The `Cache` namespace contains exception classes.

Visibility Enum

Use the `Cache.Visibility` enumeration in the `Cache.Session` or `Cache.Org` methods to indicate whether a cached value is visible only in the value's namespace or in all namespaces.

SEE ALSO:

Apex Developer Guide: Platform Cache

CacheBuilder Interface

An interface for safely retrieving and removing values from a session or org cache. Use the interface to generate a value that you want to store in the cache. The interface checks for cache misses, which means you no longer need to check for null cache values yourself.

Namespace

Cache

IN THIS SECTION:

CacheBuilder Methods

CacheBuilder Example Implementation

SEE ALSO:

Apex Developer Guide: Safely Cache Values with the CacheBuilder Interface

CacheBuilder Methods

The following are methods for `CacheBuilder`.

IN THIS SECTION:

doLoad(var)
Contains the logic that builds a cached value. You don't call this method directly. Instead, it's called indirectly when you reference the class that implements the `CacheBuilder` interface.

doLoad(var)

Contains the logic that builds a cached value. You don't call this method directly. Instead, it's called indirectly when you reference the class that implements the `CacheBuilder` interface.

Signature

```
public Object doLoad(String var)
```

Parameters

var

Type: String

A case-sensitive string value used to build a cached value. This parameter is also used as part of the unique key that identifies the cached value.

Return Value

Type: Object

The value that was cached. Cast the return value to the appropriate type.

CacheBuilder Example Implementation

This example creates a class called `UserInfoCache` that implements the `CacheBuilder` interface. The class caches the results of a SOQL query run against the User object.

```
class UserInfoCache implements Cache.CacheBuilder {
    public Object doLoad(String userid) {
        User u = (User)[SELECT Id, IsActive, username FROM User WHERE id =: userid];
        return u;
    }
}
```

This example gets a cached User record based on a user ID. If the value exists in the org cache, it is returned. If the value doesn't exist, the `doLoad(String var)` method is re-executed, and the new value is cached and returned.

```
User batman = (User) Cache.Org.get(UserInfoCache.class, '00541000000ek4c');
```

Org Class

Use the `Cache.Org` class to add, retrieve, and manage values in the org cache. Unlike the session cache, the org cache is not tied to any session and is available to the organization across requests and to all users.

Namespace

Cache

Usage

Cache Key Format

This table lists the format of the key parameter that some methods in this class take, such as `put`, `get`, and `contains`.

Key Format	Description
`namespace.partition.key`	Fully qualified key name.
`key`	Refers to a partition marked as default when the *`namespace.partition`* prefix is omitted.
`local.`*`partition.key`*	Use the `local` prefix to refer to the org's namespace when the org doesn't have a namespace defined. If the org has a namespace defined, the `local` prefix also refers to that org's namespace.

Note:

- If no default partition is specified in the org, calling a cache method without fully qualifying the key name causes a `Cache.Org.OrgCacheException` to be thrown.
- The `local` prefix in an installed managed package refers to the namespace of the subscriber org and not the package's namespace. The cache `put` calls aren't allowed in a partition that the invoking class doesn't own.

Example

This class is the controller for a sample Visualforce page (shown in the subsequent code sample). The cached values are initially added to the cache by the `init()` method, which the Visualforce page invokes when it loads through the `action` attribute. The cache keys don't contain the `namespace.partition` prefix. They all refer to the default partition in your org. To run this sample, create a partition and mark it as default.

The Visualforce page contains four output components. These components call `get` methods on the controller that returns the following values from the cache: a date, data based on the `MyData` inner class, a counter, a text value, and a list. The size of the list is also returned.

The Visualforce page also contains two buttons. The Rerender button invokes the `go()` method on the controller. This method increases the values of the counter and the custom data in the cache. When you click **Rerender**, the two counters increase by one each time. The `go()` method retrieves the values of these counters from the cache, increments their values by one, and stores them again in the cache.

The Remove datetime Key button deletes the date-time value (with key `datetime`) from the cache. As a result, the value next to `Cached datetime:` is cleared on the page.

Note: If another user logs in and runs this sample, this user gets the cache values that were last added or updated by the previous user. For example, if the counter value was five, the next user sees the counter value as increased to six.

```
public class OrgCacheController {

    // Inner class.
    // Used as the data type of a cache value.
```

```
class MyData {
    public String value { get; set; }
    public Integer counter { get; set; }

    public MyData(String value) {
        this.value = value;
        this.counter = 0;
    }

    public void inc() {
        counter++;
    }

    override public String toString() {
        return this.value + ':' + this.counter;
    }
}

// Apex List.
// Used as the data type of a cached value.
private List<String> numbers =
        new List<String> { 'ONE', 'TWO', 'THREE', 'FOUR', 'FIVE' };

// Constructor of the controller for the Visualforce page.
public OrgCacheController() {
}

// Adds various values to the cache.
// This method is called when the Visualforce page loads.
public void init() {
    // All key values are not qualified by the namespace.partition
    // prefix because they use the default partition.

    // Add counter to the cache with initial value of 0
    //  or increment it if it's already there.
    if (!Cache.Org.contains('counter')) {
        Cache.Org.put('counter', 0);
    } else {
        Cache.Org.put('counter', getCounter() + 1);
    }

    // Add the datetime value to the cache only if it's not already there.
    if (!Cache.Org.contains('datetime')) {
        DateTime dt = DateTime.now();
        Cache.Org.put('datetime', dt);
    }

    // Add the custom data to the cache only if it's not already there.
    if (!Cache.Org.contains('data')) {
        Cache.Org.put('data', new MyData('Some custom value'));
    }

    // Add a list of number to the cache if not already there.
    if (!Cache.Org.contains('list')) {
```

```
            Cache.Org.put('list', numbers);
        }

        // Add a string value to the cache if not already there.
        if (!Cache.Org.contains('output')) {
            Cache.Org.put('output', 'Cached text value');
        }
    }

    // Return counter from the cache.
    public Integer getCounter() {
        return (Integer)Cache.Org.get('counter');
    }

    // Return datetime value from the cache.
    public String getCachedDatetime() {
        DateTime dt = (DateTime)Cache.Org.get('datetime');
        return dt != null ? dt.format() : null;
    }

    // Return cached value whose type is the inner class MyData.
    public String getCachedData() {
        MyData mydata = (MyData)Cache.Org.get('data');
        return mydata != null ? mydata.toString() : null;
    }

    // Return output from the cache.
    public String getOutput() {
        return (String)Cache.Org.get('output');
    }

    // Return list from the cache.
    public List<String> getList() {
        return (List<String>)Cache.Org.get('list');
    }

    // Method invoked by the Rerender button on the Visualforce page.
    // Updates the values of various cached values.
    // Increases the values of counter and the MyData counter if those
    //   cache values are still in the cache.
    public PageReference go() {
        // Increase the cached counter value or set it to 0
        //  if it's not cached.
        if (Cache.Org.contains('counter')) {
            Cache.Org.put('counter', getCounter() + 1);
        } else {
            Cache.Org.put('counter', 0);
        }

        // Get the custom data value from the cache.
        MyData d = (MyData)Cache.Org.get('data');
        // Only if the data is already in the cache, update it.
        if (Cache.Org.contains('data')) {
            d.inc();
```

```
            Cache.Org.put('data', d);
        }

        return null;
    }

    // Method invoked by the Remove button on the Visualforce page.
    // Removes the datetime cached value from the org cache.
    public PageReference remove() {
        Cache.Org.remove('datetime');

        return null;
    }
}
```

This is the Visualforce page that corresponds to the `OrgCacheController` class.

```
<apex:page controller="OrgCacheController" action="{!init}">

    <apex:outputPanel id="output">
        <br/>Cached datetime: <apex:outputText value="{!cachedDatetime}"/>
        <br/>Cached data: <apex:outputText value="{!cachedData}"/>
        <br/>Cached counter: <apex:outputText value="{!counter}"/>
        <br/>Output: <apex:outputText value="{!output}"/>
        <br/>Repeat: <apex:repeat var="item" value="{!list}">
            <apex:outputText value="{!item}"/> 
        </apex:repeat>
        <br/>List size: <apex:outputText value="{!list.size}"/>
    </apex:outputPanel>

    <br/><br/>
    <apex:form >
        <apex:commandButton id="go" action="{!go}" value="Rerender" rerender="output"/>
        <apex:commandButton id="remove" action="{!remove}" value="Remove datetime Key"
rerender="output"/>
    </apex:form>

</apex:page>
```

This is the output of the page after clicking the **Rerender** button twice. The counter value could differ in your case if a key named `counter` was already in the cache before running this sample.

```
Cached datetime:8/11/2015 1:58 PM
Cached data:Some custom value:2
Cached counter:2
Output:Cached text value
Repeat:ONE TWO THREE FOUR FIVE
List size:5
```

IN THIS SECTION:

Org Constants
The Org class provides a constant that you can use when setting the time-to-live (TTL) value.

Org Methods

SEE ALSO:

Apex Developer Guide: Platform Cache

Org Constants

The Org class provides a constant that you can use when setting the time-to-live (TTL) value.

Constant	Description
`MAX_TTL_SECS`	Represents the maximum amount of time, in seconds, to keep the cached value in the org cache.

Org Methods

The following are methods for `Org`. All methods are static.

IN THIS SECTION:

contains(key)

Returns `true` if the org cache contains a cached value corresponding to the specified key.

contains(keys)

Returns `true` if the org cache contains values for the specified key entries.

contains(setOfKeys)

Returns `true` if the org cache contains values for a specified set of keys.

get(key)

Returns the cached value corresponding to the specified key from the org cache.

get(cacheBuilder, key)

Returns the cached value corresponding to the specified key from the org cache. Use this method if your cached value is a class that implements the `CacheBuilder` interface.

get(keys)

Returns the cached values corresponding to the specified set of keys from the org cache.

getAvgGetSize()

Returns the average item size of all the keys fetched from the org cache, in bytes.

getAvgGetTime()

Returns the average time taken to get a key from the org cache, in nanoseconds.

getAvgValueSize()

Deprecated and available only in API versions 49.0 and earlier. Returns the average item size for keys in the org cache, in bytes.

getCapacity()

Returns the percentage of org cache capacity that has been used.

getKeys()

Returns a set of all keys that are stored in the org cache and visible to the invoking namespace.

getMaxGetSize()

Returns the maximum item size of all the keys fetched from the org cache, in bytes.

getMaxGetTime()

Returns the maximum time taken to get a key from the org cache, in nanoseconds.

getMaxValueSize()

Deprecated and available only in API versions 49.0 and earlier. Returns the maximum item size for keys in the org cache, in bytes.

getMissRate()

Returns the miss rate in the org cache.

getName()

Returns the name of the default cache partition.

getNumKeys()

Returns the total number of keys in the org cache.

getPartition(partitionName)

Returns a partition from the org cache that corresponds to the specified partition name.

put(key, value)

Stores the specified key/value pair as a cached entry in the org cache. The `put` method can write only to the cache in your org's namespace.

put(key, value, visibility)

Stores the specified key/value pair as a cached entry in the org cache and sets the cached value's visibility.

put(key, value, ttlSecs)

Stores the specified key/value pair as a cached entry in the org cache and sets the cached value's lifetime.

put(key, value, ttlSecs, visibility, immutable)

Stores the specified key/value pair as a cached entry in the org cache. This method also sets the cached value's lifetime, visibility, and whether it can be overwritten by another namespace.

remove(key)

Deletes the cached value corresponding to the specified key from the org cache.

remove(cacheBuilder, key)

Deletes the cached value corresponding to the specified key from the org cache. Use this method if your cached value is a class that implements the `CacheBuilder` interface.

`contains(key)`

Returns `true` if the org cache contains a cached value corresponding to the specified key.

Signature

```
public static Boolean contains(String key)
```

Parameters

`key`

Type: String

A case-sensitive string value that uniquely identifies a cached value. For information about the format of the key name, see Usage.

Return Value

Type: Boolean

`true` if a cache entry is found. Otherwise, `false`.

`contains(keys)`

Returns `true` if the org cache contains values for the specified key entries.

Signature

```
public static List<Boolean> contains(List<String> keys)
```

Parameters

`keys`
Type: List<String>

A list of keys that identifies cached values. For information about the format of the key name, see Usage.

Return Value

Type: List<Boolean>

`true` if the key entries are found. Otherwise, `false`.

contains(setOfKeys)

Returns `true` if the org cache contains values for a specified set of keys.

Signature

```
public static Map <String, Boolean> contains (Set<String> keys)
```

Parameters

`setOfKeys`
Type: Set <String>

A set of keys that uniquely identifies cached values. For information about the format of the key name, see Usage

Return Value

Type: Map <String, Boolean>

Returns the cache key and corresponding Boolean value indicating that the key entry exists. The Boolean value is `false` if the key entry doesn't exist.

Usage

The number of input keys cannot exceed the maximum limit of 10.

Example

In this example, the code checks for the presence of multiple keys on the default partition. It fetches the cache key and the corresponding Boolean value for the key entry from the org cache of the default partition.

```
Set<String> keys = new Set<String>{'key1','key2','key3','key4','key5'};
Map<String,Boolean> result = Cache.Org.contains(keys);
for(String key : result.keySet()) {
    system.debug('key: ' + key);
    system.debug('Is Key Present in the cache : ' +  result.get(key));
}
```

In this example, the code checks for the presence of multiple keys on different partitions. It fetches the cache key and the corresponding Boolean value for the key entry from the org cache of different partitions.

```
// Assuming there are three partitions p1, p2, p3 with default 'local' namespace

Set<String> keys = new Set<String>{'local.p1.key','local.p2.key', 'local.p3.key'};
Map<String,Boolean> result = Cache.Org.contains(keys);
for(String key : result.keySet()) {
    system.debug('key: ' + key);
    system.debug('Is Key Present in the cache : +  result.get(key));
}
```

`get(key)`

Returns the cached value corresponding to the specified key from the org cache.

Signature

```
public static Object get(String key)
```

Parameters

key

Type: String

A case-sensitive string value that uniquely identifies a cached value. For information about the format of the key name, see Usage.

Return Value

Type: Object

The cached value as a generic object type. Cast the returned value to the appropriate type.

Usage

Because `Cache.Org.get()` returns an object, cast the returned value to a specific type to facilitate use of the returned value.

```
// Get a cached value
Object obj = Cache.Org.get('ns1.partition1.orderDate');
// Cast return value to a specific data type
DateTime dt2 = (DateTime)obj;
```

If a `Cache.Org.get()` call doesn't find the referenced key, it returns `null`.

get(cacheBuilder, key)

Returns the cached value corresponding to the specified key from the org cache. Use this method if your cached value is a class that implements the `CacheBuilder` interface.

Signature

```
public static Object get(System.Type cacheBuilder, String key)
```

Parameters

cacheBuilder
Type: System.Type

The Apex class that implements the `CacheBuilder` interface.

key
Type: String

A case-sensitive string value that, combined with the class name corresponding to the *cacheBuilder* parameter, uniquely identifies a cached value.

Return Value

Type: Object

The cached value as a generic object type. Cast the returned value to the appropriate type.

Usage

Because `Cache.Org.get(`*cacheBuilder*`, `*key*`)` returns an object, cast the returned value to a specific type to facilitate use of the returned value.

```
return ((DateTime)Cache.Org.get(DateCache.class, 'datetime')).format();
```

get(keys)

Returns the cached values corresponding to the specified set of keys from the org cache.

Signature

```
public static Map <String, Object> get (Set <String> keys)
```

Parameters

keys
Type: Set <String>

A set of keys that uniquely identify cached values. For information about the format of the key name, see Usage.

Return Value

Type: Map <String, Object>

Returns the cache key and corresponding value. Returns null when no corresponding value is found for an input key.

Usage

The number of input keys cannot exceed the maximum limit of 10.

Examples

Fetch multiple keys from the org cache of the default partition.

```
Set<String> keys = new Set<String>{'key1','key2','key3','key4','key5'};
Map<String,Object> result = Cache.Org.get(keys);
for(String key : result.keySet()) {
    system.debug('key: ' + key);
    system.debug('value: ' +  result.get(key));
}
```

Fetch multiple keys from the org cache of different partitions.

```
// Assuming there are three partitions p1, p2, p3 with default 'local' namespace

Set<String> keys = new Set<String>{'local.p1.key','local.p2.key', 'local.p3.key'};
Map<String,Object> result = Cache.Org.get(keys);
for(String key : result.keySet()) {
    system.debug('key: ' + key);
    system.debug('value: ' +  result.get(key));
}
```

`getAvgGetSize()`

Returns the average item size of all the keys fetched from the org cache, in bytes.

Signature

```
public static Long getAvgGetSize()
```

Return Value

Type: Long

Example

In this example the following keys and their corresponding value sizes are inserted. The code then fetches the keys: key 1, key 2, key 3 and key 4 and returns the average item size of the fetched keys.

Key	Key Value Size
key 1	42
key 2	42

Key	Key Value Size
key 3	58
key 4	36
key 5	36

```
// Inserting keys key1, key2, key3, key4, key5
Cache.Org.put('key1', 'value1');
Cache.Org.put('key2', 'value2');
Cache.Org.put('key3', 'this is a big value !!!');
Cache.Org.put('key4', 4);
Cache.Org.put('key5', 5);


// Fetching keys - key1, key2, key3, key4
Object v1 = Cache.Org.get('key1');
Object v2 = Cache.Org.get('key2');
Object v3 = Cache.Org.get('key3');
Object v4 = Cache.Org.get('key4');

// Fetching average get size
Long val = Cache.Org.getAvgGetSize();
// Avg item size returned is 44 ( average of 42(key1), 42(key2), 58(key3) and 36(key4)
keys that were fetched )
System.debug('Avg Get Size :' + val);
```

`getAvgGetTime()`

Returns the average time taken to get a key from the org cache, in nanoseconds.

Signature

```
public static Long getAvgGetTime()
```

Return Value

Type: Long

`getAvgValueSize()`

Deprecated and available only in API versions 49.0 and earlier. Returns the average item size for keys in the org cache, in bytes.

Signature

```
public static Long getAvgValueSize()
```

Return Value

Type: Long

getCapacity()

Returns the percentage of org cache capacity that has been used.

Signature

```
public static Double getCapacity()
```

Return Value

Type: Double

Used cache as a percentage number.

getKeys()

Returns a set of all keys that are stored in the org cache and visible to the invoking namespace.

Signature

```
public static Set<String> getKeys()
```

Return Value

Type: Set<String>

A set containing all cache keys.

getMaxGetSize()

Returns the maximum item size of all the keys fetched from the org cache, in bytes.

Signature

```
public static Long getMaxGetSize()
```

Return Value

Type: Long

Example

In this example the following keys and their corresponding value sizes are inserted. The code fetches the keys: key 1, key 2 and key 4 and returns the maximum key value size from the fetched keys.

Key	Key Value Size
key 1	42
key 2	42
key 3	58
key 4	36

Key	Key Value Size
key 5	36

```
// Inserting keys key1, key2, key3, key4, key5
Cache.Org.put('key1', 'value1');
Cache.Org.put('key2', 'value2');
Cache.Org.put('key3', 'this is a big value !!!');
Cache.Org.put('key4', 4);
Cache.Org.put('key5', 5);


// Fetching keys - key1, key2, key4
Object v1 = Cache.Org.get('key1');
Object v2 = Cache.Org.get('key2');
Object v4 = Cache.Org.get('key4');

// Fetching max get size
Long val = Cache.Org.getMaxGetSize();
// Max item size returned is 42 ( max of 42(key1), 42(key2), and 36(key4) keys that were
fetched )
System.debug('Max Get Size :' + val);
```

`getMaxGetTime()`

Returns the maximum time taken to get a key from the org cache, in nanoseconds.

Signature

```
public static Long getMaxGetTime()
```

Return Value

Type: Long

`getMaxValueSize()`

Deprecated and available only in API versions 49.0 and earlier. Returns the maximum item size for keys in the org cache, in bytes.

Signature

```
public static Long getMaxValueSize()
```

Return Value

Type: Long

`getMissRate()`

Returns the miss rate in the org cache.

Signature

```
public static Double getMissRate()
```

Return Value

Type: Double

`getName()`

Returns the name of the default cache partition.

Signature

```
public String getName()
```

Return Value

Type: String

The name of the default cache partition.

`getNumKeys()`

Returns the total number of keys in the org cache.

Signature

```
public static Long getNumKeys()
```

Return Value

Type: Long

`getPartition(partitionName)`

Returns a partition from the org cache that corresponds to the specified partition name.

Signature

```
public static cache.OrgPartition getPartition(String partitionName)
```

Parameters

`partitionName`
Type: String

A partition name that is qualified by the namespace, for example, *`namespace.partition`*.

Return Value

Type: Cache.OrgPartition

Example

After you get the org partition, you can add and retrieve the partition's cache values.

```
// Get partition
Cache.OrgPartition orgPart = Cache.Org.getPartition('myNs.myPartition');
// Retrieve cache value from the partition
if (orgPart.contains('BookTitle')) {
    String cachedTitle = (String)orgPart.get('BookTitle');
}

// Add cache value to the partition
orgPart.put('OrderDate', Date.today());

// Or use dot notation to call partition methods
String cachedAuthor = (String)Cache.Org.getPartition('myNs.myPartition').get('BookAuthor');
```

put(key, value)

Stores the specified key/value pair as a cached entry in the org cache. The `put` method can write only to the cache in your org's namespace.

Signature

```
public static void put(String key, Object value)
```

Parameters

key
Type: String

A case-sensitive string value that uniquely identifies a cached value. For information about the format of the key name, see Usage.

value
Type: Object

The value to store in the cache. The cached value must be serializable.

Return Value

Type: void

put(key, value, visibility)

Stores the specified key/value pair as a cached entry in the org cache and sets the cached value's visibility.

Signature

```
public static void put(String key, Object value, Cache.Visibility visibility)
```

Parameters

key
Type: String

A case-sensitive string value that uniquely identifies a cached value. For information about the format of the key name, see Usage.

value
Type: Object

The value to store in the cache. The cached value must be serializable.

visibility
Type: Cache.Visibility

Indicates whether the cached value is available only to Apex code that is executing in the same namespace or to Apex code executing from any namespace.

Return Value

Type: void

put(key, value, ttlSecs)

Stores the specified key/value pair as a cached entry in the org cache and sets the cached value's lifetime.

Signature

```
public static void put(String key, Object value, Integer ttlSecs)
```

Parameters

key
Type: String

A case-sensitive string value that uniquely identifies a cached value. For information about the format of the key name, see Usage.

value
Type: Object

The value to store in the cache. The cached value must be serializable.

ttlSecs
Type: Integer

The amount of time, in seconds, to keep the cached value in the org cache. The maximum is 172,800 seconds (48 hours). The minimum value is 300 seconds or 5 minutes. The default value is 86,400 seconds (24 hours).

Return Value

Type: void

put(key, value, ttlSecs, visibility, immutable)

Stores the specified key/value pair as a cached entry in the org cache. This method also sets the cached value's lifetime, visibility, and whether it can be overwritten by another namespace.

Signature

```
public static void put(String key, Object value, Integer ttlSecs, cache.Visibility
visibility, Boolean immutable)
```

Parameters

key
Type: String

A case-sensitive string value that uniquely identifies a cached value. For information about the format of the key name, see Usage.

value
Type: Object

The value to store in the cache. The cached value must be serializable.

ttlSecs
Type: Integer

The amount of time, in seconds, to keep the cached value in the org cache. The maximum is 172,800 seconds (48 hours). The minimum value is 300 seconds or 5 minutes. The default value is 86,400 seconds (24 hours).

visibility
Type: Cache.Visibility

Indicates whether the cached value is available only to Apex code that is executing in the same namespace or to Apex code executing from any namespace.

immutable
Type: Boolean

Indicates whether the cached value can be overwritten by another namespace (`false`) or not (`true`).

Return Value

Type: void

remove(key)

Deletes the cached value corresponding to the specified key from the org cache.

Signature

```
public static Boolean remove(String key)
```

Parameters

key
Type: String

A case-sensitive string value that uniquely identifies a cached value. For information about the format of the key name, see Usage.

Return Value

Type: Boolean

`true` if the cache value was successfully removed. Otherwise, `false`.

`remove(cacheBuilder, key)`

Deletes the cached value corresponding to the specified key from the org cache. Use this method if your cached value is a class that implements the `CacheBuilder` interface.

Signature

```
public static Boolean remove(System.Type cacheBuilder, String key)
```

Parameters

`cacheBuilder`
Type: System.Type

The Apex class that implements the `CacheBuilder` interface.

`key`
Type: String

A case-sensitive string value that, combined with the class name corresponding to the *`cacheBuilder`* parameter, uniquely identifies a cached value.

Return Value

Type: Boolean

`true` if the cache value was successfully removed. Otherwise, `false`.

OrgPartition Class

Contains methods to manage cache values in the org cache of a specific partition. Unlike the session cache, the org cache is not tied to any session. It's available to the organization across requests and to all users.

Namespace

Cache

Usage

This class extends Cache.Partition and inherits all its non-static methods. Utility methods for creating and validating keys aren't supported and can be called only from the `Cache.Partition` parent class. For a list of `Cache.Partition` methods, see Partition Methods.

To get an org partition, call `Cache.Org.getPartition` and pass in a fully qualified partition name, as follows.

```
Cache.OrgPartition orgPartition = Cache.Org.getPartition('namespace.myPartition');
```

See Cache Key Format for Partition Methods.

Example

This class is the controller for a sample Visualforce page (shown in the subsequent code sample). The controller shows how to use the methods of `Cache.OrgPartition` to manage a cache value on a particular partition. The controller takes inputs from the Visualforce page for the partition name, key name for a counter, and initial counter value. The controller contains default values for these inputs.

When you click **Rerender** on the Visualforce page, the `go()` method is invoked and increases the counter by one. When you click **Remove Key**, the counter key is removed from the cache. The counter value gets reset to its initial value when it's re-added to the cache.

Note: If another user logs in and runs this sample, the user gets the cache values that were last added or updated by the previous user. For example, if the counter value was five, the next user sees the counter value as increased to six.

```
public class OrgPartitionController {

    // Name of a partition
    String partitionInput = 'local.myPartition';
    // Name of the key
    String counterKeyInput = 'counter';
    // Key initial value
    Integer counterInitValue = 0;
    // Org partition object
    Cache.OrgPartition orgPartition;

    // Constructor of the controller for the Visualforce page.
    public OrgPartitionController() {
    }

    // Adds counter value to the cache.
    // This method is called when the Visualforce page loads.
    public void init() {
        // Create the partition instance based on the partition name
        orgPartition = getPartition();

        // Create the partition instance based on the partition name
        // given in the Visualforce page or the default value.
        orgPartition = Cache.Org.getPartition(partitionInput);

        // Add counter to the cache with an initial value
        //  or increment it if it's already there.
        if (!orgPartition.contains(counterKeyInput)) {
            orgPartition.put(counterKeyInput, counterInitValue);
        } else {
            orgPartition.put(counterKeyInput, getCounter() + 1);
        }
    }

    // Returns the org partition based on the partition name
    // given in the Visualforce page or the default value.
    private Cache.OrgPartition getPartition() {
       if (orgPartition == null) {
            orgPartition = Cache.Org.getPartition(partitionInput);
       }

       return orgPartition;
     }


    // Return counter from the cache.
    public Integer getCounter() {
        return (Integer)getPartition().get(counterKeyInput);
    }
```

```
// Invoked by the Submit button to save input values
//  supplied by the user.
public PageReference save() {
    // Reset the initial key value in the cache
    getPartition().put(counterKeyInput, counterInitValue);

    return null;
}

// Method invoked by the Rerender button on the Visualforce page.
// Updates the values of various cached values.
// Increases the values of counter and the MyData counter if those
//   cache values are still in the cache.
public PageReference go() {
    // Get the org partition object
    orgPartition = getPartition();
    // Increase the cached counter value or set it to 0
    //  if it's not cached.
    if (orgPartition.contains(counterKeyInput)) {
        orgPartition.put(counterKeyInput, getCounter() + 1);
    } else {
        orgPartition.put(counterKeyInput, counterInitValue);
    }

    return null;
}

// Method invoked by the Remove button on the Visualforce page.
// Removes the datetime cached value from the org cache.
public PageReference remove() {
    getPartition().remove(counterKeyInput);

    return null;
}

// Get and set methods for accessing variables
// that correspond to the input text fields on
// the Visualforce page.
public String getPartitionInput() {
    return partitionInput;
}

public String getCounterKeyInput() {
    return counterKeyInput;
}

public Integer getCounterInitValue() {
    return counterInitValue;
}

public void setPartitionInput(String partition) {
    this.partitionInput = partition;
}
```

```
    public void setCounterKeyInput(String keyName) {
        this.counterKeyInput = keyName;
    }

    public void setCounterInitValue(Integer counterValue) {
        this.counterInitValue = counterValue;
    }
}
```

This is the Visualforce page that corresponds to the `OrgPartitionController` class.

```
<apex:page controller="OrgPartitionController" action="{!init}">

    <apex:form >
        <br/>Partition with Namespace Prefix: <apex:inputText value="{!partitionInput}"/>

        <br/>Counter Key Name: <apex:inputText value="{!counterKeyInput}"/>
        <br/>Counter Initial Value: <apex:inputText value="{!counterInitValue}"/>
        <apex:commandButton action="{!save}" value="Save Key Input Values"/>
    </apex:form>

    <apex:outputPanel id="output">
        <br/>Cached Counter: <apex:outputText value="{!counter}"/>
    </apex:outputPanel>

    <br/>
    <apex:form >
        <apex:commandButton id="go" action="{!go}" value="Rerender" rerender="output"/>
        <apex:commandButton id="remove" action="{!remove}" value="Remove Key"
rerender="output"/>
    </apex:form>

</apex:page>
```

SEE ALSO:

Apex Developer Guide: Platform Cache

Partition Class

Base class of `Cache.OrgPartition` and `Cache.SessionPartition`. Use the subclasses to manage the cache partition for org caches and session caches.

Namespace

Cache

Cache Key Format for Partition Methods

After you obtain the partition object (an instance of `Cache.OrgPartition` or `Cache.SessionPartition`), the methods to add, retrieve, and manage the cache values in a partition take the key name. The key name that you supply to these methods (`get()`, `put()`, `remove()`, and `contains()`) doesn't include the `namespace.partition` prefix.

IN THIS SECTION:

Partition Methods

SEE ALSO:

OrgPartition Class

SessionPartition Class

Apex Developer Guide: Platform Cache

Partition Methods

The following are methods for `Partition`.

IN THIS SECTION:

contains(key)

Returns `true` if the cache partition contains a cached value corresponding to the specified key.

contains(setOfKeys)

Returns `true` if the cache partition contains values for a specified set of keys.

createFullyQualifiedKey(namespace, partition, key)

Generates a fully qualified key from the passed-in key components. The format of the generated key string is `namespace.partition.key`.

createFullyQualifiedPartition(namespace, partition)

Generates a fully qualified partition name from the passed-in namespace and partition. The format of the generated partition string is `namespace.partition`.

get(key)

Returns the cached value corresponding to the specified key from the cache partition.

get(keys)

Returns the cached values corresponding to the specified set of keys from the cache partition.

get(cacheBuilder, key)

Returns the cached value corresponding to the specified key from the partition cache. Use this method if your cached value is a class that implements the `CacheBuilder` interface.

getAvgGetSize()

Returns the average item size of all the keys fetched from the partition, in bytes.

getAvgGetTime()

Returns the average time taken to get a key from the partition, in nanoseconds.

getAvgValueSize()

Deprecated and available only in API versions 49.0 and earlier. Returns the average item size for keys in the partition, in bytes.

getCapacity()

Returns the percentage of cache used of the total capacity for this partition.

getKeys()

Returns a set of all keys that are stored in the cache partition and visible to the invoking namespace.

getMaxGetSize()

Returns the maximum item size of all the keys fetched from the partition, in bytes.

getMaxGetTime()

Returns the maximum time taken to get a key from the partition, in nanoseconds.

getMaxValueSize()

Deprecated and available only in API versions 49.0 and earlier. Returns the maximum item size for keys in the partition, in bytes.

getMissRate()

Returns the miss rate in the partition.

getName()

Returns the name of this cache partition.

getNumKeys()

Returns the total number of keys in the partition.

isAvailable()

Returns `true` if the Salesforce session is available. Only applies to `Cache.SessionPartition`. The session cache isn't available when an active session isn't present, such as in asynchronous Apex or code called by asynchronous Apex. For example, if batch Apex causes an Apex trigger to execute, the session cache isn't available in the trigger because the trigger runs in asynchronous context.

put(key, value)

Stores the specified key/value pair as a cached entry in the cache partition. The `put` method can write only to the cache in your org's namespace.

put(key, value, visibility)

Stores the specified key/value pair as a cached entry in the cache partition and sets the cached value's visibility.

put(key, value, ttlSecs)

Stores the specified key/value pair as a cached entry in the cache partition and sets the cached value's lifetime.

put(key, value, ttlSecs, visibility, immutable)

Stores the specified key/value pair as a cached entry in the cache partition. This method also sets the cached value's lifetime, visibility, and whether it can be overwritten by another namespace.

remove(key)

Deletes the cached value corresponding to the specified key from this cache partition.

remove(cacheBuilder, key)

Deletes the cached value corresponding to the specified key from the partition cache. Use this method if your cached value is a class that implements the `CacheBuilder` interface.

validateCacheBuilder(cacheBuilder)

Validates that the specified class implements the `CacheBuilder` interface.

validateKey(isDefault, key)

Validates a cache key. This method throws a `Cache.InvalidParamException` if the key is not valid. A valid key is not `null` and contains alphanumeric characters.

validateKeyValue(isDefault, key, value)

Validates a cache key and ensures that the cache value is non-null. This method throws a `Cache.InvalidParamException` if the key or value is not valid. A valid key is not `null` and contains alphanumeric characters.

validateKeys(isDefault, keys)

Validates the specified cache keys. This method throws a `Cache.InvalidParamException` if the key is not valid. A valid key is not `null` and contains alphanumeric characters.

validatePartitionName(name)

Validates the partition name — for example, that it is not null.

`contains(key)`

Returns `true` if the cache partition contains a cached value corresponding to the specified key.

Signature

```
public Boolean contains(String key)
```

Parameters

key

Type: String

A case-sensitive string value that uniquely identifies a cached value.

Return Value

Type: Boolean

`true` if a cache entry is found. Othewise, `false`.

contains(setOfKeys)

Returns `true` if the cache partition contains values for a specified set of keys.

Signature

```
public Map <String, Boolean> contains (Set<String> keys)
```

Parameters

setOfKeys

Type: Set <String>

A set of keys that uniquely identifies cached values. For information about the format of the key name, see Usage.

Return Value

Type: Map <String, Boolean>

Returns the cache key and corresponding Boolean value indicating that the key entry exists. The Boolean value is `false` if the key entry doesn't exist.

Usage

The number of input keys cannot exceed the maximum limit of 10.

Example

In this example, the code checks for the presence of multiple keys on a partition. It fetches the cache key and the corresponding Boolean value for the key entry from the org cache of the partition.

```
// Assuming there is a partition p1 in the default 'local' namespace

Set<String> keys = new Set<String>{'key1','key2','key3','key4','key5'};
Cache.OrgPartition orgPart = Cache.Org.getPartition('local.p1');
Map<String,Boolean> result = orgPart.contains(keys);
for(String key : result.keySet()) {
    system.debug('key: ' + key);
    system.debug('Is Key Present in the cache:' +  result.get(key));
}
```

In this example, the code checks for the presence of multiple keys on a partition. It fetches the cache key and the corresponding Boolean value for the key entry from the session cache of the partition.

```
// Assuming there are three partitions p1, p2, p3 with default 'local' namespace

Set<String> keys = new Set<String>{'key1','key2','key3','key4','key5'};
Cache.SessionPartition sessionPart = Cache.Session.getPartition('local.p1');
Map<String,Boolean> result = sessionPart.contains(keys);
for(String key : result.keySet()) {
    system.debug('key: ' + key);
    system.debug('value: ' +  result.get(key));
}
```

`createFullyQualifiedKey(namespace, partition, key)`

Generates a fully qualified key from the passed-in key components. The format of the generated key string is `namespace.partition.key`.

Signature

```
public static String createFullyQualifiedKey(String namespace, String partition, String key)
```

Parameters

`namespace`
Type: String

The namespace of the cache key.

`partition`
Type: String

The partition of the cache key.

`key`
Type: String

The name of the cache key.

Return Value

Type: String

createFullyQualifiedPartition(namespace, partition)

Generates a fully qualified partition name from the passed-in namespace and partition. The format of the generated partition string is `namespace.partition`.

Signature

```
public static String createFullyQualifiedPartition(String namespace, String partition)
```

Parameters

namespace
Type: String

The namespace of the cache key.

partition
Type: String

The partition of the cache key.

Return Value

Type: String

get(key)

Returns the cached value corresponding to the specified key from the cache partition.

Signature

```
public Object get(String key)
```

Parameters

key
Type: String

A case-sensitive string value that uniquely identifies a cached value.

Return Value

Type: Object

The cached value as a generic object type. Cast the returned value to the appropriate type.

get(keys)

Returns the cached values corresponding to the specified set of keys from the cache partition.

Signature

```
public Map <String, Object> get (Set <String> keys)
```

Parameters

keys
Type: Set <String>

A set of keys that uniquely identify cached values. For information about the format of the key name, see Usage.

Return Value

Type: Map <String, Object>

Returns the cache key and corresponding value. Returns null when no corresponding value is found for an input key.

Usage

The number of input keys cannot exceed the maximum limit of 10.

Examples

Fetch multiple keys from the org cache of a partition.

```
// Assuming there is a partition p1 in the default 'local' namespace

Set<String> keys = new Set<String>{'key1','key2','key3','key4','key5'};
Cache.OrgPartition orgPart = Cache.Org.getPartition('local.p1');
Map<String,Object> result = orgPart.get(keys);
for(String key : result.keySet()) {
    system.debug('key: ' + key);
    system.debug('value: ' +  result.get(key));
}
```

Fetch multiple keys from the session cache of a partition.

```
// Assuming there is a partition p1 in the default 'local' namespace

Set<String> keys = new Set<String>{'key1','key2','key3','key4','key5'};
Cache.SessionPartition sessionPart = Cache.Session.getPartition('local.p1');
Map<String,Object> result = sessionPart.get(keys);
for(String key : result.keySet()) {
    system.debug('key: ' + key);
    system.debug('value: ' +  result.get(key));
}
```

get(cacheBuilder, key)

Returns the cached value corresponding to the specified key from the partition cache. Use this method if your cached value is a class that implements the `CacheBuilder` interface.

Signature

```
public Object get(System.Type cacheBuilder, String key)
```

Parameters

cacheBuilder
Type: System.Type

The Apex class that implements the `CacheBuilder` interface.

key
Type: String

A case-sensitive string value that, combined with the class name corresponding to the *cacheBuilder* parameter, uniquely identifies a cached value.

Return Value

Type: Object

The cached value as a generic object type. Cast the returned value to the appropriate type.

getAvgGetSize()

Returns the average item size of all the keys fetched from the partition, in bytes.

Signature

```
public Long getAvgGetSize()
```

Return Value

Type: Long

getAvgGetTime()

Returns the average time taken to get a key from the partition, in nanoseconds.

Signature

```
public Long getAvgGetTime()
```

Return Value

Type: Long

getAvgValueSize()

Deprecated and available only in API versions 49.0 and earlier. Returns the average item size for keys in the partition, in bytes.

Signature

```
public Long getAvgValueSize()
```

Return Value

Type: Long

getCapacity()

Returns the percentage of cache used of the total capacity for this partition.

Signature

```
public Double getCapacity()
```

Return Value

Type: Double

Used partition cache as a percentage number.

getKeys()

Returns a set of all keys that are stored in the cache partition and visible to the invoking namespace.

Signature

```
public Set<String> getKeys()
```

Return Value

Type: Set<String>

A set containing all cache keys.

getMaxGetSize()

Returns the maximum item size of all the keys fetched from the partition, in bytes.

Signature

```
public Long getMaxGetSize()
```

Return Value

Type: Long

getMaxGetTime()

Returns the maximum time taken to get a key from the partition, in nanoseconds.

Signature

```
public Long getMaxGetTime()
```

Return Value

Type: Long

getMaxValueSize()

Deprecated and available only in API versions 49.0 and earlier. Returns the maximum item size for keys in the partition, in bytes.

Signature

```
public Long getMaxValueSize()
```

Return Value

Type: Long

getMissRate()

Returns the miss rate in the partition.

Signature

```
public Double getMissRate()
```

Return Value

Type: Double

getName()

Returns the name of this cache partition.

Signature

```
public String getName()
```

Return Value

Type: String

The name of this cache partition.

getNumKeys()

Returns the total number of keys in the partition.

Signature

```
public Long getNumKeys()
```

Return Value

Type: Long

isAvailable()

Returns `true` if the Salesforce session is available. Only applies to `Cache.SessionPartition`. The session cache isn't available when an active session isn't present, such as in asynchronous Apex or code called by asynchronous Apex. For example, if batch Apex causes an Apex trigger to execute, the session cache isn't available in the trigger because the trigger runs in asynchronous context.

Signature

```
public Boolean isAvailable()
```

Return Value

Type: Boolean

put(key, value)

Stores the specified key/value pair as a cached entry in the cache partition. The `put` method can write only to the cache in your org's namespace.

Signature

```
public void put(String key, Object value)
```

Parameters

key
Type: String

A case-sensitive string value that uniquely identifies a cached value.

value
Type: Object

The value to store in the cache. The cached value must be serializable.

Return Value

Type: void

put(key, value, visibility)

Stores the specified key/value pair as a cached entry in the cache partition and sets the cached value's visibility.

Signature

```
public void put(String key, Object value, cache.Visibility visibility)
```

Parameters

key
Type: String

A case-sensitive string value that uniquely identifies a cached value.

value
Type: Object

The value to store in the cache. The cached value must be serializable.

visibility
Type: Cache.Visibility

Indicates whether the cached value is available only to Apex code that is executing in the same namespace or to Apex code executing from any namespace.

Return Value

Type: void

put(key, value, ttlSecs)

Stores the specified key/value pair as a cached entry in the cache partition and sets the cached value's lifetime.

Signature

```
public void put(String key, Object value, Integer ttlSecs)
```

Parameters

key
Type: String

A case-sensitive string value that uniquely identifies a cached value.

value
Type: Object

The value to store in the cache. The cached value must be serializable.

ttlSecs
Type: Integer

The amount of time, in seconds, to keep the cached value in the cache.

Return Value

Type: void

put(key, value, ttlSecs, visibility, immutable)

Stores the specified key/value pair as a cached entry in the cache partition. This method also sets the cached value's lifetime, visibility, and whether it can be overwritten by another namespace.

Signature

```
public void put(String key, Object value, Integer ttlSecs, cache.Visibility visibility,
Boolean immutable)
```

Parameters

key
Type: String

A case-sensitive string value that uniquely identifies a cached value.

value
Type: Object

The value to store in the cache. The cached value must be serializable.

ttlSecs
Type: Integer

The amount of time, in seconds, to keep the cached value in the cache.

visibility
Type: Cache.Visibility

Indicates whether the cached value is available only to Apex code that is executing in the same namespace or to Apex code executing from any namespace.

immutable
Type: Boolean

Indicates whether the cached value can be overwritten by another namespace (`false`) or not (`true`).

Return Value

Type: void

remove(key)

Deletes the cached value corresponding to the specified key from this cache partition.

Signature

```
public Boolean remove(String key)
```

Parameters

key
Type: String

A case-sensitive string value that uniquely identifies a cached value.

Return Value

Type: Boolean

`true` if the cache value was successfully removed. Otherwise, `false`.

`remove(cacheBuilder, key)`

Deletes the cached value corresponding to the specified key from the partition cache. Use this method if your cached value is a class that implements the `CacheBuilder` interface.

Signature

```
public Boolean remove(System.Type cacheBuilder, String key)
```

Parameters

`cacheBuilder`
Type: System.Type

The Apex class that implements the `CacheBuilder` interface.

`key`
Type: String

A case-sensitive string value that, combined with the class name corresponding to the *`cacheBuilder`* parameter, uniquely identifies a cached value.

Return Value

Type: Boolean

`true` if the cache value was successfully removed. Otherwise, `false`.

`validateCacheBuilder(cacheBuilder)`

Validates that the specified class implements the `CacheBuilder` interface.

Signature

```
public static void validateCacheBuilder(System.Type cacheBuilder)
```

Parameters

`cacheBuilder`
Type: System.Type

The class to validate.

Return Value

Type: void

`validateKey(isDefault, key)`

Validates a cache key. This method throws a `Cache.InvalidParamException` if the key is not valid. A valid key is not `null` and contains alphanumeric characters.

Signature

```
public static void validateKey(Boolean isDefault, String key)
```

Parameters

isDefault
Type: Boolean

Set to `true` if the key references a default partition. Otherwise, set to `false`.

key
Type: String

The key to validate.

Return Value

Type: void

validateKeyValue(isDefault, key, value)

Validates a cache key and ensures that the cache value is non-null. This method throws a `Cache.InvalidParamException` if the key or value is not valid. A valid key is not `null` and contains alphanumeric characters.

Signature

```
public static void validateKeyValue(Boolean isDefault, String key, Object value)
```

Parameters

isDefault
Type: Boolean

Set to `true` if the key references a default partition. Otherwise, set to `false`.

key
Type: String

The key to validate.

value
Type: Object

The cache value to validate.

Return Value

Type: void

validateKeys(isDefault, keys)

Validates the specified cache keys. This method throws a `Cache.InvalidParamException` if the key is not valid. A valid key is not `null` and contains alphanumeric characters.

Signature

```
public static void validateKeys(Boolean isDefault, Set<String> keys)
```

Parameters

isDefault
Type: Boolean

Set to `true` if the key references a default partition. Otherwise, set to `false`.

keys
Type: Set<String>

A set of key string values to validate.

Return Value

Type: void

validatePartitionName(name)

Validates the partition name — for example, that it is not null.

Signature

```
public static void validatePartitionName(String name)
```

Parameters

name
Type: String

The name of the partition to validate.

Return Value

Type: void

Session Class

Use the `Cache.Session` class to add, retrieve, and manage values in the session cache. The session cache is active as long as the user's Salesforce session is valid (the user is logged in, and the session is not expired).

Namespace

Cache

Usage

Cache Key Format

This table lists the format of the key parameter that some methods in this class take, such as `put`, `get`, and `contains`.

Key Format	Description
namespace.partition.key	Fully qualified key name.
key	Refers to a partition marked as default when the *namespace.partition* prefix is omitted.
local.*partition.key*	Use the `local` prefix to refer to the org's namespace when the org doesn't have a namespace defined. If the org has a namespace defined, the `local` prefix also refers to that org's namespace.

Note:

- If no default partition is specified in the org, calling a cache method without fully qualifying the key name causes a `Cache.Session.SessionCacheException` to be thrown.
- The `local` prefix in an installed managed package refers to the namespace of the subscriber org and not the package's namespace. The cache `put` calls are not allowed in a partition that the invoking class doesn't own.

Example

This class is the controller for a sample Visualforce page (shown in the subsequent code sample). The cached values are initially added to the cache by the `init()` method, which the Visualforce page invokes when it loads through the `action` attribute. The cache keys don't contain the `namespace.partition` prefix. They all refer to a default partition in your org. The Visualforce page expects a partition named `myPartition`. To run this sample, create a default partition in your org with the name `myPartition`.

The Visualforce page contains four output components. The first three components call `get` methods on the controller that return the following values from the cache: a date, data based on the `MyData` inner class, and a counter. The next output component uses the `$Cache.Session` global variable to get the cached string value for the key named `output`. Next, the `$Cache.Session` global variable is used again in the Visualforce page to iterate over the elements of a cached value of type `List`. The size of the list is also returned.

The Visualforce page also contains two buttons. The Rerender button invokes the `go()` method on the controller. This method increases the values of the counter and the custom data in the cache. If you click **Rerender**, the two counters increase by one each time. The `go()` method retrieves the values of these counters from the cache, increments their values by one, and stores them again in the cache.

The Remove button deletes the date-time value (with key `datetime`) from the cache. As a result, the value next to `Cached datetime:` is cleared on the page.

```
public class SessionCacheController {

    // Inner class.
    // Used as the data type of a cache value.
    class MyData {
        public String value { get; set; }
        public Integer counter { get; set; }

        public MyData(String value) {
            this.value = value;
            this.counter = 0;
        }
```

```
        public void inc() {
            counter++;
        }

        override public String toString() {
            return this.value + ':' + this.counter;
        }
    }

    // Apex List.
    // Used as the data type of a cached value.
    private List<String> numbers =
            new List<String> { 'ONE', 'TWO', 'THREE', 'FOUR', 'FIVE' };

    // Constructor of the controller for the Visualforce page.
    public SessionCacheController() {
    }

    // Adds various values to the cache.
    // This method is called when the Visualforce page loads.
    public void init() {
        // All key values are not qualified by the namespace.partition
        // prefix because they use the default partition.

        // Add counter to the cache with initial value of 0
        //  or increment it if it's already there.
        if (!Cache.Session.contains('counter')) {
            Cache.Session.put('counter', 0);
        } else {
            Cache.Session.put('counter', getCounter() + 1);
        }

        // Add the datetime value to the cache only if it's not already there.
        if (!Cache.Session.contains('datetime')) {
            DateTime dt = DateTime.now();
            Cache.Session.put('datetime', dt);
        }

        // Add the custom data to the cache only if it's not already there.
        if (!Cache.Session.contains('data')) {
            Cache.Session.put('data', new MyData('Some custom value'));
        }

        // Add a list of number to the cache if not already there.
        if (!Cache.Session.contains('list')) {
            Cache.Session.put('list', numbers);
        }

        // Add a string value to the cache if not already there.
        if (!Cache.Session.contains('output')) {
            Cache.Session.put('output', 'Cached text value');
        }
    }
```

```
    // Return counter from the cache.
    public Integer getCounter() {
        return (Integer)Cache.Session.get('counter');
    }

    // Return datetime value from the cache.
    public String getCachedDatetime() {
        DateTime dt = (DateTime)Cache.Session.get('datetime');
        return dt != null ? dt.format() : null;
    }

    // Return cached value whose type is the inner class MyData.
    public String getCachedData() {
        MyData mydata = (MyData)Cache.Session.get('data');
        return mydata != null ? mydata.toString() : null;
    }

    // Method invoked by the Rerender button on the Visualforce page.
    // Updates the values of various cached values.
    // Increases the values of counter and the MyData counter if those
    //   cache values are still in the cache.
    public PageReference go() {
        // Increase the cached counter value or set it to 0
        //  if it's not cached.
        if (Cache.Session.contains('counter')) {
            Cache.Session.put('counter', getCounter() + 1);
        } else {
            Cache.Session.put('counter', 0);
        }

        // Get the custom data value from the cache.
        MyData d = (MyData)Cache.Session.get('data');
        // Only if the data is already in the cache, update it.
        if (Cache.Session.contains('data')) {
            d.inc();
            Cache.Session.put('data', d);
        }

        return null;
    }

    // Method invoked by the Remove button on the Visualforce page.
    // Removes the datetime cached value from the session cache.
    public PageReference remove() {
        Cache.Session.remove('datetime');

        return null;
    }
}
```

This is the Visualforce page that corresponds to the `SessionCacheController` class.

```
<apex:page controller="SessionCacheController" action="{!init}">

    <apex:outputPanel id="output">
```

```
        <br/>Cached datetime: <apex:outputText value="{!cachedDatetime}"/>
        <br/>Cached data: <apex:outputText value="{!cachedData}"/>
        <br/>Cached counter: <apex:outputText value="{!counter}"/>
       <br/>Output: <apex:outputText value="{!$Cache.Session.local.myPartition.output}"/>

        <br/>Repeat: <apex:repeat var="item"
value="{!$Cache.Session.local.myPartition.list}">
            <apex:outputText value="{!item}"/> 
        </apex:repeat>
        <br/>List size: <apex:outputText
value="{!$Cache.Session.local.myPartition.list.size}"/>
    </apex:outputPanel>

    <br/><br/>
    <apex:form >
        <apex:commandButton id="go" action="{!go}" value="Rerender" rerender="output"/>
        <apex:commandButton id="remove" action="{!remove}" value="Remove datetime Key"
rerender="output"/>
    </apex:form>

</apex:page>
```

This is the output of the page after clicking the Rerender button twice. The counter value could differ in your case if a key named `counter` was already in the cache before running this sample.

```
Cached datetime:8/11/2015 1:58 PM
Cached data:Some custom value:2
Cached counter:2
Output:Cached text value
Repeat:ONE TWO THREE FOUR FIVE
List size:5
```

IN THIS SECTION:

Session Constants

The Session class provides a constant that you can use when setting the time-to-live (TTL) value.

Session Methods

SEE ALSO:

Apex Developer Guide: Platform Cache

Session Constants

The Session class provides a constant that you can use when setting the time-to-live (TTL) value.

Constant	Description
`MAX_TTL_SECS`	Represents the maximum amount of time, in seconds, to keep the cached value in the session cache.

Session Methods

The following are methods for `Session`. All methods are static.

IN THIS SECTION:

contains(key)

Returns `true` if the session cache contains a cached value corresponding to the specified key.

contains(setOfKeys)

Returns `true` if the cache contains values for a specified set of keys.

get(key)

Returns the cached value corresponding to the specified key from the session cache.

get(keys)

Returns the cached values corresponding to the specified set of keys from the session cache.

get(cacheBuilder, key)

Returns the cached value corresponding to the specified key from the session cache. Use this method if your cached value is a class that implements the `CacheBuilder` interface.

getAvgGetSize()

Returns the average item size of all the keys fetched from the session cache, in bytes.

getAvgGetTime()

Returns the average time taken to get a key from the session cache, in nanoseconds.

getAvgValueSize()

Deprecated and available only in API versions 49.0 and earlier. Returns the average item size for keys in the session cache, in bytes.

getCapacity()

Returns the percentage of session cache capacity that has been used.

getKeys()

Returns all keys that are stored in the session cache and visible to the invoking namespace.

getMaxGetSize()

Returns the maximum item size of all the keys fetched from the session cache, in bytes.

getMaxGetTime()

Returns the maximum time taken to get a key from the session cache, in nanoseconds.

getMaxValueSize()

Deprecated and available only in API versions 49.0 and earlier. Returns the maximum item size for keys in the session cache, in bytes.

getMissRate()

Returns the miss rate in the session cache.

getName()

Returns the name of the default cache partition.

getNumKeys()

Returns the total number of keys in the session cache.

getPartition(partitionName)

Returns a partition from the session cache that corresponds to the specified partition name.

isAvailable()

Returns `true` if the session cache is available for use. The session cache isn't available when an active session isn't present, such as in asynchronous Apex or code called by asynchronous Apex. For example, if batch Apex causes an Apex trigger to execute, the session cache isn't available in the trigger because the trigger runs in asynchronous context.

put(key, value)

Stores the specified key/value pair as a cached entry in the session cache. The `put` method can write only to the cache in your org's namespace.

put(key, value, visibility)

Stores the specified key/value pair as a cached entry in the session cache and sets the cached value's visibility.

put(key, value, ttlSecs)

Stores the specified key/value pair as a cached entry in the session cache and sets the cached value's lifetime.

put(key, value, ttlSecs, visibility, immutable)

Stores the specified key/value pair as a cached entry in the session cache. This method also sets the cached value's lifetime, visibility, and whether it can be overwritten by another namespace.

remove(key)

Deletes the cached value corresponding to the specified key from the session cache.

remove(cacheBuilder, key)

Deletes the cached value corresponding to the specified key from the session cache. Use this method if your cached value is a class that implements the `CacheBuilder` interface.

`contains(key)`

Returns `true` if the session cache contains a cached value corresponding to the specified key.

Signature

```
public static Boolean contains(String key)
```

Parameters

`key`

Type: String

A case-sensitive string value that uniquely identifies a cached value. For information about the format of the key name, see Usage.

Return Value

Type: Boolean

`true` if a cache entry is found. Othewise, `false`.

contains(setOfKeys)

Returns `true` if the cache contains values for a specified set of keys.

Signature

```
public static Map <String, Boolean> contains (Set<String> keys)
```

Parameters

setOfKeys
Type: Set <String>

A set of keys that uniquely identifies cached values. For information about the format of the key name, see Usage.

Return Value

Type: Map <String, Boolean>

Returns the cache key and corresponding Boolean value indicating that the key entry exists. The Boolean value is `false` if the key entry doesn't exist.

Usage

The number of input keys cannot exceed the maximum limit of 10.

Example

In this example, the code checks for the presence of multiple keys on the default partition. It fetches the cache key and the corresponding Boolean value for the key entry from the session cache of the default partition.

```
Set<String> keys = new Set<String>{'key1','key2','key3','key4','key5'};
Map<String,Boolean> result = Cache.Session.contains(keys);
for(String key : result.keySet()) {
    system.debug('key: ' + key);
    system.debug('Is Key Present in the cache : ' +  result.get(key));
}
```

In this example, the code checks for the presence of multiple keys on different partitions. It fetches the cache key and the corresponding Boolean value for the key entry from the session cache of different partitions.

```
// Assuming there are three partitions p1, p2, p3 with default 'local' namespace

Set<String> keys = new Set<String>{'local.p1.key','local.p2.key', 'local.p3.key'};
Map<String,Boolean> result = Cache.Session.contains(keys);
for(String key : result.keySet()) {
    system.debug('key: ' + key);
    system.debug('Is Key Present in the cache : +  result.get(key));
}
```

get(key)

Returns the cached value corresponding to the specified key from the session cache.

Signature

```
public static Object get(String key)
```

Parameters

key
Type: String

A case-sensitive string value that uniquely identifies a cached value. For information about the format of the key name, see Usage.

Return Value

Type: Object

The cached value as a generic object type. Cast the returned value to the appropriate type.

Usage

Because `Cache.Session.get()` returns an object, we recommend that you cast the returned value to a specific type to facilitate use of the returned value.

```
// Get a cached value
Object obj = Cache.Session.get('ns1.partition1.orderDate');
// Cast return value to a specific data type
DateTime dt2 = (DateTime)obj;
```

If a `Cache.Session.get()` call doesn't find the referenced key, it returns `null`.

`get(keys)`

Returns the cached values corresponding to the specified set of keys from the session cache.

Signature

```
public static Map <String, Object> get (Set <String> keys)
```

Parameters

keys
Type: Set <String>

A set of keys that uniquely identify cached values. For information about the format of the key name, see Usage.

Return Value

Type: Map <String, Object>

Returns the cache key and corresponding value. Returns null when no corresponding value is found for an input key.

Usage

The number of input keys cannot exceed the maximum limit of 10.

Example

Fetch multiple keys from the session cache of the default partition.

```
Set<String> keys = new Set<String>{'key1','key2','key3','key4','key5'};
Map<String,Object> result = Cache.Session.get(keys);
for(String key : result.keySet()) {
    system.debug('key: ' + key);
    system.debug('value: ' +  result.get(key));
}
```

Fetch multiple keys from the session cache of different partitions.

```
// Assuming there are three partitions p1, p2, p3 with default 'local' namespace

Set<String> keys = new Set<String>{'local.p1.key','local.p2.key', 'local.p3.key'};
Map<String,Object> result = Cache.Session.get(keys);
for(String key : result.keySet()) {
    system.debug('key: ' + key);
    system.debug('value: ' +  result.get(key));
}
```

`get(cacheBuilder, key)`

Returns the cached value corresponding to the specified key from the session cache. Use this method if your cached value is a class that implements the `CacheBuilder` interface.

Signature

```
public static Object get(System.Type cacheBuilder, String key)
```

Parameters

`cacheBuilder`
Type: System.Type

The Apex class that implements the `CacheBuilder` interface.

`key`
Type: String

A case-sensitive string value that, combined with the class name corresponding to the *`cacheBuilder`* parameter, uniquely identifies a cached value.

Return Value

Type: Object

The cached value as a generic object type. Cast the returned value to the appropriate type.

Usage

Because `Cache.Session.get(`*`cacheBuilder`*`, `*`key`*`)` returns an object, cast the returned value to a specific type to facilitate use of the returned value.

```
return ((DateTime)Cache.Session.get(DateCache.class, 'datetime')).format();
```

`getAvgGetSize()`

Returns the average item size of all the keys fetched from the session cache, in bytes.

Signature

```
public static Long getAvgGetSize()
```

Return Value

Type: Long

`getAvgGetTime()`

Returns the average time taken to get a key from the session cache, in nanoseconds.

Signature

```
public static Long getAvgGetTime()
```

Return Value

Type: Long

`getAvgValueSize()`

Deprecated and available only in API versions 49.0 and earlier. Returns the average item size for keys in the session cache, in bytes.

Signature

```
public static Long getAvgValueSize()
```

Return Value

Type: Long

`getCapacity()`

Returns the percentage of session cache capacity that has been used.

Signature

```
public static Double getCapacity()
```

Return Value

Type: Double

Used cache as a percentage number.

getKeys()

Returns all keys that are stored in the session cache and visible to the invoking namespace.

Signature

```
public static Set<String> getKeys()
```

Return Value

Type: Set<String>

A set containing all cache keys.

getMaxGetSize()

Returns the maximum item size of all the keys fetched from the session cache, in bytes.

Signature

```
public static Long getMaxGetSize()
```

Return Value

Type: Long

getMaxGetTime()

Returns the maximum time taken to get a key from the session cache, in nanoseconds.

Signature

```
public static Long getMaxGetTime()
```

Return Value

Type: Long

getMaxValueSize()

Deprecated and available only in API versions 49.0 and earlier. Returns the maximum item size for keys in the session cache, in bytes.

Signature

```
public static Long getMaxValueSize()
```

Return Value

Type: Long

getMissRate()

Returns the miss rate in the session cache.

Signature

```
public static Double getMissRate()
```

Return Value

Type: Double

getName()

Returns the name of the default cache partition.

Signature

```
public String getName()
```

Return Value

Type: String

The name of the default cache partition.

getNumKeys()

Returns the total number of keys in the session cache.

Signature

```
public static Long getNumKeys()
```

Return Value

Type: Long

getPartition(partitionName)

Returns a partition from the session cache that corresponds to the specified partition name.

Signature

```
public static cache.SessionPartition getPartition(String partitionName)
```

Parameters

partitionName

Type: String

A partition name that is qualified by the namespace, for example, *namespace.partition*.

Return Value

Type: Cache.SessionPartition

Example

After you get the session partition, you can add and retrieve the partition's cache values.

```
// Get partition
Cache.SessionPartition sessionPart = Cache.Session.getPartition('myNs.myPartition');
// Retrieve cache value from the partition
if (sessionPart.contains('BookTitle')) {
    String cachedTitle = (String)sessionPart.get('BookTitle');
}

// Add cache value to the partition
sessionPart.put('OrderDate', Date.today());

// Or use dot notation to call partition methods
String cachedAuthor =
(String)Cache.Session.getPartition('myNs.myPartition').get('BookAuthor');
```

isAvailable()

Returns `true` if the session cache is available for use. The session cache isn't available when an active session isn't present, such as in asynchronous Apex or code called by asynchronous Apex. For example, if batch Apex causes an Apex trigger to execute, the session cache isn't available in the trigger because the trigger runs in asynchronous context.

Signature

```
public static Boolean isAvailable()
```

Return Value

Type: Boolean

`true` if the session cache is available. Otherwise, `false`.

put(key, value)

Stores the specified key/value pair as a cached entry in the session cache. The `put` method can write only to the cache in your org's namespace.

Signature

```
public static void put(String key, Object value)
```

Parameters

key
Type: String

A string that uniquely identifies the value to be cached. For information about the format of the key name, see Usage.

value
Type: Object

The value to store in the cache. The cached value must be serializable.

Return Value

Type: void

put(key, value, visibility)

Stores the specified key/value pair as a cached entry in the session cache and sets the cached value's visibility.

Signature

```
public static void put(String key, Object value, Cache.Visibility visibility)
```

Parameters

key
Type: String

A string that uniquely identifies the value to be cached. For information about the format of the key name, see Usage.

value
Type: Object

The value to store in the cache. The cached value must be serializable.

visibility
Type: Cache.Visibility

Indicates whether the cached value is available only to Apex code that is executing in the same namespace or to Apex code executing from any namespace.

Return Value

Type: void

put(key, value, ttlSecs)

Stores the specified key/value pair as a cached entry in the session cache and sets the cached value's lifetime.

Signature

```
public static void put(String key, Object value, Integer ttlSecs)
```

Parameters

key
Type: String

A string that uniquely identifies the value to be cached. For information about the format of the key name, see Usage.

value
Type: Object

The value to store in the cache. The cached value must be serializable.

ttlSecs
Type: Integer

The amount of time, in seconds, to keep the cached value in the session cache. The cached values remain in the cache as long as the Salesforce session hasn't expired. The maximum value is 28,800 seconds or eight hours. The minimum value is 300 seconds or five minutes.

Return Value

Type: void

put(key, value, ttlSecs, visibility, immutable)

Stores the specified key/value pair as a cached entry in the session cache. This method also sets the cached value's lifetime, visibility, and whether it can be overwritten by another namespace.

Signature

```
public static void put(String key, Object value, Integer ttlSecs, cache.Visibility visibility, Boolean immutable)
```

Parameters

key
Type: String

A string that uniquely identifies the value to be cached. For information about the format of the key name, see Usage.

value
Type: Object

The value to store in the cache. The cached value must be serializable.

ttlSecs
Type: Integer

The amount of time, in seconds, to keep the cached value in the session cache. The cached values remain in the cache as long as the Salesforce session hasn't expired. The maximum value is 28,800 seconds or eight hours. The minimum value is 300 seconds or five minutes.

visibility
Type: Cache.Visibility

Indicates whether the cached value is available only to Apex code that is executing in the same namespace or to Apex code executing from any namespace.

immutable
Type: Boolean

Indicates whether the cached value can be overwritten by another namespace (`false`) or not (`true`).

Return Value

Type: void

remove(key)

Deletes the cached value corresponding to the specified key from the session cache.

Signature

```
public static Boolean remove(String key)
```

Parameters

key
Type: String

A case-sensitive string value that uniquely identifies a cached value. For information about the format of the key name, see Usage.

Return Value

Type: Boolean

`true` if the cache value was successfully removed. Otherwise, `false`.

remove(cacheBuilder, key)

Deletes the cached value corresponding to the specified key from the session cache. Use this method if your cached value is a class that implements the `CacheBuilder` interface.

Signature

```
public static Boolean remove(System.Type cacheBuilder, String key)
```

Parameters

cacheBuilder
Type: System.Type

The Apex class that implements the `CacheBuilder` interface.

key
Type: String

A case-sensitive string value that, combined with the class name corresponding to the *cacheBuilder* parameter, uniquely identifies a cached value.

Return Value

Type: Boolean

`true` if the cache value was successfully removed. Otherwise, `false`.

SessionPartition Class

Contains methods to manage cache values in the session cache of a specific partition.

Namespace

Cache

Usage

This class extends Cache.Partition and inherits all of its non-static methods. Utility methods for creating and validating keys are not supported and can be called only from the `Cache.Partition` parent class. For a list of `Cache.Partition` methods, see Partition Methods.

To get a session partition, call `Cache.Session.getPartition` and pass in a fully qualified partition name, as follows.

```
Cache.SessionPartition sessionPartition =
Cache.Session.getPartition('namespace.myPartition');
```

See Cache Key Format for Partition Methods.

Example

This class is the controller for a sample Visualforce page (shown in the subsequent code sample). The controller shows how to use the methods of `Cache.SessionPartition` to manage a cache value on a particular partition. The controller takes inputs from the Visualforce page for the partition name, key name for a counter, and initial counter value. The controller contains default values for these inputs. When you click **Rerender** on the Visualforce page, the `go()` method is invoked and increases the counter by one. When you click **Remove Key**, the counter key is removed from the cache. The counter value gets reset to its initial value when it's re-added to the cache.

```
public class SessionPartitionController {

   // Name of a partition in the local namespace
   String partitionInput = 'local.myPartition';
   // Name of the key
   String counterKeyInput = 'counter';
   // Key initial value
   Integer counterInitValue = 0;
   // Session partition object
   Cache.SessionPartition sessionPartition;

    // Constructor of the controller for the Visualforce page.
    public SessionPartitionController() {
    }

    // Adds counter value to the cache.
    // This method is called when the Visualforce page loads.
    public void init() {
        // Create the partition instance based on the partition name
        sessionPartition = getPartition();

        // Add counter to the cache with an initial value
        //  or increment it if it's already there.
```

```
        if (!sessionPartition.contains(counterKeyInput)) {
            sessionPartition.put(counterKeyInput, counterInitValue);
        } else {
            sessionPartition.put(counterKeyInput, getCounter() + 1);
        }
    }

    // Returns the session partition based on the partition name
    // given in the Visualforce page or the default value.
    private Cache.SessionPartition getPartition() {
       if (sessionPartition == null) {
            sessionPartition = Cache.Session.getPartition(partitionInput);
       }

       return sessionPartition;
     }

    // Return counter from the cache.
    public Integer getCounter() {
        return (Integer)getPartition().get(counterKeyInput);
    }

    // Invoked by the Submit button to save input values
    //  supplied by the user.
    public PageReference save() {
        // Reset the initial key value in the cache
        getPartition().put(counterKeyInput, counterInitValue);

        return null;
    }

    // Method invoked by the Rerender button on the Visualforce page.
    // Updates the values of various cached values.
    // Increases the values of counter and the MyData counter if those
    //   cache values are still in the cache.
    public PageReference go() {
        // Get the partition object
        sessionPartition = getPartition();
        // Increase the cached counter value or set it to 0
        //  if it's not cached.
        if (sessionPartition.contains(counterKeyInput)) {
            sessionPartition.put(counterKeyInput, getCounter() + 1);
        } else {
            sessionPartition.put(counterKeyInput, counterInitValue);
        }

        return null;
    }

    // Method invoked by the Remove button on the Visualforce page.
    // Removes the datetime cached value from the session cache.
    public PageReference remove() {
        getPartition().remove(counterKeyInput);
```

```
        return null;
    }

    // Get and set methods for accessing variables
    // that correspond to the input text fields on
    // the Visualforce page.
    public String getPartitionInput() {
        return partitionInput;
    }

    public String getCounterKeyInput() {
        return counterKeyInput;
    }

    public Integer getCounterInitValue() {
        return counterInitValue;
    }

    public void setPartitionInput(String partition) {
        this.partitionInput = partition;
    }

    public void setCounterKeyInput(String keyName) {
        this.counterKeyInput = keyName;
    }

    public void setCounterInitValue(Integer counterValue) {
        this.counterInitValue = counterValue;
    }
}
```

This is the Visualforce page that corresponds to the `SessionPartitionController` class.

```
<apex:page controller="SessionPartitionController" action="{!init}">

    <apex:form >
        <br/>Partition with Namespace Prefix: <apex:inputText value="{!partitionInput}"/>

        <br/>Counter Key Name: <apex:inputText value="{!counterKeyInput}"/>
        <br/>Counter Initial Value: <apex:inputText value="{!counterInitValue}"/>
        <apex:commandButton action="{!save}" value="Save Key Input Values"/>
    </apex:form>

    <apex:outputPanel id="output">
        <br/>Cached Counter: <apex:outputText value="{!counter}"/>
    </apex:outputPanel>

    <br/>
    <apex:form >
        <apex:commandButton id="go" action="{!go}" value="Rerender" rerender="output"/>
        <apex:commandButton id="remove" action="{!remove}" value="Remove Key"
rerender="output"/>
    </apex:form>
```

```
</apex:page>
```

SEE ALSO:

Apex Developer Guide: Platform Cache

Cache Exceptions

The `Cache` namespace contains exception classes.

All exception classes support built-in methods for returning the error message and exception type. See Exception Class and Built-In Exceptions on page 3499 in the *Apex Developer Guide*.

The `Cache` namespace contains these exceptions.

Exception	Thrown when
`Cache.Session.SessionCacheException`	An error occurred while adding or retrieving a value in the session cache.
`Cache.Session.SessionCacheNoSessionException`	An attempt is made to access the cache when the session cache isn't available.
`Cache.Org.OrgCacheException`	An attempt is made to access a partition that doesn't exist or whose name is invalid.
`Cache.InvalidParamException`	An invalid parameter value is passed into a method of `Cache.Session` or `Cache.Org`. This error occurs when: • The key referenced is null or empty or is not alphanumeric. • The key isn't qualified with the namespace and partition in the format `<namespace>.<partition>.<key>`. • The key isn't qualified in the format `<key>` for the default partition, or for a key inserted through the partition object. • The namespace referenced is null or empty. • The partition name is null or empty or is not alphanumeric. • Another referenced value is null.
`Cache.ItemSizeLimitExceededException`	A cache `put` call is made with an item that exceeds the maximum size limit. To fix this error, break the item into multiple, smaller items.
`Cache.BulkApiKeysLimitExceededException`	The number of key parameters passed into a bulk method - `get(keys)` or `contains(setOfKeys)` exceeds the maximum limit of 10.
`Cache.PlatformCacheInvalidOperationException`	A cache `put` or `remove` call is made that is not allowed. For example, when calling `put` or `remove` inside a Visualforce constructor.
`Cache.CacheBuilderExecutionException`	This error occurs when the execution of the CacheBuilder fails; this could be due to an error in parsing, a permissions error while accessing records, or an issue with Apex callouts.

Exception	Thrown when
`Cache.InvalidCacheBuilderException`	A `get(CacheBuilder cb, String key)`, `remove(CacheBuilder cb, String key)`, or `validateCacheBuilder(CacheBuilder cb)` method is called but the `cb` parameter is a class that does not implement the `Cache.CacheBuilder` interface.

Visibility Enum

Use the `Cache.Visibility` enumeration in the `Cache.Session` or `Cache.Org` methods to indicate whether a cached value is visible only in the value's namespace or in all namespaces.

Enum Values

The following are the values of the `Cache.Visibility` enum.

Value	Description
`ALL`	The cached value is available to Apex code executing from any namespace. This is the default state.
`NAMESPACE`	The cached value is available to Apex code executing from the same namespace. If a key has the `Visibility.NAMESPACE` attribute, a `get` method initiated from a different namespace returns `null`.

Canvas Namespace

The `Canvas` namespace provides an interface and classes for canvas apps in Salesforce.

The following are the interfaces and classes in the `Canvas` namespace.

IN THIS SECTION:

ApplicationContext Interface
Use this interface to retrieve application context information, such as the application version or URL.

CanvasLifecycleHandler Interface
Implement this interface to control context information and add custom behavior during the application render phase.

ContextTypeEnum Enum
Describes context data that can be excluded from canvas app context data. You specify which context types to exclude in the `excludeContextTypes()` method in your `CanvasLifecycleHandler` implementation.

EnvironmentContext Interface
Use this interface to retrieve environment context information, such as the app display location or the configuration parameters.

RenderContext Interface

A wrapper interface that is used to retrieve application and environment context information.

Test Class

Contains methods for automated testing of your Canvas classes.

Canvas Exceptions

The `Canvas` namespace contains exception classes.

SEE ALSO:

Canvas Developer Guide

ApplicationContext Interface

Use this interface to retrieve application context information, such as the application version or URL.

Namespace

Canvas

Usage

The `ApplicationContext` interface provides methods to retrieve application information about the canvas app that's being rendered. Most of the methods are read-only. For this interface, you don't need to create an implementation. Use the default implementation that Salesforce provides.

IN THIS SECTION:

ApplicationContext Methods

ApplicationContext Methods

The following are methods for `ApplicationContext`.

IN THIS SECTION:

getCanvasUrl()

Retrieves the fully qualified URL of the canvas app.

getDeveloperName()

Retrieves the internal API name of the canvas app.

getName()

Retrieves the name of the canvas app.

getNamespace()

Retrieves the namespace prefix of the canvas app.

getVersion()

Retrieves the current version of the canvas app.

setCanvasUrlPath(newPath)

Overrides the URL of the canvas app for the current request.

getCanvasUrl()

Retrieves the fully qualified URL of the canvas app.

Signature

```
public String getCanvasUrl()
```

Return Value

Type: String

Usage

Use this method to get the URL of the canvas app, for example: `http://instance.salesforce.com:8080/canvas_app_path/canvas_app.jsp`.

getDeveloperName()

Retrieves the internal API name of the canvas app.

Signature

```
public String getDeveloperName()
```

Return Value

Type: String

Usage

Use this method to get the API name of the canvas app. You specify this value in the `API Name` field when you expose the canvas app by creating a connected app.

getName()

Retrieves the name of the canvas app.

Signature

```
public String getName()
```

Return Value

Type: String

Usage

Use this method to get the name of the canvas app.

`getNamespace()`

Retrieves the namespace prefix of the canvas app.

Signature

```
public String getNamespace()
```

Return Value

Type: String

Usage

Use this method to get the Salesforce namespace prefix that's associated with the canvas app.

`getVersion()`

Retrieves the current version of the canvas app.

Signature

```
public String getVersion()
```

Return Value

Type: String

Usage

Use this method to get the current version of the canvas app. This value changes after you update and republish a canvas app in an organization. If you are in a Developer Edition organization, using this method always returns the latest version.

`setCanvasUrlPath(newPath)`

Overrides the URL of the canvas app for the current request.

Signature

```
public void setCanvasUrlPath(String newPath)
```

Parameters

`newPath`
Type: String

The URL (not including domain) that you need to use to override the canvas app URL.

Return Value

Type: Void

Usage

Use this method to override the URL path and query string of the canvas app. Do not provide a fully qualified URL, because the provided URL string will be appended to the original canvas URL domain.

For example, if the current canvas app URL is `https://myserver.com:6000/myAppPath` and you call `setCanvasUrlPath('/alternatePath/args?arg1=1&arg2=2')`, the adjusted canvas app URL will be `https://myserver.com:6000/alternatePath/args?arg1=1&arg2=2`.

If the provided path results in a malformed URL, or a URL that exceeds 2,048 characters, a System.CanvasException will be thrown.

This method overrides the canvas app URL for the current request and does not permanently change the canvas app URL as configured in the UI for the Salesforce canvas app settings.

CanvasLifecycleHandler Interface

Implement this interface to control context information and add custom behavior during the application render phase.

Namespace

Canvas

Usage

Use this interface to specify what canvas context information is provided to your app by implementing the `excludeContextTypes()` method. Use this interface to call custom code when the app is rendered by implementing the `onRender()` method.

If you provide an implementation of this interface, you must implement `excludeContextTypes()` and `onRender()`.

Example Implementation

The following example shows a simple implementation of CanvasLifecycleHandler that specifies that organization context information will be excluded and prints a debug message when the app is rendered.

```
public class MyCanvasListener
implements Canvas.CanvasLifecycleHandler{
    public Set<Canvas.ContextTypeEnum> excludeContextTypes(){
        Set<Canvas.ContextTypeEnum> excluded = new Set<Canvas.ContextTypeEnum>();
        excluded.add(Canvas.ContextTypeEnum.ORGANIZATION);
        return excluded;
    }

    public void onRender(Canvas.RenderContext renderContext){
        System.debug('Canvas lifecycle called.');
    }
}
```

IN THIS SECTION:

CanvasLifecycleHandler Methods

SEE ALSO:

Canvas Developer Guide: Customizing Your App Lifecycle

CanvasLifecycleHandler Methods

The following are methods for `CanvasLifecycleHandler`.

IN THIS SECTION:

excludeContextTypes()
Lets the implementation exclude parts of the CanvasRequest context, if the application does not need it.

onRender(renderContext)
Invoked when a canvas app is rendered. Provides the ability to set and retrieve canvas application and environment context information during the application render phase.

excludeContextTypes()

Lets the implementation exclude parts of the CanvasRequest context, if the application does not need it.

Signature

```
public Set<Canvas.ContextTypeEnum> excludeContextTypes()
```

Return Value

Type: SET<Canvas.ContextTypeEnum>

This method must return `null` or a set of zero or more ContextTypeEnum values. Returning `null` enables all attributes by default. ContextTypeEnum values that can be set are:

- Canvas.ContextTypeEnum.ORGANIZATION
- Canvas.ContextTypeEnum.RECORD_DETAIL
- Canvas.ContextTypeEnum.USER

See ContextTypeEnum on page 272 for more details on these values.

Usage

Implement this method to specify which attributes to disable in the context of the canvas app. A disabled attribute will set the associated canvas context information to null.

Disabling attributes can help improve performance by reducing the size of the signed request and canvas context. Also, disabled attributes do not need to be retrieved by Salesforce, which further improves performance.

See the *Canvas Developer Guide* for more information on context information in the Context object that's provided in the CanvasRequest.

Example

This example implementation specifies that the organization information will be disabled in the canvas context.

```
public Set<Canvas.ContextTypeEnum> excludeContextTypes() {
    Set<Canvas.ContextTypeEnum> excluded = new Set<Canvas.ContextTypeEnum>();
    excluded.add(Canvas.ContextTypeEnum.ORGANIZATION);
    return excluded;
}
```

SEE ALSO:

Canvas Developer Guide: Filtering CanvasRequest Context Data

onRender(renderContext)

Invoked when a canvas app is rendered. Provides the ability to set and retrieve canvas application and environment context information during the application render phase.

Signature

```
public void onRender(Canvas.RenderContext renderContext)
```

Parameters

renderContext
Type: Canvas.RenderContext

Return Value

Type: Void

Usage

If implemented, this method is called whenever the canvas app is rendered. The implementation can set and retrieve context information by using the provided Canvas.RenderContext.

This method is called whenever signed request or context information is retrieved by the client. See the *Canvas Developer Guide* for more information on signed request authentication.

Example

This example implementation prints 'Canvas lifecycle called.' to the debug log when the canvas app is rendered.

```
public void onRender(Canvas.RenderContext renderContext) {
    System.debug('Canvas lifecycle called.');
}
```

SEE ALSO:

Canvas Developer Guide: Controlling App Behavior

ContextTypeEnum Enum

Describes context data that can be excluded from canvas app context data. You specify which context types to exclude in the `excludeContextTypes()` method in your `CanvasLifecycleHandler` implementation.

Namespace

Canvas

Enum Values

Value	Description
`ORGANIZATION`	Exclude context information about the organization in which the canvas app is running.
`RECORD_DETAIL`	Exclude context information about the object record on which the canvas app appears.
`USER`	Exclude context information about the current user.

EnvironmentContext Interface

Use this interface to retrieve environment context information, such as the app display location or the configuration parameters.

Namespace

Canvas

Usage

The `EnvironmentContext` interface provides methods to retrieve environment information about the current canvas app. For this interface, you don't need to create an implementation. Use the default implementation that Salesforce provides.

IN THIS SECTION:

EnvironmentContext Methods

EnvironmentContext Methods

The following are methods for `EnvironmentContext`.

IN THIS SECTION:

addEntityField(fieldName)
Adds a field to the list of object fields that are returned in the signed request Record object when the component appears on a Visualforce page that's placed on an object.

addEntityFields(fieldNames)

Adds a set of fields to the list of object fields that are returned in the signed request Record object when the component appears on a Visualforce page that's placed on an object.

getDisplayLocation()

Retrieves the display location where the canvas app is being called from. For example, a value of Visualforce page.

getEntityFields()

Retrieves the list of object fields that are returned in the signed request Record object when the component appears on a Visualforce page that's placed on an object.

getLocationUrl()

Retrieves the location URL of the canvas app.

getParametersAsJSON()

Retrieves the current custom parameters for the canvas app. Parameters are returned as a JSON string.

getSublocation()

Retrieves the display sublocation where the canvas app is being called from.

setParametersAsJSON(jsonString)

Sets the custom parameters for the canvas app.

addEntityField(fieldName)

Adds a field to the list of object fields that are returned in the signed request Record object when the component appears on a Visualforce page that's placed on an object.

Signature

```
public void addEntityField(String fieldName)
```

Parameters

fieldName

Type: String

The object field name that you need to add to the list of returned fields., Using '*' adds all fields that the user has permission to view.

Return Value

Type: Void

Usage

When you use the `<apex:canvasApp>` component to display a canvas app on a Visualforce page, and that page is associated with an object (placed on the page layout, for example), you can specify fields to be returned from the related object. See the Canvas Developer Guide for more information on the Record object.

Use `addEntityField()` to add a field to the list of object fields that are returned in the signed request Record object. By default the list of fields includes ID. You can add fields by name or add all fields that the user has permission to view by calling `addEntityField('*')`.

You can inspect the configured list of fields by using Canvas.EnvironmentContext.`getEntityFields()`.

Example

This example adds the Name and BillingAddress fields to the list of object fields. This example assumes the canvas app will appear in a Visualforce page that's associated with the Account page layout.

```
Canvas.EnvironmentContext env = renderContext.getEnvironmentContext();

// Add Name and BillingAddress to fields (assumes we'll run from the Account detail page)
env.addEntityField('Name');
env.addEntityField('BillingAddress');
```

addEntityFields(fieldNames)

Adds a set of fields to the list of object fields that are returned in the signed request Record object when the component appears on a Visualforce page that's placed on an object.

Signature

```
public void addEntityFields(Set<String> fieldNames)
```

Parameters

fieldNames
Type: SET<String>

The set of object field names that you need to add to the list of returned fields. If an item in the set is '*', all fields that the user has permission to view are added.

Return Value

Type: Void

Usage

When you use the `<apex:canvasApp>` component to display a canvas app on a Visualforce page, and that page is associated with an object (placed on the page layout, for example), you can specify fields to be returned from the related object. See the *Canvas Developer Guide* for more information on the Record object.

Use `addEntityFields()` to add a set of one or more fields to the list of object fields that are returned in the signed request Record object. By default the list of fields includes ID. You can add fields by name or add all fields that the user has permission to view by adding a set that includes '*' as one of the strings.

You can inspect the configured list of fields by using Canvas.EnvironmentContext.`getEntityFields()`.

Example

This example adds the Name, BillingAddress, and YearStarted fields to the list of object fields. This example assumes that the canvas app will appear in a Visualforce page that's associated with the Account page layout.

```
Canvas.EnvironmentContext env = renderContext.getEnvironmentContext();

// Add Name, BillingAddress and YearStarted to fields (assumes we'll run from the Account
 detail page)
```

```
Set<String> fields = new Set<String>{'Name','BillingAddress','YearStarted'};
env.addEntityFields(fields);
```

getDisplayLocation()

Retrieves the display location where the canvas app is being called from. For example, a value of Visualforce page.

Signature

```
public String getDisplayLocation()
```

Return Value

Type: String

The return value can be one of the following strings:

- Chatter—The canvas app was called from the Chatter tab.
- ChatterFeed—The canvas app was called from a Chatter canvas feed item.
- MobileNav—The canvas app was called from the navigation menu.
- OpenCTI—The canvas app was called from an Open CTI component.
- PageLayout—The canvas app was called from an element within a page layout. If the displayLocation is PageLayout, one of the subLocation values might be returned.
- Publisher—The canvas app was called from a canvas custom quick action.
- ServiceDesk—The canvas app was called from a Salesforce Console component.
- Visualforce—The canvas app was called from a Visualforce page.
- None—The canvas app was called from the Canvas App Previewer.

Usage

Use this method to obtain the display location for the canvas app.

getEntityFields()

Retrieves the list of object fields that are returned in the signed request Record object when the component appears on a Visualforce page that's placed on an object.

Signature

```
public List<String> getEntityFields()
```

Return Value

Type: LIST<String>

Usage

When you use the `<apex:canvasApp>` component to display a canvas app on a Visualforce page, and that page is associated with an object (placed on the page layout, for example), you can specify fields to be returned from the related object. See the *Canvas Developer Guide* for more information on the Record object.

Use getEntityFields() to retrieve the list of object fields that are returned in the signed request Record object. By default the list of fields includes ID. The list of fields can be configured by using the Canvas.EnvironmentContext.addEntityField(fieldName) or Canvas.EnvironmentContext.addEntityFields(fieldNames) methods.

Example

This example gets the current list of object fields and retrieves each item in the list, printing each field name to the debug log.

```
Canvas.EnvironmentContext env = renderContext.getEnvironmentContext();

List<String> entityFields = env.getEntityFields();
for (String fieldVal : entityFields) {
    System.debug('Environment Context entityField: ' + fieldVal);
}
```

If the canvas app that's using this lifecycle code was run from the detail page of an Account, the debug log output might look like:

```
Environment Context entityField: Id
```

getLocationUrl()

Retrieves the location URL of the canvas app.

Signature

```
public String getLocationUrl()
```

Return Value

Type: String

Usage

Use this method to obtain the URL of the page where the user accessed the canvas app. For example, if the user accessed your app by clicking a link on the Chatter tab, this method returns the URL of the Chatter tab, which would be similar to 'https://*MyDomainName*.my.salesforce.com/_ui/core/chatter/ui/ChatterPage'.

getParametersAsJSON()

Retrieves the current custom parameters for the canvas app. Parameters are returned as a JSON string.

Signature

```
public String getParametersAsJSON()
```

Return Value

Type: String

Usage

Use this method to get the current custom parameters for the canvas app. The parameters are returned in a JSON string that can be de-serialized by using the System.JSON.`deserializeUntyped(jsonString)` method.

Custom parameters can be modified by using the Canvas.EnvironmentContext.`setParametersAsJSON(jsonString)` string.

Example

This example gets the current custom parameters, de-serializes them into a map, and prints the results to the debug log.

```
Canvas.EnvironmentContext env = renderContext.getEnvironmentContext();

// Get current custom params
Map<String, Object> currentParams =
    (Map<String, Object>) JSON.deserializeUntyped(env.getParametersAsJSON());
System.debug('Environment Context custom paramters: ' + currentParams);
```

`getSublocation()`

Retrieves the display sublocation where the canvas app is being called from.

Signature

```
public String getSublocation()
```

Return Value

Type: String

The return value can be one of the following strings:

- S1MobileCardFullview—The canvas app was called from a mobile card.
- S1MobileCardPreview—The canvas app was called from a mobile card preview. The user must click the preview to open the app.
- S1RecordHomePreview—The canvas app was called from a record detail page preview. The user must click the preview to open the app.
- S1RecordHomeFullview—The canvas app was called from a page layout.

Usage

Use this method to obtain the display sublocation for the canvas app. Use only if the primary display location can be displayed on mobile devices.

`setParametersAsJSON(jsonString)`

Sets the custom parameters for the canvas app.

Signature

```
public void setParametersAsJSON(String jsonString)
```

Parameters

jsonString
Type: String

The custom parameters that you need to set, serialized into a JSON format string.

Return Value

Type: Void

Usage

Use this method to set the current custom parameters for the canvas app. The parameters must be provided in a JSON string. You can use the System.JSON.`serialize(objectToSerialize)` method to serialize a map into a JSON string.

Setting the custom parameters will overwrite the custom parameters that are set for the current request. If you need to modify the current custom parameters, first get the current set of custom parameters by using `getParametersAsJSON()`, modify the retrieved parameter set as needed, and then use this modified set in your call to setParametersAsJSON().

If the provided JSON string exceeds 32KB, a System.CanvasException will be thrown.

Example

This example gets the current custom parameters, adds a new `newCustomParam` parameter with a value of 'TESTVALUE', and sets the current custom parameters.

```
Canvas.EnvironmentContext env = renderContext.getEnvironmentContext();

// Get current custom params
Map<String, Object> previousParams =
    (Map<String, Object>) JSON.deserializeUntyped(env.getParametersAsJSON());

// Add a new custom param
previousParams.put('newCustomParam','TESTVALUE');

// Now replace the parameters with the current parameters plus our new custom param
env.setParametersAsJSON(JSON.serialize(previousParams));
```

RenderContext Interface

A wrapper interface that is used to retrieve application and environment context information.

Namespace

Canvas

Usage

Use this interface to retrieve application and environment context information for your canvas app. For this interface, you don't need to create an implementation. Use the default implementation that Salesforce provides.

IN THIS SECTION:

RenderContext Methods

RenderContext Methods

The following are methods for `RenderContext`.

IN THIS SECTION:

getApplicationContext()
Retrieves the application context information.

getEnvironmentContext()
Retrieves the environment context information.

`getApplicationContext()`

Retrieves the application context information.

Signature

```
public Canvas.ApplicationContext getApplicationContext()
```

Return Value

Type: Canvas.ApplicationContext

Usage

Use this method to get the application context information for your canvas app.

Example

The following example implementation of the CanvasLifecycleHandler onRender() method uses the provided RenderContext to retrieve the application context information and then checks the namespace, version, and app URL.

```
public void onRender(Canvas.RenderContext renderContext){
    Canvas.ApplicationContext app = renderContext.getApplicationContext();
    if (!'MyNamespace'.equals(app.getNamespace())){
        // This application is installed, add code as needed
        ...
    }

    // Check the application version
    Double currentVersion = Double.valueOf(app.getVersion());

    if (currentVersion <= 5){
        // Add version specific code as needed
        ...
        // Tell the canvas application to operate in deprecated mode
        app.setCanvasUrlPath('/canvas?deprecated=true');
```

```
    }
}
```

getEnvironmentContext()

Retrieves the environment context information.

Signature

```
public Canvas.EnvironmentContext getEnvironmentContext()
```

Return Value

Type: Canvas.EnvironmentContext

Usage

Use this method to get the environment context information for your canvas app.

Example

The following example implementation of the CanvasLifecycleHandler onRender() method uses the provided RenderContext to retrieve the environment context information and then modifies the custom parameters.

```
public void onRender(Canvas.RenderContext renderContext) {
   Canvas.EnvironmentContext env =
        renderContext.getEnvironmentContext();

    // Retrieve the custom params
    Map<String, Object> previousParams = (Map<String, Object>)
          JSON.deserializeUntyped(env.getParametersAsJSON());

    previousParams.put('param1',1);
    previousParams.put('param2',3.14159);

    ...

    // Now, add in some opportunity record IDs
    Opportunity[] o = [select id, name from opportunity];
    previousParams.put('opportunities',o);

    // Now, replace the parameters
    env.setParametersAsJSON(JSON.serialize(previousParams));
}
```

Test Class

Contains methods for automated testing of your Canvas classes.

Namespace

Canvas

Usage

Use this class to test your implementation of Canvas.CanvasLifecycleHandler with mock test data. You can create a test Canvas.RenderContext with mock application and environment context data and use this data to verify that your CanvasLifecycleHandler is being invoked correctly.

IN THIS SECTION:

Test Constants

The Test class provides constants that are used as keys when you set mock application and environment context data.

Test Methods

The Test class provides methods for creating test contexts and invoking your CanvasLifecycleHandler with mock data.

SEE ALSO:

Canvas Developer Guide: Testing Your CanvasLifecycleHandler Implementation

Test Constants

The Test class provides constants that are used as keys when you set mock application and environment context data.

When you call Canvas.Test.`mockRenderContext(applicationContextTestValues, environmentContextTestValues)`, you need to provide maps of key-value pairs to represent your mock application and environment context data. The Test class provides static constant strings that you can use as keys for various parts of the application and environment context.

Constant	Description
`KEY_CANVAS_URL`	Represents the canvas app URL key in the ApplicationContext.
`KEY_DEVELOPER_NAME`	Represents the canvas app developer or API name key in the ApplicationContext.
`KEY_DISPLAY_LOCATION`	Represents the canvas app display location key in the EnvironmentContext.
`KEY_LOCATION_URL`	Represents the canvas app location URL key in the EnvironmentContext.
`KEY_NAME`	Represents the canvas app name key in the ApplicationContext.
`KEY_NAMESPACE`	Represents the canvas app namespace key in the ApplicationContext.
`KEY_SUB_LOCATION`	Represents the canvas app sublocation key in the EnvironmentContext.
`KEY_VERSION`	Represents the canvas app version key in the ApplicationContext.

Test Methods

The Test class provides methods for creating test contexts and invoking your CanvasLifecycleHandler with mock data.

The following are methods for `Test`. All are static methods.

IN THIS SECTION:

mockRenderContext(applicationContextTestValues, environmentContextTestValues)
Creates and returns a test Canvas.RenderContext based on the provided application and environment context parameters.

testCanvasLifecycle(lifecycleHandler, mockRenderContext)
Calls the Canvas test framework to invoke a CanvasLifecycleHandler with the provided RenderContext.

mockRenderContext(applicationContextTestValues, environmentContextTestValues)

Creates and returns a test Canvas.RenderContext based on the provided application and environment context parameters.

Signature

```
public static Canvas.RenderContext mockRenderContext(Map<String,String>
applicationContextTestValues, Map<String,String> environmentContextTestValues)
```

Parameters

applicationContextTestValues
Type: Map<String,String>

Specifies a map of key-value pairs that provide mock application context data. Use constants that are provided by Canvas.Test as keys. If `null` is provided for this parameter, the canvas framework generates some default mock application context values.

environmentContextTestValues
Type: Map<String,String>

Specifies a map of key-value pairs that provide mock environment context data. Use constants provided by Canvas.Test as keys. If `null` is provided for this parameter, the canvas framework generates some default mock environment context values.

Return Value

Type: Canvas.RenderContext

Usage

Use this method to create a mock Canvas.RenderContext. Use the returned RenderContext in calls to Canvas.Test.`testCanvasLifecycle(lifecycleHandler, mockRenderContext)` for testing Canvas.CanvasLifecycleHandler implementations.

Example

The following example creates maps to represent mock application and environment context data and generates a test Canvas.RenderContext. This test RenderContext can be used in a call to Canvas.Test.`testCanvasLifecycle(lifecycleHandler, mockRenderContext)`.

```
Map<String,String> appValues = new Map<String,String>();
appValues.put(Canvas.Test.KEY_NAMESPACE,'alternateNamespace');
appValues.put(Canvas.Test.KEY_VERSION,'3.0');

Map<String,String> envValues = new Map<String,String>();
envValues.put(Canvas.Test.KEY_DISPLAY_LOCATION,'Chatter');
envValues.put(Canvas.Test.KEY_LOCATION_URL,'https://MyDomainName.my.salesforce.com/_ui/core/chatter/ui/ChatterPage');
```

```
Canvas.RenderContext mock = Canvas.Test.mockRenderContext(appValues,envValues);
```

SEE ALSO:

Canvas Developer Guide: Testing Your CanvasLifecycleHandler Implementation

`testCanvasLifecycle(lifecycleHandler, mockRenderContext)`

Calls the Canvas test framework to invoke a CanvasLifecycleHandler with the provided RenderContext.

Signature

```
public static Void testCanvasLifecycle(Canvas.CanvasLifecycleHandler
lifecycleHandler,Canvas.RenderContext mockRenderContext)
```

Parameters

`lifecycleHandler`

Type: Canvas.CanvasLifecycleHandler

Specifies the CanvasLifecycleHandler implementation that you need to invoke.

`mockRenderContext`

Type: Canvas.RenderContext

Specifies the RenderContext information that you need to provide to the invoked CanvasLifecycleHandler. If `null` is provided for this parameter, the canvas framework generates and uses a default mock RenderContext.

Return Value

Type: Void

Usage

Use this method to invoke an implementation of Canvas.CanvasLifecycleHandler.`onRender(renderContext)` with a mock Canvas.RenderContext that you provide.

Example

The following example creates an Apex test class that uses maps to represent mock application and environment context data. The mock RenderContext object is then used to invoke a CanvasLifecycleHandler object. In this example, the CanvasLifecycleHandler is defined as MyCanvasListener, which is example implementation provided in Canvas.RenderContext.

```
@IsTest
global class CanvasRendercontextTest {
    @IsTest
    static void testRenderContext(){
        // Set some application context data in a Map
        Map<String,String> appValues = new Map<String,String>();
        appValues.put(Canvas.Test.KEY_NAMESPACE,'alternateNamespace');
        appValues.put(Canvas.Test.KEY_VERSION,'3.0');
```

```
        // Set some environment context data in a MAp
        Map<String,String> envValues = new Map<String,String>();
        envValues.put(Canvas.Test.KEY_DISPLAY_LOCATION,'Chatter');

envValues.put(Canvas.Test.KEY_LOCATION_URL,'https://MyDomainName.my.salesforce.com/_ui/core/chatter/ui/ChatterPage');


        // Create a mock RenderContext using the test application and environment context
 data Maps
        Canvas.RenderContext mock = Canvas.Test.mockRenderContext(appValues,envValues);

        // Set some custom params on the mock RenderContext

mock.getEnvironmentContext().setParametersAsJSON('{\"param1\":1,\"boolParam\":true,\"stringParam\":\"test
 string\"}');

        // Create a CanvasLifecycleHandler
        MyCanvasListener handler = new MyCanvasListener();

        // Use the mock RenderContext to invoke the CanvasLifecycleHandler
        Canvas.Test.testCanvasLifecycle(handler,mock);
 }
}
```

SEE ALSO:

Canvas Developer Guide: Testing Your CanvasLifecycleHandler Implementation

Canvas Exceptions

The `Canvas` namespace contains exception classes.

All exception classes support built-in methods for returning the error message and exception type. See Exception Class and Built-In Exceptions.

The `Canvas` namespace contains this exception:

Exception	Description
`Canvas.CanvasRenderException`	Use this class in your implementation of `Canvas.CanvasLifecycleHandler.onRender(renderContext)`. To show an error to the user in your `onRender()` implementation, throw a `Canvas.CanvasRenderException`, and the canvas framework will render the error message to the user. This exception will be managed only within the `onRender()` method.

Example

The following example implementation of `onRender()` catches a CanvasException that was thrown because a canvas URL was set with a string that exceeded the maximum length. A CanvasRenderException is created and thrown to display the error to the user.

```
public class MyCanvasListener
implements Canvas.CanvasLifecycleHandler {

    public void onRender(Canvas.RenderContext renderContext) {
        Canvas.ApplicationContext app = renderContext.getApplicationContext();

        // Code to generate a URL string that is too long

        // ...

        // Try to set the canvas app URL using the invalid URL string
        try {
            app.setCanvasUrlPath(aUrlPathThatIsTooLong);
        } catch (CanvasException e) {
            // Display error to user by throwing a new CanvasRenderException
            throw new Canvas.CanvasRenderException(e.getMessage());
        }
    }
}
```

See the *Canvas Developer Guide* for additional examples that use CanvasRenderException.

ChatterAnswers Namespace

The `ChatterAnswers` namespace provides an interface for creating Account records.

The following is the interface in the `ChatterAnswers` namespace.

IN THIS SECTION:

AccountCreator Interface
Creates Account records that will be associated with Chatter Answers users.

AccountCreator Interface

Creates Account records that will be associated with Chatter Answers users.

Namespace

ChatterAnswers

Usage

The `ChatterAnswers.AccountCreator` is specified in the `registrationClassName` attribute of a `chatteranswers:registration` Visualforce component. This interface is called by Chatter Answers and allows for custom creation of Account records used for portal users.

To implement the `ChatterAnswers.AccountCreator` interface, you must first declare a class with the `implements` keyword as follows:

```
public class ChatterAnswersRegistration implements ChatterAnswers.AccountCreator {
```

Next, your class must provide an implementation for the following method:

```
public String createAccount(String firstname, String lastname, Id siteAdminId) {
    // Your code here
}
```

The implemented method must be declared as `global` or `public`.

IN THIS SECTION:

AccountCreator Methods

AccountCreator Example Implementation

AccountCreator Methods

The following are methods for `AccountCreator`.

IN THIS SECTION:

createAccount(firstName, lastName, siteAdminId)

Accepts basic user information and creates an Account record. The implementation of this method returns the account ID.

`createAccount(firstName, lastName, siteAdminId)`

Accepts basic user information and creates an Account record. The implementation of this method returns the account ID.

Signature

```
public String createAccount(String firstName, String lastName, Id siteAdminId)
```

Parameters

`firstName`
Type: String

The first name of the user who is registering.

`lastName`
Type: String

The last name of the user who is registering.

`siteAdminId`
Type: ID

The user ID of the Site administrator, used for notification if any exceptions occur.

Return Value

Type: String

AccountCreator Example Implementation

This is an example implementation of the `ChatterAnswers.AccountCreator` interface. The `createAccount` method implementation accepts user information and creates an Account record. The method returns a String value for the Account ID.

```
public class ChatterAnswersRegistration implements ChatterAnswers.AccountCreator {
    public String createAccount(String firstname, String lastname, Id siteAdminId) {
        Account a = new Account(name = firstname + ' ' + lastname, ownerId = siteAdminId);

         insert a;
         return a.Id;
    }
}
```

This example tests the code above.

```
@isTest
private class ChatterAnswersCreateAccountTest {
    static testMethod void validateAccountCreation() {
       User[] user = [SELECT Id, Firstname, Lastname from User WHERE UserType='Standard'];

        if (user.size() == 0) { return; }
        String firstName = user[0].FirstName;
        String lastName = user[0].LastName;
        String userId = user[0].Id;
        String accountId = new ChatterAnswersRegistration().createAccount(firstName,
lastName, userId);
        Account acct = [SELECT name, ownerId from Account where Id =: accountId];
        System.assertEquals(firstName + ' ' + lastName, acct.name);
        System.assertEquals(userId, acct.ownerId);
  }
}
```

CommerceExtension Namespace

Use the `CommerceExtension` namespace to define resolution strategies for registered Commerce extensions.

The following are the classes in the `CommerceExtension` namespace.

IN THIS SECTION:

ExtensionInfo Class

Contains static methods to expose extension-related context information.

Resolution Class

Resolution of a resolution strategy, which conditionally invokes default domain logic, logic provided by an extension provider, or no logic.

ResolutionException Class

Exception indicating a problem with the execution of a resolution strategy.

ResolutionStates Enum

Potential resolution states for a resolution strategy.

ResolutionStrategy Interface
Interface for a resolution strategy.

ExtensionInfo Class

Contains static methods to expose extension-related context information.

Namespace

CommerceExtension on page 287

Example

```
        // The Sample Extension Provider registered with developer name as
       // 'tax_extension_provider_for_us' will be selected for execution for en_US locale

        if(CommerceExtension.ExtensionInfo.getLocaleString() == 'en_US') {
            return new CommerceExtension.Resolution('tax_extension_provider_for_us');
        }
        // The Sample Extension Provider registered with developer name as
        // 'tax_extension_provider_for_canada' will be selected for execution for en_CA
locale
        if(CommerceExtension.ExtensionInfo.getLocaleString() == 'en_CA') {
            return new CommerceExtension.Resolution('tax_extension_provider_for_canada');

        }
       // The default Salesforce Internal Tax Api will return an empty response for German
 locale
        if(CommerceExtension.ExtensionInfo.getLocaleString() == 'de') {
           return new CommerceExtension.Resolution(CommerceExtension.ResolutionStates.OFF);

        }
```

IN THIS SECTION:

ExtensionInfo Methods

ExtensionInfo Methods

The following are methods for `ExtensionInfo`.

IN THIS SECTION:

getClientApiVersion()
Returns the version number of the Client API for the extension context.

getCustomParameterField(fieldName)
Returns a custom parameter field value, if available, for the extension context.

getLocaleString()
Returns the locale for the extension context.

isCustomParametersAvailable()
Indicates whether custom parameters are available for the extension context.

getClientApiVersion()

Returns the version number of the Client API for the extension context.

Signature

```
public static Double getClientApiVersion()
```

Return Value

Type: Double

Version number of the Client API for the extension context.

getCustomParameterField(fieldName)

Returns a custom parameter field value, if available, for the extension context.

Signature

```
public static String getCustomParameterField(String fieldName)
```

Parameters

fieldName
Type: String

Custom parameter field name.

Return Value

Type: String

Custom parameter field value for the extension context.

getLocaleString()

Returns the locale for the extension context.

Signature

```
public static String getLocaleString()
```

Return Value

Type: String

Locale for the extension context.

`isCustomParametersAvailable()`

Indicates whether custom parameters are available for the extension context.

Signature

```
public static Boolean isCustomParametersAvailable()
```

Return Value

Type: Boolean

Value indicating if custom parameters are available in the extension context (`true`) or not (`false`).

Resolution Class

Resolution of a resolution strategy, which conditionally invokes default domain logic, logic provided by an extension provider, or no logic.

Namespace

CommerceExtension on page 287

Example

```
public class TaxServiceExtensionResolverSample extends commercestoretax.TaxService implements
 CommerceExtension.ResolutionStrategy {
    public CommerceExtension.Resolution resolve() {
        // The Sample Extension Provider registered with developer name as
'tax_extension_provider_for_us' will be selected for execution for en_US locale
        if(CommerceExtension.ExtensionInfo.getLocaleString() == 'en_US') {
            return new CommerceExtension.Resolution('tax_extension_provider_for_us');
        }
        // The Sample Extension Provider registered with developer name as
'tax_extension_provider_for_canada' will be selected for execution for en_CA locale
        if(CommerceExtension.ExtensionInfo.getLocaleString() == 'en_CA') {
            return new CommerceExtension.Resolution('tax_extension_provider_for_canada');

        }
       // The default Salesforce Internal Tax Api will return an empty response for German
 locale
        if(CommerceExtension.ExtensionInfo.getLocaleString() == 'de') {
           return new CommerceExtension.Resolution(CommerceExtension.ResolutionStates.OFF);

        }
        // The default Salesforce Internal Tax Api will be selected for execution for all
 other locales than US, Canada and Germany
        return new CommerceExtension.Resolution();
    }
}
```

IN THIS SECTION:

Resolution Constructors

Resolution Methods

Resolution Constructors

The following are constructors for `Resolution`.

IN THIS SECTION:

Resolution(resolutionState)

Constructor that takes a CommerceExtension.ResolutionStates object as an argument.

Resolution(providerName)

Constructor that takes the name of an extension provider as an argument.

Resolution()

Default constructor for the Resolution class.

`Resolution(resolutionState)`

Constructor that takes a CommerceExtension.ResolutionStates object as an argument.

Signature

```
public Resolution(CommerceExtension.ResolutionStates resolutionState)
```

Parameters

`resolutionState`

Type: CommerceExtension.ResolutionStates on page 295

Resolution state.

`Resolution(providerName)`

Constructor that takes the name of an extension provider as an argument.

Signature

```
public Resolution(String providerName)
```

Parameters

`providerName`

Type: String

Name of the extension provider.

`Resolution()`

Default constructor for the Resolution class.

Signature

```
public Resolution()
```

Resolution Methods

The following are methods for `Resolution`.

IN THIS SECTION:

getProviderName()
Returns the name of an extension provider.

getResolutionState()
Returns the resolution state of the resolution.

`getProviderName()`

Returns the name of an extension provider.

Signature

```
public String getProviderName()
```

Return Value

Type: String

Name of an extension provider.

`getResolutionState()`

Returns the resolution state of the resolution.

Signature

```
public CommerceExtension.ResolutionStates getResolutionState()
```

Return Value

Type: CommerceExtension.ResolutionStates on page 295

Resolution state of the resolution.

ResolutionException Class

Exception indicating a problem with the execution of a resolution strategy.

Namespace

CommerceExtension on page 287

IN THIS SECTION:

ResolutionException Constructors

ResolutionException Methods

ResolutionException Constructors

The following are constructors for `ResolutionException`.

IN THIS SECTION:

ResolutionException(errorMessage, exception)
Constructor that takes two arguments: an error message and an exception.

ResolutionException(exception)
Constructor that takes an exception as an argument,

ResolutionException(errorMessage)
Constructor that takes an error message as an argument.

ResolutionException()
Default constructor for the ResolutionException class.

`ResolutionException(errorMessage, exception)`

Constructor that takes two arguments: an error message and an exception.

Signature

```
public ResolutionException(String errorMessage, Exception exception)
```

Parameters

`errorMessage`
Type: String

Error message.

`exception`
Type: Exception

Exception.

`ResolutionException(exception)`

Constructor that takes an exception as an argument,

Signature

```
public ResolutionException(Exception exception)
```

Parameters

exception
Type: Exception

Exception.

ResolutionException(errorMessage)

Constructor that takes an error message as an argument.

Signature

```
public ResolutionException(String errorMessage)
```

Parameters

errorMessage
Type: String

Error message.

ResolutionException()

Default constructor for the ResolutionException class.

Signature

```
public ResolutionException()
```

ResolutionException Methods

The following are methods for the `ResolutionException` class.

IN THIS SECTION:

getTypeName()
Returns the type of the exception.

getTypeName()

Returns the type of the exception.

Signature

```
public String getTypeName()
```

Return Value

Type: String

The type of the Exception.

ResolutionStates Enum

Potential resolution states for a resolution strategy.

Enum Values

The following are the values of the `CommerceExtension.ResolutionStates` enum.

Value	Description
`EXECUTE_DEFAULT`	Run the default domain logic (without running extension provider logic).
`EXECUTE_REGISTERED`	Run the extension provider logic provided by the Apex class registered for the endpoint provider name.
`OFF`	Don't run any domain logic (default logic or logic provided by an extension provider).

ResolutionStrategy Interface

Interface for a resolution strategy.

Namespace

CommerceExtension on page 287

Usage

When you implement this interface, you can register your apex class just like an extension provider class. Your class can then conditionally decide how to handle each extension invocation. You can delegate to a specific extension provider, you can execute default domain logic, or you can execute no logic at all.

IN THIS SECTION:

ResolutionStrategy Methods

ResolutionStrategy Example Implementation

ResolutionStrategy Methods

The following are methods for `ResolutionStrategy`.

IN THIS SECTION:

resolve()

Returns a resolution object, which indicates how the resolution strategy was resolved. The resolution indicates whether default logic, extension provider logic, or no logic is executed.

`resolve()`

Returns a resolution object, which indicates how the resolution strategy was resolved. The resolution indicates whether default logic, extension provider logic, or no logic is executed.

Signature

```
public CommerceExtension.Resolution resolve()
```

Return Value

Type: CommerceExtension.Resolution on page 290

Resolution object that indicates how the resolution strategy was resolved.

ResolutionStrategy Example Implementation

This is an example implementation of the `CommerceExtension.ResolutionStrategy` interface.

```
// This sample is for the situation when different tax behaviors need to be
// implemented for different locales.
//
// These tax behaviors can be -
// 1. ResolutionState - EXECUTE_DEFAULT (the default Salesforce Internal Tax Api).
// 2. ResolutionState - EXECUTE_REGISTERED (extended or overridden implementations
//    via the extension point from the default Salesforce Internal Tax Api)
// 3. ResolutionState - OFF (In this case, the default Salesforce Internal Tax Api
//    will return an empty response).
//
// An Extension Provider is a custom apex class which extends or overrides the
// default Salesforce Internal Tax Api.
//
// An Extension Resolver is a custom apex class which selects different resolution
// states (EXECUTE_DEFAULT, EXECUTE_REGISTERED and OFF) for different locales
// to execute respective implementations (Extension Providers or the Default
// Salesforce Internal Tax Api).

// Your custom apex extension providers and the resolver must be registered with
// the tax extension point and then the resolver must be registered and mapped to
// the web store via appropriate setup.
//
// You can have as many Extension Providers registered as per your use case and
// select them in your resolver for different locales.
//
// Please follow the corresponding salesforce documentation on how to use locales.
// For more information related to that, please see the corresponding documentation.

// This must implement the commercestoretax.TaxService class in order to be
// processed by the tax service flow. It must also implement the
// CommerceExtension.ResolutionStrategy in order to work as a extension resolver
// and get the different locales and resolutions.
//
public class TaxServiceExtensionResolverSample
    extends commercestoretax.TaxService
```

```
    implements CommerceExtension.ResolutionStrategy {

    public CommerceExtension.Resolution resolve() {
        // The Sample Extension Provider registered with developer name as
       // 'tax_extension_provider_for_us' will be selected for execution for en_US locale

        if (CommerceExtension.ExtensionInfo.getLocaleString() == 'en_US') {
            return new CommerceExtension.Resolution('tax_extension_provider_for_us');
        }
        // The Sample Extension Provider registered with developer name as
        // 'tax_extension_provider_for_canada' will be selected for execution for en_CA
locale
        if (CommerceExtension.ExtensionInfo.getLocaleString() == 'en_CA') {
            return new CommerceExtension.Resolution('tax_extension_provider_for_canada');

        }
       // The default Salesforce Internal Tax Api will return an empty response for German
 locale
        if (CommerceExtension.ExtensionInfo.getLocaleString() == 'de') {
            return new CommerceExtension.Resolution(
                CommerceExtension.ResolutionStates.OFF
            );
        }
        // The default Salesforce Internal Tax Api will be selected for execution for
        // all other locales than US, Canada and Germany
        return new CommerceExtension.Resolution();
    }
}
```

CommerceOrders Namespace

The `CommerceOrders` namespace provides classes and methods to place orders with integrated pricing, configuration, and validation.

See CommerceOrders namespace for more information about the available classes and methods.

CommercePayments Namespace

Use the `CommercePayments` namespace to provide a safe and customizable platform for managing customer payments and refunds.

To review `CommercePayments` use cases and walkthroughs, go to Use Cases for the CommercePayments Namespace.

The following are the classes in the `CommercePayments` namespace.

IN THIS SECTION:

AbstractResponse Class

Contains the normalized response fields from payment gateways that are common to all the other gateway responses.

AbstractTransactionResponse Class

Abstract class for storing normalized information sent from payment gateways about a payment transaction. Holds the common response fields sent from payment gateways for authorization, sale, capture, and refund transactions.

AddressRequest Class

Contains address request data that is sent to a gateway adapter during a service call.

AlternativePaymentMethodRequest Class

The class contains information about the alternative payment method that are required for a gateway to process the request.

AlternativePaymentMethodResponse Class

The class contains the response details of the alternative payment method.

AuditParamsRequest

`AuditParamsRequest` is used for audit parameters in a transaction request. This is an abstract request class that is extended by the `BaseRequest` class.

AuthApiPaymentMethodRequest Class

Sends information about a payment method to a gateway adapter during an authorization service call.

AuthorizationRequest Class

Sends information about an authorization request to a gateway adapter during a service call. This class extends the `BaseRequest` class and inherits all its methods.

AuthorizationResponse Class

Response sent by the payment gateway adapter for an authorization service.

AuthorizationReversalRequest Class

Sends information about an authorization reversal request to a gateway adapter during a service call.

AuthorizationReversalResponse Class

Response sent by the payment gateway following a payment authorization reversal service.

BaseApiPaymentMethodRequest Class

Abstract class used to send information about a payment method to a gateway adapter during a service call.

BaseNotification Class

Abstract class for storing notification information sent from payment gateways.

BasePaymentMethodRequest Class

Abstract class for storing information about payment methods.

BaseRequest Class

`BaseRequest` is extended by all the request classes.

CaptureNotification Class

When a payment gateway sends a notification for a capture transaction, the payment gateway adapter creates the `CaptureNotification` object to store information about the notification.

CaptureRequest Class

Represents a capture request. This class extends the `BaseRequest` class and inherits all its methods.

CaptureResponse Class

The payment gateway adapter sends this response for the capture request type. This class extends `AbstractResponse` and inherits its methods.

CardCategory Enum

Defines whether the payment method represents a credit card or a debit card.

CardPaymentMethodRequest Class

Sends data related to a card payment method to a gateway adapter during a service call.

CardPaymentMethodResponse Class

This class contains details about the card payment method.

CardType Enum

Specifies the credit card issuer.

CustomMetadataTypeInfo Class

Access information about custom metadata. The `PaymentGatewayAdapter` can send `CustomMetadataTypeInfo` to transaction requests through the response object's `SalesforceResultCodeInfo`.

GatewayErrorResponse Class

Use to respond with an error indication following errors from the `PaymentGateway` adapter, such as request-forbidden responses, custom validation errors, or expired API tokens.

GatewayNotificationResponse Class

When the payment gateway sends a notification to the payments platform, the platform responds with a `GatewayNotificationResponse` indicating whether the platform succeeded or failed at receiving the notification.

GatewayResponse Interface

Generic payment gateway response interface. This class extends the `CaptureResponse` on page 361, `AbstractTransactionResponse` on page 304, and `AbstractResponse` on page 300 classes and inherits all their properties. It has no unique methods or parameters.

NotificationClient Class

Communicates with the payment platform regarding the gateway's notification.

NotificationSaveResult Class

Contains the result of the payment platform's attempt to record data from the gateway's notification.

NotificationStatus Enum

Shows whether the payments platform successfully received the notification from the gateway.

PaymentGatewayAdapter Interface

`PaymentGatewayAdapters` can implement this interface in order to process requests.

PaymentGatewayAsyncAdapter Interface

Implement the interface to allow customers to process payments asynchronously.

PaymentGatewayContext Class

Wraps the information related to a payment request.

PaymentGatewayNotificationContext Class

Wraps the information related to a gateway notification.

PaymentGatewayNotificationRequest Class

Contains the notification request data from the gateway.

PaymentMethodDetailsResponse Class

This class contains the details about the payment method.

PaymentMethodTokenizationRequest Class

Stores data about a request to tokenize a card payment method. The tokenization process occurs in the payment gateway. This process replaces sensitive customer data, such as a card number or CVV, with unique identification symbols. The symbols are used while the data is handled by Salesforce, the payment gateway, and the customer bank, allowing Salesforce to store the token without storing sensitive customer data.

PaymentMethodTokenizationResponse Class

Gateway response sent by payment gateway adapters for the payment method tokenization request. The response includes the payment method's token ID value.

PaymentsHttp Class

Makes an HTTP request to start the interaction with the payment gateway.

PostAuthApiPaymentMethodRequest Class

Sends information about a payment method to a gateway adapter during a postauthorization service call.

PostAuthorizationRequest Class

Sends information about a postauthorization request to a gateway adapter during a service call.

PostAuthorizationResponse Class

Response sent by the payment gateway adapter for a postauthorization service.

ReferencedRefundNotification Class

When a payment gateway sends a notification for a refund transaction, the payment gateway adapter creates the `ReferencedRefundNotification` object to store information about notification.

ReferencedRefundRequest

Access information about the referenced refund requests. Extends the `RefundRequest` class.

ReferencedRefundResponse Class

The payment gateway adapter sends this response for the `ReferencedRefund` request type.

RefundRequest Class

Sends data related to a refund to the payment gateway adapter.

RequestType Enum

Defines the type of payment transaction request made to the payment gateway.

SaleApiPaymentMethodRequest Class

Sends data related to a card payment method to a gateway adapter during a sale service call.

SaleRequest Class

Stores information about a sales request.

SaleResponse Class

Response sent by payment gateway adapters for a sales service.

SalesforceResultCode Enum

Defines the gateway call status values in Salesforce based on the call status values that the payment gateway returned.

SalesforceResultCodeInfo

Stores Salesforce result code information from payment gateway adapters.

AbstractResponse Class

Contains the normalized response fields from payment gateways that are common to all the other gateway responses.

Namespace

CommercePayments

Usage

You must specify the `CommercePayments` namespace when creating an instance of this class. The constructor of this class takes no arguments. For example:

```
CommercePayments.AbstractResponse abr = new CommercePayments.AbstractResponse();
```

This class can't be instantiated on its own. This class implements the GatewayResponse class. Other GatewayResponse classes extend this class to inherit common properties.

IN THIS SECTION:

AbstractResponse Methods

AbstractResponse Methods

The following are methods for `AbstractResponse`.

IN THIS SECTION:

setGatewayAvsCode(gatewayAvsCode)
Sets the AVS (address verification system) result code information that the gateway returned. Maximum length of 64 characters.

setGatewayDate(gatewayDate)
Sets the date that the transaction occurred. Some gateways don't send this value.

setGatewayMessage(gatewayMessage)
Sets error messages that the gateway returned for the payment request. Maximum length of 255 characters.

setGatewayResultCode(gatewayResultCode)
Sets a gateway-specific result code. The code may be mapped to a Salesforce-specific result code. Maximum length of 64 characters.

setGatewayResultCodeDescription(gatewayResultCodeDescription)
Sets a description of the gateway-specific result code that a payment gateway returned. Maximum length of 1000 characters.

setSalesforceResultCodeInfo(salesforceResultCodeInfo)
Sets the Salesforce-specific result code information. Payment gateways have many response codes for payment calls. Salesforce uses the result code information to map payment gateway codes to a predefined set of standard Salesforce result codes.

`setGatewayAvsCode(gatewayAvsCode)`

Sets the AVS (address verification system) result code information that the gateway returned. Maximum length of 64 characters.

Signature

```
global void setGatewayAvsCode(String gatewayAvsCode)
```

Parameters

`gatewayAvsCode`
Type: String

Code sent by gateways that use an address verification system.

Return Value

Type: void

`setGatewayDate(gatewayDate)`

Sets the date that the transaction occurred. Some gateways don't send this value.

Signature

```
global void setGatewayDate(Datetime gatewayDate)
```

Parameters

gatewayDate
Type: Datetime

Date and time of the gateway communication.

Return Value

Type: void

`setGatewayMessage(gatewayMessage)`

Sets error messages that the gateway returned for the payment request. Maximum length of 255 characters.

Signature

```
global void setGatewayMessage(String gatewayMessage)
```

Parameters

gatewayMessage
Type: String

Information on error messages sent from the gateway.

Return Value

Type: void

`setGatewayResultCode(gatewayResultCode)`

Sets a gateway-specific result code. The code may be mapped to a Salesforce-specific result code. Maximum length of 64 characters.

Signature

```
global void setGatewayResultCode(String gatewayResultCode)
```

Parameters

gatewayResultCode
Type: String

Gateway-specific result code. Must be used to map a Salesforce-specific result code.

Return Value

Type: void

setGatewayResultCodeDescription(gatewayResultCodeDescription)

Sets a description of the gateway-specific result code that a payment gateway returned. Maximum length of 1000 characters.

Signature

```
global void setGatewayResultCodeDescription(String gatewayResultCodeDescription)
```

Parameters

gatewayResultCodeDescription
Type: String

Description of the gateway's result code. Use this field to learn more about why the gateway returned a certain result code.

Return Value

Type: void

setSalesforceResultCodeInfo(salesforceResultCodeInfo)

Sets the Salesforce-specific result code information. Payment gateways have many response codes for payment calls. Salesforce uses the result code information to map payment gateway codes to a predefined set of standard Salesforce result codes.

Signature

```
global void setSalesforceResultCodeInfo(commercepayments.SalesforceResultCodeInfo
salesforceResultCodeInfo)
```

Parameters

salesforceResultCodeInfo
Type: commercepayments.SalesforceResultCodeInfo on page 446

Description of the Salesforce result code value.

Return Value

Type: void

AbstractTransactionResponse Class

Abstract class for storing normalized information sent from payment gateways about a payment transaction. Holds the common response fields sent from payment gateways for authorization, sale, capture, and refund transactions.

Namespace

CommercePayments

Usage

Specify the `CommercePayments` namespace when creating an instance of this class. The constructor of this class takes no arguments. For example:

```
CommercePayments.AbstractTransactionResponse atr = new
CommercePayments.AbstractTransactionResponse();
```

IN THIS SECTION:

AbstractTransactionResponse Methods

AbstractTransactionResponse Methods

The following are methods for `AbstractTransactionResponse`.

IN THIS SECTION:

setAmount(amount)
Sets the transaction amount. Must be a non-negative value.

setGatewayAvsCode(gatewayAvsCode)
Sets the AVS (address verification system) result code that the gateway returned. Maximum length of 64 characters.

setGatewayDate(gatewayDate)
Sets the date that the notification occurred. Some gateways don't send this value.

setGatewayMessage(gatewayMessage)
Sets error messages that the gateway returned for the notification request. Maximum length of 255 characters.

setGatewayReferenceDetails(gatewayReferenceDetails)
Sets the payment gateway's reference details.

setGatewayReferenceNumber(gatewayReferenceNumber)
Sets the payment gateway's reference number.

setGatewayResultCode(gatewayResultCode)
Sets a gateway-specific result code. You can map the result code to a Salesforce-specific result code. Maximum length of 64 characters.

setGatewayResultCodeDescription(gatewayResultCodeDescription)
Sets a description of the gateway-specific result code that a payment gateway returned. Maximum length of 1000 characters.

setSalesforceResultCodeInfo(salesforceResultCodeInfo)
Sets the Salesforce-specific result code information.

`setAmount(amount)`

Sets the transaction amount. Must be a non-negative value.

Signature

```
global void setAmount(Double amount)
```

Parameters

amount
Type: Double

The amount of the transaction.

Return Value

Type: void

`setGatewayAvsCode(gatewayAvsCode)`

Sets the AVS (address verification system) result code that the gateway returned. Maximum length of 64 characters.

Signature

```
global void setGatewayAvsCode(String gatewayAvsCode)
```

Parameters

gatewayAvsCode
Type: String

Used to verify the address mapped to a payment method when the payments platform requests tokenization from the payment gateway.

Return Value

Type: void

`setGatewayDate(gatewayDate)`

Sets the date that the notification occurred. Some gateways don't send this value.

Signature

```
global void setGatewayDate(Datetime gatewayDate)
```

Parameters

gatewayDate
Type: Datetime

The date that the transaction occurred.

Return Value

Type: void

`setGatewayMessage(gatewayMessage)`

Sets error messages that the gateway returned for the notification request. Maximum length of 255 characters.

Signature

```
global void setGatewayMessage(String gatewayMessage)
```

Parameters

`gatewayMessage`
Type: String

The message that the gateway returned with the transaction request. Contains additional information about the transaction.

Return Value

Type: void

`setGatewayReferenceDetails(gatewayReferenceDetails)`

Sets the payment gateway's reference details.

Signature

```
global void setGatewayReferenceDetails(String gatewayReferenceDetails)
```

Parameters

`gatewayReferenceDetails`
Type: String

Provides information about the gateway communication.

Return Value

Type: void

`setGatewayReferenceNumber(gatewayReferenceNumber)`

Sets the payment gateway's reference number.

Signature

```
global void setGatewayReferenceNumber(String gatewayReferenceNumber)
```

Parameters

gatewayReferenceNumber
Type: String

Unique transaction ID created by the payment gateway.

Return Value

Type: void

setGatewayResultCode(gatewayResultCode)

Sets a gateway-specific result code. You can map the result code to a Salesforce-specific result code. Maximum length of 64 characters.

Signature

```
global void setGatewayResultCode(String gatewayResultCode)
```

Parameters

gatewayResultCode
Type: String

Gateway-specific result code. Must be mapped to a Salesforce-specific result code.

Return Value

Type: void

setGatewayResultCodeDescription(gatewayResultCodeDescription)

Sets a description of the gateway-specific result code that a payment gateway returned. Maximum length of 1000 characters.

Signature

```
global void setGatewayResultCodeDescription(String gatewayResultCodeDescription)
```

Parameters

gatewayResultCodeDescription
Type: String

Provides additional information about the result code and why the gateway returned the specific code. Descriptions vary between different gateways.

Return Value

Type: void

setSalesforceResultCodeInfo(salesforceResultCodeInfo)

Sets the Salesforce-specific result code information.

Signature

```
global void setSalesforceResultCodeInfo(commercepayments.SalesforceResultCodeInfo
salesforceResultCodeInfo)
```

Parameters

salesforceResultCodeInfo

Type: commercepayments.SalesforceResultCodeInfo on page 446

Payment gateways have many response codes for payment calls. Salesforce uses the result code information to map payment gateway codes to a predefined set of standard Salesforce result codes.

Return Value

Type: void

AddressRequest Class

Contains address request data that is sent to a gateway adapter during a service call.

Namespace

CommercePayments

Usage

Contains information about the payment method's address. Use this information in authorization, sale, and tokenization requests. The payment gateway adapter uses information in an AddressRequest object to construct a JSON request to send to the payment gateway.

The constructor of this class takes no arguments. For example:

```
CommercePayments.AddressRequest adr = new CommercePayments.AddressRequest();
```

IN THIS SECTION:

AddressRequest Constructors

AddressRequest Properties

AddressRequest Methods

AddressRequest Constructors

The following are constructors for `AddressRequest`.

IN THIS SECTION:

AddressRequest(street, city, state, country, postalCode)

Constructs a sample address. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

`AddressRequest(street, city, state, country, postalCode)`

Constructs a sample address. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Signature

```
global AddressRequest(String street, String city, String state, String country, String postalCode)
```

Parameters

street
Type: String

Street for the payment method's address.

city
Type: String

City for the payment method's address.

state
Type: String

State for the payment method's address.

country
Type: String

Country for the payment method's address.

postalCode
Type: String

Postal code for the payment method's address.

AddressRequest Properties

The following are properties for `AddressRequest`.

IN THIS SECTION:

city
City of the payment method address.

companyName
Company name of the payment method address.

country
Country for the payment method address.

postalCode
Postal code for the payment method address.

state
State for the payment method address.

street
Street for the payment method address.

city

City of the payment method address.

Signature

```
global String city {get; set;}
```

Property Value

Type: String

companyName

Company name of the payment method address.

Signature

```
global String companyName {get; set;}
```

Property Value

Type: String

country

Country for the payment method address.

Signature

```
global String country {get; set;}
```

Property Value

Type: String

postalCode

Postal code for the payment method address.

Signature

```
global String postalCode {get; set;}
```

Property Value

Type: String

state

State for the payment method address.

Signature

```
global String state {get; set;}
```

Property Value

Type: String

`street`

Street for the payment method address.

Signature

```
global String street {get; set;}
```

Property Value

Type: String

AddressRequest Methods

The following are methods for `AddressRequest`.

IN THIS SECTION:

equals(obj)
Maintains the integrity of lists of type `AddressRequest` by determining the equality of external objects in a list. This method is dynamic and is based on the equals method in Java.

hashCode()
Maintains the integrity of lists of type `AddressRequest`.

toString()
Converts a date to a string.

`equals(obj)`

Maintains the integrity of lists of type `AddressRequest` by determining the equality of external objects in a list. This method is dynamic and is based on the equals method in Java.

Signature

```
global Boolean equals(Object obj)
```

Parameters

obj

Type: Object

External object whose key is to be validated.

Return Value

Type: Boolean

`hashCode()`

Maintains the integrity of lists of type `AddressRequest`.

Signature

```
global Integer hashCode()
```

Return Value

Type: Integer

`toString()`

Converts a date to a string.

Signature

```
global String toString()
```

Return Value

Type: String

AlternativePaymentMethodRequest Class

The class contains information about the alternative payment method that are required for a gateway to process the request.

Namespace

CommercePayments

Example

```
commercepayments.PostAuthApiPaymentMethodRequest apiPaymentMethod
=(commercepayments.PostAuthApiPaymentMethodRequest) postAuthRequest.paymentMethod;
commercepayments.AlternativePaymentMethodRequest alternativePaymentMethod=
(commercepayments.AlternativePaymentMethodRequest) apiPaymentMethod.alternativePaymentMethod;
String gatewayToken = (String)alternativePaymentMethod.gatewayToken;
String gatewayTokenDetails = (String)alternativePaymentMethod.gatewayTokenDetails;
String name = (String)alternativePaymentMethod.name;
String accountId = (String)alternativePaymentMethod.accountId;
String email = (String)alternativePaymentMethod.email;
```

IN THIS SECTION:

AlternativePaymentMethodRequest Constructors

AlternativePaymentMethodRequest Properties

AlternativePaymentMethodRequest Methods

AlternativePaymentMethodRequest Constructors

The following are constructors for `AlternativePaymentMethodRequest`.

IN THIS SECTION:

AlternativePaymentMethodRequest(gatewayToken)
Creates a new instance of the `CommercePayments.AlternativePaymentMethodRequest` class.

`AlternativePaymentMethodRequest(gatewayToken)`

Creates a new instance of the `CommercePayments.AlternativePaymentMethodRequest` class.

Signature

```
public AlternativePaymentMethodRequest(String gatewayToken)
```

Parameters

`gatewayToken`
Type: String

A unique, alphanumeric ID, called a token, that a payment gateway generates when it first processes a payment. The token replaces the actual payment data so that the data is kept secure. This token is stored as encrypted text, and can be used for recurring payments.

AlternativePaymentMethodRequest Properties

The following are properties for `AlternativePaymentMethodRequest`.

IN THIS SECTION:

accountId
Salesforce account ID to which this payment method is linked.

email
Email address of the card holder.

gatewayToken
A unique, alphanumeric ID, that a payment gateway generates when it first processes a payment.

gatewayTokenDetails
Information about the gateway token.

name
Name that you assign to the PaymentMethod object.

accountId

Salesforce account ID to which this payment method is linked.

Signature

```
public String accountId {get; set;}
```

Property Value

Type: String

email

Email address of the card holder.

Signature

```
public String email {get; set;}
```

Property Value

Type: String

gatewayToken

A unique, alphanumeric ID, that a payment gateway generates when it first processes a payment.

The token replaces the actual payment data so that the data is kept secure. This token is stored as encrypted text, and can be used for recurring payments.

Signature

```
public String gatewayToken {get; set;}
```

Property Value

Type: String

gatewayTokenDetails

Information about the gateway token.

Signature

```
public String gatewayTokenDetails {get; set;}
```

Property Value

Type: String

name

Name that you assign to the PaymentMethod object.

Signature

```
public String name {get; set;}
```

Property Value

Type: String

AlternativePaymentMethodRequest Methods

The following are methods for `AlternativePaymentMethodRequest`.

IN THIS SECTION:

equals(obj)
Maintains the integrity of lists of type `AlternativePaymentMethodRequest` by determining the equality of external objects in a list. This method is dynamic and based on the equals method in Java.

hashCode()
Maintains the integrity of lists of type `AlternativePaymentMethodRequest` by determining the uniqueness of the external object records in a list.

toString()
Converts a date to a string.

equals(obj)

Maintains the integrity of lists of type `AlternativePaymentMethodRequest` by determining the equality of external objects in a list. This method is dynamic and based on the equals method in Java.

Signature

```
public Boolean equals(Object obj)
```

Parameters

obj
Type: Object

External object whose key is to be validated.

Return Value

Type: Boolean

`hashCode()`

Maintains the integrity of lists of type `AlternativePaymentMethodRequest` by determining the uniqueness of the external object records in a list.

Signature

```
public Integer hashCode()
```

Return Value

Type: Integer

`toString()`

Converts a date to a string.

Signature

```
public String toString()
```

Return Value

Type: String

AlternativePaymentMethodResponse Class

The class contains the response details of the alternative payment method.

Namespace

CommercePayments

Example

```
commercepayments.AlternativePaymentMethodResponse response = new
commercepayments.AlternativePaymentMethodResponse();
response.setEmail('alternativePaymentMethod');
response.setEmail('foo@foo.com');
response.setGatewayToken('NMoPoIOnTZSaRaWcV7gUUXe');
response.setGatewayTokenDetails('gateway token details');
```

IN THIS SECTION:

AlternativePaymentMethodResponse Methods

AlternativePaymentMethodResponse Methods

The following are methods for `AlternativePaymentMethodResponse`.

IN THIS SECTION:

setAccountId(accountId)
Sets the ID of the Salesforce payments account to which the payment method is linked.

setComments(comments)
Sets the notes about the payment method added by users.

setEmail(email)
Sets the email ID of the card holder.

setGatewayToken(gatewayToken)
Sets the token ID that a payment gateway generates when it first processes a payment.

setGatewayTokenDetails(gatewayTokenDetails)
Sets the details about the payment gateway token.

setName(name)
Sets the name that is assigned to the PaymentMethod object.

`setAccountId(accountId)`

Sets the ID of the Salesforce payments account to which the payment method is linked.

Signature

```
public void setAccountId(Id accountId)
```

Parameters

`accountId`
Type: Id

Salesforce payments account ID.

Return Value

Type: void

`setComments(comments)`

Sets the notes about the payment method added by users.

Signature

```
public void setComments(String comments)
```

Parameters

`comments`
Type: String

Notes about the payment method added by users, maximum 1000 characters.

Return Value

Type: void

`setEmail(email)`

Sets the email ID of the card holder.

Signature

```
public void setEmail(String email)
```

Parameters

email
Type: String

Email ID of the card holder.

Return Value

Type: void

`setGatewayToken(gatewayToken)`

Sets the token ID that a payment gateway generates when it first processes a payment.

Signature

```
public void setGatewayToken(String gatewayToken)
```

Parameters

gatewayToken
Type: String

A unique, alphanumeric ID, called a token, that a payment gateway generates when it first processes a payment. The token replaces the actual payment data so that the data is kept secure. This token is stored as encrypted text, and can be used for recurring payments.

Return Value

Type: void

`setGatewayTokenDetails(gatewayTokenDetails)`

Sets the details about the payment gateway token.

Signature

```
public void setGatewayTokenDetails(String gatewayTokenDetails)
```

Parameters

gatewayTokenDetails
Type: String

Detailed information about the gateway token.

Return Value

Type: void

setName(name)

Sets the name that is assigned to the PaymentMethod object.

Signature

```
public void setName(String name)
```

Parameters

name
Type: String

Name that you assign to the payment method object.

Return Value

Type: void

AuditParamsRequest

`AuditParamsRequest` is used for audit parameters in a transaction request. This is an abstract request class that is extended by the `BaseRequest` class.

Namespace

CommercePayments

Usage

`AuditParamsRequest` is an abstract class that holds attributes related to audit parameters such as email, IP address, MAC address, and phone number. This class can't be instantiated on its own. All `CommercePayments` request classes extend this class.

IN THIS SECTION:

AuditParamsRequest Constructors

AuditParamsRequest Properties

AuditParamsRequest Constructors

The following are constructors for `AuditParamsRequest`.

IN THIS SECTION:

AuditParamsRequest(email, macAddress, ipAddress, phone)
This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

`AuditParamsRequest(email, macAddress, ipAddress, phone)`

This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Signature

```
AuditParamsRequest(String email, String macAddress, String ipAddress, String phone)
```

Parameters

email
Type: String

Email of the client that initiated the request.

macAddress
Type: String

Mac address of the customer's device. Gateways often use this data in risk checks.

ipAddress
Type: String

The customer's IP address. Gateways often use this data in risk checks.

phone
Type: String

Phone number of the client that initiated the request.

AuditParamsRequest Properties

The following are properties for `AuditParamsRequest`.

IN THIS SECTION:

email
Email of the client that initiated the request.

ipAddress
The customer's IP address. Gateways often use this data in risk checks.

macAddress
Mac address of the customer's device. Gateways often use this data in risk checks.

phone
Phone number of the client that initiated the request.

email

Email of the client that initiated the request.

Property Value

Type: String

ipAddress

The customer's IP address. Gateways often use this data in risk checks.

Property Value

Type: String

macAddress

Mac address of the customer's device. Gateways often use this data in risk checks.

Property Value

Type: String

phone

Phone number of the client that initiated the request.

Property Value

Type: String

AuthApiPaymentMethodRequest Class

Sends information about a payment method to a gateway adapter during an authorization service call.

Namespace

CommercePayments

Usage

Contains information about the payment method that is used for an authorization request. It contains all available payment methods as fields, but populates only one field for each request. The gateway adapter uses this class when constructing an authorization request. An object of this class is available through the `paymentMethod` field on the `AuthorizationRequest Class` object.

IN THIS SECTION:

AuthApiPaymentMethodRequest Constructors

AuthApiPaymentMethodRequest Properties

AuthApiPaymentMethodRequest Constructors

The following are constructors for `AuthApiPaymentMethodRequest`.

IN THIS SECTION:

AuthApiPaymentMethodRequest(cardPaymentMethodRequest)
Constructs a sample `cardPaymentMethodRequest`. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

AuthApiPaymentMethodRequest()
Constructor for `AuthApiPaymentMethodRequest`.

`AuthApiPaymentMethodRequest(cardPaymentMethodRequest)`

Constructs a sample `cardPaymentMethodRequest`. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Signature

```
global AuthApiPaymentMethodRequest(commercepayments.CardPaymentMethodRequest
cardPaymentMethodRequest)
```

Parameters

`cardPaymentMethodRequest`
Type: commercepayments.CardPaymentMethodRequest on page 366

Contains information about the card payment method. Used to send information to a gateway adapter during a service call.

`AuthApiPaymentMethodRequest()`

Constructor for `AuthApiPaymentMethodRequest`.

Signature

```
global AuthApiPaymentMethodRequest()
```

AuthApiPaymentMethodRequest Properties

The following are properties for `AuthApiPaymentMethodRequest`.

IN THIS SECTION:

cardPaymentMethod
The card payment method object used in a payment method request.

`cardPaymentMethod`

The card payment method object used in a payment method request.

Signature

```
global commercepayments.CardPaymentMethodRequest cardPaymentMethod {get; set;}
```

Property Value

Type: commercepayments.CardPaymentMethodRequest on page 366

AuthorizationRequest Class

Sends information about an authorization request to a gateway adapter during a service call. This class extends the `BaseRequest` class and inherits all its methods.

Namespace

CommercePayments

Usage

This class contains information about a transaction authorization request. The gateway adapter reads fields from this class while constructing an authorization JSON request to send to the payment gateway. An object of this class is available by calling `getPaymentRequest()` in the `PaymentGatewayContext Class`.

Example

Creating a `buildAuthRequest` class to store information about the authorization request.

```
private String buildAuthRequest(commercepayments.AuthorizationRequest authRequest) {
        // Multiply amount by 100.0 to convert to cents
        String requestBody =
createRequestBody(String.ValueOf((authRequest.amount*100.0).intValue()), authRequest);
        return requestBody;

       private String createRequestBody(String amount, commercepayments.AuthorizationRequest
 authRequest) {
        JSONGenerator jsonGeneratorInstance = JSON.createGenerator(true);
        String currencyIso = authRequest.currencyIsoCode;
        commercepayments.AuthApiPaymentMethodRequest paymentMethod =
authRequest.paymentMethod;
        commercepayments.GatewayErrorResponse error;
        // Write data to the JSON string.
        jsonGeneratorInstance.writeStartObject();
       jsonGeneratorInstance.writeStringField('merchantAccount', '{!$Credential.Username}');

        jsonGeneratorInstance.writeStringField('reference', authRequest.comments == null
? 'randomstring' : authRequest.comments);

        if(currencyIso == null) {
            currencyIso = UserInfo.getDefaultCurrency();
        }

        jsonGeneratorInstance.writeFieldName('amount');
```

```
        jsonGeneratorInstance.writeStartObject();
        jsonGeneratorInstance.writeStringField('value', amount);
        jsonGeneratorInstance.writeStringField('currency', currencyIso);
        jsonGeneratorInstance.writeEndObject();

        commercepayments.CardPaymentMethodRequest cardPaymentMethod;
        if(paymentMethod != null) {
            cardPaymentMethod = paymentMethod.cardPaymentMethod;
            if (cardPaymentMethod != null) {
                if (cardPaymentMethod.CardCategory != null) {
                    if (commercepayments.CardCategory.CreditCard ==
cardPaymentMethod.CardCategory) {
                        jsonGeneratorInstance.writeFieldName('card');
                        jsonGeneratorInstance.writeStartObject();
                        if (cardPaymentMethod.cvv != null)
                            jsonGeneratorInstance.writeStringField('cvc',
String.ValueOf(cardPaymentMethod.cvv));
                        if (cardPaymentMethod.cardholdername != null)
                            jsonGeneratorInstance.writeStringField('holderName',
cardPaymentMethod.cardholdername);
                        if (cardPaymentMethod.cardnumber != null)
                            jsonGeneratorInstance.writeStringField('number',
cardPaymentMethod.cardnumber);
                        if (cardPaymentMethod.expiryMonth != null &&
cardPaymentMethod.expiryYear != null ) {
                            String expMonth =
((String.ValueOf(cardPaymentMethod.expiryMonth)).length() == 1 ? '0' : '') +
String.ValueOf(cardPaymentMethod.expiryMonth);
                         jsonGeneratorInstance.writeStringField('expiryMonth', expMonth);

                            jsonGeneratorInstance.writeStringField('expiryYear',
String.ValueOf(cardPaymentMethod.expiryYear));
                        }
                        jsonGeneratorInstance.writeEndObject();
                    } else {
                    //Support for other card type
                    }
                } else {
                    throw new SampleValidationException('Required Field Missing :
CardCategory');
                }
            } else {
                throw new SampleValidationException('Required Field Missing :
CardPaymentMethod');
            }
        } else {
           throw new SampleValidationException('Required Field Missing : PaymentMethod');

        }
        jsonGeneratorInstance.writeEndObject();
        return jsonGeneratorInstance.getAsString();
    }
```

IN THIS SECTION:

AuthorizationRequest Constructors

AuthorizationRequest Properties

AuthorizationRequest Methods

AuthorizationRequest Constructors

The following are constructors for `AuthorizationRequest`.

IN THIS SECTION:

AuthorizationRequest(amount)
Constructor for building the amount in an authorization request. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

`AuthorizationRequest(amount)`

Constructor for building the amount in an authorization request. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Signature

```
global AuthorizationRequest(Double amount)
```

Parameters

amount
Type: Double

The amount of the authorization.

AuthorizationRequest Properties

The following are properties for `AuthorizationRequest`.

IN THIS SECTION:

accountId
The customer account where the authorization is performed.

amount
The total amount of the authorization. Can be positive or negative.

comments
Comments about the authorization. Users can enter comments to provide additional information.

currencyIsoCode
The ISO currency code for the authorization request.

paymentMethod
The payment method used to process the authorization in the authorization request.

accountId

The customer account where the authorization is performed.

Signature

```
global String accountId {get; set;}
```

Property Value

Type: String

amount

The total amount of the authorization. Can be positive or negative.

Signature

```
global Double amount {get; set;}
```

Property Value

Type: Double

comments

Comments about the authorization. Users can enter comments to provide additional information.

Signature

```
global String comments {get; set;}
```

Property Value

Type: String

currencyIsoCode

The ISO currency code for the authorization request.

Signature

```
global String currencyIsoCode {get; set;}
```

Property Value

Type: String

paymentMethod

The payment method used to process the authorization in the authorization request.

Signature

```
global AuthApiPaymentMethodRequest paymentMethod {get; set;}
```

Property Value

Type: AuthApiPaymentMethodRequest on page 321

AuthorizationRequest Methods

The following are methods for `AuthorizationRequest`.

IN THIS SECTION:

equals(obj)

Maintains the integrity of lists of type `AuthorizationRequest` by determining the equality of external objects in a list. This method is dynamic and based on the equals method in Java.

hashCode()

Maintains the integrity of lists of type `AuthorizationRequest` by determining the uniqueness of the external object in a list.

toString()

Converts a date to a string.

`equals(obj)`

Maintains the integrity of lists of type `AuthorizationRequest` by determining the equality of external objects in a list. This method is dynamic and based on the equals method in Java.

Signature

```
global Boolean equals(Object obj)
```

Parameters

obj

Type: Object

External object whose key is to be validated.

Return Value

Type: Boolean

`hashCode()`

Maintains the integrity of lists of type `AuthorizationRequest` by determining the uniqueness of the external object in a list.

Signature

```
global Integer hashCode()
```

Return Value

Type: Integer

`toString()`

Converts a date to a string.

Signature

```
global String toString()
```

Return Value

Type: String

AuthorizationResponse Class

Response sent by the payment gateway adapter for an authorization service.

Namespace

CommercePayments

Usage

The constructor of this class takes no arguments. For example:

```
CommercePayments.AuthorizationResponse authr = new
CommercePayments.AuthorizationResponse();
```

Contains information about the payment gateway's response following an authorization transaction. The gateway adapter uses the payment gateway's response to populate the `AuthorizationResponse` fields. The payments platform uses the information from this class to construct the authorization gateway response shown to the user.

Example

```
private commercepayments.GatewayResponse createAuthResponse(HttpResponse response, Double
 amount) {
        Map<String, Object> mapOfResponseValues = (Map
                            <String, Object>) JSON.deserializeUntyped(response.getBody());


        commercepayments.AuthorizationResponse authResponse = new
commercepayments.AuthorizationResponse();

        String resultCode = (String)mapOfResponseValues.get('resultCode');

        if(resultCode != null){
            system.debug('Response - success');
            if(resultCode.equals('Authorised')){
                system.debug('status - authorised');
```

```
            authResponse.setGatewayAuthCode((String)mapOfResponseValues.get('authCode'));

                authResponse.setSalesforceResultCodeInfo(new
commercepayments.SalesforceResultCodeInfo(commercepayments.SalesforceResultCode.Success));

            } else {
                //Sample returns 200 with refused status in some cases
                system.debug('status - refused');

authResponse.setGatewayResultCodeDescription((String)mapOfResponseValues.get('refusalReason'));

                authResponse.setSalesforceResultCodeInfo(new
commercepayments.SalesforceResultCodeInfo(commercepayments.SalesforceResultCode.Decline));

            }

authResponse.setGatewayReferenceNumber((String)mapOfResponseValues.get('pspReference'));
            authResponse.setAmount(amount);
            authResponse.setGatewayDate(system.now());
            return authResponse;
        } else {
            system.debug('Response - failed');
            system.debug('Validation error');
            String statusCode = (String)mapOfResponseValues.get('errorType');
            String message = (String)mapOfResponseValues.get('message');
            commercepayments.GatewayErrorResponse error = new
commercepayments.GatewayErrorResponse(statusCode, message);
            return error;
        }
    }
```

IN THIS SECTION:

AuthorizationResponse Methods

AuthorizationResponse Methods

The following are methods for `AuthorizationResponse`.

IN THIS SECTION:

setAmount(amount)

Sets the amount of the authorization. Must be a non-zero value.

setAuthorizationExpirationDate(authExpDate)

Sets the expiration date of the authorization request.

setGatewayAuthCode(gatewayAuthCode)

Sets the authorization code that the gateway returned. Maximum length of 64 characters.

setGatewayAvsCode(gatewayAvsCode)

Sets the AVS (address verification system) result code information that the gateway returned. Maximum length of 64 characters.

setGatewayDate(gatewayDate)

Sets the date that the authorization occurred. Some gateways don't send this value.

setGatewayMessage(gatewayMessage)

Sets error messages that the gateway returned for the authorization request. Maximum length of 255 characters.

setGatewayReferenceDetails(gatewayReferenceDetails)

Stores data that you can use for subsequent authorizations. You can use any data that isn't normalized in financial entities. This field has a maximum length of 1000 characters and can store data as JSON or XML.

setGatewayReferenceNumber(gatewayReferenceNumber)

Sets the unique gateway reference number for the transaction that the gateway returned. Maximum length of 255 characters.

setGatewayResultCode(gatewayResultCode)

Sets a gateway-specific result code. The code can be mapped to a Salesforce-specific result code. Maximum length of 64 characters.

setGatewayResultCodeDescription(gatewayResultCodeDescription)

Sets a description of the gateway-specific result code that a payment gateway returned. Maximum length of 1000 characters.

setPaymentMethodTokenizationResponse(paymentMethodTokenizationResponse)

Sets information from the gateway about the tokenized payment method.

setSalesforceResultCodeInfo(salesforceResultCodeInfo)

Sets the Salesforce-specific result code information. Payment gateways have many response codes for payment calls. Salesforce uses the result code information to map payment gateway codes to a predefined set of standard Salesforce result codes.

`setAmount(amount)`

Sets the amount of the authorization. Must be a non-zero value.

Signature

```
global void setAmount(Double amount)
```

Parameters

amount

Type: Double

Return Value

Type: void

`setAuthorizationExpirationDate(authExpDate)`

Sets the expiration date of the authorization request.

Signature

```
global void setAuthorizationExpirationDate(Datetime authExpDate)
```

Parameters

authExpDate
Type: Datetime

Return Value

Type: void

setGatewayAuthCode(gatewayAuthCode)

Sets the authorization code that the gateway returned. Maximum length of 64 characters.

Signature

```
global void setGatewayAuthCode(String gatewayAuthCode)
```

Parameters

gatewayAuthCode
Type: String

The authorization code returned by the gateway.

Return Value

Type: void

setGatewayAvsCode(gatewayAvsCode)

Sets the AVS (address verification system) result code information that the gateway returned. Maximum length of 64 characters.

Signature

```
global void setGatewayAvsCode(String gatewayAvsCode)
```

Parameters

gatewayAvsCode
Type: String

Used to verify the address mapped to a payment method when the payments platform requests tokenization from the payment gateway.

Return Value

Type: void

setGatewayDate(gatewayDate)

Sets the date that the authorization occurred. Some gateways don't send this value.

Signature

```
global void setGatewayDate(Datetime gatewayDate)
```

Parameters

gatewayDate
Type: Datetime

Return Value

Type: void

setGatewayMessage(gatewayMessage)

Sets error messages that the gateway returned for the authorization request. Maximum length of 255 characters.

Signature

```
global void setGatewayMessage(String gatewayMessage)
```

Parameters

gatewayMessage
Type: String

Return Value

Type: void

setGatewayReferenceDetails(gatewayReferenceDetails)

Stores data that you can use for subsequent authorizations. You can use any data that isn't normalized in financial entities. This field has a maximum length of 1000 characters and can store data as JSON or XML.

Signature

```
global void setGatewayReferenceDetails(String gatewayReferenceDetails)
```

Parameters

gatewayReferenceDetails
Type: String

Return Value

Type: void

setGatewayReferenceNumber(gatewayReferenceNumber)

Sets the unique gateway reference number for the transaction that the gateway returned. Maximum length of 255 characters.

Signature

```
global void setGatewayReferenceNumber(String gatewayReferenceNumber)
```

Parameters

gatewayReferenceNumber
Type: String

Unique authorization ID created by the payment gateway.

Return Value

Type: void

setGatewayResultCode(gatewayResultCode)

Sets a gateway-specific result code. The code can be mapped to a Salesforce-specific result code. Maximum length of 64 characters.

Signature

```
global void setGatewayResultCode(String gatewayResultCode)
```

Parameters

gatewayResultCode
Type: String

Gateway-specific result code. Must be used to map a Salesforce-specific result code.

Return Value

Type: void

setGatewayResultCodeDescription(gatewayResultCodeDescription)

Sets a description of the gateway-specific result code that a payment gateway returned. Maximum length of 1000 characters.

Signature

```
global void setGatewayResultCodeDescription(String gatewayResultCodeDescription)
```

Parameters

gatewayResultCodeDescription
Type: String

Description of the gateway's result code. Use this field to learn more about why the gateway returned a certain result code.

Return Value

Type: void

`setPaymentMethodTokenizationResponse(paymentMethodTokenizationResponse)`

Sets information from the gateway about the tokenized payment method.

Signature

```
global void
setPaymentMethodTokenizationResponse(commercepayments.PaymentMethodTokenizationResponse
paymentMethodTokenizationResponse)
```

Parameters

paymentMethodTokenizationResponse
PaymentMethodTokenizationResponse on page 404

Gateway response sent by payment gateway adapters for the payment method tokenization request.

Return Value

Type: void

`setSalesforceResultCodeInfo(salesforceResultCodeInfo)`

Sets the Salesforce-specific result code information. Payment gateways have many response codes for payment calls. Salesforce uses the result code information to map payment gateway codes to a predefined set of standard Salesforce result codes.

Signature

```
global void setSalesforceResultCodeInfo(commercepayments.SalesforceResultCodeInfo
salesforceResultCodeInfo)
```

Parameters

salesforceResultCodeInfo
Type: SalesforceResultCodeInfo on page 446

Description of the Salesforce result code value.

Return Value

Type: void

AuthorizationReversalRequest Class

Sends information about an authorization reversal request to a gateway adapter during a service call.

Namespace

CommercePayments on page 297

Example

Add your reversal classes to your payment gateway adapter. We recommend adding `AuthorizationReversal` as a possible requestType value when calling processRequest on the gateway's response.

```
global commercepayments.GatewayResponse processRequest(commercepayments.paymentGatewayContext
 gatewayContext) {
       commercepayments.RequestType requestType = gatewayContext.getPaymentRequestType();

        commercepayments.GatewayResponse response;

        try {
        //add other requestType values here
        //..
        else if (requestType == commercepayments.RequestType.AuthorizationReversal) {
                response =
createAuthReversalResponse((commercepayments.AuthorizationReversalRequest)gatewayContext.getPaymentRequest());}


        return response;
```

Then, add a class that sets the amount of the authorization reversal request, as well as gateway information and the Salesforce result code.

```
global commercepayments.GatewayResponse
createAuthReversalResponse(commercepayments.AuthorizationReversalRequest authReversalRequest)
 {
        commercepayments.AuthorizationReversalResponse authReversalResponse = new
commercepayments.AuthorizationReversalResponse();
        if(authReversalRequest.amount!=null )
        {
            authReversalResponse.setAmount(authReversalRequest.amount);
        }
        else
        {
             throw new SalesforceValidationException('Required Field Missing : Amount');

        }

        system.debug('Response - success');
        authReversalResponse.setGatewayDate(system.now());
        authReversalResponse.setGatewayResultCode('00');
        authReversalResponse.setGatewayResultCodeDescription('Transaction Normal');
        authReversalResponse.setGatewayReferenceNumber('SF'+getRandomNumber(6));

authReversalResponse.setSalesforceResultCodeInfo(SUCCESS_SALESFORCE_RESULT_CODE_INFO);
        return authReversalResponse;
    }
```

IN THIS SECTION:

AuthorizationReversalRequest Constructors

AuthorizationReversalRequest Properties

AuthorizationReversalRequest Methods

AuthorizationReversalRequest Constructors

The following are constructors for `AuthorizationReversalRequest`.

IN THIS SECTION:

AuthorizationReversalRequest(amount, authorizationId)
Constructor for building the amount in an authorization reversal request. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

`AuthorizationReversalRequest(amount, authorizationId)`

Constructor for building the amount in an authorization reversal request. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Signature

```
global AuthorizationReversalRequest(Double amount, String authorizationId)
```

Parameters

amount
Type: Double

The amount of the authorization reversal request.

authorizationId
Type: String

The authorization request to be reversed.

AuthorizationReversalRequest Properties

The following are properties for `AuthorizationReversalRequest`.

IN THIS SECTION:

accountId
References the customer account for the transaction where the authorization reversal was performed.
amount
The total amount of the authorization reversal request. Can be positive or negative.
paymentAuthorizationId
References the payment authorization to be reversed.

`accountId`

References the customer account for the transaction where the authorization reversal was performed.

Signature

```
global String accountId {get; set;}
```

Property Value

Type: String

`amount`

The total amount of the authorization reversal request. Can be positive or negative.

Signature

```
global Double amount {get; set;}
```

Property Value

Type: Double

`paymentAuthorizationId`

References the payment authorization to be reversed.

Signature

```
global String paymentAuthorizationId {get; set;}
```

Property Value

Type: String

AuthorizationReversalRequest Methods

The following are methods for `AuthorizationReversalRequest`.

IN THIS SECTION:

equals(obj)
Maintains the integrity of lists of type `AuthorizationReversalRequest` by determining the equality of external objects in a list. This method is dynamic and based on the equals method in Java.

hashCode()
Maintains the integrity of lists of type `AuthorizationReversalRequest` by determining the uniqueness of the external object in a list.

toString()
Converts a date to a string.

`equals(obj)`

Maintains the integrity of lists of type `AuthorizationReversalRequest` by determining the equality of external objects in a list. This method is dynamic and based on the equals method in Java.

Signature

```
global Boolean equals(Object obj)
```

Parameters

obj
Type: Object

External object whose key is to be validated.

Return Value

Type: Boolean

`hashCode()`

Maintains the integrity of lists of type `AuthorizationReversalRequest` by determining the uniqueness of the external object in a list.

Signature

```
global Integer hashCode()
```

Return Value

Type: Integer

`toString()`

Converts a date to a string.

Signature

```
global String toString()
```

Return Value

Type: String

AuthorizationReversalResponse Class

Response sent by the payment gateway following a payment authorization reversal service.

Namespace

CommercePayments

Usage

The constructor of this class takes no arguments. For example:

```
CommercePayments.AuthorizationReversalResponse authRevRes = new
CommercePayments.AuthorizationResponse();
```

Contains information about the payment gateway's response following an authorization reversal transaction. The gateway adapter uses the payment gateway's response to populate the `AuthorizationReversalResponse` fields. The payments platform uses the information from this class to construct the authorization gateway response shown to the user.

Example

This class builds an authorization reversal response that contains the amount of the original reversal request, gateway information, and the Salesforce result code.

```
global commercepayments.GatewayResponse
createAuthReversalResponse(commercepayments.AuthorizationReversalRequest authReversalRequest)
 {
        commercepayments.AuthorizationReversalResponse authReversalResponse = new
commercepayments.AuthorizationReversalResponse();
        if(authReversalRequest.amount!=null )
        {
            authReversalResponse.setAmount(authReversalRequest.amount);
        }
        else
        {
             throw new SalesforceValidationException('Required Field Missing : Amount');

        }

        system.debug('Response - success');
        authReversalResponse.setGatewayDate(system.now());
        authReversalResponse.setGatewayResultCode('00');
        authReversalResponse.setGatewayResultCodeDescription('Transaction Normal');
        authReversalResponse.setGatewayReferenceNumber('SF'+getRandomNumber(6));

authReversalResponse.setSalesforceResultCodeInfo(SUCCESS_SALESFORCE_RESULT_CODE_INFO);
        return authReversalResponse;
    }
```

IN THIS SECTION:

AuthorizationReversalResponse Methods

AuthorizationReversalResponse Methods

The following are methods for `AuthorizationReversalResponse`.

IN THIS SECTION:

setAmount(amount)
Contains the amount of the authorization reversal. Must be a non-zero value.

setGatewayAvsCode(gatewayAvsCode)
Sets the AVS (Address Verification System) result code that the gateway returned. Maximum length of 64 characters.

setGatewayDate(gatewayDate)
Sets the date that the authorization reversal request occurred in the payment gateway. Some gateways don't send this value.

setGatewayMessage(gatewayMessage)
Sets error messages that the gateway returned for the authorization reversal request. Maximum length of 255 characters.

setGatewayReferenceDetails(gatewayReferenceDetails)
Stores data that you can use for subsequent authorizations. You can use any data that isn't normalized in financial entities. This field has a maximum length of 1000 characters and can store data as JSON or XML.

setGatewayReferenceNumber(gatewayReferenceNumber)
Sets a unique gateway reference number for the transaction that the gateway returned. Maximum length of 255 characters.

setGatewayResultCode(gatewayResultCode)
Sets a gateway-specific result code. The code can be mapped to a Salesforce-specific result code. Maximum length of 64 characters.

setGatewayResultCodeDescription(gatewayResultCodeDescription)
Sets a description of the gateway-specific result code that a payment gateway returned. Maximum length of 1000 characters.

setSalesforceResultCodeInfo(salesforceResultCodeInfo)
Sets the Salesforce-specific result code information. Payment gateways have many response codes for payment calls. Salesforce uses the result code information to map payment gateway codes to a predefined set of standard Salesforce result codes.

`setAmount(amount)`

Contains the amount of the authorization reversal. Must be a non-zero value.

Signature

```
global void setAmount(Double amount)
```

Parameters

amount
Type: Double

Return Value

Type: void

`setGatewayAvsCode(gatewayAvsCode)`

Sets the AVS (Address Verification System) result code that the gateway returned. Maximum length of 64 characters.

Signature

```
global void setGatewayAvsCode(String gatewayAvsCode)
```

Parameters

gatewayAvsCode
Type: String

Used to verify the address mapped to a payment method when the payments platform requests tokenization from the payment gateway.

Return Value

Type: void

setGatewayDate(gatewayDate)

Sets the date that the authorization reversal request occurred in the payment gateway. Some gateways don't send this value.

Signature

```
global void setGatewayDate(Datetime gatewayDate)
```

Parameters

gatewayDate
Type: Datetime

Return Value

Type: void

setGatewayMessage(gatewayMessage)

Sets error messages that the gateway returned for the authorization reversal request. Maximum length of 255 characters.

Signature

```
global void setGatewayMessage(String gatewayMessage)
```

Parameters

gatewayMessage
Type: String

Return Value

Type: void

setGatewayReferenceDetails(gatewayReferenceDetails)

Stores data that you can use for subsequent authorizations. You can use any data that isn't normalized in financial entities. This field has a maximum length of 1000 characters and can store data as JSON or XML.

Signature

```
global void setGatewayReferenceDetails(String gatewayReferenceDetails)
```

Parameters

gatewayReferenceDetails
Type: String

Return Value

Type: void

setGatewayReferenceNumber(gatewayReferenceNumber)

Sets a unique gateway reference number for the transaction that the gateway returned. Maximum length of 255 characters.

Signature

```
global void setGatewayReferenceNumber(String gatewayReferenceNumber)
```

Parameters

gatewayReferenceNumber
Type: String

Unique reference ID created by the payment gateway.

Return Value

Type: void

setGatewayResultCode(gatewayResultCode)

Sets a gateway-specific result code. The code can be mapped to a Salesforce-specific result code. Maximum length of 64 characters.

Signature

```
global void setGatewayResultCode(String gatewayResultCode)
```

Parameters

gatewayResultCode
Type: String

Gateway-specific result code. Must be used to map a Salesforce-specific result code.

Return Value

Type: void

setGatewayResultCodeDescription(gatewayResultCodeDescription)

Sets a description of the gateway-specific result code that a payment gateway returned. Maximum length of 1000 characters.

Signature

```
global void setGatewayResultCodeDescription(String gatewayResultCodeDescription)
```

Parameters

gatewayResultCodeDescription
Type: String

Description of the gateway's result code. Use this field to learn more about why the gateway returned a certain result code.

Return Value

Type: void

setSalesforceResultCodeInfo(salesforceResultCodeInfo)

Sets the Salesforce-specific result code information. Payment gateways have many response codes for payment calls. Salesforce uses the result code information to map payment gateway codes to a predefined set of standard Salesforce result codes.

Signature

```
global void setSalesforceResultCodeInfo(commercepayments.SalesforceResultCodeInfo
salesforceResultCodeInfo)
```

Parameters

salesforceResultCodeInfo
Type: SalesforceResultCodeInfo

Description of the Salesforce result code value.

Return Value

Type: void

BaseApiPaymentMethodRequest Class

Abstract class used to send information about a payment method to a gateway adapter during a service call.

Namespace

CommercePayments

Usage

`BaseApiPaymentMethodRequest` is the base class for `SaleApiPaymentMethodRequest` and `AuthApiPaymentMethodRequest`.

IN THIS SECTION:

BaseApiPaymentMethodRequest Constructors

BaseApiPaymentMethodRequest Properties

BaseApiPaymentMethodRequest Methods

BaseApiPaymentMethodRequest Constructors

The following are constructors for `BaseApiPaymentMethodRequest`.

IN THIS SECTION:

BaseApiPaymentMethodRequest(address, id, saveForFuture)
Constructs a payment method. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

`BaseApiPaymentMethodRequest(address, id, saveForFuture)`

Constructs a payment method. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Signature

```
global BaseApiPaymentMethodRequest(commercepayments.AddressRequest address, String id,
Boolean saveForFuture)
```

Parameters

address
Type: commercepayments.AddressRequest on page 308

Sends data related on address request to a gateway adapter during a service call.

id
Type: String

saveForFuture
Type: Boolean

Indicates whether Salesforce saves the payment method for future use.

BaseApiPaymentMethodRequest Properties

The following are properties for `BaseApiPaymentMethodRequest`.

IN THIS SECTION:

address
The payment method's address.

id
ID of the payment method request.

saveForFuture
Indicates whether the payment method is saved as a record in Salesforce for future use.

address

The payment method's address.

Signature

```
global commercepayments.AddressRequest address {get; set;}
```

Property Value

Type: AddressRequest on page 308

id

ID of the payment method request.

Signature

```
global String id {get; set;}
```

Property Value

Type: String

saveForFuture

Indicates whether the payment method is saved as a record in Salesforce for future use.

Signature

```
global Boolean saveForFuture {get; set;}
```

Property Value

Type: Boolean

BaseApiPaymentMethodRequest Methods

The following are methods for `BaseApiPaymentMethodRequest`.

IN THIS SECTION:

equals(obj)
Maintains the integrity of lists of type `BaseApiPaymentMethodRequest` by determining the equality of external objects in a list. This method is dynamic and is based on the equals method in Java.

hashCode()
Maintains the integrity of lists of type `BaseApiPaymentMethodRequest` by determining the uniqueness of the external object records in a list.

toString()
Converts a date to a string.

equals(obj)

Maintains the integrity of lists of type `BaseApiPaymentMethodRequest` by determining the equality of external objects in a list. This method is dynamic and is based on the equals method in Java.

Signature

```
global Boolean equals(Object obj)
```

Parameters

obj
Type: Object

External object whose key is to be validated.

Return Value

Type: Boolean

hashCode()

Maintains the integrity of lists of type `BaseApiPaymentMethodRequest` by determining the uniqueness of the external object records in a list.

Signature

```
global Integer hashCode()
```

Return Value

Type: Integer

toString()

Converts a date to a string.

Signature

```
global String toString()
```

Return Value

Type: String

BaseNotification Class

Abstract class for storing notification information sent from payment gateways.

Namespace

CommercePayments

Usage

An abstract class that contains the common fields from payment gateways. `BaseNotification` can't be instantiated on its own.

The constructor of this class takes no arguments. For example:

```
CommercePayments.BaseNotification bnt = new CommercePayments.BaseNotification();
```

Example

```
commercepayments.BaseNotification notification = null;
        if ('CAPTURE'.equals(eventCode)) {
            notification = new commercepayments.CaptureNotification();
        } else if ('REFUND'.equals(eventCode)) {
            notification = new commercepayments.ReferencedRefundNotification();
        }
```

IN THIS SECTION:

BaseNotification Methods

BaseNotification Methods

The following are methods for `BaseNotification`.

IN THIS SECTION:

setAmount(amount)
Sets the transaction amount. Must be a non-negative value.

setGatewayDate(gatewayDate)
Sets the date that the notification occurred. Some gateways don't send this value.

setGatewayMessage(gatewayMessage)
Sets error messages that the gateway returned for the notification request. Maximum length of 255 characters.

setGatewayReferenceDetails(gatewayReferenceDetails)
Sets the payment gateway's reference details.

setGatewayReferenceNumber(gatewayReferenceNumber)
Sets the payment gateway's reference number.

setGatewayResultCode(gatewayResultCode)
Sets a gateway-specific result code. The code can be mapped to a Salesforce-specific result code. Maximum length of 64 characters.

setGatewayResultCodeDescription(gatewayResultCodeDescription)
Sets a description of the gateway-specific result code that a payment gateway returned. Maximum length of 1000 characters.

setId(id)
Sets the ID of the notification sent by the gateway.

setSalesforceResultCodeInfo(salesforceResultCodeInfo)
Sets the information about the Salesforce-specific result code used to match a result code from a payment gateway.

setStatus(status)
Sets the status of the notification sent by the gateway.

`setAmount(amount)`

Sets the transaction amount. Must be a non-negative value.

Signature

```
global void setAmount(Double amount)
```

Parameters

amount
Type: Double

The amount of the transaction.

Return Value

Type: void

`setGatewayDate(gatewayDate)`

Sets the date that the notification occurred. Some gateways don't send this value.

Signature

```
global void setGatewayDate(Datetime gatewayDate)
```

Parameters

gatewayDate
Type: Datetime

The date that the notification occurred.

Return Value

Type: void

setGatewayMessage(gatewayMessage)

Sets error messages that the gateway returned for the notification request. Maximum length of 255 characters.

Signature

```
global void setGatewayMessage(String gatewayMessage)
```

Parameters

gatewayMessage
Type: String

The message that the gateway returned with the notification request. Contains additional information about the notification.

Return Value

Type: void

setGatewayReferenceDetails(gatewayReferenceDetails)

Sets the payment gateway's reference details.

Signature

```
global void setGatewayReferenceDetails(String gatewayReferenceDetails)
```

Parameters

gatewayReferenceDetails
Type: String

Provides information about the gateway communication.

Return Value

Type: void

setGatewayReferenceNumber(gatewayReferenceNumber)

Sets the payment gateway's reference number.

Signature

```
global void setGatewayReferenceNumber(String gatewayReferenceNumber)
```

Parameters

gatewayReferenceNumber
Type: String

Unique transaction ID created by the payment gateway.

Return Value

Type: void

`setGatewayResultCode(gatewayResultCode)`

Sets a gateway-specific result code. The code can be mapped to a Salesforce-specific result code. Maximum length of 64 characters.

Signature

```
global void setGatewayResultCode(String gatewayResultCode)
```

Parameters

`gatewayResultCode`
Type: String

Gateway-specific result code. Must be used to map a Salesforce-specific result code.

Return Value

Type: void

`setGatewayResultCodeDescription(gatewayResultCodeDescription)`

Sets a description of the gateway-specific result code that a payment gateway returned. Maximum length of 1000 characters.

Signature

```
global void setGatewayResultCodeDescription(String gatewayResultCodeDescription)
```

Parameters

`gatewayResultCodeDescription`
Type: String

Provides additional information about the result code and why the gateway returned the code. Descriptions vary between different gateways.

Return Value

Type: void

`setId(id)`

Sets the ID of the notification sent by the gateway.

Signature

```
global void setId(String id)
```

Parameters

id
Type: String

Return Value

Type: void

setSalesforceResultCodeInfo(salesforceResultCodeInfo)

Sets the information about the Salesforce-specific result code used to match a result code from a payment gateway.

Signature

```
global void setSalesforceResultCodeInfo(commercepayments.SalesforceResultCodeInfo
salesforceResultCodeInfo)
```

Parameters

salesforceResultCodeInfo
Type: commercepayments.SalesforceResultCodeInfo on page 446

Payment gateways have many response codes for payment calls. Salesforce uses the result code information to map payment gateway codes to a predefined set of standard Salesforce result codes.

Return Value

Type: void

setStatus(status)

Sets the status of the notification sent by the gateway.

Signature

```
global void setStatus(commercepayments.NotificationStatus status)
```

Parameters

status
Type: commercepayments.NotificationStatus on page 389

Shows whether the payments platform successfully received the notification from the gateway.

Return Value

Type: void

BasePaymentMethodRequest Class

Abstract class for storing information about payment methods.

Namespace

CommercePayments

Usage

The `BasePaymentMethodRequest` class contains fields common to `CardPaymentMethodRequest` on page 366

.

IN THIS SECTION:

BasePaymentMethodRequest Methods

BasePaymentMethodRequest Methods

The following are methods for `BasePaymentMethodRequest`.

IN THIS SECTION:

equals(obj)
Maintains the integrity of lists of type `BasePaymentMethodRequest` by determining the equality of external objects in a list. This method is dynamic and based on the equals method in Java.

hashCode()
Maintains the integrity of lists of type `BasePaymentMethodRequest` by determining the uniqueness of the external object records in a list.

toString()
Converts a date to a string.

`equals(obj)`

Maintains the integrity of lists of type `BasePaymentMethodRequest` by determining the equality of external objects in a list. This method is dynamic and based on the equals method in Java.

Signature

```
global Boolean equals(Object obj)
```

Parameters

obj
Type: Object

External object whose key is to be validated.

Return Value

Type: Boolean

`hashCode()`

Maintains the integrity of lists of type `BasePaymentMethodRequest` by determining the uniqueness of the external object records in a list.

Signature

```
global Integer hashCode()
```

Return Value

Type: Integer

`toString()`

Converts a date to a string.

Signature

```
global String toString()
```

Return Value

Type: String

BaseRequest Class

`BaseRequest` is extended by all the request classes.

Namespace

CommercePayments

IN THIS SECTION:

BaseRequest Methods

BaseRequest Methods

The following are methods for `BaseRequest`.

IN THIS SECTION:

BaseRequest(AdditionalData, IdempotencyKey)
Used for testing.

`BaseRequest(AdditionalData, IdempotencyKey)`

Used for testing.

Signature

```
global Void BaseRequest(String AdditionalData, Map<String, String> IdempotencyKey)
```

Parameters

AdditionalData
Type: String

Contains additional data that may be required for a payment request. The `additionalData` object consists of key-value pairs. You can retrieve the `additionalData` object from the request object: `Map<String, String> additionalData=request.additionalData`

IdempotencyKey
Type: Map<String, String>

Unique value that's generated by a client and sent to the server in the request. The server stores the value and uses the it to keep track of the request status.

Return Value

Type: Void

CaptureNotification Class

When a payment gateway sends a notification for a capture transaction, the payment gateway adapter creates the `CaptureNotification` object to store information about the notification.

Namespace

CommercePayments

Usage

`CaptureNotification` is used in asynchronous payment gateway adapters.

Specify the `CommercePayments` namespace when creating an instance of this class. The constructor of this class takes no arguments. For example:

```
CommercePayments.CaptureNotification crn = new CommercePayments.CaptureNotification();
```

Example

```
commercepayments.BaseNotification notification = null;
        if ('CAPTURE'.equals(eventCode)) {
            notification = new commercepayments.CaptureNotification();
        } else if ('REFUND'.equals(eventCode)) {
            notification = new commercepayments.ReferencedRefundNotification();
        }
```

IN THIS SECTION:

CaptureNotification Methods

CaptureNotification Methods

The following are methods for `CaptureNotification`.

IN THIS SECTION:

setAmount(amount)
Sets the transaction amount. Must be a non-negative value.

setGatewayDate(gatewayDate)
Sets the date that the transaction occurred. Some gateways don't send this value.

setGatewayMessage(gatewayMessage)
Sets error messages that the gateway returned for the payment request. Maximum length of 255 characters.

setGatewayReferenceDetails(gatewayReferenceDetails)
Sets additional data that you can use for subsequent transactions. You can use any data that isn't normalized in financial entities. This field has a maximum length of 1000 characters and can store data as JSON or XML.

setGatewayReferenceNumber(gatewayReferenceNumber)
Sets the unique gateway reference number for the transaction that the gateway returned. Maximum length of 255 characters.

setGatewayResultCode(gatewayResultCode)
Sets a gateway-specific result code. The code can be mapped to a Salesforce-specific result code. Maximum length of 64 characters.

setGatewayResultCodeDescription(gatewayResultCodeDescription)
Sets a description of the gateway-specific result code that a gateway returned. Maximum length of 1000 characters.

setId(id)
Sets the ID of a notification sent by the payment gateway.

setSalesforceResultCodeInfo(salesforceResultCodeInfo)
Sets the Salesforce-specific result code information. Payment gateways have many response codes for payment calls. Salesforce uses the result code information to map payment gateway codes to a predefined set of standard Salesforce result codes.

setStatus(status)
Sets the notification status to the same value that was sent by the gateway.

`setAmount(amount)`

Sets the transaction amount. Must be a non-negative value.

Signature

```
global void setAmount(Double amount)
```

Parameters

amount
Type: Double

The amount to be debited or captured.

Return Value

Type: void

setGatewayDate(gatewayDate)

Sets the date that the transaction occurred. Some gateways don't send this value.

Signature

```
global void setGatewayDate(Datetime gatewayDate)
```

Parameters

gatewayDate
Type: Datetime

Date and time of the gateway communication.

Return Value

Type: void

setGatewayMessage(gatewayMessage)

Sets error messages that the gateway returned for the payment request. Maximum length of 255 characters.

Signature

```
global void setGatewayMessage(String gatewayMessage)
```

Parameters

gatewayMessage
Type: String

Information on error messages sent from the gateway.

Return Value

Type: void

setGatewayReferenceDetails(gatewayReferenceDetails)

Sets additional data that you can use for subsequent transactions. You can use any data that isn't normalized in financial entities. This field has a maximum length of 1000 characters and can store data as JSON or XML.

Signature

```
global void setGatewayReferenceDetails(String gatewayReferenceDetails)
```

Parameters

gatewayReferenceDetails
Type: String

Return Value

Type: void

setGatewayReferenceNumber(gatewayReferenceNumber)

Sets the unique gateway reference number for the transaction that the gateway returned. Maximum length of 255 characters.

Signature

```
global void setGatewayReferenceNumber(String gatewayReferenceNumber)
```

Parameters

gatewayReferenceNumber
Type: String

Unique transaction ID created by the payment gateway.

Return Value

Type: void

setGatewayResultCode(gatewayResultCode)

Sets a gateway-specific result code. The code can be mapped to a Salesforce-specific result code. Maximum length of 64 characters.

Signature

```
global void setGatewayResultCode(String gatewayResultCode)
```

Parameters

gatewayResultCode
Type: String

Gateway-specific result code. Map this value to a Salesforce-specific result code.

Return Value

Type: void

setGatewayResultCodeDescription(gatewayResultCodeDescription)

Sets a description of the gateway-specific result code that a gateway returned. Maximum length of 1000 characters.

Signature

```
global void setGatewayResultCodeDescription(String gatewayResultCodeDescription)
```

Parameters

gatewayResultCodeDescription
Type: String

Description of the gateway's result code. Use this field to learn more about why the gateway returned a certain result code.

Return Value

Type: void

setId(id)

Sets the ID of a notification sent by the payment gateway.

Signature

```
global void setId(String id)
```

Parameters

id
Type: String

Return Value

Type: void

setSalesforceResultCodeInfo(salesforceResultCodeInfo)

Sets the Salesforce-specific result code information. Payment gateways have many response codes for payment calls. Salesforce uses the result code information to map payment gateway codes to a predefined set of standard Salesforce result codes.

Signature

```
global void setSalesforceResultCodeInfo(commercepayments.SalesforceResultCodeInfo
salesforceResultCodeInfo)
```

Parameters

salesforceResultCodeInfo
Type: commercepayments.SalesforceResultCodeInfo on page 446

Description of the Salesforce result code value.

Return Value

Type: void

setStatus(status)

Sets the notification status to the same value that was sent by the gateway.

Signature

```
global void setStatus(commercepayments.NotificationStatus status)
```

Parameters

status
Type: NotificationStatus on page 389

Sets the Salesforce-specific result code information. Payment gateways have many response codes for payment calls. Salesforce uses the result code information to map payment gateway codes to a predefined set of standard Salesforce result codes.

Return Value

Type: void

CaptureRequest Class

Represents a capture request. This class extends the `BaseRequest` class and inherits all its methods.

Namespace

CommercePayments on page 297

Usage

The `CaptureRequest` class's `buildCaptureRequest` method creates a CaptureRequest object to store payment information, such as value and currency, as JSON strings.

Example

Builds a CaptureRequest object for a multicurrency org.

```
    private String buildCaptureRequest(commercepayments.CaptureRequest captureRequest) {
        Boolean IS_MULTICURRENCY_ORG = UserInfo.isMultiCurrencyOrganization();
        QueryUtils qBuilderForAuth = new QueryUtils(PaymentAuthorization.SObjectType);
        // Add required fields
        qBuilderForAuth.getSelectClause().addField('GatewayRefNumber', false);
        if (IS_MULTICURRENCY_ORG) {
            // addField also takes a boolean to enable translation (uses label instead of
actual value)
            qBuilderForAuth.getSelectClause().addField('CurrencyIsoCode', false);
        }
```

IN THIS SECTION:

CaptureRequest Constructors

CaptureRequest Properties

CaptureRequest Constructors

The following are constructors for `CaptureRequest`.

IN THIS SECTION:

CaptureRequest(amount, authorizationId)
This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

`CaptureRequest(amount, authorizationId)`

This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Parameters

`amount`
Type: Double

The amount to be debited or captured.

`authorizationId`
Type: String

Represents a payment authorization record.

CaptureRequest Properties

The following are properties for `CaptureRequest`.

IN THIS SECTION:

accountId
Account ID value. References an account record.

amount
Amount of currency that needs to be captured.

paymentAuthorizationId
ID value that references a PaymentAuthorization.

`accountId`

Account ID value. References an account record.

Property Value

Type: String

`amount`

Amount of currency that needs to be captured.

Property Value

Type: Double

`paymentAuthorizationId`

ID value that references a PaymentAuthorization.

Property Value

Type: String

CaptureResponse Class

The payment gateway adapter sends this response for the capture request type. This class extends `AbstractResponse` and inherits its methods.

Namespace

CommercePayments on page 297

Usage

You must specify the `CommercePayments` namespace when creating an instance of this class. The constructor of this class takes no arguments. For example:

```
CommercePayments.Capture Response ctr = new CommercePayments.CaptureResponse();
```

IN THIS SECTION:

CaptureResponse Methods

CaptureResponse Methods

The following are methods for `CaptureResponse`.

IN THIS SECTION:

setAmount(amount)
Sets the transaction amount.

setAsync(async)
Indicates whether the payment gateway adapter used in the payment capture was asynchronous (`True`) or synchronous (`False`).

setGatewayAvsCode(gatewayAvsCode)
Sets the payment gateway's AVS (address verification system) code.

setGatewayDate(gatewayDate)
Sets the payment gateway's date.

setGatewayMessage(gatewayMessage)
Sets information or messages that the gateway returned.

setGatewayReferenceDetails(gatewayReferenceDetails)
Sets the payment gateway's reference details.

setGatewayReferenceNumber(gatewayReferenceNumber)
Sets the payment gateway's reference number.

setGatewayResultCode(gatewayResultCode)
Sets the payment gateway's result code.

setGatewayResultCodeDescription(gatewayResultCodeDescription)
Sets the payment gateway's result code description.

setSalesforceResultCodeInfo(salesforceResultCodeInfo)
Sets Salesforce result code information.

`setAmount(amount)`

Sets the transaction amount.

Signature

```
global void setAmount(Double amount)
```

Parameters

amount
Type: Double

The amount to be debited or captured.

Return Value

Type: void

`setAsync(async)`

Indicates whether the payment gateway adapter used in the payment capture was asynchronous (`True`) or synchronous (`False`).

Signature

```
global void setAsync(Boolean async)
```

Parameters

async
Type: Boolean

Return Value

Type: void

setGatewayAvsCode(gatewayAvsCode)

Sets the payment gateway's AVS (address verification system) code.

Signature

```
global void setGatewayAvsCode(String gatewayAvsCode)
```

Parameters

gatewayAvsCode
Type: String

Payment gateways that use an AVS system return this code.

Return Value

Type: void

setGatewayDate(gatewayDate)

Sets the payment gateway's date.

Signature

```
global void setGatewayDate(Datetime gatewayDate)
```

Parameters

gatewayDate
Type: Datetime

The date that communication happened with the gateway.

Return Value

Type: void

setGatewayMessage(gatewayMessage)

Sets information or messages that the gateway returned.

Signature

```
global void setGatewayMessage(String gatewayMessage)
```

Parameters

gatewayMessage
Type: String

Information or error messages returned by the gateway.

Return Value

Type: void

setGatewayReferenceDetails(gatewayReferenceDetails)

Sets the payment gateway's reference details.

Signature

```
global void setGatewayReferenceDetails(String gatewayReferenceDetails)
```

Parameters

gatewayReferenceDetails
Type: String

Provides information about the gateway communication.

Return Value

Type: void

setGatewayReferenceNumber(gatewayReferenceNumber)

Sets the payment gateway's reference number.

Signature

```
global void setGatewayReferenceNumber(String gatewayReferenceNumber)
```

Parameters

gatewayReferenceNumber
Type: String

Unique transaction ID created by the payment gateway.

Return Value

Type: void

setGatewayResultCode(gatewayResultCode)

Sets the payment gateway's result code.

Signature

```
global void setGatewayResultCode(String gatewayResultCode)
```

Parameters

gatewayResultCode
Type: String

The gateway result code. You must map this to a Salesforce result code.

Return Value

Type: void

setGatewayResultCodeDescription(gatewayResultCodeDescription)

Sets the payment gateway's result code description.

Signature

```
global void setGatewayResultCodeDescription(String gatewayResultCodeDescription)
```

Parameters

gatewayResultCodeDescription
Type: String

Description of the GatewayResultCode. Provides additional context about the result code that the gateway returned.

Return Value

Type: void

setSalesforceResultCodeInfo(salesforceResultCodeInfo)

Sets Salesforce result code information.

Signature

```
global void setSalesforceResultCodeInfo(commercepayments.SalesforceResultCodeInfo
salesforceResultCodeInfo)
```

Parameters

salesforceResultCodeInfo
SalesforceResultCodeInfoType: commercepayments.SalesforceResultCodeInfo

Description of the Salesforce result code value.

Return Value

Type: void

CardCategory Enum

Defines whether the payment method represents a credit card or a debit card.

Namespace

CommercePayments on page 297

Enum Values

The following are the values of the `commercepayments.CardCategory` enum.

Value	Description
`CreditCard`	Shows that the payment method is a credit card.
`DebitCard`	Shows that the payment method is a debit card.

CardPaymentMethodRequest Class

Sends data related to a card payment method to a gateway adapter during a service call.

Namespace

CommercePayments on page 297

Usage

This class contains details about the card used as a payment method for authorization, sale, or tokenization transaction requests. The gateway adapter reads the fields of this class object while constructing a transaction JSON request to send to the payment gateway. The object of this class is available as the `cardPaymentMethod` field in the `SaleApiPaymentMethodRequest Class`, `AuthApiPaymentMethodRequest Class`, and `PaymentMethodTokenizationRequest Class`.

Example: This code sample retrieves the `cardPaymentMethodRequest` object from the `paymentMethod` class.

```
commercepayments.CardPaymentMethodRequest cardPaymentMethod =
paymentMethod.cardPaymentMethod;
```

IN THIS SECTION:

CardPaymentMethodRequest Constructors

CardPaymentMethodRequest Properties

CardPaymentMethodRequest Methods

CardPaymentMethodRequest Constructors

The following are constructors for `CardPaymentMethodRequest`.

IN THIS SECTION:

CardPaymentMethodRequest(cardCategory)

Sets the `cardCategory` value for the card payment method request.

`CardPaymentMethodRequest(cardCategory)`

Sets the `cardCategory` value for the card payment method request.

Signature

```
global CardPaymentMethodRequest(commercepayments.CardCategory cardCategory)
```

Parameters

`cardCategory`
Type: CardCategory on page 365

Defines whether the card payment method is a credit card or a debit card.

CardPaymentMethodRequest Properties

The following are properties for `CardPaymentMethodRequest`.

IN THIS SECTION:

accountId
Customer account for this payment method.

autoPay
Indicates whether a token is being requested so that the payment method can be used for recurring payments.

cardCategory
Indicates whether a card payment method is for a credit card or debit card.

cardHolderFirstName
The first name of the cardholder for the card payment method.

cardHolderLastName
The last name of the cardholder for the card payment method.

cardHolderName
Full name of the cardholder on the card payment method.

cardNumber
System-defined unique ID for the card payment method.

cardType
Defines the credit card bank. Possible values are `AmericanExpress`, `DinersClub`, `JCB`, `Maestro`, `MasterCard`, and`Visa`.

cvv
The card security code for the credit or debit card on a card payment method.

email
Email address of the cardholder for the credit or debit card on a card payment method.

expiryMonth
Expiration month for the credit or debit card on a card payment method.

expiryYear
Expiration year of the credit or debit card for the card payment method.

inputCardType
Input field for the card type. This field doesn't store the card type directly, but instead populates CardBin, LastFour, and DisplayCardNumber based on the value entered in `inputCardType`.

startMonth
The credit or debit card becomes valid on the first day of the `startMonth` in the `startYear`

startYear
Year during which the credit or debit card becomes valid.

accountId

Customer account for this payment method.

Signature

```
global String accountId {get; set;}
```

Property Value

Type: String

autoPay

Indicates whether a token is being requested so that the payment method can be used for recurring payments.

Signature

```
global Boolean autoPay {get; set;}
```

Property Value

Type: Boolean

cardCategory

Indicates whether a card payment method is for a credit card or debit card.

Signature

```
global commercepayments.CardCategory cardCategory {get; set;}
```

Property Value

Type: CardCategory on page 365

cardHolderFirstName

The first name of the cardholder for the card payment method.

Signature

```
global String cardHolderFirstName {get; set;}
```

Property Value

Type: String

cardHolderLastName

The last name of the cardholder for the card payment method.

Signature

```
global String cardHolderLastName {get; set;}
```

Property Value

Type: String

cardHolderName

Full name of the cardholder on the card payment method.

Signature

```
global String cardHolderName {get; set;}
```

Property Value

Type: String

cardNumber

System-defined unique ID for the card payment method.

Signature

```
global String cardNumber {get; set;}
```

Property Value

Type: String

cardType

Defines the credit card bank. Possible values are `AmericanExpress`, `DinersClub`, `JCB`, `Maestro`, `MasterCard`, and `Visa`.

Signature

```
global commercepayments.CardType cardType {get; set;}
```

Property Value

Type: CardType

cvv

The card security code for the credit or debit card on a card payment method.

Signature

```
global String cvv {get; set;}
```

Property Value

Type: String

email

Email address of the cardholder for the credit or debit card on a card payment method.

Signature

```
global String email {get; set;}
```

Property Value

Type: String

expiryMonth

Expiration month for the credit or debit card on a card payment method.

Signature

```
global Integer expiryMonth {get; set;}
```

Property Value

Type: Integer

expiryYear

Expiration year of the credit or debit card for the card payment method.

Signature

```
global Integer expiryYear {get; set;}
```

Property Value

Type: Integer

inputCardType

Input field for the card type. This field doesn't store the card type directly, but instead populates CardBin, LastFour, and DisplayCardNumber based on the value entered in `inputCardType`.

Signature

```
global String inputCardType {get; set;}
```

Property Value

Type: String

startMonth

The credit or debit card becomes valid on the first day of the `startMonth` in the `startYear`

Signature

```
global Integer startMonth {get; set;}
```

Property Value

Type: Integer

startYear

Year during which the credit or debit card becomes valid.

Signature

```
global Integer startYear {get; set;}
```

Property Value

Type: Integer

CardPaymentMethodRequest Methods

The following are methods for `CardPaymentMethodRequest`.

IN THIS SECTION:

equals(obj)

Maintains the integrity of lists of type `CardPaymentMethodRequest` by determining the equality of external objects in a list. This method is dynamic and based on the equals method in Java.

hashCode()

Maintains the integrity of lists of type `CardPaymentMethodRequest`.

toString()

Converts a date to a string.

equals(obj)

Maintains the integrity of lists of type `CardPaymentMethodRequest` by determining the equality of external objects in a list. This method is dynamic and based on the equals method in Java.

Signature

```
global Boolean equals(Object obj)
```

Parameters

obj
Type: Object

External object whose key is to be validated.

Return Value

Type: Boolean

hashCode()

Maintains the integrity of lists of type `CardPaymentMethodRequest`.

Signature

```
global Integer hashCode()
```

Return Value

Type: Integer

toString()

Converts a date to a string.

Signature

```
global String toString()
```

Return Value

Type: String

CardPaymentMethodResponse Class

This class contains details about the card payment method.

Namespace

CommercePayments

IN THIS SECTION:

CardPaymentMethodResponse Methods

CardPaymentMethodResponse Methods

The following are methods for `CardPaymentMethodResponse`.

IN THIS SECTION:

setAccountId(accountId)
Sets the Salesforce payments account to which this payment method is linked.

setAutoPay(autoPay)
Sets whether a token for recurring payments is being requested or not.

setCardBin(cardBin)
Sets the card Bank Identification Number (BIN).

setCardCategory(cardCategory)
Sets the card category.

setCardHolderFirstName(cardHolderFirstName)
Sets the first name of the card holder.

setCardHolderLastName(cardHolderLastName)
Sets the last name of the card holder.

setCardHolderName(cardHolderName)
Sets the name of the card holder.

setCardLastFour(cardLastFour)
Sets the last four digits of the card.

setCardType(cardType)
Specifies the type of the credit card issuer.

setCardTypeCategory(cardTypeCategory)
Sets the credit card issuer.

setComments(comments)
Sets the notes added by a user for card payment.

setDisplayCardNumber(displayCardNumber)
Sets the display card number.

setEmail(email)
Sets the email address of the card holder.

setExpiryMonth(expiryMonth)
Sets the month of expiry of the card.

setExpiryYear(expiryYear)
Sets the year of expiry of the card.

setNickName(nickName)
Sets the nickname of the card.

setStartMonth(startMonth)
Sets the month the card becomes active.

setStartYear(startYear)
Sets the year the card becomes active.

setAccountId(accountId)

Sets the Salesforce payments account to which this payment method is linked.

Signature

```
public void setAccountId(Id accountId)
```

Parameters

accountId
Type: Id

Salesforce Payments account to which this payment method is linked.

Return Value

Type: void

setAutoPay(autoPay)

Sets whether a token for recurring payments is being requested or not.

Signature

```
public void setAutoPay(Boolean autoPay)
```

Parameters

autoPay
Type: Boolean

Indicates whether a token for recurring payments is being requested (`true`) or not (`false`). The token lets the payment method be used for recurring payments.

Return Value

Type: void

setCardBin(cardBin)

Sets the card Bank Identification Number (BIN).

Signature

```
public void setCardBin(String cardBin)
```

Parameters

cardBin
Type: String

Bank Identification Number (BIN). The BIN is the first 4-6 numbers on a payment card that identifies the card issuer.

Return Value

Type: void

setCardCategory(cardCategory)

Sets the card category.

Signature

```
public void setCardCategory(commercepayments.CardCategory cardCategory)
```

Parameters

cardCategory
Type: CommercePayments.CardCategory

Specifies whether it is a credit card or debit card.

Return Value

Type: void

setCardHolderFirstName(cardHolderFirstName)

Sets the first name of the card holder.

Signature

```
public void setCardHolderFirstName(String cardHolderFirstName)
```

Parameters

cardHolderFirstName
Type: String

First name of the card holder.

Return Value

Type: void

setCardHolderLastName(cardHolderLastName)

Sets the last name of the card holder.

Signature

```
public void setCardHolderLastName(String cardHolderLastName)
```

Parameters

cardHolderLastName
Type: String

Last name of the card holder.

Return Value

Type: void

setCardHolderName(cardHolderName)

Sets the name of the card holder.

Signature

```
public void setCardHolderName(String cardHolderName)
```

Parameters

cardHolderName
Type: String

Card holder name.

Return Value

Type: void

setCardLastFour(cardLastFour)

Sets the last four digits of the card.

Signature

```
public void setCardLastFour(String cardLastFour)
```

Parameters

cardLastFour
Type: String

Last four digits of the card.

Return Value

Type: void

`setCardType(cardType)`

Specifies the type of the credit card issuer.

Signature

```
public void setCardType(String cardType)
```

Parameters

`cardType`
Type: String

Type of the credit card issuer.

Return Value

Type: void

`setCardTypeCategory(cardTypeCategory)`

Sets the credit card issuer.

Signature

```
public void setCardTypeCategory(commercepayments.CardType cardTypeCategory)
```

Parameters

`cardTypeCategory`
Type: CommercePayments.CardType

Credit card issuer.

Return Value

Type: void

`setComments(comments)`

Sets the notes added by a user for card payment.

Signature

```
public void setComments(String comments)
```

Parameters

`comments`
Type: String

Details about a record added by a user, maximum is 1000 characters.

Return Value

Type: void

`setDisplayCardNumber(displayCardNumber)`

Sets the display card number.

Signature

```
public void setDisplayCardNumber(String displayCardNumber)
```

Parameters

displayCardNumber
Type: String

Displayed card number.

Return Value

Type: void

`setEmail(email)`

Sets the email address of the card holder.

Signature

```
public void setEmail(String email)
```

Parameters

email
Type: String

Email address of the card holder.

Return Value

Type: void

`setExpiryMonth(expiryMonth)`

Sets the month of expiry of the card.

Signature

```
public void setExpiryMonth(Integer expiryMonth)
```

Parameters

expiryMonth
Type: Integer

Month of expiry of the card.

Return Value

Type: void

setExpiryYear(expiryYear)

Sets the year of expiry of the card.

Signature

```
public void setExpiryYear(Integer expiryYear)
```

Parameters

expiryYear
Type: Integer

Year of expiry of the card.

Return Value

Type: void

setNickName(nickName)

Sets the nickname of the card.

Signature

```
public void setNickName(String nickName)
```

Parameters

nickName
Type: String

Card nickname.

Return Value

Type: void

setStartMonth(startMonth)

Sets the month the card becomes active.

Signature

```
public void setStartMonth(Integer startMonth)
```

Parameters

startMonth
Type: Integer

Determines from which month the card becomes active.

Return Value

Type: void

setStartYear(startYear)

Sets the year the card becomes active.

Signature

```
public void setStartYear(Integer startYear)
```

Parameters

startYear
Type: Integer

Determines from which year the card becomes active.

Return Value

Type: void

CardType Enum

Specifies the credit card issuer.

Enum Values

The following are the values of the `commercepayments.CardType` enum.

Value	Description
`AmericanExpress`	American Express card
`DinersClub`	Diners Club card
`Jcb`	Japan Credit Bureau (JCB) card
`Maestro`	Maestro card
`MasterCard`	Master card

Value	Description
`Visa`	Visa card

CustomMetadataTypeInfo Class

Access information about custom metadata. The `PaymentGatewayAdapter` can send `CustomMetadataTypeInfo` to transaction requests through the response object's `SalesforceResultCodeInfo`.

Namespace

CommercePayments on page 297

IN THIS SECTION:

CustomMetadataTypeInfo Constructors

CustomMetadataTypeInfo Methods

CustomMetadataTypeInfo Constructors

The following are constructors for `CustomMetadataTypeInfo`.

IN THIS SECTION:

CustomMetadataTypeInfo(cmtRecordId, cmtSfResultCodeFieldName)
Constructor for providing custom metadata type information.

`CustomMetadataTypeInfo(cmtRecordId, cmtSfResultCodeFieldName)`

Constructor for providing custom metadata type information.

Signature

```
global CustomMetadataTypeInfo(String cmtRecordId, String cmtSfResultCodeFieldName)
```

Parameters

`cmtRecordId`
Type: String

ID of the matchedcustom metadata type record

`cmtSfResultCodeFieldName`
Type: String

Field that contains the Salesforce result code values. Belongs to the custom metadata type.

CustomMetadataTypeInfo Methods

The following are methods for `CustomMetadataTypeInfo`.

GatewayErrorResponse Class

Use to respond with an error indication following errors from the `PaymentGateway` adapter, such as request-forbidden responses, custom validation errors, or expired API tokens.

Namespace

CommercePayments on page 297

Usage

Use `GatewayErrorResponse` to create an object that stores information about error responses sent by the payment gateway adapter.

Example

If `GatewayResponse` receives an exception rather than a valid request, it calls `GatewayErrorResponse` to create an error object with information about the exception.

```
global commercepayments.GatewayResponse processRequest(commercepayments.paymentGatewayContext
 gatewayContext) {
       commercepayments.RequestType requestType = gatewayContext.getPaymentRequestType();

        commercepayments.GatewayResponse response;
        try {
            if (requestType == commercepayments.RequestType.Authorize) {
                response =
createAuthResponse((commercepayments.AuthorizationRequest)gatewayContext.getPaymentRequest());

            } else if (requestType == commercepayments.RequestType.Capture) {
                response =
createCaptureResponse((commercepayments.CaptureRequest)gatewayContext.getPaymentRequest())
 ;
            } else if (requestType == commercepayments.RequestType.ReferencedRefund) {
                response =
createRefundResponse((commercepayments.ReferencedRefundRequest)gatewayContext.getPaymentRequest());

            }
            return response;
        } catch(SalesforceValidationException e) {
             commercepayments.GatewayErrorResponse error = new
commercepayments.GatewayErrorResponse('400', e.getMessage());
             return error;
        }
    }
```

IN THIS SECTION:

GatewayErrorResponse Constructors

GatewayErrorResponse Constructors

The following are constructors for `GatewayErrorResponse`.

IN THIS SECTION:

GatewayErrorResponse(errorCode, errorMessage)
Constructor to create a GatewayErrorResponse object that accepts `errorCode` and `errorMessage`.

`GatewayErrorResponse(errorCode, errorMessage)`

Constructor to create a GatewayErrorResponse object that accepts `errorCode` and `errorMessage`.

Signature

```
global GatewayErrorResponse(String errorCode, String errorMessage)
```

Parameters

`errorCode`
Type: String

Should match with the HTTP status code to be returned to the user. Here are a few examples.

- If the status code is for a bad request, the errorCode should be 400.
- If the status code is for a forbidden request, errorCode should be 403.
- If errorCode isn't a valid HTTP status code, a 500 internal server error is returned.

Note: *`errorCode`* must have a value, otherwise the platform throws an error.

`errorMessage`
Type: String

The message response to users following an error.

Note: *`errorMessage`* must have a value, otherwise the platform throws an error.

GatewayNotificationResponse Class

When the payment gateway sends a notification to the payments platform, the platform responds with a `GatewayNotificationResponse` indicating whether the platform succeeded or failed at receiving the notification.

Namespace

CommercePayments on page 297

Usage

You must specify the `CommercePayments` namespace when creating an instance of this class. The constructor of this class takes no arguments. For example:

```
CommercePayments.GatewayNotificationResponse gnr = new
CommercePayments.GatewayNotificationResponse();
```

When an asynchronous payment gateway sends a notification, the gateway requires the platform to acknowledge that it has either succeeded or failed in receiving the notification. Payment gateway adapters use this class to construct the acknowledgment response, which gateways expect for a notification. `GatewayNotificationResponse` is the return type of the `processNotification` method.

Example

```
commercepayments.GatewayNotificationResponse gnr = new
commercepayments.GatewayNotificationResponse();
if (saveResult.isSuccess()) {
    system.debug('Notification accepted by platform');
} else {
    system.debug('Errors in the result '+ Blob.valueOf(saveResult.getErrorMessage()));
}
gnr.setStatusCode(200);
gnr.setResponseBody(Blob.valueOf('[accepted]'));
return gnr;
```

IN THIS SECTION:

GatewayNotificationResponse Methods

GatewayNotificationResponse Methods

The following are methods for `GatewayNotificationResponse`.

IN THIS SECTION:

setResponseBody(responseBody)

Sets the body of the response to the gateway. Some gateways expect the payments platform to acknowledge the notification with a response regardless of whether the notification was accepted.

setStatusCode(statusCode)

Sets the HTTP status code sent to the gateway as part of the payments platform's response notification.

`setResponseBody(responseBody)`

Sets the body of the response to the gateway. Some gateways expect the payments platform to acknowledge the notification with a response regardless of whether the notification was accepted.

Signature

```
global void setResponseBody(Blob responseBody)
```

Parameters

`responseBody`
Type: Blob

Common response values include `accepted` for successfully receiving the response. For example:

```
commercepayments.GatewayNotificationResponse gnr = new
commercepayments.GatewayNotificationResponse();
if (saveResult.isSuccess()) {
    system.debug('Notification accepted by platform');
} else {
    system.debug('Errors in the result '+ Blob.valueOf(saveResult.getErrorMessage()));
}
gnr.setStatusCode(200);
gnr.setResponseBody(Blob.valueOf('[accepted]'));
return gnr;
```

Return Value

Type: void

setStatusCode(statusCode)

Sets the HTTP status code sent to the gateway as part of the payments platform's response notification.

Signature

```
global void setStatusCode(Integer statusCode)
```

Parameters

statusCode

Type: Integer

The status code will vary based on the type of payments platform response. Users should configure their `GatewayNotificationResponse` class to account for all values that their payments platform can possibly return. For example:

```
commercepayments.GatewayNotificationResponse gnr = new
commercepayments.GatewayNotificationResponse();
if (saveResult.isSuccess()) {
    system.debug('Notification accepted by platform');
} else {
    system.debug('Errors in the result '+ Blob.valueOf(saveResult.getErrorMessage()));
}
gnr.setStatusCode(200);
gnr.setResponseBody(Blob.valueOf('[accepted]'));
return gnr;
```

Return Value

Type: void

GatewayResponse Interface

Generic payment gateway response interface. This class extends the `CaptureResponse` on page 361, `AbstractTransactionResponse` on page 304, and `AbstractResponse` on page 300 classes and inherits all their properties. It has no unique methods or parameters.

Namespace

CommercePayments on page 297

IN THIS SECTION:

GatewayResponse Example Implementation

GatewayResponse Example Implementation

This is an example implementation of the `commercepayments.GatewayResponse` interface.

```
/**
     * Abstract function to build gateway response for a Transaction
     * The input is the response from gateway
     * It creates and returns GatewayResponse from the HttpResponse
     */
    public abstract commercepayments.GatewayResponse buildResponse(HttpResponse response);


    /**
     * Function to process transaction requests
     * Steps involved are:
     * 1. Build HttpRequest with the input Request from gateway context
     * 2. Send request and get the response from gateway
     * 3. Parse the response from gateway and return GatewayResponse
     */
    public commercepayments.GatewayResponse execute(){
                HttpRequest req;
        try{
            //Building a new request
            req = buildRequest();
        } catch(PayeezeValidationException e) {
            return getValidationExceptionError(e);
        }
        commercepayments.PaymentsHttp http = new commercepayments.PaymentsHttp();
        HttpResponse res = null;
        try{
            //Sending the request
            res = http.send(req);
        } catch(CalloutException ce) {
            return getCalloutExceptionError(ce);
        }
        try{
            //Parsing the response from gateway
            return buildResponse(res);
        } catch(Exception e) {
```

```
            return getParseExceptionError(e);
        }
    }
```

For additional context, review the complete Sample Gateway Adapter in the CommercePayments Gateway Reference Implementation.

NotificationClient Class

Communicates with the payment platform regarding the gateway's notification.

Namespace

CommercePayments on page 297

Usage

Specify the `CommercePayments` namespace when creating an instance of this class. The constructor of this class takes no arguments. For example:

```
CommercePayments.NotificationClient ntc = new CommercePayments.NotificationClient();
```

This class is used in asynchronous payment gateway adapters. The notification client contains API for communicating with the payments platform regarding the gateway's notification. When the gateway sends a notification, the gateway adapter invokes the `record` method in `NotificationClient` to request that the platform updates notification details.

Example

The `NotificationSaveResult` class creates a saveResult object to store the result of the save request made to the payment gateway.

```
commercepayments.NotificationSaveResult saveResult =
commercepayments.NotificationClient.record(notification);
```

IN THIS SECTION:

NotificationClient Methods

NotificationClient Methods

The following are methods for `NotificationClient`.

IN THIS SECTION:

record(notification)
Stores the results of a notification request.

`record(notification)`

Stores the results of a notification request.

Signature

```
global static commercepayments.NotificationSaveResult
record(commercepayments.BaseNotification notification)
```

Parameters

notification
Type: BaseNotification on page 347

Return Value

Type: NotificationSaveResult on page 388

NotificationSaveResult Class

Contains the result of the payment platform's attempt to record data from the gateway's notification.

Namespace

CommercePayments on page 297

Usage

This class is used with asynchronous payments. It is the return type of the `NotificiationClient.record` operation and contains the result of the payment platform's attempt to save notification details.

The constructor of this class takes no arguments. For example:

```
CommercePayments.NotificationSaveResult nsr = new
CommercePayments.NotificationSaveResult();
```

Example

```
commercepayments.NotificationSaveResult saveResult =
commercepayments.NotificationClient.record(notification);
```

IN THIS SECTION:

NotificationSaveResult Methods

NotificationSaveResult Methods

The following are methods for `NotificationSaveResult`.

IN THIS SECTION:

getErrorMessage()
Gets the error message, if any, from the payment platform regarding its attempt to save the notification sent from the payment gateway.

getStatusCode()
Gets the status code from the payment platform's attempt to save the notification sent from the payment gateway.

isSuccess()
Gets the status of whether the payment platform successfully saved the notification sent from the payment gateway.

`getErrorMessage()`

Gets the error message, if any, from the payment platform regarding its attempt to save the notification sent from the payment gateway.

Signature

```
global String getErrorMessage()
```

Return Value

Type: String

`getStatusCode()`

Gets the status code from the payment platform's attempt to save the notification sent from the payment gateway.

Signature

```
global Integer getStatusCode()
```

Return Value

Type: Integer

`isSuccess()`

Gets the status of whether the payment platform successfully saved the notification sent from the payment gateway.

Signature

```
global Boolean isSuccess()
```

Return Value

Type: Boolean

NotificationStatus Enum

Shows whether the payments platform successfully received the notification from the gateway.

Usage

When the gateway sends a notification for a payment request, the payments platform delegates the notification request to the gateway adapter. First, the adapter evaluates the signature from the notification request. If the signature is valid, the adapter builds a notification

object to store information about the notification. During this process, the adapter sets the `NotificationStatus` to `Failed` or `Success` based on information from the notification request.

Enum Values

The following are the values of the `commercepayments.NotificationStatus` enum.

Value	Description
`Failed`	The payments platform couldn't receive the notification due to an error.
`Success`	The payments platform received the notification.

PaymentGatewayAdapter Interface

`PaymentGatewayAdapters` can implement this interface in order to process requests.

Namespace

CommercePayments on page 297

IN THIS SECTION:

PaymentGatewayAdapter Methods

PaymentGatewayAdapter Methods

The following are methods for `PaymentGatewayAdapter`.

IN THIS SECTION:

processRequest(var1)

The entry point for processing payment requests. Returns the response from the payment gateway.

`processRequest(var1)`

The entry point for processing payment requests. Returns the response from the payment gateway.

Signature

```
global commercepayments.GatewayResponse
processRequest(commercepayments.PaymentGatewayContext var1)
```

Parameters

var1

Type: commercepayments.PaymentGatewayContext

You can retrieve the request type and the request from the Context object.

Return Value

Type: commercepayments.GatewayResponse

The response from the payment gateway.

PaymentGatewayAsyncAdapter Interface

Implement the interface to allow customers to process payments asynchronously.

Namespace

CommercePayments on page 297

Usage

Implementing an asynchronous adapter also requires the `processNotification` method from the GatewayNotificationResponse on page 383 class.

Example

```
global with sharing class SampleAsyncAdapter
        implements commercepayments.PaymentGatewayAsyncAdapter,
        commercepayments.PaymentGatewayAdapter {

        global SampleAsyncAdapter() {
        }

        global commercepayments.GatewayResponse processRequest(
        commercepayments.paymentGatewayContext gatewayContext) {
        }

        global commercepayments.GatewayNotificationResponse processNotification(
        commercepayments.PaymentGatewayNotificationContext gatewayNotificationContext) {
        }
        }
```

IN THIS SECTION:

PaymentGatewayAsyncAdapter Methods

PaymentGatewayAsyncAdapter Example Implementation

PaymentGatewayAsyncAdapter Methods

The following are methods for `PaymentGatewayAsyncAdapter`.

IN THIS SECTION:

processNotification(paymentGatewayNotificationContext)

Entry point for processing notifications from payment gateways.

processNotification(paymentGatewayNotificationContext)

Entry point for processing notifications from payment gateways.

Signature

```
global commercepayments.GatewayNotificationResponse
processNotification(commercepayments.PaymentGatewayNotificationContext var1)
```

Parameters

paymentGatewayNotificationContext
Type: PaymentGatewayNotificationContext on page 396

The `PaymentGatewayNotificationContext` object wraps all the information related to a gateway notification.

Return Value

Type: GatewayNotificationResponse on page 383

When the payment gateway sends a notification to the payments platform, the platform responds with a `GatewayNotificationResponse` indicating whether the platform succeeded or failed at receiving the notification.

PaymentGatewayAsyncAdapter Example Implementation

This is a sample implementation of the `commercepayments.PaymentGatewayAsyncAdapter` interface.

```
global with sharing class AdyenAdapter implements
commercepayments.PaymentGatewayAsyncAdapter, commercepayments.PaymentGatewayAdapter {
    global AdyenAdapter() {}

    global commercepayments.GatewayResponse
processRequest(commercepayments.paymentGatewayContext gatewayContext) {
    }

    global commercepayments.GatewayNotificationResponse
processNotification(commercepayments.PaymentGatewayNotificationContext
gatewayNotificationContext) {
    }
}

commercepayments.RequestType requestType = gatewayContext.getPaymentRequestType();
if (requestType == commercepayments.RequestType.Capture) {
    req.setEndpoint('/pal/servlet/Payment/v52/capture');
    body =
buildCaptureRequest((commercepayments.CaptureRequest)gatewayContext.getPaymentRequest());
} else if (requestType == commercepayments.RequestType.ReferencedRefund) {
    req.setEndpoint('/pal/servlet/Payment/v52/refund');
    body =
buildRefundRequest((commercepayments.ReferencedRefundRequest)gatewayContext.getPaymentRequest());
}

req.setBody(body);
req.setMethod('POST');
```

```
commercepayments.PaymentsHttp http = new commercepayments.PaymentsHttp();
HttpResponse res = null;
try {
    res = http.send(req);
} catch(CalloutException ce) {
    commercepayments.GatewayErrorResponse error = new
commercepayments.GatewayErrorResponse('500', ce.getMessage());
    return error;
}

if ( requestType == commercepayments.RequestType.Capture) {
    response =  createCaptureResponse(res);
} else if ( requestType == commercepayments.RequestType.ReferencedRefund) {
    response =  createRefundResponse(res);
}
return response;

commercepayments.PaymentGatewayNotificationRequest notificationRequest =
gatewayNotificationContext.getPaymentGatewayNotificationRequest();
Blob request = notificationRequest.getRequestBody();
Map<String, Object> jsonReq = (Map<String,
Object>)JSON.deserializeUntyped(request.toString());
List<Object> notificationItems = (List<Object>)jsonReq.get('notificationItems');
Map<String, Object> notificationRequestItem =
    (Map<String, Object>)((Map<String,
Object>)notificationItems[0]).get('NotificationRequestItem');
Boolean success = Boolean.valueOf(notificationRequestItem.get('success'));
String pspReference = (String)notificationRequestItem.get('pspReference');
String eventCode = (String)notificationRequestItem.get('eventCode');
Double amount = (Double)((Map<String,
Object>)notificationRequestItem.get('amount')).get('value');

commercepayments.NotificationStatus notificationStatus = null;
if (success) {
    notificationStatus = commercepayments.NotificationStatus.Success;
} else {
    notificationStatus = commercepayments.NotificationStatus.Failed;
}
commercepayments.BaseNotification notification = null;
if ('CAPTURE'.equals(eventCode)) {
    notification = new commercepayments.CaptureNotification();
} else if ('REFUND'.equals(eventCode)) {
    notification = new commercepayments.ReferencedRefundNotification();
}
notification.setStatus(notificationStatus);
notification.setGatewayReferenceNumber(pspReference);
notification.setAmount(amount);

commercepayments.NotificationSaveResult saveResult =
commercepayments.NotificationClient.record(notification);

commercepayments.GatewayNotificationResponse gnr = new
commercepayments.GatewayNotificationResponse();
if (saveResult.isSuccess()) {
```

```
    system.debug('Notification accepted by platform');
} else {
    system.debug('Errors in the result '+ Blob.valueOf(saveResult.getErrorMessage()));
}
gnr.setStatusCode(200);
gnr.setResponseBody(Blob.valueOf('[accepted]'));
return gnr;
```

PaymentGatewayContext Class

Wraps the information related to a payment request.

Namespace

CommercePayments on page 297

Usage

The constructor of this class takes no arguments. For example:

```
CommercePayments.PaymentGatewayContext pgc = new
CommercePayments.PaymentGatewayContext();
```

Example

```
global commercepayments.GatewayResponse processRequest(commercepayments.PaymentGatewayContext
 gatewayContext) {
    commercepayments.RequestType requestType = gatewayContext.getPaymentRequestType();
    if (requestType == commercepayments.RequestType.Capture) {
        commercepayments.CaptureRequest captureRequest = (commercepayments.CaptureRequest)
 gatewayContext.getPaymentRequest();
    }
}
```

IN THIS SECTION:

PaymentGatewayContext Constructors

PaymentGatewayContext Methods

PaymentGatewayContext Constructors

The following are constructors for `PaymentGatewayContext`.

IN THIS SECTION:

PaymentGatewayContext(request, requestType)

Constructor to enable instance creation. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

`PaymentGatewayContext(request, requestType)`

Constructor to enable instance creation. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Signature

```
global PaymentGatewayContext(commercepayments.PaymentGatewayRequest request, String requestType)
```

Parameters

request
Type: commercepayments.PaymentGatewayRequest

Raw payload. Sensitive attributes are masked to ensure PCI compliance.

requestType
Type: commercepayments.RequestType Enum

Defines the type of request made to the gateway

PaymentGatewayContext Methods

The following are methods for `PaymentGatewayContext`.

IN THIS SECTION:

getPaymentRequest()
Returns the payment request object.

getPaymentRequestType()
Returns the payment request type.

`getPaymentRequest()`

Returns the payment request object.

Signature

```
global commercepayments.PaymentGatewayRequest getPaymentRequest()
```

Return Value

Type: PaymentGatewayRequest

`getPaymentRequestType()`

Returns the payment request type.

Signature

```
global String getPaymentRequestType()
```

Return Value

Type: String

PaymentGatewayNotificationContext Class

Wraps the information related to a gateway notification.

Namespace

CommercePayments on page 297

Usage

This class is used with asynchronous payments. It wraps all of the information related to a notification from the payment gateway. The payments platform provides its context to the payment gateway adapters.

The constructor of this class takes no arguments. For example:

```
CommercePayments.PaymentGatewayNotificationContext pgnc = new
CommercePayments.PaymentGatewayNotificationContext();
```

Example

```
global commercepayments.GatewayNotificationResponse
processNotification(commercepayments.PaymentGatewayNotificationContext
gatewayNotificationContext) {
commercepayments.PaymentGatewayNotificationRequest notificationRequest =
gatewayNotificationContext.getPaymentGatewayNotificationRequest();
}
```

IN THIS SECTION:

PaymentGatewayNotificationContext Methods

PaymentGatewayNotificationContext Methods

The following are methods for `PaymentGatewayNotificationContext`.

IN THIS SECTION:

getPaymentGatewayNotificationRequest()

Returns the payment gateway's notification request.

`getPaymentGatewayNotificationRequest()`

Returns the payment gateway's notification request.

Signature

```
global commercepayments.PaymentGatewayNotificationRequest
getPaymentGatewayNotificationRequest()
```

Return Value

Type: PaymentGatewayNotificationRequest on page 397

PaymentGatewayNotificationRequest Class

Contains the notification request data from the gateway.

Namespace

CommercePayments on page 297

Usage

When the payment gateway sends a notification for a payment request, the payments platform sends the notification request to the gateway adapter. If the notification payload contains an `eventCode` of `CAPTURE`, the adapter constructs a `CaptureNotification`. If the notification payload contains an `eventCode` of `REFUND`, the adapter constructs a `ReferencedRefundNotification`. If the notification payload contains `eventCode` of `AUTHORISATION`, the adapter constructs a `GatewayNotificationResponse`.

You can obtain a notification request from `PaymentGatewayNotificationContext` on page 396 by invoking its `getPaymentGatewayNotificationRequest` method.

Example

```
global commercepayments.GatewayNotificationResponse
    processNotification(commercepayments.PaymentGatewayNotificationContext
gatewayNotificationContext) {
        commercepayments.PaymentGatewayNotificationRequest notificationRequest =
gatewayNotificationContext.getPaymentGatewayNotificationRequest();
}
```

IN THIS SECTION:

PaymentGatewayNotificationRequest Properties

PaymentGatewayNotificationRequest Methods

PaymentGatewayNotificationRequest Properties

The following are properties for `PaymentGatewayNotificationRequest`.

IN THIS SECTION:

requestBody

Body of the notification request sent by the payment gateway.

requestBody

Body of the notification request sent by the payment gateway.

Signature

```
global Blob requestBody {get; set;}
```

Property Value

Type: Blob

PaymentGatewayNotificationRequest Methods

The following are methods for `PaymentGatewayNotificationRequest`.

IN THIS SECTION:

getHeaders()
Gets HTTP headers from the notification request sent by the payment gateway.

getRequestBody()
Stores the notification request body information from the payment gateway's notification request.

getHeaders()

Gets HTTP headers from the notification request sent by the payment gateway.

Signature

```
global Map<String,String> getHeaders()
```

Return Value

Type: Map<String,String>

getRequestBody()

Stores the notification request body information from the payment gateway's notification request.

Signature

```
global Blob getRequestBody()
```

Return Value

Type: Blob

PaymentMethodDetailsResponse Class

This class contains the details about the payment method.

Namespace

CommercePayments

Example

```
commercepayments.AlternativePaymentMethodResponse alternativePaymentMethodResponse = new
commercepayments.AlternativePaymentMethodResponse();
alternativePaymentMethodResponse.setEmail('alternativePaymentMethod');
alternativePaymentMethodResponse.setEmail('foo@foo.com');
alternativePaymentMethodResponse.setGatewayToken('NMoPoIOnTZSaRaWcV7gUUXe');
alternativePaymentMethodResponse.setGatewayTokenDetails('gateway token details');
commercepayments.PaymentMethodDetailsResponse response = new
commercepayments.PaymentMethodDetailsResponse();
response.setAlternativePaymentMethod(alternativePaymentMethodResponse);
```

IN THIS SECTION:

PaymentMethodDetailsResponse Methods

PaymentMethodDetailsResponse Methods

The following are methods for `PaymentMethodDetailsResponse`.

IN THIS SECTION:

setAlternativePaymentMethod(alternativePaymentMethod)
Sets the alternative payment method details.

setCardPaymentMethod(cardPaymentMethod)
Sets the details about the card payment method.

`setAlternativePaymentMethod(alternativePaymentMethod)`

Sets the alternative payment method details.

Signature

```
public void setAlternativePaymentMethod(commercepayments.AlternativePaymentMethodResponse
alternativePaymentMethod)
```

Parameters

`alternativePaymentMethod`
Type: CommercePayments.AlternativePaymentMethodResponse

Details of the alternative payment method.

Return Value

Type: void

`setCardPaymentMethod(cardPaymentMethod)`

Sets the details about the card payment method.

Signature

```
public void setCardPaymentMethod(commercepayments.CardPaymentMethodResponse
cardPaymentMethod)
```

Parameters

`cardPaymentMethod`
Type: CommercePayments.CardPaymentMethodResponse

Details about the card payment method.

Return Value

Type: void

PaymentMethodTokenizationRequest Class

Stores data about a request to tokenize a card payment method. The tokenization process occurs in the payment gateway. This process replaces sensitive customer data, such as a card number or CVV, with unique identification symbols. The symbols are used while the data is handled by Salesforce, the payment gateway, and the customer bank, allowing Salesforce to store the token without storing sensitive customer data.

Namespace

CommercePayments on page 297

Usage

The constructor of this class takes no arguments. For example:

```
CommercePayments.PaymentMethodTokenizationRequest pmtr = new
CommercePayments.PaymentMethodTokenizationRequest();
```

This class holds all the required details about the tokenize request. Gateway adapters read the information in this class while constructing a tokenization JSON request, which is sent to the payment gateway.

Example

The following code is used within your payment gateway adapter Apex class.

Use the `GatewayResponse` class's `processRequest` method to build responses based on the request type that it receives from an instance of `PaymentGatewayContext on page 394`. If the request type is Tokenize, `GatewayResponse on page 386` calls the `createTokenizeResponse` method and passes an instance of the

`PaymentMethodTokenizationRequest` class. The passed `PaymentMethodTokenizationRequest` object contains the address and cardPaymentMethod information that the payment gateway needs to manage the tokenization process. For example:

```
global commercepayments.GatewayResponse processRequest(commercepayments.paymentGatewayContext
 gatewayContext) {
        commercepayments.RequestType requestType = gatewayContext.getPaymentRequestType();

         commercepayments.GatewayResponse response;
         try
         {
             if (requestType == commercepayments.RequestType.Tokenize) {
                     response =
createTokenizeResponse((commercepayments.PaymentMethodTokenizationRequest)gatewayContext.getPaymentRequest());

             }
             //Add other else if statements for different request types as needed.
             return response;
         }
         catch(SalesforceValidationException e)
         {
              commercepayments.GatewayErrorResponse error = new
commercepayments.GatewayErrorResponse('400', e.getMessage());
              return error;
         }
     }
```

Configure the `createTokenizeResponse` method to accept an instance of `PaymentMethodTokenizationRequest`. Then, build an instance of `PaymentMethodTokenizationResponse` based on the values received from the payment gateway.

```
     public commercepayments.GatewayResponse
createTokenizeResponse(commercepayments.PaymentMethodTokenizationRequest tokenizeRequest)
 {
         commercepayments.PaymentMethodTokenizationResponse tokenizeResponse = new
commercepayments.PaymentMethodTokenizationResponse();
         tokenizeResponse.setGatewayTokenEncrypted(encryptedValue);
         tokenizeResponse.setGatewayTokenDetails(tokenDetails);
         tokenizeResponse.setGatewayAvsCode(avsCode);
         tokenizeResponse.setGatewayMessage(gatewayMessage);
         tokenizeResponse.setGatewayResultCode(resultcode);
         tokenizeResponse.setGatewayResultCodeDescription(resultCodeDescription);
         tokenizeResponse.setSalesforceResultCodeInfo(resultCodeInfo);
         tokenizeResponse.setGatewayDate(system.now());
         return tokenizeResponse;
     }
```

The `tokenizeResponse` contains the results of the gateway's tokenization process, and if successful, the tokenized value.

IN THIS SECTION:

PaymentMethodTokenizationRequest Constructors

PaymentMethodTokenizationRequest Properties

PaymentMethodTokenizationRequest Methods

PaymentMethodTokenizationRequest Constructors

The following are constructors for `PaymentMethodTokenizationRequest`.

IN THIS SECTION:

PaymentMethodTokenizationRequest(paymentGatewayId)

Payment gateway ID constructor used with `paymentMethodTokenizationRequest`. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

PaymentMethodTokenizationRequest()

The following are constructors for `PaymentMethodTokenizationRequest`.

`PaymentMethodTokenizationRequest(paymentGatewayId)`

Payment gateway ID constructor used with `paymentMethodTokenizationRequest`. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Signature

```
global PaymentMethodTokenizationRequest(String paymentGatewayId)
```

Parameters

`paymentGatewayId`

Type: String

The payment method's payment gateway ID that will be tokenized.

`PaymentMethodTokenizationRequest()`

The following are constructors for `PaymentMethodTokenizationRequest`.

Signature

```
global PaymentMethodTokenizationRequest()
```

PaymentMethodTokenizationRequest Properties

The following are properties for `PaymentMethodTokenizationRequest`.

IN THIS SECTION:

address

The card payment method address to be tokenized.

cardPaymentMethod

The card payment method containing data to be tokenized.

`address`

The card payment method address to be tokenized.

Signature

```
global commercepayments.AddressRequest address {get; set;}
```

Property Value

Type: AddressRequest on page 308

cardPaymentMethod

The card payment method containing data to be tokenized.

Signature

```
global commercepayments.CardPaymentMethodRequest cardPaymentMethod {get; set;}
```

Property Value

Type: CardPaymentMethodRequest on page 366

PaymentMethodTokenizationRequest Methods

The following are methods for `PaymentMethodTokenizationRequest`.

IN THIS SECTION:

equals(obj)
Maintains the integrity of lists of type `PaymentMethodTokenizationRequest` by determining the equality of external objects in a list. This method is dynamic and is based on the equals method in Java.

hashCode()
Maintains the integrity of lists of type `PaymentMethodTokenizationRequest` by determining the uniquness of the external object records in a list.

toString()
Converts a date to a string.

equals(obj)

Maintains the integrity of lists of type `PaymentMethodTokenizationRequest` by determining the equality of external objects in a list. This method is dynamic and is based on the equals method in Java.

Signature

```
global Boolean equals(Object obj)
```

Parameters

obj
Type: Object

External object whose key is to be validated.

Return Value

Type: Boolean

`hashCode()`

Maintains the integrity of lists of type `PaymentMethodTokenizationRequest` by determining the uniquness of the external object records in a list.

Signature

```
global Integer hashCode()
```

Return Value

Type: Integer

`toString()`

Converts a date to a string.

Signature

```
global String toString()
```

Return Value

Type: String

PaymentMethodTokenizationResponse Class

Gateway response sent by payment gateway adapters for the payment method tokenization request. The response includes the payment method's token ID value.

Namespace

CommercePayments on page 297

Usage

The constructor of this class takes no arguments. For example:

```
CommercePayments.PaymentMethodTokenizationResponse pmtr = new
CommercePayments.PaymentMethodTokenizationResponse();
```

After the payment gateway processes a tokenization request, the fields of `PaymentMethodTokenizationResponse` receive and store information from the gateway's response. The gateway's response shows whether the tokenization request was successful, the token value, and any additional messages or information about the tokenization process. You can then pass an instance of `PaymentMethodTokenizationResponse` to an authorization response or a sale response. This class is mapped to a response class in the Java layer.

Example

This constructor builds a new instance of the `PaymentMethodTokenizationResponse` class.

```
commercepayments.PaymentMethodTokenizationResponse tokenizeResponse = new
commercepayments.PaymentMethodTokenizationResponse();
```

`PaymentMethodTokenizationResponse` contains only setter methods. Each setter accepts a value from the payment gateway and use it to set an attribute of `PaymentMethodTokenizationResponse`.

The most important method in `PaymentMethodTokenizationResponse` is `setGatewayTokenEncrypted`, which uses Salesforce encryption to set an encrypted token value for a payment method. The `setGatewayTokenEncrypted` method is available in Salesforce API v52.0 and later. We recommend using it to ensure your tokenized payment method values are encrypted and secure. While the `setGatewayToken` method (available in earlier API versions) also returns a payment method token, the tokenized value isn't encrypted.

If the instantiated class already has a gateway token, `setGatewayTokenEncrypted` throws an error.

```
    /** @description Method to set Gateway token to persist in Encrypted Text */
    global void setGatewayTokenEncrypted(String gatewayTokenEncrypted) {
        if (gatewayTokenSet)  {
            throwTokenError();
        }
        this.delegate.setGatewayTokenEncrypted(gatewayTokenEncrypted);
        gatewayTokenEncryptedSet = true;
    }
```

A typical instantiation of `PaymentMethodTokenizationResponse` sets the encrypted gateway token alongside the other tokenization response values sent by the gateway.

```
    public commercepayments.GatewayResponse
createTokenizeResponse(commercepayments.PaymentMethodTokenizationRequest tokenizeRequest)
 {
        commercepayments.PaymentMethodTokenizationResponse tokenizeResponse = new
commercepayments.PaymentMethodTokenizationResponse();
        tokenizeResponse.setGatewayTokenEncrypted(gatewayTokenEncrypted);
        tokenizeResponse.setGatewayTokenDetails(gatewayTokenDetails);
        tokenizeResponse.setGatewayAvsCode(gatewayAvsCode);
        tokenizeResponse.setGatewayMessage(gatewayMessage);
        tokenizeResponse.setGatewayResultCode(gatewayResultCode);
        tokenizeResponse.setGatewayResultCodeDescription(gatewayResultCodeDescription);
       tokenizeResponse.setSalesforceResultCodeInfo(SUCCESS_SALESFORCE_RESULT_CODE_INFO);

        tokenizeResponse.setGatewayDate(system.now());
        return tokenizeResponse;
    }
```

After you've built a PaymentMethodTokenizationResponse object and set the encrypted gateway token, pass the object to the `setPaymentMethodTokenizationResponse` method of an authorization response or a sale response.

Authorization Response

```
    public commercepayments.GatewayResponse
createAuthResponse(commercepayments.AuthorizationRequest authRequest) {
        commercepayments.AuthorizationResponse authResponse = new
commercepayments.AuthorizationResponse();
        commercepayments.PaymentMethodTokenizationResponse
```

```
paymentMethodTokenizationResponse = new
commercepayments.PaymentMethodTokenizationResponse();
         if(authRequest.amount!=null )
         {
             authResponse.setAmount(authRequest.amount);
         }
         else
         {
            throw new SalesforceValidationException('Required Field Missing : Amount');

         }

         authResponse.setGatewayResultCode('00');
         authResponse.setGatewayResultCodeDescription('Transaction Normal');
         authResponse.setGatewayAuthCode('SF'+getRandomNumber(6));
         authResponse.setGatewayReferenceNumber(getRandomNumber(10));
        authResponse.setSalesforceResultCodeInfo(SUCCESS_SALESFORCE_RESULT_CODE_INFO);

         authResponse.setGatewayDate(system.now());

paymentMethodTokenizationResponse.setGatewayTokenEncrypted(gatewayTokenEncrypted);

authResponse.setPaymentMethodTokenizationResponse(paymentMethodTokenizationResponse);
         return authResponse;
     }
```

Sale Response

```
     public commercepayments.GatewayResponse
createSaleResponse(commercepayments.SaleRequest saleRequest) {
        commercepayments.SaleResponse saleResponse = new commercepayments.SaleResponse();

         commercepayments.PaymentMethodTokenizationResponse
paymentMethodTokenizationResponse = new
commercepayments.PaymentMethodTokenizationResponse();
         if(saleRequest.amount!=null )
         {
             saleResponse.setAmount(saleRequest.amount);
         }
         else
         {
            throw new SalesforceValidationException('Required Field Missing : Amount');

         }

         system.debug('Response - success');
         saleResponse.setGatewayDate(system.now());
         saleResponse.setGatewayResultCode('00');
         saleResponse.setGatewayResultCodeDescription('Transaction Normal');
         saleResponse.setGatewayReferenceNumber('SF'+getRandomNumber(6));
        saleResponse.setSalesforceResultCodeInfo(SUCCESS_SALESFORCE_RESULT_CODE_INFO);


paymentMethodTokenizationResponse.setGatewayTokenEncrypted(gatewayTokenEncrypted);
```

```
saleResponse.setPaymentMethodTokenizationResponse(paymentMethodTokenizationResponse);
        return saleResponse;
    }
```

IN THIS SECTION:

PaymentMethodTokenizationResponse Methods

PaymentMethodTokenizationResponse Methods

The following are methods for `PaymentMethodTokenizationResponse`.

IN THIS SECTION:

setGatewayAvsCode(gatewayAvsCode)
Sets the AVS (address verification system) result code information that the gateway returned. Maximum length of 64 characters.

setGatewayDate(gatewayDate)
Sets the date that the tokenization occurred. Some gateways don't send this value.

setGatewayMessage(gatewayMessage)
Sets error messages that the gateway returned for the tokenization request. Maximum length of 255 characters.

setGatewayResultCode(gatewayResultCode)
Sets a gateway-specific result code. The code may be mapped to a Salesforce-specific result code. Maximum length of 64 characters.

setGatewayResultCodeDescription(gatewayResultCodeDescription)
Sets a description of the gateway-specific result code that a payment gateway returned. Maximum length of 1000 characters.

setGatewayToken(gatewayToken)
Sets the gateway token value that the gateway returned.

setGatewayTokenDetails(gatewayTokenDetails)
Sets any additional information that the gateway returned about the payment token.

setGatewayTokenEncrypted(gatewayTokenEncrypted)
Sets the value of the `gatewayTokenEncrypted` field on a CardPaymentMethod or DigitalWallet object.

setSalesforceResultCodeInfo(salesforceResultCodeInfo)
Sets the Salesforce-specific result code information. Payment gateways have many response codes for payment calls. Salesforce uses the result code information to map payment gateway codes to a predefined set of standard Salesforce result codes.

`setGatewayAvsCode(gatewayAvsCode)`

Sets the AVS (address verification system) result code information that the gateway returned. Maximum length of 64 characters.

Signature

```
global void setGatewayAvsCode(String gatewayAvsCode)
```

Parameters

`gatewayAvsCode`
Type: String

Used to verify the address mapped to a payment method when the payments platform requests tokenization from the payment gateway.

Return Value

Type: void

`setGatewayDate(gatewayDate)`

Sets the date that the tokenization occurred. Some gateways don't send this value.

Signature

```
global void setGatewayDate(Datetime gatewayDate)
```

Parameters

gatewayDate
Type: Datetime

Return Value

Type: void

`setGatewayMessage(gatewayMessage)`

Sets error messages that the gateway returned for the tokenization request. Maximum length of 255 characters.

Signature

```
global void setGatewayMessage(String gatewayMessage)
```

Parameters

gatewayMessage
Type: String

Return Value

Type: void

`setGatewayResultCode(gatewayResultCode)`

Sets a gateway-specific result code. The code may be mapped to a Salesforce-specific result code. Maximum length of 64 characters.

Signature

```
global void setGatewayResultCode(String gatewayResultCode)
```

Parameters

gatewayResultCode
Type: String

Gateway-specific result code. Must be used to map a Salesforce-specific result code.

Return Value

Type: void

setGatewayResultCodeDescription(gatewayResultCodeDescription)

Sets a description of the gateway-specific result code that a payment gateway returned. Maximum length of 1000 characters.

Signature

```
global void setGatewayResultCodeDescription(String gatewayResultCodeDescription)
```

Parameters

gatewayResultCodeDescription
Type: String

Provides additional information about the result code and why the gateway returned the specific code. Descriptions will vary between different gateways.

Return Value

Type: void

setGatewayToken(gatewayToken)

Sets the gateway token value that the gateway returned.

Signature

```
global void setGatewayToken(String gatewayToken)
```

Parameters

gatewayToken
Type: String

The gateway token that the payment gateway sends following a tokenization request.

For the CardPaymentMethod and DigitalWallet objects, use the *gatewyTokenEncrypted* parameter, which encrypts the token value.

Return Value

Type: void

setGatewayTokenDetails(gatewayTokenDetails)

Sets any additional information that the gateway returned about the payment token.

Signature

```
global void setGatewayTokenDetails(String gatewayTokenDetails)
```

Parameters

gatewayTokenDetails
Type: String

Return Value

Type: void

setGatewayTokenEncrypted(gatewayTokenEncrypted)

Sets the value of the `gatewayTokenEncrypted` field on a CardPaymentMethod or DigitalWallet object.

Signature

```
global void setGatewayTokenEncrypted(String gatewayTokenEncrypted)
```

Parameters

gatewayTokenEncrypted
Type: String

The gateway token that the payment gateway sends following a tokenization request. Salesforce Payments uses Salesforce encryption to encrypt the token value.

Return Value

Type: void

setSalesforceResultCodeInfo(salesforceResultCodeInfo)

Sets the Salesforce-specific result code information. Payment gateways have many response codes for payment calls. Salesforce uses the result code information to map payment gateway codes to a predefined set of standard Salesforce result codes.

Signature

```
global void setSalesforceResultCodeInfo(commercepayments.SalesforceResultCodeInfo
salesforceResultCodeInfo)
```

Parameters

salesforceResultCodeInfo
Type: SalesforceResultCodeInfo on page 446

Description of the Salesforce result code value.

Return Value

Type: void

PaymentsHttp Class

Makes an HTTP request to start the interaction with the payment gateway.

Namespace

CommercePayments on page 297

Usage

You must specify the `CommercePayments` namespace when creating an instance of this class. The constructor of this class takes no arguments. For example:

```
CommercePayments.PaymentsHttp payhttp = new CommercePayments.PaymentsHttp();
```

IN THIS SECTION:

PaymentsHttp Methods

PaymentsHttp Constructors

PaymentsHttp Methods

The following are methods for `PaymentsHttp`. All methods are instance methods.

IN THIS SECTION:

send(Request)
Sends an HttpRequest and returns the response.

`send(Request)`

Sends an HttpRequest and returns the response.

Signature

```
global HttpResponse send(HttpRequest request)
```

Parameters

`request`
Type: System.HttpRequest

Return Value

Type: System.HttpResponse

PaymentsHttp Constructors

The following are constructors for `PaymentsHttp`.

IN THIS SECTION:

PaymentsHttp()
Initiates an HTTP request and response.

`PaymentsHttp()`

Initiates an HTTP request and response.

Signature

```
global PaymentsHttp()
```

PostAuthApiPaymentMethodRequest Class

Sends information about a payment method to a gateway adapter during a postauthorization service call.

Namespace

CommercePayments

Usage

Contains information about the payment method that is used for a postauthorization request. It contains all available payment methods as fields, but populates only one field for each request. The gateway adapter uses this class when constructing a postauthorization request. An object of this class is available through the `paymentMethod` field on the `PostAuthorizationRequest Class` object.

IN THIS SECTION:

PostAuthApiPaymentMethodRequest Constructors
Lists the constructors for the PostAuthApiPaymentMethodRequest.

PostAuthApiPaymentMethodRequest Properties
Lists the properties for PostAuthApiPaymentMethodRequest.

PostAuthApiPaymentMethodRequest Constructors

Lists the constructors for the PostAuthApiPaymentMethodRequest.

The following are constructors for `PostAuthApiPaymentMethodRequest`.

IN THIS SECTION:

PostAuthApiPaymentMethodRequest(cardPaymentMethodRequest)
Constructs a sample `cardPaymentMethodRequest`. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

PostAuthApiPaymentMethodRequest(AlternativePaymentMethodRequest)
Constructs a sample `alternativePaymentMethodRequest`. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

PostAuthApiPaymentMethodRequest()
Constructor for `PostAuthApiPaymentMethodRequest`.

PostAuthApiPaymentMethodRequest(cardPaymentMethodRequest)

Constructs a sample `cardPaymentMethodRequest`. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Signature

```
global PostAuthApiPaymentMethodRequest(commercepayments.CardPaymentMethodRequest
cardPaymentMethodRequest)
```

Parameters

cardPaymentMethodRequest
Type: commercepayments.CardPaymentMethodRequest on page 366

Contains information about the card payment method. Used to send information to a gateway adapter during a service call.

PostAuthApiPaymentMethodRequest(AlternativePaymentMethodRequest)

Constructs a sample `alternativePaymentMethodRequest`. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Signature

```
global
PostAuthApiPaymentMethodRequest(commercepayments.AlternativePaymentMethodRequestPaymentMethodRequest)
```

Parameters

alternativePaymentMethodRequest
Type: commercepayments.AlternativePaymentMethodRequest on page 366

Contains information about the alternative payment method. Used to send information to a gateway adapter during a service call.

PostAuthApiPaymentMethodRequest()

Constructor for `PostAuthApiPaymentMethodRequest`.

Signature

```
global PostAuthApiPaymentMethodRequest()
```

PostAuthApiPaymentMethodRequest Properties

Lists the properties for PostAuthApiPaymentMethodRequest.

The following are properties for `PostAuthApiPaymentMethodRequest`.

IN THIS SECTION:

cardPaymentMethod
The card payment method object used in a postauthorizaiton payment method request.

alternativePaymentMethod
The alternative payment method object used in a postauthorizaiton payment method request.

`cardPaymentMethod`

The card payment method object used in a postauthorizaiton payment method request.

Signature

```
global commercepayments.CardPaymentMethodRequest cardPaymentMethod {get; set;}
```

Property Value

Type: commercepayments.CardPaymentMethodRequest on page 366

`alternativePaymentMethod`

The alternative payment method object used in a postauthorizaiton payment method request.

Signature

```
global commercepayments.AlternativePaymentMethodRequest PaymentMethod {get; set;}
```

Property Value

Type: commercepayments.alternativePaymentMethodRequest

PostAuthorizationRequest Class

Sends information about a postauthorization request to a gateway adapter during a service call.

Namespace

CommercePayments

Usage

This class extends `BaseRequest` and contains information about a transaction postauthorization request. The gateway adapter reads fields from this class to validate the client-side transaction with the payment gateway. An object of this class is available by calling `getPaymentRequest()` in the `PaymentGatewayContext Class`).

```
((commercepayments.PostAuthorizationRequest)gatewayContext.getPaymentRequest());
```

IN THIS SECTION:

PostAuthorizationRequest Constructors
Lists the constructors for postauthorization requests.

PostAuthorizationRequest Properties
Lists properties for a postauthorizaiton request.

PostAuthorizationRequest Constructors

Lists the constructors for postauthorization requests.

The following are constructors for `PostAuthorizationRequest`.

IN THIS SECTION:

PostAuthorizationRequest(amount)
Constructor for building the amount in a postauthorization request. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

`PostAuthorizationRequest(amount)`

Constructor for building the amount in a postauthorization request. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Signature

```
global PostAuthorizationRequest(Double amount)
```

Parameters

amount
Type: Double

The amount of the authorization.

PostAuthorizationRequest Properties

Lists properties for a postauthorizaiton request.

The following are properties for a `PostAuthorizationRequest`.

IN THIS SECTION:

accountId
The customer account that is settled when the postauthorization is performed.

amount
The total amount of the postauthorization request.

comments
Comments about the postauthorization. Users can enter comments to provide additional information.

currencyIsoCode
The ISO currency code for the postauthorization request.

paymentMethod
The payment method used to process the postauthorization request.

accountId

The customer account that is settled when the postauthorization is performed.

Signature

```
global String accountId {get; set;}
```

Property Value

Type: String

amount

The total amount of the postauthorization request.

Signature

```
global Double amount {get; set;}
```

Property Value

Type: Double

comments

Comments about the postauthorization. Users can enter comments to provide additional information.

Signature

```
global String comments {get; set;}
```

Property Value

Type: String

currencyIsoCode

The ISO currency code for the postauthorization request.

Signature

```
global String currencyIsoCode {get; set;}
```

Property Value

Type: String

paymentMethod

The payment method used to process the postauthorization request.

Signature

```
global PostAuthApiPaymentMethodRequest paymentMethod {get; set;}
```

Property Value

Type: AuthApiPaymentMethodRequest on page 321

PostAuthorizationResponse Class

Response sent by the payment gateway adapter for a postauthorization service.

Namespace

CommercePayments

Usage

This class extends AbstractTransactionResponse. The constructor of this class takes no arguments. For example:

```
CommercePayments.PostAuthorizationResponse authr = new
CommercePayments.PostAuthorizationResponse();
```

Contains information about the payment gateway's response following an authorization transaction. The gateway adapter uses the payment gateway's response to populate the `PostAuthorizationResponse` fields. The payments platform uses the information from this class to settle the transaction.

IN THIS SECTION:

PostAuthorizationResponse Methods
Lists the methods for the `PostAuthorizationResponse`.

PostAuthorizationResponse Methods

Lists the methods for the `PostAuthorizationResponse`.

The following are methods for `PostAuthorizationResponse`.

IN THIS SECTION:

setPaymentMethodDetailsResponse(PaymentMethodDetailsResponsepaymentMethodDetails)
Sets details from the gateway about the authorized payment method.

setCardPaymentMethodResponse(CardPaymentMethodResponsecardPaymentMethod)
Sets details from the gateway about the authorized card payment method.

setAlternativePaymentMethodResponse(AlternativePaymentMethodResponsealternativePaymentMethod)
Sets details from the gateway about the authorized alternative payment method.

setPaymentMethodTokenizationResponse(paymentMethodTokenizationResponse)
Sets information from the gateway about the tokenized payment method.

setGatewayAuthCode(gatewayAuthCode)
Sets the authorization code that the gateway returned. Maximum length of 64 characters.

setAuthorizationExpirationDate(authExpDate)
Sets the expiration date of the authorization request.

setAsync(async)
Sets whether the payment capture or authorization is asynchronous (`True`) or synchronous (`False`). If `True`, then the payment or payment authorization record created has a status of `Pending`.

`setPaymentMethodDetailsResponse(PaymentMethodDetailsResponsepaymentMethodDetails)`

Sets details from the gateway about the authorized payment method.

Signature

```
global void setPaymentMethodDetailsResponse(commercepayments.PaymentMethodDetailsResponse
paymentMethodDetailsResponse)
```

Parameters

`paymentMethodDetailsResponse`
Gateway response sent by payment gateway adapters for the payment method details request.

Return Value

Type: void

`setCardPaymentMethodResponse(CardPaymentMethodResponsecardPaymentMethod)`

Sets details from the gateway about the authorized card payment method.

Signature

```
global void setCardPaymentMethodResponse(commercepaymentsCardPaymentMethodResponse
cardpaymentMethodResponse)
```

Parameters

cardPaymentMethodResponse
Gateway response sent by payment gateway adapter for the card payment method request.

Return Value

Type: void

setAlternativePaymentMethodResponse(AlternativePaymentMethodResponsealternativePaymentMethod)

Sets details from the gateway about the authorized alternative payment method.

Signature

```
global void
setAlternativePaymentMethodResponse(commercepayments.AlternativePaymentMethodResponse
paymentMethodResponse)
```

Parameters

alternativePaymentMethodResponse
Gateway response sent by payment gateway adapter for the alternative payment method request.

Return Value

Type: void

setPaymentMethodTokenizationResponse(paymentMethodTokenizationResponse)

Sets information from the gateway about the tokenized payment method.

Signature

```
global void
setPaymentMethodTokenizationResponse(commercepayments.PaymentMethodTokenizationResponse
paymentMethodTokenizationResponse)
```

Parameters

paymentMethodTokenizationResponse
PaymentMethodTokenizationResponse on page 404

Gateway response sent by payment gateway adapters for the payment method tokenization request.

Return Value

Type: void

setGatewayAuthCode(gatewayAuthCode)

Sets the authorization code that the gateway returned. Maximum length of 64 characters.

Signature

```
global void setGatewayAuthCode(String gatewayAuthCode)
```

Parameters

gatewayAuthCode
Type: String

The authorization code returned by the gateway.

Return Value

Type: void

setAuthorizationExpirationDate(authExpDate)

Sets the expiration date of the authorization request.

Signature

```
global void setAuthorizationExpirationDate(Datetime authExpDate)
```

Parameters

authExpDate
Type: Datetime

Return Value

Type: void

setAsync(async)

Sets whether the payment capture or authorization is asynchronous (`True`) or synchronous (`False`). If `True`, then the payment or payment authorization record created has a status of `Pending`.

Signature

```
global void setAsync(Boolean async)
```

Parameters

async
Type: Boolean

Return Value

Type: void

ReferencedRefundNotification Class

When a payment gateway sends a notification for a refund transaction, the payment gateway adapter creates the `ReferencedRefundNotification` object to store information about notification.

Namespace

CommercePayments on page 297

Usage

This class is used with asynchronous payments. When a payment gateway sends a notification for a refund transcation, the gateway adapter creates an object of type `ReferencedRefundNotification` to populate the respective values.

The constructor of this class takes no arguments. For example:

```
CommercePayments.ReferencedRefundNotification rrn = new
CommercePayments.ReferencedRefundNotification();
```

Example

```
commercepayments.NotificationStatus notificationStatus = null;
        if (success) {
            notificationStatus = commercepayments.NotificationStatus.Success;
        } else {
            notificationStatus = commercepayments.NotificationStatus.Failed;
        }
        commercepayments.BaseNotification notification = null;
        if ('CAPTURE'.equals(eventCode)) {
            notification = new commercepayments.CaptureNotification();
        } else if ('REFUND'.equals(eventCode)) {
            notification = new commercepayments.ReferencedRefundNotification();
        }
```

IN THIS SECTION:

ReferencedRefundNotification Methods

ReferencedRefundNotification Methods

The following are methods for `ReferencedRefundNotification`.

IN THIS SECTION:

setAmount(amount)
Sets the transaction amount. Can be positive, negative, or zero.

setGatewayDate(gatewayDate)
Sets the date that communication for the refund notification occurred with the payment gateway.

setGatewayMessage(gatewayMessage)
Sets information or messages that the gateway returned.

setGatewayReferenceDetails(gatewayReferenceDetails)
Sets the payment gateway's reference details.

setGatewayReferenceNumber(gatewayReferenceNumber)
Sets the payment gateway's reference number.

setGatewayResultCode(gatewayResultCode)
Sets the payment gateway's result code.

setGatewayResultCodeDescription(gatewayResultCodeDescription)
Sets the payment gateway's result code description.

setId(id)
Sets the ID of a notification sent by the payment gateway.

setSalesforceResultCodeInfo(salesforceResultCodeInfo)
Sets Salesforce result code information.

setStatus(status)
Sets the notification status value on the notification object.

`setAmount(amount)`

Sets the transaction amount. Can be positive, negative, or zero.

Signature

```
global void setAmount(Double amount)
```

Parameters

amount
Type: Double

The amount to be debited or captured.

Return Value

Type: void

`setGatewayDate(gatewayDate)`

Sets the date that communication for the refund notification occurred with the payment gateway.

Signature

```
global void setGatewayDate(Datetime gatewayDate)
```

Parameters

gatewayDate
Type: Datetime

The date that communication happened with the gateway.

Return Value

Type: void

`setGatewayMessage(gatewayMessage)`

Sets information or messages that the gateway returned.

Signature

```
global void setGatewayMessage(String gatewayMessage)
```

Parameters

gatewayMessage
Type: String

Return Value

Type: void

`setGatewayReferenceDetails(gatewayReferenceDetails)`

Sets the payment gateway's reference details.

Signature

```
global void setGatewayReferenceDetails(String gatewayReferenceDetails)
```

Parameters

gatewayReferenceDetails
Type: String

Provides information about the gateway communication.

Return Value

Type: void

`setGatewayReferenceNumber(gatewayReferenceNumber)`

Sets the payment gateway's reference number.

Signature

```
global void setGatewayReferenceNumber(String gatewayReferenceNumber)
```

Parameters

gatewayReferenceNumber
Type: String

Unique transaction ID created by the payment gateway.

Return Value

Type: void

setGatewayResultCode(gatewayResultCode)

Sets the payment gateway's result code.

Signature

```
global void setGatewayResultCode(String gatewayResultCode)
```

Parameters

gatewayResultCode
Type: String

The gateway result code. You must map this to a Salesforce-specific result code.

Return Value

Type: void

setGatewayResultCodeDescription(gatewayResultCodeDescription)

Sets the payment gateway's result code description.

Signature

```
global void setGatewayResultCodeDescription(String gatewayResultCodeDescription)
```

Parameters

gatewayResultCodeDescription
Type: String

Description of the gateway result code. Provides additional context about the result code .

Return Value

Type: void

setId(id)

Sets the ID of a notification sent by the payment gateway.

Signature

```
global void setId(String id)
```

Parameters

id
Type: String

Return Value

Type: void

setSalesforceResultCodeInfo(salesforceResultCodeInfo)

Sets Salesforce result code information.

Signature

```
global void setSalesforceResultCodeInfo(commercepayments.SalesforceResultCodeInfo
salesforceResultCodeInfo)
```

Parameters

salesforceResultCodeInfo
Type: SalesforceResultCodeInfo on page 446

Description of the Salesforce result code value.

Return Value

Type: void

setStatus(status)

Sets the notification status value on the notification object.

Signature

```
global void setStatus(commercepayments.NotificationStatus status)
```

Parameters

status
Type: NotificationStatus on page 389

Indicates whether the payments platform successfully received the notification from the payment gateway.

Return Value

Type: void

ReferencedRefundRequest

Access information about the referenced refund requests. Extends the `RefundRequest` class.

Namespace

CommercePayments on page 297

Example

```
global commercepayments.GatewayResponse processRequest(commercepayments.PaymentGatewayContext
 gatewayContext) {
    commercepayments.RequestType requestType = gatewayContext.getPaymentRequestType();
    if (requestType == commercepayments.RequestType.ReferencedRefund) {
        commercepayments.*ReferencedRefundRequest* refundRequest =
(commercepayments.*ReferencedRefundRequest*) gatewayContext.getPaymentRequest();
    }
}
```

IN THIS SECTION:

ReferencedRefundRequest Constructors

ReferencedRefundRequest Properties

ReferencedRefundRequest Methods

ReferencedRefundRequest Constructors

The following are constructors for `ReferencedRefundRequest`.

IN THIS SECTION:

ReferencedRefundRequest(amount, paymentId)
This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

`ReferencedRefundRequest(amount, paymentId)`

This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Parameters

`amount`
Type: Double

The amount to be debited or captured.

`paymentId`
Type: String

The payment record.

ReferencedRefundRequest Properties

The following are properties for `ReferencedRefundRequest`.

IN THIS SECTION:

PaymentId
References a payment object.

accountId
References an account.

amount
References an amount.

`PaymentId`

References a payment object.

Property Value

Type: String

`accountId`

References an account.

Property Value

Type: String

`amount`

References an amount.

Property Value

Type: Double

ReferencedRefundRequest Methods

The following are methods for `ReferencedRefundRequest`.

ReferencedRefundResponse Class

The payment gateway adapter sends this response for the `ReferencedRefund` request type.

Namespace

CommercePayments on page 297

Usage

The constructor of this class takes no arguments. For example:

```
CommercePayments.ReferencedRefundResponse refr = new
CommercePayments.ReferencedRefundResponse();
```

IN THIS SECTION:

ReferencedRefundResponse Methods

ReferencedRefundResponse Methods

The following are methods for `ReferencedRefundResponse`.

IN THIS SECTION:

setAmount(amount)
Sets the transaction amount. The value must be a postive number.

setGatewayAvsCode(gatewayAvsCode)
Sets the payment gateway's address verification system (AVS) code.

setGatewayDate(gatewayDate)
Sets the payment gateway's date.

setGatewayMessage(gatewayMessage)
Sets information or messages that the gateway returned.

setGatewayReferenceDetails(gatewayReferenceDetails)
Sets the payment gateway's reference details.

setGatewayReferenceNumber(gatewayReferenceNumber)
Sets the payment gateway's reference number.

setGatewayResultCode(gatewayResultCode)
Sets the payment gateway's result code.

setGatewayResultCodeDescription(gatewayResultCodeDescription)
Sets the payment gateway's result code description.

setSalesforceResultCodeInfo(salesforceResultCodeInfo)
Set the Salesforce result code info.

`setAmount(amount)`

Sets the transaction amount. The value must be a postive number.

Signature

```
global void setAmount(Double amount)
```

Parameters

`amount`
Type: Double

The amount to be debited or captured.

Return Value

Type: void

setGatewayAvsCode(gatewayAvsCode)

Sets the payment gateway's address verification system (AVS) code.

Signature

```
global void setGatewayAvsCode(String gatewayAvsCode)
```

Parameters

gatewayAvsCode
Type: String

Code sent by gateways that use an address verification system.

Return Value

Type: void

setGatewayDate(gatewayDate)

Sets the payment gateway's date.

Signature

```
global void setGatewayDate(Datetime gatewayDate)
```

Parameters

gatewayDate
Type: Datetime

Date and time of the gateway communication.

Return Value

Type: void

setGatewayMessage(gatewayMessage)

Sets information or messages that the gateway returned.

Signature

```
global void setGatewayMessage(String gatewayMessage)
```

Parameters

gatewayMessage
Type: String

Information or error messages returned by the gateway.

Return Value

Type: void

setGatewayReferenceDetails(gatewayReferenceDetails)

Sets the payment gateway's reference details.

Signature

```
global void setGatewayReferenceDetails(String gatewayReferenceDetails)
```

Parameters

gatewayReferenceDetails
Type: String

Information about the gateway communication.

Return Value

Type: void

setGatewayReferenceNumber(gatewayReferenceNumber)

Sets the payment gateway's reference number.

Signature

```
global void setGatewayReferenceNumber(String gatewayReferenceNumber)
```

Parameters

gatewayReferenceNumber
Type: String

Unique transaction ID created by the payment gateway.

Return Value

Type: void

setGatewayResultCode(gatewayResultCode)

Sets the payment gateway's result code.

Signature

```
global void setGatewayResultCode(String gatewayResultCode)
```

Parameters

gatewayResultCode
Type: String

The gateway result code. Must be mapped to a Salesforce result code.

Return Value

Type: void

setGatewayResultCodeDescription(gatewayResultCodeDescription)

Sets the payment gateway's result code description.

Signature

```
global void setGatewayResultCodeDescription(String gatewayResultCodeDescription)
```

Parameters

gatewayResultCodeDescription
Type: String

Description of the `GatewayResultCode`. Provides more information about the result code returned by the gateway.

Return Value

Type: void

setSalesforceResultCodeInfo(salesforceResultCodeInfo)

Set the Salesforce result code info.

Signature

```
global void setSalesforceResultCodeInfo(commercepayments.SalesforceResultCodeInfo
salesforceResultCodeInfo)
```

Parameters

salesforceResultCodeInfo
Type: commercepayments.SalesforceResultCodeInfo on page 446

Describes the Salesforce result code value.

Return Value

Type: void

RefundRequest Class

Sends data related to a refund to the payment gateway adapter.

Namespace

CommercePayments on page 432

Usage

The constructor of this class takes no arguments. For example:

```
CommercePayments.RefundRequest rrq = new CommercePayments.RefundRequest();
```

Example

```
commercepayments.ReferencedRefundRequest refundRequest = new
commercepayments.ReferencedRefundRequest(80, pmt.id);
```

IN THIS SECTION:

RefundRequest Methods

RefundRequest Methods

The following are methods for `RefundRequest`.

IN THIS SECTION:

equals(obj)
Maintains the integrity of lists of type `RefundRequest` by determining the equality of external objects in a list. This method is dynamic and is based on the equals method in Java.

hashCode()
Maintains the integrity of lists of type `RefundRequest` by determining the uniqueness of the external object records in a list.

`equals(obj)`

Maintains the integrity of lists of type `RefundRequest` by determining the equality of external objects in a list. This method is dynamic and is based on the equals method in Java.

Signature

```
global Boolean equals(Object obj)
```

Parameters

obj
Type: Object

Return Value

Type: Boolean

`hashCode()`

Maintains the integrity of lists of type `RefundRequest` by determining the uniqueness of the external object records in a list.

Signature

```
global Integer hashCode()
```

Return Value

Type: Integer

RequestType Enum

Defines the type of payment transaction request made to the payment gateway.

Enum Values

The following are the values of the `commercepayments.RequestType` enum.

Value	Description
`Authorize`	Payment authorization request
`PostAuth`	Post authorization request
`Capture`	Payment capture request
`AuthorizationReversal`	Authorization Reversal request
`ReferencedRefund`	Payment refund request
`Sale` `commercepayments.RequestType, Sale`	Sale request
`Tokenize` `commercepayments.RequestType, Tokenize`	Payment tokenize request

SaleApiPaymentMethodRequest Class

Sends data related to a card payment method to a gateway adapter during a sale service call.

Namespace

CommercePayments on page 297

Usage

This class holds information about a payment method that was used for a Sale request. `SaleApiPaymentMethodRequest` contains all the possible payment methods as fields, but only one value is populated for a given request. Gateway adapters use this class when constructing a sale request. The object of this class is obtained through the `paymentMethod` field on the `SaleRequest` object.

Example: This code sample retrieves the `SaleApiPaymentMethodRequest` object from the `SaleRequest` class.

```
commercepayments.SaleApiPaymentMethodRequest paymentMethod = saleRequest.paymentMethod;
```

IN THIS SECTION:

SaleApiPaymentMethodRequest Constructors

SaleApiPaymentMethodRequest Properties

SaleApiPaymentMethodRequest Methods

SaleApiPaymentMethodRequest Constructors

The following are constructors for `SaleApiPaymentMethodRequest`.

IN THIS SECTION:

SaleApiPaymentMethodRequest(cardPaymentMethodRequest)

Sends data related to a card payment method to a gateway adapter during a sale service call.

SaleApiPaymentMethodRequest()

Constructor for building a sale payment method request. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

`SaleApiPaymentMethodRequest(cardPaymentMethodRequest)`

Sends data related to a card payment method to a gateway adapter during a sale service call.

Signature

```
global SaleApiPaymentMethodRequest(commercepayments.CardPaymentMethodRequest
cardPaymentMethodRequest)
```

Parameters

`cardPaymentMethodRequest`

Type: CardPaymentMethodRequest on page 366

SaleApiPaymentMethodRequest()

Constructor for building a sale payment method request. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Signature

```
global SaleApiPaymentMethodRequest()
```

SaleApiPaymentMethodRequest Properties

The following are properties for `SaleApiPaymentMethodRequest`.

IN THIS SECTION:

cardPaymentMethod
Contains details of the card used in a payment method.

cardPaymentMethod

Contains details of the card used in a payment method.

Signature

```
global commercepayments.CardPaymentMethodRequest cardPaymentMethod {get; set;}
```

Property Value

Type: CardPaymentMethodRequest on page 366

SaleApiPaymentMethodRequest Methods

The following are methods for `SaleApiPaymentMethodRequest`.

IN THIS SECTION:

equals(obj)
Maintains the integrity of lists of type `SaleApiPaymentMethodRequest` by determining the equality of external objects in a list. This method is dynamic and is based on the equals method in Java.

hashCode()
Maintains the integrity of lists of type `SaleApiPaymentMethodRequest` by determining the uniqueness of the external object records in a list.

toString()
Converts a date to a string.

equals(obj)

Maintains the integrity of lists of type `SaleApiPaymentMethodRequest` by determining the equality of external objects in a list. This method is dynamic and is based on the equals method in Java.

Signature

```
global Boolean equals(Object obj)
```

Parameters

obj
Type: Object

Return Value

Type: Boolean

hashCode()

Maintains the integrity of lists of type `SaleApiPaymentMethodRequest` by determining the uniqueness of the external object records in a list.

Signature

```
global Integer hashCode()
```

Return Value

Type: Integer

toString()

Converts a date to a string.

Signature

```
global String toString()
```

Return Value

Type: String

SaleRequest Class

Stores information about a sales request.

Namespace

CommercePayments on page 297

Usage

This class holds all the required details about a sale request. Gateway adapters read the fields of this class object while constructing a sale JSON request thatis sent to the payment gateway. The object of this class is made available through `commercepayments.paymentGatewayContext` by calling `getPaymentRequest()`.

Example

This code sample retrieves the SaleRequest object from the `PaymentGatewayContext` class.

```
commercepayments.SaleRequest =
(commercepayments.SaleRequest)gatewayContext.getPaymentRequest()
```

IN THIS SECTION:

SaleRequest Constructors

SaleRequest Properties

SaleRequest Methods

SaleRequest Constructors

The following are constructors for `SaleRequest`.

IN THIS SECTION:

SaleRequest(amount)

Constructor for defining an amount for the sale request. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

`SaleRequest(amount)`

Constructor for defining an amount for the sale request. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Signature

```
global SaleRequest(Double amount)
```

Parameters

amount

Type: Double

Amount of the sale request.

SaleRequest Properties

The following are properties for `SaleRequest`.

IN THIS SECTION:

accountId
Customer account ID for the sale request.

amount
Amount of the sale request. Can be positive only.

comments
Additional information about the sale request.

currencyIsoCode
Currency code for the sale request.

paymentMethod
Payment method used in the sale request.

accountId

Customer account ID for the sale request.

Signature

```
global String accountId {get; set;}
```

Property Value

Type: String

amount

Amount of the sale request. Can be positive only.

Signature

```
global Double amount {get; set;}
```

Property Value

Type: Double

comments

Additional information about the sale request.

Signature

```
global String comments {get; set;}
```

Property Value

Type: String

currencyIsoCode

Currency code for the sale request.

Signature

```
global String currencyIsoCode {get; set;}
```

Property Value

Type: String

paymentMethod

Payment method used in the sale request.

Signature

```
global commercepayments.SaleApiPaymentMethodRequest paymentMethod {get; set;}
```

Property Value

Type: SaleApiPaymentMethodRequest on page 433

SaleRequest Methods

The following are methods for `SaleRequest`.

IN THIS SECTION:

equals(obj)
Compares this object with the specified object and returns `true` if both objects are equal; otherwise, returns `false`.

hashCode()
Maintains the integrity of lists of type `SaleRequest` by determining the uniqueness of the external object records in a list.

toString()
Converts a date to a string.

equals(obj)

Compares this object with the specified object and returns `true` if both objects are equal; otherwise, returns `false`.

Signature

```
global Boolean equals(Object obj)
```

Parameters

obj
Type: Object

Return Value

Type: Boolean

`hashCode()`

Maintains the integrity of lists of type `SaleRequest` by determining the uniqueness of the external object records in a list.

Signature

```
global Integer hashCode()
```

Return Value

Type: Integer

`toString()`

Converts a date to a string.

Signature

```
global String toString()
```

Return Value

Type: String

SaleResponse Class

Response sent by payment gateway adapters for a sales service.

Namespace

CommercePayments on page 297

Usage

The constructor of this class takes no arguments. For example:

```
CommercePayments.SaleResponse slr CommercePayments.SaleResponse();
```

This class contains details about a customer card that was used as a payment method for Authorization, Sale, or Tokenization request. The gateway adapter reads the fields of this class while constructing a transaction JSON request, which it then sends to the payment gateway. The object of this class is made available by the `cardPaymentMethod` field in `SaleApiPaymentMethodRequest on page 433` and `AuthApiPaymentMethodRequest on page 321`.

Example

This code sample builds a `SaleResponse` object.

```
commercepayments.SaleResponse saleResponse = new commercepayments.SaleResponse();
saleResponse.setGatewayReferenceDetails("refDetailString");
saleResponse.setGatewayResultCode("res_code");
saleResponse.setGatewayResultCodeDescription("");
saleResponse.setGatewayReferenceNumber("");
saleResponse.setSalesforceResultCodeInfo(getSalesforceResultCodeInfo(commercepayments.SalesforceResultCode.SUCCESS.name()));
```

IN THIS SECTION:

SaleResponse Methods

SaleResponse Methods

The following are methods for `SaleResponse`.

IN THIS SECTION:

setAmount(amount)
Sets the transaction amount. Must be a non-negative value.

setGatewayAvsCode(gatewayAvsCode)
Sets the AVS (address verification system) result code information that the gateway returned. Maximum length of 64 characters.

setGatewayDate(gatewayDate)
Sets the date that the sale occurred. Some gateways don't send this value.

setGatewayMessage(gatewayMessage)
Sets error messages that the gateway returned for the sale request. Maximum length of 255 characters.

setGatewayReferenceDetails(gatewayReferenceDetails)
Sets additional data that you can use for subsequent sales. You can use any data that isn't normalized in financial entities. This field has a maximum length of 1000 characters and can store data as JSON or XML.

setGatewayReferenceNumber(gatewayReferenceNumber)
Sets the unique gateway reference number for the transaction that the gateway returned. Maximum length of 255 characters.

setGatewayResultCode(gatewayResultCode)
Sets a gateway-specific result code. The code may be mapped to a Salesforce-specific result code. Maximum length of 64 characters.

setGatewayResultCodeDescription(gatewayResultCodeDescription)
Sets a description of the gateway-specific result code that a payment gateway returned. Maximum length of 1000 characters.

setPaymentMethodTokenizationResponse(paymentMethodTokenizationResponse)
Sets information from the gateway about the tokenized payment method.

setSalesforceResultCodeInfo(salesforceResultCodeInfo)
Sets the Salesforce-specific result code information. Payment gateways have many response codes for payment calls. Salesforce uses the result code information to map payment gateway codes to a predefined set of standard Salesforce result codes.

`setAmount(amount)`

Sets the transaction amount. Must be a non-negative value.

Signature

```
global void setAmount(Double amount)
```

Parameters

amount
Type: Double

The amount of the transaction.

Return Value

Type: void

`setGatewayAvsCode(gatewayAvsCode)`

Sets the AVS (address verification system) result code information that the gateway returned. Maximum length of 64 characters.

Signature

```
global void setGatewayAvsCode(String gatewayAvsCode)
```

Parameters

gatewayAvsCode
Type: String

Used to verify the address mapped to a payment method when the payments platform requests tokenization from the payment gateway.

Return Value

Type: void

`setGatewayDate(gatewayDate)`

Sets the date that the sale occurred. Some gateways don't send this value.

Signature

```
global void setGatewayDate(Datetime gatewayDate)
```

Parameters

gatewayDate
Type: Datetime

Return Value

Type: void

`setGatewayMessage(gatewayMessage)`

Sets error messages that the gateway returned for the sale request. Maximum length of 255 characters.

Signature

```
global void setGatewayMessage(String gatewayMessage)
```

Parameters

gatewayMessage
Type: String

Return Value

Type: void

`setGatewayReferenceDetails(gatewayReferenceDetails)`

Sets additional data that you can use for subsequent sales. You can use any data that isn't normalized in financial entities. This field has a maximum length of 1000 characters and can store data as JSON or XML.

Signature

```
global void setGatewayReferenceDetails(String gatewayReferenceDetails)
```

Parameters

gatewayReferenceDetails
Type: String

Return Value

Type: void

`setGatewayReferenceNumber(gatewayReferenceNumber)`

Sets the unique gateway reference number for the transaction that the gateway returned. Maximum length of 255 characters.

Signature

```
global void setGatewayReferenceNumber(String gatewayReferenceNumber)
```

Parameters

gatewayReferenceNumber
Type: String

Unique authorization ID created by the payment gateway.

Return Value

Type: void

setGatewayResultCode(gatewayResultCode)

Sets a gateway-specific result code. The code may be mapped to a Salesforce-specific result code. Maximum length of 64 characters.

Signature

```
global void setGatewayResultCode(String gatewayResultCode)
```

Parameters

gatewayResultCode
Type: String

Gateway-specific result code. Must be used to map a Salesforce-specific result code.

Return Value

Type: void

setGatewayResultCodeDescription(gatewayResultCodeDescription)

Sets a description of the gateway-specific result code that a payment gateway returned. Maximum length of 1000 characters.

Signature

```
global void setGatewayResultCodeDescription(String gatewayResultCodeDescription)
```

Parameters

gatewayResultCodeDescription
Type: String

Description of the gateway's result code. Use this field to learn more about why the gateway returned a certain result code.

Return Value

Type: void

setPaymentMethodTokenizationResponse(paymentMethodTokenizationResponse)

Sets information from the gateway about the tokenized payment method.

Signature

```
global void
setPaymentMethodTokenizationResponse(commercepayments.PaymentMethodTokenizationResponse
paymentMethodTokenizationResponse)
```

Parameters

paymentMethodTokenizationResponse
Type: PaymentMethodTokenizationResponse on page 404

Gateway response sent by payment gateway adapters for the payment method tokenization request. The response includes the payment method's token ID value.

Return Value

Type: void

setSalesforceResultCodeInfo(salesforceResultCodeInfo)

Sets the Salesforce-specific result code information. Payment gateways have many response codes for payment calls. Salesforce uses the result code information to map payment gateway codes to a predefined set of standard Salesforce result codes.

Signature

```
global void setSalesforceResultCodeInfo(commercepayments.SalesforceResultCodeInfo
salesforceResultCodeInfo)
```

Parameters

salesforceResultCodeInfo
Type: SalesforceResultCodeInfo on page 446

Sets the Salesforce-specific result code information. Payment gateways have many response codes for payment calls. Salesforce uses the result code information to map payment gateway codes to a predefined set of standard Salesforce result codes.

Return Value

Type: void

SalesforceResultCode Enum

Defines the gateway call status values in Salesforce based on the call status values that the payment gateway returned.

Usage

Payment gateways can return many different responses. Salesforce maps these responses into one of seven possible Salesforce response values.

Enum Values

The following are the values of the `commercepayments.SalesforceResultCode` enum.

Value	Description
`Decline`	The gateway call failed, but it may still work if you try again. For example, the customer had insufficient funds or briefly lost their connection to the internet. This is also known as a "soft decline."
`Indeterminate`	The gateway didn't respond to the call and the user has to check the transaction request's status. Indeterminate responses often occur following server timeouts, system failure, or any action that interrupts the gateway's ability to process the payment.
`PermanentFail`	The customer's bank recognized the payment account as closed, terminated, or fraudulent. The gateway won't further calls from the payment method associate with the transaction. After a permanent fail response, the transaction changes its gateway status to Permanent Fail.
`RequiresReview`	The gateway call initially failed, but the payment method may still work after further evaluation. This response often happens when the customer bank requires more information about the payment request. In this case, the bank provides an authorization code manually when the payment manager calls the processor.
`Success`	The gateway processed the transaction successfully.
`SystemError`	Salesforce ended the payment request call before receiving a gateway response. System error responses often occur due to gateway server errors, invalid customer credentials, or anytime the request times out before receiving a gateway response. The failure occurs before the request reaches the gateway, so there's no risk of an unaccounted payment remaining in the gateway. You can continue with the transaction by manually creating a payment.
`ValidationError`	The gateway received incorrect customer payment information, such as misspelled credit card names or a CVV with missing numbers.

SalesforceResultCodeInfo

Stores Salesforce result code information from payment gateway adapters.

Namespace

CommercePayments on page 297

Usage

The constructor of this class takes no arguments. For example:

```
CommercePayments.SalesforceResultCodeInfo srci = new
CommercePayments.SalesforceResultCodeInfo();
```

Gateways can return the transaction result as either `CustomMetadata` or `SalesforceResultCode`.

IN THIS SECTION:

SalesforceResultCodeInfo Constructors

SalesforceResultCodeInfo Constructors

The following are constructors for `SalesforceResultCodeInfo`.

IN THIS SECTION:

SalesforceResultCodeInfo(customMetadataTypeInfo)
Constructor for providing the `customMetadataTypeInfo` for the result of the transaction.

SalesforceResultCodeInfo(salesforceResultCode)
Constructor that provides the `salesforceResultCode` for the transaction result.

`SalesforceResultCodeInfo(customMetadataTypeInfo)`

Constructor for providing the `customMetadataTypeInfo` for the result of the transaction.

Signature

```
global SalesforceResultCodeInfo(commercepayments.CustomMetadataTypeInfo
customMetadataTypeInfo)
```

Parameters

`customMetadataTypeInfo`
Type: CustomMetadataTypeInfo on page 381

Information about the metadata type.

`SalesforceResultCodeInfo(salesforceResultCode)`

Constructor that provides the `salesforceResultCode` for the transaction result.

Signature

```
global SalesforceResultCodeInfo(commercepayments.SalesforceResultCode
salesforceResultCode)
```

Parameters

`salesforceResultCode`
Type: SalesforceResultCode on page 445

The enum value for the result code.

CommerceTax Namespace

Manage the communication between Salesforce and an external tax engine.

The `CommerceTax` namespace includes these classes.

IN THIS SECTION:

AbstractTransactionResponse Class

Abstract class that contains methods for setting tax fields based on the external tax provider's response. Response classes that extend `AbstractTransactionResponse` inherit these methods.

AddressesResponse Class

Sets the tax address fields based on a response from the external tax engine. Contains setter methods for the Ship From, Ship To, and Sold To addresses.

AddressResponse Class

Contains a location code sent from the external tax engine.

AmountDetailsResponse Class

Sets tax amount fields based on a response from the external tax engine.

CalculateTaxRequest Class

Represents a request to an external tax engine to calculate tax. Extends the TaxTransactionRequest class and is the top-level request class.

CalculateTaxResponse Class

Sets the values of the tax transaction following a response from the external tax engine. Extends the AbstractTransactionResponse class and is the top-level response class.

CalculateTaxType Enum

Shows whether a tax calculation request is for estimated or actual tax.

ErrorResponse Class

Use to respond with an error after receiving errors from the PaymentGatewayAdapter methods of the CommercePayments namespace, such as request-forbidden responses, custom validation errors, or expired API tokens.

HeaderTaxAddressesRequest Class

Captures the address values that are applicable for the quote or order transaction.

ImpositionResponse Class

Stores details of tax impositions from the external tax engine.

JurisdictionResponse Class

Stores details from the external tax engine about the tax jurisdiction used in the tax calculation process. A tax jurisdiction represents a government entity that collects tax.

LineItemResponse Class

Response class that stores details of a list of one or more line items on which the tax engine has calculated tax.

LineTaxAddressesRequest Class

Stores details of the addresses applied per line item in a tax calculation request.

RequestType Enum

Shows the type of tax request made to the tax engine.

ResultCode Enum

Code that represents the results of a tax request made to the tax engine.

RuleDetailsResponse Class

Contains details about the tax rules used for tax calculation.

TaxAddressesRequest Class

Contains methods to get and set tax address values.

TaxAddressRequest Class

Contains address details used for tax calculation.

TaxApiException Class

Contains details about any exceptions during the tax calculation process. Extends the `ApexBaseException` class.

TaxCustomerDetailsRequest Class

Contains customer details used in tax calculation.

TaxDetailsResponse Class

Stores details of the tax values that an external tax engine calculates in response to a tax calculation request.

TaxEngineAdapter Interface

Retrieves information from the tax engine and evaluates the information to define tax details.

TaxEngineContext Class

Wrapper class that stores details about the type of a tax calculation request.

TaxLineItemRequest Class

Contains line item details of a tax request.

TaxSellerDetailsRequest Class

Contains tax code details used in the tax calculation request.

TaxTransactionRequest Class

Abstract class for storing customer details used in tax calculation and estimation requests.

TaxTransactionStatus Enum

Shows whether the tax transaction has been committed or uncommitted.

TaxTransactionType Enum

Shows whether the tax transaction is for a credit or debit transaction.

AbstractTransactionResponse Class

Abstract class that contains methods for setting tax fields based on the external tax provider's response. Response classes that extend `AbstractTransactionResponse` inherit these methods.

Namespace

CommerceTax

IN THIS SECTION:

AbstractTransactionResponse Methods

Learn more about the methods for AbstractTransactionResponse class.

AbstractTransactionResponse Methods

Learn more about the methods for AbstractTransactionResponse class.

The `AbstractTransactionResponse` class includes these methods.

IN THIS SECTION:

setAddresses(addresses)
Uses an instance of `AddressesResponse` to set the values of tax address fields.

setAmountDetails(amountDetails)
Uses an instance of `AmountDetailsResponse` to set tax amount fields such as exemption amount and tax amount.

setCurrencyIsoCode(currencyIsoCode)
Sets the currencyIsoCode field.

setDescription(dscptn)
Sets the Description field.

setDocumentCode(documentCode)
Sets the DocumentCode field. Document codes are often used to reference tax documents that the external tax engine uses in the tax calculation process. Document code acts as a unique link to chain-related transactions, such as amendment or refunds.

setEffectiveDate(effectiveDate)
Sets the EffectiveDate field. Effective Date fields are optional fields that store the date that a transaction takes effect. We provide these fields only for recordkeeping purposes – for example, if you must report an effective date to an external general ledger system. Salesforce doesn't use them to calculate any tax or payment values.

setLineItems(lineItems)
Uses an instance of the `LineItemResponse` class to set a list of line items. Each line item represents an item sent to an external tax engine for tax calculation.

setReferenceDocumentCode(referenceDocumentCode)
Sets the ReferenceDocumentCode field. Use this field to store the code of an additional document used in the tax calculation process. For example, use this field in case of a refund for a previously taxed purchase.

setReferenceEntityId(referenceEntityId)
Sets the ID of a reference entity. In Commerce Tax, a reference entity represents a record related to the items sent to the external tax engine for tax calculation. For example, if you sent order items for tax calculation, you could define the parent order as the reference entity.

setTaxTransactionId(taxTrxnId)
Sets the TaxTransactionId field using the ID of a tax transaction record. In Commerce Tax, a tax transaction record stores information about a specific tax calculation process.

setTransactionDate(transactionDate)
Sets the TransactionDate field.

`setAddresses(addresses)`

Uses an instance of `AddressesResponse` to set the values of tax address fields.

Signature

```
global void setAddresses(commercetax.AddressesResponse addresses)
```

Parameters

`addresses`
Type: AddressesResponse

Class that contains methods to set the Ship To, Ship From, and Sold To address information.

Return Value

Type: void

setAmountDetails(amountDetails)

Uses an instance of `AmountDetailsResponse` to set tax amount fields such as exemption amount and tax amount.

Signature

```
global void setAmountDetails(commercetax.AmountDetailsResponse amountDetails)
```

Parameters

amountDetails
Type: AmountDetailsResponse

Class that contains methods to set the tax exemption amount, tax amount, total amount, and total amount with tax.

Return Value

Type: void

setCurrencyIsoCode(currencyIsoCode)

Sets the currencyIsoCode field.

Signature

```
global void setCurrencyIsoCode(String currencyIsoCode)
```

Parameters

currencyIsoCode
Type: String

Three-letter ISO 4217 currency code associated with a tax object.

Return Value

Type: void

setDescription(dscptn)

Sets the Description field.

Signature

```
global void setDescription(String dscptn)
```

Parameters

dscptn
Type: String

Optional field for providing additional information about a record.

Return Value

Type: void

setDocumentCode(documentCode)

Sets the DocumentCode field. Document codes are often used to reference tax documents that the external tax engine uses in the tax calculation process. Document code acts as a unique link to chain-related transactions, such as amendment or refunds.

Signature

```
global void setDocumentCode(String documentCode)
```

Parameters

documentCode
Type: String

Code for a tax document used in the tax calculation process.

Return Value

Type: void

setEffectiveDate(effectiveDate)

Sets the EffectiveDate field. Effective Date fields are optional fields that store the date that a transaction takes effect. We provide these fields only for recordkeeping purposes – for example, if you must report an effective date to an external general ledger system. Salesforce doesn't use them to calculate any tax or payment values.

Signature

```
global void setEffectiveDate(Datetime effectiveDate)
```

Parameters

effectiveDate
Type: Datetime

Optional field that stores the date that a transaction takes effect.

Return Value

Type: void

`setLineItems(lineItems)`

Uses an instance of the `LineItemResponse` class to set a list of line items. Each line item represents an item sent to an external tax engine for tax calculation.

Signature

```
global void setLineItems(List<commercetax.LineItemResponse> lineItems)
```

Parameters

`lineItems`
Type: List<LineItemResponse>

A list of line items sent to an external tax engine for tax calculation.

Return Value

Type: void

`setReferenceDocumentCode(referenceDocumentCode)`

Sets the ReferenceDocumentCode field. Use this field to store the code of an additional document used in the tax calculation process. For example, use this field in case of a refund for a previously taxed purchase.

Signature

```
global void setReferenceDocumentCode(String referenceDocumentCode)
```

Parameters

`referenceDocumentCode`
Type: String

The code for a document used in the tax calculation process.

Return Value

Type: void

`setReferenceEntityId(referenceEntityId)`

Sets the ID of a reference entity. In Commerce Tax, a reference entity represents a record related to the items sent to the external tax engine for tax calculation. For example, if you sent order items for tax calculation, you could define the parent order as the reference entity.

Signature

```
global void setReferenceEntityId(String referenceEntityId)
```

Parameters

referenceEntityId
Type: String

ID of a record related to the items sent for tax calculation.

Return Value

Type: void

setTaxTransactionId(taxTrxnId)

Sets the TaxTransactionId field using the ID of a tax transaction record. In Commerce Tax, a tax transaction record stores information about a specific tax calculation process.

Signature

```
global void setTaxTransactionId(String taxTrxnId)
```

Parameters

taxTrxnId
Type: String

The ID of a tax transaction record in Commerce Tax.

Return Value

Type: void

setTransactionDate(transactionDate)

Sets the TransactionDate field.

Signature

```
global void setTransactionDate(Datetime transactionDate)
```

Parameters

transactionDate
Type: Datetime

Date that a tax transaction occurred.

Return Value

Type: void

AddressesResponse Class

Sets the tax address fields based on a response from the external tax engine. Contains setter methods for the Ship From, Ship To, and Sold To addresses.

Namespace

CommerceTax

Usage

Because `AddressesResponse` contains multiple addresses, we recommend using multiple instances of `AddressResponse` to set unique values for each address.

Example

This code sample represents a portion of the code used in a mock tax adapter. In this example, you create three `AddressResponse` classes, set their location codes, and pass them to the `Ship To`, `Ship From`, and `Sold To` setter methods in `AddressesResponse`. In an actual implementation, your `AddressResponse` classes already have a location code based on the response from the external tax engine.

```
commercetax.AddressesResponse addressesRes = new commercetax.AddressesResponse();

//AddressResponse containing ShipTo information
commercetax.AddressResponse shipToAddress = new commercetax.AddressResponse();
shipToAddress.setLocationCode('1234567');

//AddressResponse containing ShipFrom information
commercetax.AddressResponse shipFromAddress = new commercetax.AddressResponse();
shipFromAddress.setLocationCode('84720385');

//AddressResponse containing Sold To information
commercetax.AddressResponse soldToAddress = new commercetax.AddressResponse();
soldToAddress.setLocationCode('92381749');

//set values of addressesRes
addressesRes.setShipFrom(shipFromAddress);
addressesRes.setShipTo(shipToAddress);
addressesRes.setSoldTo(soldToAddress);
```

IN THIS SECTION:

AddressesResponse Methods
Learn more about the methods for AddressesResponse class.

AddressesResponse Methods

Learn more about the methods for AddressesResponse class.

The `AddressesResponse` class includes these methods.

IN THIS SECTION:

setShipFrom(shipFrom)
Sets the value of a ShipFrom address field.

setShipTo(shipTo)
Sets the value of a ShipTo address field.

setSoldTo(soldTo)
Sets the value of a SoldTo address field.

setShipFrom(shipFrom)

Sets the value of a ShipFrom address field.

Signature

```
global void setShipFrom(commercetax.AddressResponse shipFrom)
```

Parameters

shipFrom
Type: AddressResponse

A single address. Use this generic address parameter to store any type of address, such as Ship From, Ship To, and Sold To details. Users set the specific address in an `AddressResponse` instance and then pass that instance to the `AddressesResponse`'s `setShipTo()`, `setShipFrom()`, and `setSoldTo()` methods as needed.

Return Value

Type: void

setShipTo(shipTo)

Sets the value of a ShipTo address field.

Signature

```
global void setShipTo(commercetax.AddressResponse shipTo)
```

Parameters

shipTo
Type: AddressResponse

Stores a single address. This is a generic address parameter and can be used to store any type of address, such as Ship From, Ship To, and Sold To details. Users set the specific address in an `AddressResponse` instance and then pass that instance to the `AddressesResponse`'s `setShipTo()`, `setShipFrom()`, and `setSoldTo()` methods as needed.

Return Value

Type: void

`setSoldTo(soldTo)`

Sets the value of a SoldTo address field.

Signature

```
global void setSoldTo(commercetax.AddressResponse soldTo)
```

Parameters

`soldTo`
Type: AddressResponse

Stores a single address. This is a generic address parameter and can be used to store any type of address, such as Ship From, Ship To, Sold To details. Users set the specific address in an AddressResponse instance and then pass that instance to the `AddressesResponse`'s `setShipTo()`, `setShipFrom()`, and `setSoldTo()` methods as needed.

Return Value

Type: void

AddressResponse Class

Contains a location code sent from the external tax engine.

Namespace

CommerceTax

Usage

Use the `AddressResponse` class to set unique values for each address.

```
commercetax.AddressesResponse addressesRes = new commercetax.AddressesResponse();

//AddressResponse containing ShipTo information
commercetax.AddressResponse shipToAddress = new commercetax.AddressResponse();
shipToAddress.setLocationCode('1234567');

//AddressResponse containing ShipFrom information
commercetax.AddressResponse shipFromAddress = new commercetax.AddressResponse();
shipFromAddress.setLocationCode('84720385');

//AddressResponse containing Sold To information
commercetax.AddressResponse soldToAddress = new commercetax.AddressResponse();
soldToAddress.setLocationCode('92381749');

//set values of addressesRes
addressesRes.setShipFrom(shipFromAddress);
addressesRes.setShipTo(shipToAddress);
addressesRes.setSoldTo(soldToAddress);
```

IN THIS SECTION:

AddressResponse Methods
Learn more about the available methods with the `AddressResponse` class.

AddressResponse Methods

Learn more about the available methods with the `AddressResponse` class.

The `AddressResponse` class includes these methods.

IN THIS SECTION:

setLocationCode(locationCode)
Sets the value of a LocationCode field.

`setLocationCode(locationCode)`

Sets the value of a LocationCode field.

Signature

```
global void setLocationCode(String locationCode)
```

Parameters

`locationCode`
Type: String

A code that contains address information. This value can be passed to a method that sets the value of an address field.

Return Value

Type: void

AmountDetailsResponse Class

Sets tax amount fields based on a response from the external tax engine.

Namespace

CommerceTax

Example

In this example, an instance of `AmountDetailsResponse` class in a mock adapter calculates several tax amount fields. The `totalTax` and `totalAmount` parameters were defined in an instance of `LineItemResponse` class. The adapter then assigns the instance to `lineItemResponse`.

```
commercetax.AmountDetailsResponse amountResponse = new commercetax.AmountDetailsResponse();
amountResponse.setTotalAmountWithTax(totalTax+totalAmount);
```

```
amountResponse.setExemptAmount(0);
amountResponse.setTotalAmount(totalAmount);
amountResponse.setTaxAmount(totalTax);
lineItemResponse.setAmountDetails(amountResponse);
```

IN THIS SECTION:

AmountDetailsResponse Methods
Learn more about the methods available from the `AmountDetailsResponse` class.

AmountDetailsResponse Methods

Learn more about the methods available from the `AmountDetailsResponse` class.

The following are methods for `AmountDetailsResponse`.

IN THIS SECTION:

setExemptAmount(exemptAmount)
Sets the value of the ExemptAmount field.

setTaxAmount(taxAmount)
Sets the value of the TaxAmount field.

setTotalAmount(totalAmount)
Sets the value of the TotalAmount field.

setTotalAmountWithTax(totalAmtWithTax)
Sets the value of the TotalAmountWithTax field.

`setExemptAmount(exemptAmount)`

Sets the value of the ExemptAmount field.

Signature

```
global void setExemptAmount(Double exemptAmount)
```

Parameters

`exemptAmount`
Type: Double

The amount of a line item's total amount that's exempt from tax calculation.

Return Value

Type: void

`setTaxAmount(taxAmount)`

Sets the value of the TaxAmount field.

Signature

```
global void setTaxAmount(Double taxAmount)
```

Parameters

taxAmount
Type: Double

The calculated amount of tax for a line item.

Return Value

Type: void

setTotalAmount(totalAmount)

Sets the value of the TotalAmount field.

Signature

```
global void setTotalAmount(Double totalAmount)
```

Parameters

totalAmount
Type: Double

The total amount of a line item, excluding tax.

Return Value

Type: void

setTotalAmountWithTax(totalAmtWithTax)

Sets the value of the TotalAmountWithTax field.

Signature

```
global void setTotalAmountWithTax(Double totalAmtWithTax)
```

Parameters

totalAmtWithTax
Type: Double

The total amount of a line item combined with the calculated tax for that line item.

Return Value

Type: void

CalculateTaxRequest Class

Represents a request to an external tax engine to calculate tax. Extends the TaxTransactionRequest class and is the top-level request class.

Namespace

CommerceTax

Example

See TaxEngineAdapter Example Implementation for more details on how to access information from the `CalculateTaxRequest` class.

IN THIS SECTION:

CalculateTaxRequest Constructors
Learn more about the constructors that are available with the `CalculateTaxRequest` class. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

CalculateTaxRequest Properties
Learn more about the available properties with the `CalculateTaxRequest` class.

CalculateTaxRequest Methods
Learn more about the available methods with the `CalculateTaxRequest` class.

CalculateTaxRequest Constructors

Learn more about the constructors that are available with the `CalculateTaxRequest` class. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

The `CalculateTaxRequest` class includes these constructors.

IN THIS SECTION:

CalculateTaxRequest(taxType)
This constructor is intended for test usage only and throws an exception if used outside of the Apex test context.

`CalculateTaxRequest(taxType)`

This constructor is intended for test usage only and throws an exception if used outside of the Apex test context.

Signature

```
global CalculateTaxRequest(commercetax.CalculateTaxType taxType)
```

Parameters

`taxType`
Type: CalculateTaxType

Indicates whether the tax calculation is for estimated tax or actual tax.

CalculateTaxRequest Properties

Learn more about the available properties with the `CalculateTaxRequest` class.

The `CalculateTaxRequest` class includes these properties.

IN THIS SECTION:

isCommit
Indicates whether the tax calculation has to be committed or reported to government authorities.

taxTransactionType
Shows whether the tax transaction is for a credit or debit transaction.

taxType
Shows whether the tax calculation is for estimated or actual tax wherein only actual tax can be submitted.

`isCommit`

Indicates whether the tax calculation has to be committed or reported to government authorities.

Signature

```
global Boolean isCommit {get; set;}
```

Property Value

Type: Boolean

`taxTransactionType`

Shows whether the tax transaction is for a credit or debit transaction.

Signature

```
global commercetax.TaxTransactionType taxTransactionType {get; set;}
```

Property Value

Type: TaxTransactionType

`taxType`

Shows whether the tax calculation is for estimated or actual tax wherein only actual tax can be submitted.

Signature

```
global commercetax.CalculateTaxType taxType {get; set;}
```

Property Value

Type: CalculateTaxType

CalculateTaxRequest Methods

Learn more about the available methods with the `CalculateTaxRequest` class.

The `CalculateTaxRequest` class includes these methods.

IN THIS SECTION:

equals(obj)
Maintains the integrity of lists of type `CalculateTaxRequest` by determining the equality of external objects in a list. This method is dynamic and is based on the `equals()` method in Java.

hashCode()
Maintains the integrity of lists of type `CalculateTaxRequest` by determining the uniqueness of the external object records in a list.

toString()
Converts a value to a string.

`equals(obj)`

Maintains the integrity of lists of type `CalculateTaxRequest` by determining the equality of external objects in a list. This method is dynamic and is based on the `equals()` method in Java.

Signature

```
global Boolean equals(Object obj)
```

Parameters

obj
Type: Object

External object whose key is to be validated.

Return Value

Type: Boolean

`hashCode()`

Maintains the integrity of lists of type `CalculateTaxRequest` by determining the uniqueness of the external object records in a list.

Signature

```
global Integer hashCode()
```

Return Value

Type: Integer

`toString()`

Converts a value to a string.

Signature

```
global String toString()
```

Return Value

Type: String

CalculateTaxResponse Class

Sets the values of the tax transaction following a response from the external tax engine. Extends the AbstractTransactionResponse class and is the top-level response class.

Namespace

CommerceTax

Example

```
if(requestType == commercetax.RequestType.CalculateTax){
              commercetax.calculatetaxtype type = request.taxtype;
              String docCode='';
              if(request.DocumentCode == 'simulateEmptyDocumentCode')
                  docCode = '';
              else if(request.DocumentCode != null)
                  docCode =request.DocumentCode;
            else if(request.ReferenceEntityId != null) docCode = request.ReferenceEntityId;

              else docCode =  String.valueOf(getRandomInteger(0,2147483647));
              commercetax.CalculateTaxResponse response = new
commercetax.CalculateTaxResponse();
              if(request.isCommit == true) {
                  response.setStatus(commercetax.TaxTransactionStatus.Committed);
              } else {
                  response.setStatus(commercetax.TaxTransactionStatus.Uncommitted);
              }
              response.setDocumentCode(docCode);
              response.setReferenceDocumentCode(request.referenceDocumentCode);
              response.setTaxType(type);
              response.setStatusDescription('statusDescription');
              if(request.sellerDetails.code == 'testSellerCode') {
                  response.setDescription('SellerCode fetched from TaxEngine entity');
              }
              else {
                  response.setDescription('description');
              }
             response.setEffectiveDate(system.now());
             if(request.transactionDate == null) {
```

```
                response.setTransactionDate(system.now());
            } else {
                response.setTransactionDate(request.transactionDate);
            }
            if(request.taxTransactionType == null) {
                response.setTaxTransactionType(commercetax.TaxTransactionType.Debit);
            } else {
                response.setTaxTransactionType(request.taxTransactionType);
            }
             if(request.currencyIsoCode == null || request.currencyIsoCode == '') {
                response.setCurrencyIsoCode('USD');
             } else {
                response.setCurrencyIsoCode(request.currencyIsoCode);
             }
             response.setReferenceEntityId(request.ReferenceEntityId);
}
```

IN THIS SECTION:

CalculateTaxResponse Methods

Learn more about the available methods with the `CalculateTaxResponse` class.

CalculateTaxResponse Methods

Learn more about the available methods with the `CalculateTaxResponse` class.

The `CalculateTaxResponse` class includes these methods.

IN THIS SECTION:

setAddresses(addresses)

Sets the value of the Addresses field using the addresses contained in an instance of the AddressesResponse class.

setAmountDetails(amountDetails)

Sets the value of the AmountDetails field using an instance of `AmountDetailsResponse`.

setCurrencyIsoCode(currencyIsoCode)

Sets the value of the CurrencyIsoCode field of the `CalculateTaxResponse` object.

setDescription(dscptn)

Sets the value of the Description field of the `CalculateTaxResponse` object.

setDocumentCode(documentCode)

Sets the value of the DocumentCode field of the `CalculateTaxResponse` object.

setEffectiveDate(effectiveDate)

Sets the value of the EffectiveDate field of the `CalculateTaxResponse` object.

setLineItems(lineItems)

Sets the value of the LineItems field of the `CalculateTaxResponse` object.

setReferenceDocumentCode(referenceDocumentCode)

Sets the value of the ReferenceDocumentCode field of the `CalculateTaxResponse` object.

setReferenceEntityId(referenceEntityId)
Sets the value of the ReferenceEntityId field of the `CalculateTaxResponse` object.

setStatus(status)
Sets the value of the Status field of the `CalculateTaxResponse` object.

setStatusDescription(statusDescription)
Sets the value of the StatusDescription field of the `CalculateTaxResponse` object.

setTaxTransactionId(taxTrxnId)
Sets the value of the TaxTransactionId field of the `CalculateTaxResponse` object.

setTaxTransactionType(taxTransactionType)
Sets the value of the TaxTransactionType field of the `CalculateTaxResponse` object.

setTaxType(taxType)
Sets the value of the TaxType field of the `CalculateTaxResponse` object.

setTransactionDate(transactionDate)
Sets the value of the TransactionDate field of the `CalculateTaxResponse` object.

`setAddresses(addresses)`

Sets the value of the Addresses field using the addresses contained in an instance of the AddressesResponse class.

Signature

```
global void setAddresses(commercetax.AddressesResponse addresses)
```

Parameters

addresses
Type: AddressesResponse

Contains Ship To, Ship From, and Sold To addresses.

Return Value

Type: void

`setAmountDetails(amountDetails)`

Sets the value of the AmountDetails field using an instance of `AmountDetailsResponse`.

Signature

```
global void setAmountDetails(commercetax.AmountDetailsResponse amountDetails)
```

Parameters

amountDetails
Type: AmountDetailsResponse

The tax amount details for a line item on which tax was calculated.

Return Value

Type: void

`setCurrencyIsoCode(currencyIsoCode)`

Sets the value of the CurrencyIsoCode field of the `CalculateTaxResponse` object.

Signature

```
global void setCurrencyIsoCode(String currencyIsoCode)
```

Parameters

`currencyIsoCode`
Type: String

Three-letter ISO 4217 currency code associated with a tax object.

Return Value

Type: void

`setDescription(dscptn)`

Sets the value of the Description field of the `CalculateTaxResponse` object.

Signature

```
global void setDescription(String dscptn)
```

Parameters

`dscptn`
Type: String

Optional description for providing more information about the calculate tax response.

Return Value

Type: void

`setDocumentCode(documentCode)`

Sets the value of the DocumentCode field of the `CalculateTaxResponse` object.

Signature

```
global void setDocumentCode(String documentCode)
```

Parameters

documentCode
Type: String

Code for a tax document that's created by the tax engine for the calculation process.

Return Value

Type: void

setEffectiveDate(effectiveDate)

Sets the value of the EffectiveDate field of the `CalculateTaxResponse` object.

Signature

```
global void setEffectiveDate(Datetime effectiveDate)
```

Parameters

effectiveDate
Type: Datetime

The date a tax calculation action takes effect. This parameter is optional and is provided only for recordkeeping purpose. Additionally, this parameter is used to determine the tax rates or rules and overrides the transaction date. For example, if the tax calculation request is placed on January 3 and the transaction date is January 1, you can set the effective date as January 1.

Return Value

Type: void

setLineItems(lineItems)

Sets the value of the LineItems field of the `CalculateTaxResponse` object.

Signature

```
global void setLineItems(List<commercetax.LineItemResponse> lineItems)
```

Parameters

lineItems
Type: List<LineItemResponse>

Response object that the tax adapter populates from the response of the external tax engine.

Return Value

Type: void

`setReferenceDocumentCode(referenceDocumentCode)`

Sets the value of the ReferenceDocumentCode field of the `CalculateTaxResponse` object.

Signature

```
global void setReferenceDocumentCode(String referenceDocumentCode)
```

Parameters

`referenceDocumentCode`
Type: String

Code for a reference document used in the tax calculation process.

Return Value

Type: void

`setReferenceEntityId(referenceEntityId)`

Sets the value of the ReferenceEntityId field of the `CalculateTaxResponse` object.

Signature

```
global void setReferenceEntityId(String referenceEntityId)
```

Parameters

`referenceEntityId`
Type: String

ID of an entity related to the line items submitted for tax calculation. For example, if order items were sent for tax calculation, you could use the ID of their parent order.

Return Value

Type: void

`setStatus(status)`

Sets the value of the Status field of the `CalculateTaxResponse` object.

Signature

```
global void setStatus(commercetax.TaxTransactionStatus status)
```

Parameters

`status`
Type: TaxTransactionStatus

Indicates whether a tax transaction has been committed.

Return Value

Type: void

setStatusDescription(statusDescription)

Sets the value of the StatusDescription field of the `CalculateTaxResponse` object.

Signature

```
global void setStatusDescription(String statusDescription)
```

Parameters

statusDescription
Type: String

Optional value for providing more information about a tax transaction's status.

Return Value

Type: void

setTaxTransactionId(taxTrxnId)

Sets the value of the TaxTransactionId field of the `CalculateTaxResponse` object.

Signature

```
public void setTaxTransactionId(String taxTrxnId)
```

Parameters

taxTrxnId
Type: String

The ID of the Salesforce tax transaction entity that stores information about the tax calculation transaction.

Return Value

Type: void

setTaxTransactionType(taxTransactionType)

Sets the value of the TaxTransactionType field of the `CalculateTaxResponse` object.

Signature

```
global void setTaxTransactionType(commercetax.TaxTransactionType taxTransactionType)
```

Parameters

taxTransactionType
Type: TaxTransactionType

Whether the tax transaction was for a credit or debit transacation.

Return Value

Type: void

setTaxType(taxType)

Sets the value of the TaxType field of the `CalculateTaxResponse` object.

Signature

```
global void setTaxType(commercetax.CalculateTaxType taxType)
```

Parameters

taxType
Type: CalculateTaxType

Indicates whether a tax calculation request is for estimated or actual tax.

Return Value

Type: void

setTransactionDate(transactionDate)

Sets the value of the TransactionDate field of the `CalculateTaxResponse` object.

Signature

```
global void setTransactionDate(Datetime transactionDate)
```

Parameters

transactionDate
Type: Datetime

The date that the tax transaction occurred.

Return Value

Type: void

CalculateTaxType Enum

Shows whether a tax calculation request is for estimated or actual tax.

Usage

Used by the CalculateTaxRequest and CalculateTaxResponse class methods.

Enum Values

The `commercetax.CalculateTaxType` enum includes these values.

Value	Description
`Actual`	Specifies that the tax calculation service should calculate the finalized (actual) tax for the requested line items.
`Estimated`	Specifies that the tax calculation service should estimate the tax for the requested line items.

ErrorResponse Class

Use to respond with an error after receiving errors from the PaymentGatewayAdapter methods of the CommercePayments namespace, such as request-forbidden responses, custom validation errors, or expired API tokens.

Namespace

CommerceTax

Example

This example snippet of a mock tax adapter shows a hypothetical scenario to demo an error response. The adapter receives request information from `TaxEngineContext` and stores it in an instance of `CalculateTaxRequest`. If the request's `documentCode` property is null or indicates an error, the adapter returns an error response with information about the error.

```
global virtual class MockAdapter implements commercetax.TaxEngineAdapter {

     global commercetax.TaxEngineResponse processRequest(commercetax.TaxEngineContext
taxEngineContext) {
         commercetax.RequestType requestType = taxEngineContext.getRequestType();
         commercetax.CalculateTaxRequest request =
(commercetax.CalculateTaxRequest)taxEngineContext.getRequest();

if(request.documentCode == null) {
             return new commercetax.ErrorResponse(commercetax.resultcode.TaxEngineError,
'404', 'documentCode  is mandatory');
         }
         if(request.documentCode == 'TaxEngineError') {
             return new commercetax.ErrorResponse(commercetax.resultcode.TaxEngineError,
'504', 'documentCode  - not supported');
         }
         if(request.documentCode == 'simulateValidationFailureInAdapter') {
             return new commercetax.ErrorResponse(commercetax.resultcode.TaxEngineError,
'400', 'validations for documentCode failed in adapter');
```

```
        }
        if(request.documentCode == 'simulateMalformedErrorInAdapter') {
                    return new
commercetax.ErrorResponse(commercetax.resultcode.TaxEngineError, null, 'malformed adapter
 error response');
        }
        if(request.documentCode == 'simulateTaxEngineProcessFailure') {
            return new commercetax.ErrorResponse(commercetax.resultcode.TaxEngineError,
'500', 'Tax Engine couldnt process your request');
        }
```

IN THIS SECTION:

ErrorResponse Constructors

Learn more about the available constructors with the `ErrorResponse` class.

ErrorResponse Constructors

Learn more about the available constructors with the `ErrorResponse` class.

The `ErrorResponse` class includes these constructors.

IN THIS SECTION:

ErrorResponse(resultCode, errorCode, errorMessage)

Constructor to initialize an `ErrorResponse` object from the result code, error code, and error message sent from the tax engine.

`ErrorResponse(resultCode, errorCode, errorMessage)`

Constructor to initialize an `ErrorResponse` object from the result code, error code, and error message sent from the tax engine.

Signature

```
global ErrorResponse(commercetax.ResultCode resultCode, String errorCode, String
errorMessage)
```

Parameters

`resultCode`

Type: ResultCode

Code for the type of result sent by the tax engine.

`errorCode`

Type: String

Code for the type of error sent by the tax engine.

Codes must match the HTTP status codes to be returned to the user. Here are a few examples:

- If the status code is for a bad request, set `errorCode` to `400`.
- If the status code is for a forbidden request, set `errorCode` to `403`.
- If `errorCode` isn't a valid HTTP status code, a 500 internal server error is returned.

errorMessage
Type: String

The error message sent by the tax engine.

HeaderTaxAddressesRequest Class

Captures the address values that are applicable for the quote or order transaction.

Namespace

CommerceTax

IN THIS SECTION:

HeaderTaxAddressesRequest Constructors
Learn more about the constructors available with the `HeaderTaxAddressesRequest` class.

HeaderTaxAddressesRequest Properties
Learn more about the available properties with the `HeaderTaxAddressesRequest` class.

HeaderTaxAddressesRequest Methods
Learn more about the available methods with the `HeaderTaxAddressesRequest` class.

HeaderTaxAddressesRequest Constructors

Learn more about the constructors available with the `HeaderTaxAddressesRequest` class.

The `HeaderTaxAddressesRequest` class includes these constructors.

IN THIS SECTION:

HeaderTaxAddressesRequest(shipFrom, shipTo, soldTo, billTo, taxEngineAddress)
Constructor for initializing the required addresses of the tax addresses request such as the ship from, ship to, sold to, and bill to addresses. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

`HeaderTaxAddressesRequest(shipFrom, shipTo, soldTo, billTo, taxEngineAddress)`

Constructor for initializing the required addresses of the tax addresses request such as the ship from, ship to, sold to, and bill to addresses. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Signature

```
global HeaderTaxAddressesRequest(commercetax.TaxAddressRequest shipFrom,
commercetax.TaxAddressRequest shipTo, commercetax.TaxAddressRequest soldTo,
commercetax.TaxAddressRequest billTo, commercetax.TaxAddressRequest taxEngineAddress)
```

Parameters

shipFrom
Type: TaxAddressRequest

Address where a line item was shipped from.

shipTo
Type: TaxAddressRequest

Address where a line item was shipped to.

soldTo
Type: TaxAddressRequest

Address of the line item's buyer.

billTo
Type: TaxAddressRequest

Person or group who was billed for the line item.

taxEngineAddress
Type: TaxAddressRequest

Address that the tax engine uses to calculate tax.

HeaderTaxAddressesRequest Properties

Learn more about the available properties with the `HeaderTaxAddressesRequest` class.

The `HeaderTaxAddressesRequest` class includes these properties.

IN THIS SECTION:

billTo
Specifies the billTo address for a line item on which tax was calculated.

shipFrom
Specifies the shipFrom address for a line item on which tax was calculated.

shipTo
Specifies the shipTo address for a line item on which tax was calculated.

soldTo
Specifies the soldTo address for a line item on which tax was calculated.

taxEngineAddress
Address used by the tax engine when calculating tax for a line item.

billTo

Specifies the billTo address for a line item on which tax was calculated.

Signature

```
global commercetax.TaxAddressRequest billTo {get; set;}
```

Property Value

Type: TaxAddressRequest

shipFrom

Specifies the shipFrom address for a line item on which tax was calculated.

Signature

```
global commercetax.TaxAddressRequest shipFrom {get; set;}
```

Property Value

Type: TaxAddressRequest

shipTo

Specifies the shipTo address for a line item on which tax was calculated.

Signature

```
global commercetax.TaxAddressRequest shipTo {get; set;}
```

Property Value

Type: TaxAddressRequest

soldTo

Specifies the soldTo address for a line item on which tax was calculated.

Signature

```
global commercetax.TaxAddressRequest soldTo {get; set;}
```

Property Value

Type: TaxAddressRequest

taxEngineAddress

Address used by the tax engine when calculating tax for a line item.

Signature

```
global commercetax.TaxAddressRequest taxEngineAddress {get; set;}
```

Property Value

Type: TaxAddressRequest

HeaderTaxAddressesRequest Methods

Learn more about the available methods with the `HeaderTaxAddressesRequest` class.

The `HeaderTaxAddressesRequest` class includes these methods.

IN THIS SECTION:

equals(obj)
Maintains the integrity of lists of type `HeaderTaxAddressesRequest` by determining the equality of external objects in a list. This method is dynamic and is based on the `equals()` method in Java.

hashCode()
Maintains the integrity of lists of type `TaxAddressesRequest` by determining the uniqueness of the external objects in a list.

toString()
Converts a value to a string.

`equals(obj)`

Maintains the integrity of lists of type `HeaderTaxAddressesRequest` by determining the equality of external objects in a list. This method is dynamic and is based on the `equals()` method in Java.

Signature

```
global Boolean equals(Object obj)
```

Parameters

obj
Type: Object

External object whose key is to be validated.

Return Value

Type: Boolean

`hashCode()`

Maintains the integrity of lists of type `TaxAddressesRequest` by determining the uniqueness of the external objects in a list.

Signature

```
global Integer hashCode()
```

Return Value

Type: Integer

`toString()`

Converts a value to a string.

Signature

```
global String toString()
```

Return Value

Type: String

ImpositionResponse Class

Stores details of tax impositions from the external tax engine.

Namespace

CommerceTax

Example

In this mock adapter example, the adapter sets the `TaxDetailsResponse.setImposition()` method parameter to null if the request's document code indicates that the tax calculation didn't require any exceptions. Otherwise, it creates an instance of `ImpositionResponse` and sets its SubType and Type values, and then assigns it to `TaxDetailsResponse`.

```
if(request.DocumentCode == 'SetsNullForResponseWithoutException'){
  taxDetailsResponse.setImposition(null);
                      }else{
                         commercetax.ImpositionResponse imposition = new
commercetax.ImpositionResponse();
                         imposition.setSubType('subtype');
                         imposition.setType('type');
                         taxDetailsResponse.setImposition(imposition);
                    }
```

IN THIS SECTION:

ImpositionResponse Methods
Learn more about the available methods with the `ImpositionResponse` class.

ImpositionResponse Methods

Learn more about the available methods with the `ImpositionResponse` class.

The `ImpositionResponse` class includes these methods.

IN THIS SECTION:

setId(id)
Sets the ID field of the `ImpositionResponse` class.

setName(name)
Sets the Name field of the `ImpositionResponse` class.

setSubType(subType)
Sets the SubType field of the `ImpositionResponse` class.

setType(type)
Sets the Type field of the `ImpositionResponse` class.

setId(id)

Sets the ID field of the `ImpositionResponse` class.

Signature

```
global void setId(String id)
```

Parameters

id
Type: String

User-defined ID value used for referencing the tax imposition.

Return Value

Type: void

setName(name)

Sets the Name field of the `ImpositionResponse` class.

Signature

```
global void setName(String name)
```

Parameters

name
Type: String

Optional user-defined name for the tax imposition response.

Return Value

Type: void

setSubType(subType)

Sets the SubType field of the `ImpositionResponse` class.

Signature

```
global void setSubType(String subType)
```

Parameters

subType
Type: String

Many tax calculation organizations use types and subtypes to categorize their tax imposition procedures. If the tax engine you use follows this process, set the subtype with this parameter.

Return Value

Type: void

`setType(type)`

Sets the Type field of the `ImpositionResponse` class.

Signature

```
public void setType(String type)
```

Parameters

type
Type: String

Many tax calculation organizations use types and subtypes to categorize their tax imposition procedures. If the tax engine you use follows this process, set the type with this parameter.

Return Value

Type: void

JurisdictionResponse Class

Stores details from the external tax engine about the tax jurisdiction used in the tax calculation process. A tax jurisdiction represents a government entity that collects tax.

Namespace

CommerceTax

Example

In this mock adapter example, the adapter sets the `TaxDetailsResponse.setJurisdiction()` method parameter to null if the request's document code indicates that the tax calculation didn't require any exceptions. Otherwise, it creates an instance of `JurisdictionResponse` and sets its address values. Because this code represents a mock adapter, the example defines the address parameters directly. In a standard implementation, the jurisdiction's setters receive values passed from the eternal tax engine.

```
if(request.DocumentCode == 'SetsNullForResponseWithoutException'){
                        taxDetailsResponse.setJurisdiction(null);
                    }else{
                       commercetax.JurisdictionResponse jurisdiction = new
```

```
commercetax.JurisdictionResponse();
                        jurisdiction.setCountry('country');
                        jurisdiction.setRegion('region');
                        jurisdiction.setName('name');
                        jurisdiction.setStateAssignedNumber('stateAssignedNo');
                        jurisdiction.setId('id');
                        jurisdiction.setLevel('level');
                        taxDetailsResponse.setJurisdiction(jurisdiction);

}
```

IN THIS SECTION:

JurisdictionResponse Methods

Learn more about the available methods with the `JurisdictionResponse` class.

JurisdictionResponse Methods

Learn more about the available methods with the `JurisdictionResponse` class.

The `JurisdictionResponse` class includes these methods.

IN THIS SECTION:

setCountry(country)

Sets the Country field of the `JurisdictionResponse` class.

setId(id)

Sets the ID field of the `JurisdictionResponse` class.

setLevel(level)

Sets the Level field of the `JurisdictionResponse` class.

setName(name)

Sets the Name field of the `JurisdictionResponse` class.

setRegion(region)

Sets the Region value of the `JurisdictionResponse` class.

setStateAssignedNumber(stateAssignedNo)

Sets the StateAssignedNumber field of the `JurisdictionResponse` class.

`setCountry(country)`

Sets the Country field of the `JurisdictionResponse` class.

Signature

```
global void setCountry(String country)
```

Parameters

country
Type: String

The country of the tax jurisdiction entity's address.

Return Value

Type: void

setId(id)

Sets the ID field of the `JurisdictionResponse` class.

Signature

```
global void setId(String id)
```

Parameters

id
Type: String

User-defined Id value used to reference the jurisdiction response.

Return Value

Type: void

setLevel(level)

Sets the Level field of the `JurisdictionResponse` class.

Signature

```
global void setLevel(String level)
```

Parameters

level
Type: String

Level value used in the jurisdiction entity's address.

Return Value

Type: void

setName(name)

Sets the Name field of the `JurisdictionResponse` class.

Signature

```
global void setName(String name)
```

Parameters

name
Type: String

Optional user-defined name field for referencing the jurisdiction response.

Return Value

Type: void

setRegion(region)

Sets the Region value of the `JurisdictionResponse` class.

Signature

```
global void setRegion(String region)
```

Parameters

region
Type: String

Region value used in the tax jurisdiction entity's address.

Return Value

Type: void

setStateAssignedNumber(stateAssignedNo)

Sets the StateAssignedNumber field of the `JurisdictionResponse` class.

Signature

```
global void setStateAssignedNumber(String stateAssignedNo)
```

Parameters

stateAssignedNo
Type: String

State assigned number value of the tax jurisdiction entity's address.

Return Value

Type: void

LineItemResponse Class

Response class that stores details of a list of one or more line items on which the tax engine has calculated tax.

Namespace

CommerceTax

Example

This example uses a `LineItemResponse` list to store information about each line item that was processed as part of the request. For simplicity, the sample code uses a static value of 1 for the tax rate. However, most integrations typically have a more complex process for determining a tax rate. Most integrations also build a `TaxDetailsResponse` list to store the actual tax value information that they assign to each line item in the `LineItemResponse` list.

```
Double totalTax = 0.0;
             Double totalAmount = 0.0;
             List<commercetax.LineItemResponse> lineItemResponses = new
List<commercetax.LineItemResponse>();
             for(Commercetax.TaxLineItemRequest lineItem : request.lineItems){
                 commercetax.AddressesResponse addressesRes = new
commercetax.AddressesResponse();
                 if(request.DocumentCode == 'SetsNullForResponseWithoutException'){
                     addressesRes.setShipFrom(null);
                     addressesRes.setShipTO(null);
                     addressesRes.setSoldTo(null);
                 }else{
                  commercetax.AddressResponse addRes = new commercetax.AddressResponse();

                     addRes.setLocationCode('locationCode');
                     addressesRes.setShipFrom(addRes);
                     addressesRes.setShipTO(addRes);
                     addressesRes.setSoldTo(addRes);
                 }
                 commercetax.LineItemResponse lineItemResponse = new
commercetax.LineItemResponse();
                 Double totalLineTax = 0;
                 List<commercetax.TaxDetailsResponse> taxDetailsResponses = new
List<commercetax.TaxDetailsResponse>();
                 for(integer i =0;i<1;i++){
                     Integer rate  = 1;
                     Double taxableAmount = lineItem.amount;
                     commercetax.TaxDetailsResponse taxDetailsResponse = new
commercetax.TaxDetailsResponse();
                     taxDetailsResponse.setRate(Double.valueOf(rate));
                     taxDetailsResponse.setTaxableAmount(taxableAmount);
                     Double tax = taxableAmount*rate;
                     totalLineTax+=tax;
                     taxDetailsResponse.setTax(taxableAmount*rate);
                     taxDetailsResponse.setExemptAmount(0);
                     taxDetailsResponse.setExemptReason('exemptReason');
                     taxDetailsResponse.setTaxRegionId('taxRegionId');
```

```
taxDetailsResponse.setTaxId(String.valueOf(getRandomInteger(0,2323233)));
                    taxDetailsResponse.setSerCode('serCode');
                    taxDetailsResponse.setTaxAuthorityTypeId('taxAuthorityTypeId');
                    if(request.DocumentCode == 'SetsNullForResponseWithoutException'){
                        taxDetailsResponse.setImposition(null);
                    }else{
                        commercetax.ImpositionResponse imposition = new
commercetax.ImpositionResponse();
                        imposition.setSubType('subtype');
                        imposition.setType('type');
                        taxDetailsResponse.setImposition(imposition);
                    }

                    if(request.DocumentCode == 'SetsNullForResponseWithoutException'){
                        taxDetailsResponse.setJurisdiction(null);
                    }else{
                        commercetax.JurisdictionResponse jurisdiction = new
commercetax.JurisdictionResponse();
                        jurisdiction.setCountry('country');
                        jurisdiction.setRegion('region');
                        jurisdiction.setName('name');
                        jurisdiction.setStateAssignedNumber('stateAssignedNo');
                        jurisdiction.setId('id');
                        jurisdiction.setLevel('level');
                        taxDetailsResponse.setJurisdiction(jurisdiction);
                    }

                    taxDetailsResponses.add(taxDetailsResponse);
                }
                lineItemResponse.setTaxes(taxDetailsResponses);
                totalTax +=totalLineTax;
                totalAmount+=lineItem.amount;
```

IN THIS SECTION:

LineItemResponse Methods

Learn more about the available methods with the `LineItemResponse` class.

LineItemResponse Methods

Learn more about the available methods with the `LineItemResponse` class.

The `LineItemResponse` class includes these methods.

IN THIS SECTION:

setAddresses(addresses)

Sets the Addresses field on the `LineItemResponse` using an instance of `AddressesResponse` class.

setAmountDetails(amountDetails)

Sets the Amount Details field on the `LineItemResponse` using an instance of `AmountDetails`.

setEffectiveDate(effectiveDate)

Sets the EffectiveDate field on the `LineItemResponse` class. Effective Date fields are optional fields that store the date that a transaction takes effect. We provide these fields only for recordkeeping purposes – for example, if you must report an effective date to an external general ledger system. Salesforce doesn't use them to calculate any tax or payment values.

setIsTaxable(isTaxable)

Sets the IsTaxable field on the `LineItemResponse` class.

setLineNumber(lineNumber)

Sets the LineNumber field on the `LineItemResponse` class.

setProductCode(productCode)

Sets the ProductCode field on the `LineItemResponse` class.

setQuantity(quantity)

Sets the Quantity field on the `LineItemResponse` class.

setTaxCode(taxCode)

Sets the TaxCode field on the `LineItemResponse`.

setTaxes(taxes)

Sets the Taxes field on a `LineItemResponse`.

`setAddresses(addresses)`

Sets the Addresses field on the `LineItemResponse` using an instance of `AddressesResponse` class.

Signature

```
global void setAddresses(commercetax.AddressesResponse addresses)
```

Parameters

`addresses`

Type: AddressesResponse

Class that contains methods to set the Ship To, Ship From, and Sold To address information.

Return Value

Type: void

`setAmountDetails(amountDetails)`

Sets the Amount Details field on the `LineItemResponse` using an instance of `AmountDetails`.

Signature

```
global void setAmountDetails(commercetax.AmountDetailsResponse amountDetails)
```

Parameters

`amountDetails`

Type: AmountDetailsResponse

Class that contains methods to set the tax amount, total amount with tax, total amount, and exempt amount.

Return Value

Type: void

`setEffectiveDate(effectiveDate)`

Sets the EffectiveDate field on the `LineItemResponse` class. Effective Date fields are optional fields that store the date that a transaction takes effect. We provide these fields only for recordkeeping purposes – for example, if you must report an effective date to an external general ledger system. Salesforce doesn't use them to calculate any tax or payment values.

Signature

```
global void setEffectiveDate(Datetime effectiveDate)
```

Parameters

`effectiveDate`
Type: Datetime

Optional field that stores the date that a transaction takes effect.

Return Value

Type: void

`setIsTaxable(isTaxable)`

Sets the IsTaxable field on the `LineItemResponse` class.

Signature

```
global void setIsTaxable(Boolean isTaxable)
```

Parameters

`isTaxable`
Type: Boolean

Whether line items were taxed as part of the tax calculation request.

Return Value

Type: void

`setLineNumber(lineNumber)`

Sets the LineNumber field on the `LineItemResponse` class.

Signature

```
global void setLineNumber(String lineNumber)
```

Parameters

lineNumber
Type: String

User-defined number used to identify a line item.

Return Value

Type: void

setProductCode(productCode)

Sets the ProductCode field on the `LineItemResponse` class.

Signature

```
global void setProductCode(String productCode)
```

Parameters

productCode
Type: String

Code for the product that a line item represents.

Return Value

Type: void

setQuantity(quantity)

Sets the Quantity field on the `LineItemResponse` class.

Signature

```
global void setQuantity(Double quantity)
```

Parameters

quantity
Type: Double

Quantity of a line item.

Return Value

Type: void

`setTaxCode(taxCode)`

Sets the TaxCode field on the `LineItemResponse`.

Signature

```
global void setTaxCode(String taxCode)
```

Parameters

`taxCode`
Type: String

Federal code that an individual or business uses to pay their taxes to a federal or state government. The tax engine uses this code during the tax calculation process.

Return Value

Type: void

`setTaxes(taxes)`

Sets the Taxes field on a `LineItemResponse`.

Signature

```
global void setTaxes(List<commercetax.TaxDetailsResponse> taxes)
```

Parameters

`taxes`
Type: List<TaxDetailsResponse>

Tax values applied to a line item in the `LineItemResponse` list. This information is stored in a list of `TaxDetailsResponses`, which contains values such as tax, taxable amount, and tax rate.

Return Value

Type: void

LineTaxAddressesRequest Class

Stores details of the addresses applied per line item in a tax calculation request.

Namespace

CommerceTax

IN THIS SECTION:

LineTaxAddressesRequest Constructors
Learn more about the constructors available with the `LineTaxAddressesRequest` class.

LineTaxAddressesRequest Properties

Learn more about the available properties with the `LineTaxAddressesRequest` class.

LineTaxAddressesRequest Methods

Learn more about the available methods with the `LineTaxAddressesRequest` class.

LineTaxAddressesRequest Constructors

Learn more about the constructors available with the `LineTaxAddressesRequest` class.

The `LineTaxAddressesRequest` class includes these constructors.

IN THIS SECTION:

LineTaxAddressesRequest(shipFrom, shipTo, soldTo, billTo, taxEngineAddress)

Constructor for initializing the required addresses for a line item of the tax addresses request such as the ship to, ship from, and bill to addresses. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

`LineTaxAddressesRequest(shipFrom, shipTo, soldTo, billTo, taxEngineAddress)`

Constructor for initializing the required addresses for a line item of the tax addresses request such as the ship to, ship from, and bill to addresses. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Signature

```
global LineTaxAddressesRequest(commercetax.TaxAddressRequest shipFrom,
commercetax.TaxAddressRequest shipTo, commercetax.TaxAddressRequest soldTo,
commercetax.TaxAddressRequest billTo, commercetax.TaxAddressRequest taxEngineAddress)
```

Parameters

shipFrom
TaxAddressRequest

Address where a line item was shipped from.

shipTo
TaxAddressRequest

Address where a line item is shipped to.

soldTo
TaxAddressRequest

Address of the line item's buyer.

billTo
TaxAddressRequest

Person or group who was billed for the line item.

taxEngineAddress
TaxAddressRequest

Address that the tax engine uses to calculate tax.

LineTaxAddressesRequest Properties

Learn more about the available properties with the `LineTaxAddressesRequest` class.

The `LineTaxAddressesRequest` class includes these properties.

IN THIS SECTION:

billTo
The Bill To address for a line item.

shipFrom
The Ship From address for a line item.

shipTo
The Ship To address for a line item.

soldTo
The Sold To address for a line item.

billTo

The Bill To address for a line item.

Signature

```
global commercetax.TaxAddressRequest billTo {get; set;}
```

Property Value

Type: TaxAddressRequest

shipFrom

The Ship From address for a line item.

Signature

```
global commercetax.TaxAddressRequest shipFrom {get; set;}
```

Property Value

Type: TaxAddressRequest

shipTo

The Ship To address for a line item.

Signature

```
global commercetax.TaxAddressRequest shipTo {get; set;}
```

Property Value

Type: TaxAddressRequest

`soldTo`

The Sold To address for a line item.

Signature

```
global commercetax.TaxAddressRequest soldTo {get; set;}
```

Property Value

Type: TaxAddressRequest

LineTaxAddressesRequest Methods

Learn more about the available methods with the `LineTaxAddressesRequest` class.

The `LineTaxAddressesRequest` class includes these methods.

IN THIS SECTION:

equals(obj)
Maintains the integrity of lists of type `LineTaxAddressesRequest` by determining the equality of external objects in a list. This method is dynamic and is based on the `equals()` method in Java.

hashCode()
Maintains the integrity of lists of type `LineTaxAddressesRequest` by determining the uniquness of the external object records in a list.

toString()
Converts a value to a string.

`equals(obj)`

Maintains the integrity of lists of type `LineTaxAddressesRequest` by determining the equality of external objects in a list. This method is dynamic and is based on the `equals()` method in Java.

Signature

```
global Boolean equals(Object obj)
```

Parameters

obj

Type: Object

External object whose key is to be validated.

Return Value

Type: Boolean

`hashCode()`

Maintains the integrity of lists of type `LineTaxAddressesRequest` by determining the uniquness of the external object records in a list.

Signature

```
global Integer hashCode()
```

Return Value

Type: Integer

`toString()`

Converts a value to a string.

Signature

```
global String toString()
```

Return Value

Type: String

RequestType Enum

Shows the type of tax request made to the tax engine.

Usage

Used by the TaxEngineContext class method.

Enum Values

The `commercetax.RequestType` enum includes these values.

Value	Description
`CalculateTax`	Represents a request to calculate tax on a list of taxable line items.

ResultCode Enum

Code that represents the results of a tax request made to the tax engine.

Usage

Used by the ErrorResponse class method.

Enum Values

The `commercetax.ResultCode` enum includes these values.

Value	Description
`TaxEngineError`	Represents an error that occurred during the tax request process.

RuleDetailsResponse Class

Contains details about the tax rules used for tax calculation.

Namespace

CommerceTax

IN THIS SECTION:

RuleDetailsResponse Methods
Learn more about the available methods with the `RuleDetailsResponse` class.

RuleDetailsResponse Methods

Learn more about the available methods with the `RuleDetailsResponse` class.

The `RuleDetailsResponse` includes these methods.

IN THIS SECTION:

RuleDetailsResponse()
Contains information about the tax rules used when calculating tax for line items.

setNonTaxableRuleId(nonTaxableRuleId)
Sets the NonTaxableRuleId field of the `RuleDetailsResponse`.

setNonTaxableType(nonTaxableType)
Sets the NonTaxableType field of the `RuleDetailsResponse`.

setRateRuleId(rateRuleId)
Sets the RateRuleId field of the `RuleDetailsResponse`.

setRateSourceId(rateSourceId)
Sets the RateSourceId field on the `RuleDetailsResponse`.

`RuleDetailsResponse()`

Contains information about the tax rules used when calculating tax for line items.

Signature

```
global void RuleDetailsResponse()
```

Return Value

Type: void

setNonTaxableRuleId(nonTaxableRuleId)

Sets the NonTaxableRuleId field of the `RuleDetailsResponse`.

Signature

```
global void setNonTaxableRuleId(String nonTaxableRuleId)
```

Parameters

nonTaxableRuleId
Type: String

ID of the tax rule applied to non-taxable line items.

Return Value

Type: void

setNonTaxableType(nonTaxableType)

Sets the NonTaxableType field of the `RuleDetailsResponse`.

Signature

```
global void setNonTaxableType(String nonTaxableType)
```

Parameters

nonTaxableType
Type: String

Reason (from several possible types) that a line item is non-taxable.

Return Value

Type: void

setRateRuleId(rateRuleId)

Sets the RateRuleId field of the `RuleDetailsResponse`.

Signature

```
global void setRateRuleId(String rateRuleId)
```

Parameters

rateRuleId
Type: String

ID of the tax rule used to determine a tax rate.

Return Value

Type: void

setRateSourceId(rateSourceId)

Sets the RateSourceId field on the `RuleDetailsResponse`.

Signature

```
global void setRateSourceId(String rateSourceId)
```

Parameters

rateSourceId
Type: String

ID of the source object used for calculating tax rate.

Return Value

Type: void

TaxAddressesRequest Class

Contains methods to get and set tax address values.

Namespace

CommerceTax

IN THIS SECTION:

TaxAddressesRequest Constructors
Learn more about the available constructors with the `TaxAddressesRequest` class.

TaxAddressesRequest Properties
Learn more about the available properties with the `TaxAddressesRequest` class.

TaxAddressesRequest Methods
Learn more about the available methods with the `TaxAddressesRequest` class.

TaxAddressesRequest Constructors

Learn more about the available constructors with the `TaxAddressesRequest` class.

The `TaxAddressesRequest` class includes these constructors.

IN THIS SECTION:

TaxAddressesRequest(shipFrom, shipTo, soldTo, billTo, taxEngineAddress)
Constructor for defining addresses for the tax addresses request. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

`TaxAddressesRequest(shipFrom, shipTo, soldTo, billTo, taxEngineAddress)`

Constructor for defining addresses for the tax addresses request. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Signature

```
global TaxAddressesRequest(commercetax.TaxAddressRequest shipFrom,
commercetax.TaxAddressRequest shipTo, commercetax.TaxAddressRequest soldTo,
commercetax.TaxAddressRequest billTo, commercetax.TaxAddressRequest taxEngineAddress)
```

Parameters

shipFrom
TaxAddressRequest

The address where a line item was shipped from.

shipTo
TaxAddressRequest

The address where a line item is shipped to.

soldTo
TaxAddressRequest

The address of the line item's buyer.

billTo
TaxAddressRequest

The person or group who was billed for the line item.

taxEngineAddress
TaxAddressRequest

The address that the tax engine uses to calculate tax.

TaxAddressesRequest Properties

Learn more about the available properties with the `TaxAddressesRequest` class.

The `TaxAddressesRequest` class includes these properties.

IN THIS SECTION:

billTo
The Bill To address for a line item.

shipFrom
The Ship From address for a line item.

shipTo
The Ship To address for a line item.

soldTo
The Sold To address for a line item.

taxEngineAddress
The Tax Engine Address for a line item.

billTo

The Bill To address for a line item.

Signature

```
global commercetax.TaxAddressRequest billTo {get; set;}
```

Property Value

TaxAddressRequest

shipFrom

The Ship From address for a line item.

Signature

```
global commercetax.TaxAddressRequest shipFrom {get; set;}
```

Property Value

TaxAddressRequest

shipTo

The Ship To address for a line item.

Signature

```
public commercetax.TaxAddressRequest shipTo {get; set;}
```

Property Value

TaxAddressRequest

soldTo

The Sold To address for a line item.

Signature

```
global commercetax.TaxAddressRequest soldTo {get; set;}
```

Property Value

TaxAddressRequest

`taxEngineAddress`

The Tax Engine Address for a line item.

Signature

```
global commercetax.TaxAddressRequest taxEngineAddress {get; set;}
```

Property Value

TaxAddressRequest

TaxAddressesRequest Methods

Learn more about the available methods with the `TaxAddressesRequest` class.

The `TaxAddressesRequest` class includes these methods.

IN THIS SECTION:

equals(obj)
Maintains the integrity of lists of type `TaxAddressesRequest` by determining the equality of external objects in a list. This method is dynamic and is based on the `equals()` method in Java.

hashCode()
Maintains the integrity of lists of type `TaxAddressesRequest` by determining the uniqueness of the external object records in a list.

toString()
Converts a value to a string.

`equals(obj)`

Maintains the integrity of lists of type `TaxAddressesRequest` by determining the equality of external objects in a list. This method is dynamic and is based on the `equals()` method in Java.

Signature

```
global Boolean equals(Object obj)
```

Parameters

obj
Type: Object

External object whose key is to be validated.

Return Value

Type: Boolean

`hashCode()`

Maintains the integrity of lists of type `TaxAddressesRequest` by determining the uniqueness of the external object records in a list.

Signature

```
global Integer hashCode()
```

Return Value

Type: Integer

`toString()`

Converts a value to a string.

Signature

```
global String toString()
```

Return Value

Type: String

TaxAddressRequest Class

Contains address details used for tax calculation.

Namespace

CommerceTax

IN THIS SECTION:

TaxAddressRequest Constructors
Learn more about the available constructors with the `TaxAddressRequest` class.

TaxAddressRequest Properties
Learn more about the available properties with the `TaxAddressRequest` class.

TaxAddressRequest Methods
Learn more about the available methods with the `TaxAddressRequest` class.

TaxAddressRequest Constructors

Learn more about the available constructors with the `TaxAddressRequest` class.

The `TaxAddressRequest` class includes these constructors.

IN THIS SECTION:

TaxAddressRequest(city, country, latitude, longitude, postalCode, state, street, locationCode)
Initializes the `TaxAddressRequest` object using address details. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

`TaxAddressRequest(city, country, latitude, longitude, postalCode, state, street, locationCode)`

Initializes the `TaxAddressRequest` object using address details. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Signature

```
global TaxAddressRequest(String city, String country, Double latitude, Double longitude,
String postalCode, String state, String street, String locationCode)
```

Parameters

city
Type: String

City used in an address required for tax calculation.

country
Type: String

Country used in an address required for tax calculation.

latitude
Type: Double

Latitude used in an address required for tax calculation.

longitude
Type: Double

Longitude used in an address required for tax calculation.

postalCode
Type: String

Postal code used in an address required for tax calculation.

state
Type: String

State used in an address required for tax calculation.

street
Type: String

Street used in an address required for tax calculation.

locationCode
Type: String

Location code used in an address required for tax calculation.

TaxAddressRequest Properties

Learn more about the available properties with the `TaxAddressRequest` class.

The `TaxAddressRequest` class includes these properties.

IN THIS SECTION:

city
City used in an address required for tax calculation.

country
Country used in an address required for tax calculation.

latitude
Latitude used in an address required for tax calculation.

locationCode
Location code used in an address required for tax calculation.

longitude
Longitude used in an address required for tax calculation.

postalCode
Postal code used in an address required for tax calculation.

state
State used in an address required for tax calculation.

street
Street used in an address required for tax calculation.

city

City used in an address required for tax calculation.

Signature

```
global String city {get; set;}
```

Property Value

Type: String

country

Country used in an address required for tax calculation.

Signature

```
global String country {get; set;}
```

Property Value

Type: String

`latitude`

Latitude used in an address required for tax calculation.

Signature

```
global Double latitude {get; set;}
```

Property Value

Type: Double

`locationCode`

Location code used in an address required for tax calculation.

Signature

```
global String locationCode {get; set;}
```

Property Value

Type: String

`longitude`

Longitude used in an address required for tax calculation.

Signature

```
global Double longitude {get; set;}
```

Property Value

Type: Double

`postalCode`

Postal code used in an address required for tax calculation.

Signature

```
global String postalCode {get; set;}
```

Property Value

Type: String

`state`

State used in an address required for tax calculation.

Signature

```
global String state {get; set;}
```

Property Value

Type: String

`street`

Street used in an address required for tax calculation.

Signature

```
global String street {get; set;}
```

Property Value

Type: String

TaxAddressRequest Methods

Learn more about the available methods with the `TaxAddressRequest` class.

The `TaxAddressRequest` class includes these methods.

IN THIS SECTION:

equals(obj)
Maintains the integrity of lists of type TaxAddressRequest by determining the equality of external objects in a list. This method is dynamic and based on the `equals()` method in Java.

hashCode()
Maintains the integrity of lists of type `TaxAddressRequest` by determining the uniqueness of the external object in a list.

toString()
Converts a date to a string.

`equals(obj)`

Maintains the integrity of lists of type TaxAddressRequest by determining the equality of external objects in a list. This method is dynamic and based on the `equals()` method in Java.

Signature

```
global Boolean equals(Object obj)
```

Parameters

obj
Type: Object

External object whose key is to be validated.

Return Value

Type: Boolean

`hashCode()`

Maintains the integrity of lists of type `TaxAddressRequest` by determining the uniqueness of the external object in a list.

Signature

```
global Integer hashCode()
```

Return Value

Type: Integer

`toString()`

Converts a date to a string.

Signature

```
global String toString()
```

Return Value

Type: String

TaxApiException Class

Contains details about any exceptions during the tax calculation process. Extends the `ApexBaseException` class.

Namespace

CommerceTax

IN THIS SECTION:

TaxApiException Constructors
Learn more about the available constructors with the `TaxApiException` class.

TaxApiException Constructors

Learn more about the available constructors with the `TaxApiException` class.

The `TaxApiException` class includes these constructors.

IN THIS SECTION:

TaxApiException(var1, var2)

Initializes the `TaxApiException` class using an `Exception` and a string to provide more details about the exception. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

TaxApiException(var1)

Initializes the `TaxApiException` class using an `Exception`. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

TaxApiException()

Initializes the `TaxApiException` class without any initialized parameters. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

`TaxApiException(var1, var2)`

Initializes the `TaxApiException` class using an `Exception` and a string to provide more details about the exception. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Signature

```
global TaxApiException(String var1, Exception var2)
```

Parameters

var1

Type: String

Text that provides more information about the returned exception.

var2

Type: Exception

An exception denotes an error that disrupts the normal flow of code execution. You can use Apex built-in exceptions or create custom exceptions. All exceptions have common methods.

`TaxApiException(var1)`

Initializes the `TaxApiException` class using an `Exception`. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Signature

```
global TaxApiException(Exception var1)
```

Parameters

var1
Type: Exception

An exception denotes an error that disrupts the normal flow of code execution. You can use Apex built-in exceptions or create custom exceptions. All exceptions have common methods.

`TaxApiException()`

Initializes the `TaxApiException` class without any initialized parameters. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Signature

```
global TaxApiException()
```

TaxCustomerDetailsRequest Class

Contains customer details used in tax calculation.

Namespace

CommerceTax

IN THIS SECTION:

TaxCustomerDetailsRequest Constructors
Learn more about the available constructors with the `TaxCustomerDetailsRequest` class.

TaxCustomerDetailsRequest Properties
Learn more about the available properties with the `TaxCustomerDetailsRequest` class.

TaxCustomerDetailsRequest Methods
Learn more about the available methods with the `TaxCustomerDetailsRequest` class.

TaxCustomerDetailsRequest Constructors

Learn more about the available constructors with the `TaxCustomerDetailsRequest` class.

The `TaxCustomerDetailsRequest` class includes these constructors.

IN THIS SECTION:

TaxCustomerDetailsRequest(accountId, code, exemptionNo, exemptionReason)
Initializes the `TaxCustomerDetailsRequest` object. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

`TaxCustomerDetailsRequest(accountId, code, exemptionNo, exemptionReason)`

Initializes the `TaxCustomerDetailsRequest` object. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Signature

```
global TaxCustomerDetailsRequest(String accountId, String code, String exemptionNo,
String exemptionReason)
```

Parameters

accountId
Type: String

The customer account ID for the line items sent for tax calculation.

code
Type: String

The tax code used during tax calculation.

exemptionNo
Type: String

The exemption number applied to any tax exempt line items.

exemptionReason
Type: String

The reason that certain line items are tax exempt.

TaxCustomerDetailsRequest Properties

Learn more about the available properties with the `TaxCustomerDetailsRequest` class.

The `TaxCustomerDetailsRequest` class includes these properties.

IN THIS SECTION:

accountId
Customer account that contains the line items sent for tax calculation.

code
Tax code used during tax calculation.

exemptionNo
Number used to qualify a line item for tax exemption.

exemptionReason
Reason why a line item qualifies for tax exemption.

taxCertificateId
ID of a tax certificate used for tax calculation.

accountId

Customer account that contains the line items sent for tax calculation.

Signature

```
global String accountId {get; set;}
```

Property Value

Type: String

code

Tax code used during tax calculation.

Signature

```
global String code {get; set;}
```

Property Value

Type: String

exemptionNo

Number used to qualify a line item for tax exemption.

Signature

```
global String exemptionNo {get; set;}
```

Property Value

Type: String

exemptionReason

Reason why a line item qualifies for tax exemption.

Signature

```
global String exemptionReason {get; set;}
```

Property Value

Type: String

taxCertificateId

ID of a tax certificate used for tax calculation.

Signature

```
global String taxCertificateId {get; set;}
```

Property Value

Type: String

TaxCustomerDetailsRequest Methods

Learn more about the available methods with the `TaxCustomerDetailsRequest` class.

The `TaxCustomerDetailsRequest` class includes these methods.

IN THIS SECTION:

equals(obj)

Maintains the integrity of lists of type `TaxCustomerDetailsRequest` by determining the equality of external objects in a list. This method is dynamic and based on the `equals()` method in Java.

hashCode()

Maintains the integrity of lists of type `TaxCustomerDetailsRequest` by determining the uniqueness of the external objects in a list.

toString()

Converts a value to a string.

`equals(obj)`

Maintains the integrity of lists of type `TaxCustomerDetailsRequest` by determining the equality of external objects in a list. This method is dynamic and based on the `equals()` method in Java.

Signature

```
global Boolean equals(Object obj)
```

Parameters

obj

Type: Object

External object whose key is to be validated.

Return Value

Type: Boolean

`hashCode()`

Maintains the integrity of lists of type `TaxCustomerDetailsRequest` by determining the uniqueness of the external objects in a list.

Signature

```
global Integer hashCode()
```

Return Value

Type: Integer

`toString()`

Converts a value to a string.

Signature

```
global String toString()
```

Return Value

Type: String

TaxDetailsResponse Class

Stores details of the tax values that an external tax engine calculates in response to a tax calculation request.

Namespace

CommerceTax

Usage

If your tax calculation request contains multiple line items, we recommend building your adapter using a list of `TaxDetailsResponse` instances. Each instance represents the tax details calculated for a given line item.

Example

```
List<commercetax.TaxDetailsResponse> taxDetailsResponses = new
List<commercetax.TaxDetailsResponse>();
                for(integer i =0;i<1;i++){
                    Integer rate  = 1;
                    Double taxableAmount = lineItem.amount;
                    commercetax.TaxDetailsResponse taxDetailsResponse = new
commercetax.TaxDetailsResponse();
                    taxDetailsResponse.setRate(Double.valueOf(rate));
                    taxDetailsResponse.setTaxableAmount(taxableAmount);
                    Double tax = taxableAmount*rate;
                    totalLineTax+=tax;
                    taxDetailsResponse.setTax(taxableAmount*rate);
                    taxDetailsResponse.setExemptAmount(0);
                    taxDetailsResponse.setExemptReason('exemptReason');
                    taxDetailsResponse.setTaxRegionId('taxRegionId');

taxDetailsResponse.setTaxId(String.valueOf(getRandomInteger(0,2323233)));
                    taxDetailsResponse.setSerCode('serCode');
                    taxDetailsResponse.setTaxAuthorityTypeId('taxAuthorityTypeId');
                    if(request.DocumentCode == 'SetsNullForResponseWithoutException'){
                        taxDetailsResponse.setImposition(null);
                    }else{
                        commercetax.ImpositionResponse imposition = new
commercetax.ImpositionResponse();
```

```
                        imposition.setSubType('subtype');
                        imposition.setType('type');
                        taxDetailsResponse.setImposition(imposition);
                    }

                    if(request.DocumentCode == 'SetsNullForResponseWithoutException'){
                        taxDetailsResponse.setJurisdiction(null);
                    }else{
                        commercetax.JurisdictionResponse jurisdiction = new
commercetax.JurisdictionResponse();
                        jurisdiction.setCountry('country');
                        jurisdiction.setRegion('region');
                        jurisdiction.setName('name');
                        jurisdiction.setStateAssignedNumber('stateAssignedNo');
                        jurisdiction.setId('id');
                        jurisdiction.setLevel('level');
                        taxDetailsResponse.setJurisdiction(jurisdiction);
                    }

                    taxDetailsResponses.add(taxDetailsResponse);
                }
                lineItemResponse.setTaxes(taxDetailsResponses);
                totalTax +=totalLineTax;
                totalAmount+=lineItem.amount;
```

IN THIS SECTION:

TaxDetailsResponse Methods
Learn more about the available methods with the `TaxDetailsResponse` class.

TaxDetailsResponse Methods

Learn more about the available methods with the `TaxDetailsResponse` class.

The `TaxDetailsResponse` class includes these methods.

IN THIS SECTION:

setExemptAmount(exemptAmount)
Sets the ExemptAmount field of the `TaxDetailsResponse` class.

setExemptReason(reason)
Sets the ExemptReason field of the `TaxDetailsResponse` class.

setImposition(imposition)
Sets the Imposition field of the `TaxDetailsResponse` class using an instance of the `ImpositionResponse` class.

setJurisdiction(jurisdiction)
Sets the Jurisdiction field of the `TaxDetailsResponse` using an instance of the `JurisdictionResponse` class.

setRate(rate)
Sets the Rate field of the `TaxDetailsResponse` class.

setSerCode(serCode)
Sets the Service Code field of the `TaxDetailsResponse` class.

setTax(tax)
Sets the Tax field of the `TaxDetailsResponse` class.

setTaxAuthorityTypeId(taxAuthorityTypeId)
Sets the TaxAuthorityTypeId field of the `TaxDetailsResponse` class.

setTaxId(taxId)
Sets the TaxId field of the `TaxDetailsResponse` class.

setTaxRegionId(taxRegionId)
Sets the TaxRegionId field on the `TaxDetailsResponse` class.

setTaxRuleDetails(taxRuleDetails)
Sets the TaxRuleDetails field of the `TaxDetailsResponse` class.

setTaxableAmount(taxableAmount)
Sets the TaxableAmount field of the `TaxDetailsResponse class`.

`setExemptAmount(exemptAmount)`

Sets the ExemptAmount field of the `TaxDetailsResponse` class.

Signature

```
global void setExemptAmount(Double exemptAmount)
```

Parameters

exemptAmount
Type: Double

Amount of tax on a line item that is exempt from tax calculation.

Return Value

Type: void

`setExemptReason(reason)`

Sets the ExemptReason field of the `TaxDetailsResponse` class.

Signature

```
global void setExemptReason(String reason)
```

Parameters

reason
Type: String

Optional user-defined information on why a tax exemption applies to a line item.

Return Value

Type: void

`setImposition(imposition)`

Sets the Imposition field of the `TaxDetailsResponse` class using an instance of the `ImpositionResponse` class.

Signature

```
global void setImposition(commercetax.ImpositionResponse imposition)
```

Parameters

`imposition`
Type: ImpositionResponse

Contains information about why tax was imposed on a line item.

Return Value

Type: void

`setJurisdiction(jurisdiction)`

Sets the Jurisdiction field of the `TaxDetailsResponse` using an instance of the `JurisdictionResponse` class.

Signature

```
global void setJurisdiction(commercetax.JurisdictionResponse jurisdiction)
```

Parameters

`jurisdiction`
Type: JurisdictionResponse

Contains address information about the tax jurisdiction used in the tax calculation process.

Return Value

Type: void

`setRate(rate)`

Sets the Rate field of the `TaxDetailsResponse` class.

Signature

```
global void setRate(Double rate)
```

Parameters

rate
Type: Double

Tax used during tax calculation. This value is often a decimal amount, such as 0.1 or 0.06, based on the applied tax percentage.

Return Value

Type: void

setSerCode(serCode)

Sets the Service Code field of the `TaxDetailsResponse` class.

Signature

```
global void setSerCode(String serCode)
```

Parameters

serCode
Type: String

Service code used in tax calculation.

Return Value

Type: void

setTax(tax)

Sets the Tax field of the `TaxDetailsResponse` class.

Signature

```
global void setTax(Double tax)
```

Parameters

tax
Type: Double

Amount of tax for a line item.

Return Value

Type: void

setTaxAuthorityTypeId(taxAuthorityTypeId)

Sets the TaxAuthorityTypeId field of the `TaxDetailsResponse` class.

Signature

```
global void setTaxAuthorityTypeId(String taxAuthorityTypeId)
```

Parameters

taxAuthorityTypeId
Type: String

ID of the organization that oversees tax collection.

Return Value

Type: void

setTaxId(taxId)

Sets the TaxId field of the `TaxDetailsResponse` class.

Signature

```
global void setTaxId(String taxId)
```

Parameters

taxId
Type: String

ID value used to determine the tax for an individual or business.

Return Value

Type: void

setTaxRegionId(taxRegionId)

Sets the TaxRegionId field on the `TaxDetailsResponse` class.

Signature

```
global void setTaxRegionId(String taxRegionId)
```

Parameters

taxRegionId
Type: String

ID of the tax region used in tax calculation. A tax region represents a geographical area where tax is applied.

Return Value

Type: void

`setTaxRuleDetails(taxRuleDetails)`

Sets the TaxRuleDetails field of the `TaxDetailsResponse` class.

Signature

```
global void setTaxRuleDetails(commercetax.RuleDetailsResponse taxRuleDetails)
```

Parameters

`taxRuleDetails`
Type: RuleDetailsResponse

Information about the Salesforce tax rules used during tax calculation.

Return Value

Type: void

`setTaxableAmount(taxableAmount)`

Sets the TaxableAmount field of the `TaxDetailsResponse class`.

Signature

```
global void setTaxableAmount(Double taxableAmount)
```

Parameters

`taxableAmount`
Type: Double

Amount that can be taxed on a line item.

Return Value

Type: void

TaxEngineAdapter Interface

Retrieves information from the tax engine and evaluates the information to define tax details.

Namespace

CommerceTax

IN THIS SECTION:

TaxEngineAdapter Methods
Learn more about the available methods with the `TaxEngineAdapter` class.

TaxEngineAdapter Example Implementation
Refer to the example implementation of the `TaxEngineAdapter` interface to accept information from a tax engine and evaluate the information to define tax details.

TaxEngineAdapter Methods

Learn more about the available methods with the `TaxEngineAdapter` class.

The `TaxEngineAdapter` class includes these methods.

IN THIS SECTION:

processRequest(requestType)
The `processRequest` method takes an instance of `TaxEngineContext` class and returns a response with the calculated tax details through the `TaxDetailsResponse` class or an error response through the `ErrorResponse` class.

`processRequest(requestType)`

The `processRequest` method takes an instance of `TaxEngineContext` class and returns a response with the calculated tax details through the `TaxDetailsResponse` class or an error response through the `ErrorResponse` class.

Signature

```
global commercetax.TaxEngineResponse processRequest(commercetax.TaxEngineContext var1)
```

Parameters

var1
Type: TaxEngineContext

Wrapper class that stores information about the type of a tax calculation request.

Return Value

Type: TaxEngineResponse

Generic interface representing a response from a tax engine.

TaxEngineAdapter Example Implementation

Refer to the example implementation of the `TaxEngineAdapter` interface to accept information from a tax engine and evaluate the information to define tax details.

Namespace

commercetax

Usage

The `TaxEngineAdapter` interface accepts information from the tax engine through the `TaxEngineContext` class. The interface evaluates the information to define tax in the response with details, such as tax amount and addresses. The response is used to update and create entities in the Salesforce org.

Use these steps to build a sample tax adapter implementation. Each tax adapter implementation varies based on your implementation requirements. Customize this example to suit your business requirements.

Example:

- The custom adapter class implements the `TaxEngineAdapter` interface. The `processRequest` method takes an instance of `TaxEngineContext` class and returns a response with the calculated tax details through the `TaxDetailsResponse` class or an error response through the `ErrorResponse` class.

```
global virtual class AvalaraAdapter implements commercetax.TaxEngineAdapter {
    global commercetax.TaxEngineResponse processRequest(commercetax.TaxEngineContext
 taxEngineContext) {
        commercetax.RequestType requestType = taxEngineContext.getRequestType();
        if(requestType == commercetax.RequestType.CalculateTax){
            return CalculateTaxService.getTax(taxEngineContext);
        }
        else
            return null;
    }
}
```

- This example shows the `CalculateTaxService` class.

```
global class CalculateTaxService {
    // ========================================================================

    // CONSTANT
    // ========================================================================

    private static final String AVALARA_ENDPOINT_URL_SANDBOX =
'https://sandbox-rest.avatax.com/api/v2';
    // Avalara Endpoint URL Production
    private static final String AVALARA_ENDPOINT_URL_PRODUCTION =
'https://rest.avatax.com/api/v2';
   private static final String TEST_REQUEST_BODY = '{  "id": -1,  "code": "00000131",
  "companyId": -1,  "date": "2017-02-03T00:00:00",  "taxDate": "2017-02-03T00:00:00",
  "status": "Temporary",  "type": "SalesOrder",  "reconciled": false,  "totalAmount":
 4000,  "totalExempt": 0,  "totalTax": 290,  "totalTaxable": 4000,
"totalTaxCalculated": 290,  "adjustmentReason": "NotAdjusted",  "locked": false,
"version": 1,  "modifiedDate": "2017-02-03T12:18:18.7347388Z",  "modifiedUserId":
53894,  "lines": [    {      "id": -1,      "transactionId": -1,      "lineNumber":
 "80241000000jNDCAA2",      "discountAmount": 0,      "exemptAmount": 0,
"exemptCertId": 0,      "isItemTaxable": true,      "lineAmount": 1000,
"reportingDate": "2017-02-03T00:00:00",      "tax": 72.5,      "taxableAmount":
1000,      "taxCalculated": 72.5,      "taxCode": "P0000000",      "taxDate":
"2017-02-03T00:00:00",      "taxIncluded": false,      "details": [        {
    "id": -1,          "transactionLineId": -1,          "transactionId": -1,
     "country": "US",          "region": "CA",          "exemptAmount": 0,
  "jurisCode": "06",          "jurisName": "CALIFORNIA",          "stateAssignedNo":
```

```
 "",          "jurisType": "STA",          "nonTaxableAmount": 0,          "rate":
 0.06,          "tax": 60,          "taxableAmount": 1000,          "taxType":
"Sales",          "taxName": "CA STATE TAX",          "taxAuthorityTypeId": 45,
      "taxCalculated": 60,          "rateType": "General"        },        {
     "id": -1,          "transactionLineId": -1,          "transactionId": -1,
      "country": "US",          "region": "CA",          "exemptAmount": 0,
   "jurisCode": "075",          "jurisName": "SAN FRANCISCO",
"stateAssignedNo": "",          "jurisType": "CTY",          "nonTaxableAmount": 0,
          "rate": 0.0025,          "tax": 2.5,          "taxableAmount": 1000,
      "taxType": "Sales",          "taxName": "CA COUNTY TAX",
"taxAuthorityTypeId": 45,          "taxCalculated": 2.5,          "rateType":
"General"        },        {          "id": -1,          "transactionLineId": -1,
        "transactionId": -1,          "country": "US",          "region": "CA",
       "exemptAmount": 0,          "jurisCode": "EMTV0",          "jurisName": "SAN
 FRANCISCO CO LOCAL TAX SL",          "stateAssignedNo": "38",          "jurisType":
 "STJ",          "nonTaxableAmount": 0,          "rate": 0.01,          "tax": 10,
         "taxableAmount": 1000,          "taxType": "Sales",          "taxName":
"CA SPECIAL TAX",          "taxAuthorityTypeId": 45,          "taxCalculated": 10,
          "rateType": "General"        }      ]    }  ]}';

    private static String getTestResponseString(){

     List<String> jsonResponse = new List<String> {
                                    '"id": 0',
                                    '"code": "testDocCode1231245984"',
                                    '"companyId": 468039',
                                    '"date": "2020-07-15"',
                                    '"paymentDate": "2020-07-15"',
                                    '"status": "Temporary"',
                                    '"type": "SalesOrder"',
                                    '"customerVendorCode": "testDocCode1234"',
                                    '"customerCode": "testDocCode1234"',
                                    '"reconciled": false',
                                    '"totalAmount": 232',
                                    '"totalExempt": 0',
                                    '"totalDiscount": 0',
                                    '"totalTax": 23.43',
                                    '"totalTaxable": 232',
                                    '"totalTaxCalculated": 23.43',
                                    '"adjustmentReason": "NotAdjusted"',
                                    '"locked": false',
                                    '"version": 1',
                                    '"exchangeRateEffectiveDate": "2020-07-15"',
                                    '"exchangeRate": 1',
                                  '"modifiedDate": "2020-08-13T11:19:20.4836636Z"',

                                    '"modifiedUserId": 53894',
                                    '"taxDate": "2020-07-15T00:00:00"',
                                    '"lines": [{"id": 0,"transactionId":
0,"lineNumber": "1","discountAmount": 0,"exemptAmount": 0,"exemptCertId":
0,"isItemTaxable": true,"itemCode": "","lineAmount": 232,"quantity":
1,"reportingDate": "2020-07-15","tax": 23.43,"taxableAmount": 232,"taxCalculated":
 23.43,"taxCode": "P0000000","taxCodeId": 8087,"taxDate":
"2020-07-15","taxOverrideType": "None","taxOverrideAmount": 0,"taxIncluded":
```

```
false,"details": [{"id": 0,"transactionLineId": 0,"transactionId": 0,"country":
"US","region": "WA","exemptAmount": 0,"jurisCode": "53","jurisName":
"WASHINGTON","stateAssignedNo": "","jurisType": "STA","jurisdictionType":
"State","nonTaxableAmount": 0,"rate": 0.065,"tax": 15.08,"taxableAmount":
232,"taxType": "Sales","taxSubTypeId": "S","taxName": "WA STATE
TAX","taxAuthorityTypeId": 45,"taxCalculated": 15.08,"rateType":
"General","rateTypeCode": "G","unitOfBasis": "PerCurrencyUnit","isNonPassThru":
false,"isFee": false},{"id": 0,"transactionLineId": 0,"transactionId": 0,"country":
 "US","region": "WA","exemptAmount": 0,"jurisCode": "033","jurisName":
"KING","stateAssignedNo": "1700","jurisType": "CTY","jurisdictionType":
"County","nonTaxableAmount": 0,"rate": 0,"tax": 0,"taxableAmount": 232,"taxType":
"Sales","taxSubTypeId": "S","taxName": "WA COUNTY TAX","taxAuthorityTypeId":
45,"taxCalculated": 0,"rateType": "General","rateTypeCode": "G","unitOfBasis":
"PerCurrencyUnit","isNonPassThru": false,"isFee": false}],"nonPassthroughDetails":
 [],"hsCode": "","costInsuranceFreight": 0,"vatCode": "","vatNumberTypeId": 0}]',
                                    '"addresses": [{"id": 0,"transactionId":
0,"boundaryLevel": "Address","line1": "255 S. King Street","line2": "","line3":
"","city": "Seattle","region": "WA","postalCode": "98104","country":
"US","taxRegionId": 2109700,"latitude": "47.59821","longitude": "-122.33108"}]',
                                    '"summary": [{"country": "US","region":
"WA","jurisType": "State","jurisCode": "53","jurisName":
"WASHINGTON","taxAuthorityType": 45,"stateAssignedNo": "","taxType":
"Sales","taxSubType": "S","taxName": "WA STATE TAX","rateType": "General","taxable":
 232,"rate": 0.065,"tax": 15.08,"taxCalculated": 15.08,"nonTaxable": 0,"exemption":
 0},{"country": "US","region": "WA","jurisType": "County","jurisCode":
"033","jurisName": "KING","taxAuthorityType": 45,"stateAssignedNo": "1700","taxType":
 "Sales","taxSubType": "S","taxName": "WA COUNTY TAX","rateType": "General","taxable":
 232,"rate": 0,"tax": 0,"taxCalculated": 0,"nonTaxable": 0,"exemption": 0}]'
                                };
            return '{' + String.join(jsonResponse, ',') + '}';
        }

   public static commercetax.TaxEngineResponse getTax(commercetax.TaxEngineContext
 taxEngineContext)
    {
        commercetax.CalculateTaxRequest request =
(commercetax.CalculateTaxRequest)taxEngineContext.getRequest();
        commercetax.calculatetaxtype requestType = request.taxtype;
        string referenceEntity = request.ReferenceEntityId;
        try{
            List<commercetax.TaxLineItemRequest> listOfLines = request.lineItems;
            if(!listOfLines.isEmpty()){
                HttpService sendHttpRequest = new HttpService();
                sendHttpRequest.addHeader('Content-type', 'application/json');
                String requestBody =
AvalaraJSONBuilder.getInstance().frameJsonForGetTaxOrderItem(request);
                sendHttpRequest.post('/transactions/create',requestBody);
                //system.debug('Request '+requestBody);
                String responseString = '';
                if(Test.isRunningTest()){
                    responseString = getTestResponseString();
                } else{
                    responseString = sendHttpRequest.getResponse().getBody();
                }
```

```
                //system.debug(sendHttpRequest.getResponse());
                //system.debug('response'+responseString);
                //responseString = TEST_REQUEST_BODY;
                system.debug('Heap size used ' +Limits.getHeapSize());

                if(!responseString.contains('error'))
                {
                    commercetax.CalculateTaxResponse response = new
commercetax.CalculateTaxResponse();
                    JsonSuccessParser jsonSuccessParserClass =
JsonSuccessParser.parse(responseString);
                    response.setTaxTransactionType(request.taxTransactionType);
                    response.setDocumentCode(jsonSuccessParserClass.code);

response.setReferenceDocumentCode(jsonSuccessParserClass.referenceCode);
                    if(jsonSuccessParserClass.status == 'Temporary')  {

response.setStatus(commercetax.TaxTransactionStatus.Uncommitted);
                    }
                    if(jsonSuccessParserClass.status == 'Committed') {

response.setStatus(commercetax.TaxTransactionStatus.Committed);
                    }
                    response.setTaxType(requestType);
                    commercetax.AmountDetailsResponse headerAmountResponse = new
commercetax.AmountDetailsResponse();

headerAmountResponse.setTotalAmountWithTax(jsonSuccessParserClass.totalAmount +
jsonSuccessParserClass.totaltax);

headerAmountResponse.setExemptAmount(jsonSuccessParserClass.totalExempt);

headerAmountResponse.setTotalAmount(jsonSuccessParserClass.totalAmount);

headerAmountResponse.setTaxAmount(jsonSuccessParserClass.totalTax);
                    response.setAmountDetails(headerAmountResponse);

response.setStatusDescription(jsonSuccessParserClass.adjustmentReason);

response.setEffectiveDate(date.valueof(jsonSuccessParserClass.taxDate));

response.setTransactionDate(date.valueof(jsonSuccessParserClass.transactionDate));
                    response.setReferenceEntityId(referenceEntity);
                    response.setTaxTransactionId(jsonSuccessParserClass.id);
                    response.setCurrencyIsoCode(request.currencyIsoCode);
                    List<commercetax.LineItemResponse> lineItemResponses = new
List<commercetax.LineItemResponse>();
                    for(JsonSuccessParser.Lines linesToProcess:
jsonSuccessParserClass.lines)
                    {
                        commercetax.LineItemResponse lineItemResponse = new
commercetax.LineItemResponse();
                        Double rateCalculated = 0.0;
                        List<commercetax.TaxDetailsResponse> taxDetailsResponses =
```

```
 new List<commercetax.TaxDetailsResponse>();
                        for(JsonSuccessParser.details linesDetails :
linesToProcess.details)
                        {
                            commercetax.TaxDetailsResponse taxDetailsResponse = new
 commercetax.TaxDetailsResponse();
                             if(linesDetails.exemptAmount != 0){

taxDetailsResponse.setExemptAmount(linesDetails.exemptAmount);
                                 taxDetailsResponse.setExemptReason('Some reason we
 dont know');
                             }
                                 commercetax.ImpositionResponse imposition = new
commercetax.ImpositionResponse();
                                     imposition.setSubType(linesDetails.taxName);
                                     imposition.setType(linesDetails.ratetype);
                                     imposition.setSubType(linesDetails.taxName);
                                     taxDetailsResponse.setImposition(imposition);
                                commercetax.JurisdictionResponse jurisdiction = new
 commercetax.JurisdictionResponse();
                                     jurisdiction.setCountry(linesDetails.country);
                                     jurisdiction.setRegion(linesDetails.region);
                                     jurisdiction.setName(linesDetails.jurisName);

jurisdiction.setStateAssignedNumber(linesDetails.stateAssignedNo);
                                     jurisdiction.setId(linesDetails.jurisCode);
                                     jurisdiction.setLevel(linesDetails.jurisType);
                                  taxDetailsResponse.setJurisdiction(jurisdiction);

                                     rateCalculated += linesDetails.rate;
                                 taxDetailsResponse.setRate(rateCalculated);
                              taxDetailsResponse.setTax(linesDetails.taxCalculated);


taxDetailsResponse.setTaxableAmount(linesDetails.taxableAmount);

taxDetailsResponse.setTaxAuthorityTypeId(String.valueOf(linesDetails.taxAuthorityTypeId));

                                 taxDetailsResponse.setTaxId(linesDetails.id);

taxDetailsResponse.setTaxRegionId(linesDetails.region);
                                 taxDetailsResponses.add(taxDetailsResponse);

                        }
                             lineItemResponse.setTaxes(taxDetailsResponses);

lineItemResponse.setEffectiveDate(date.valueof(linesToProcess.taxDate));
                             lineItemResponse.setIsTaxable(true);
                                 commercetax.AmountDetailsResponse amountResponse =
 new commercetax.AmountDetailsResponse();

amountResponse.setTaxAmount(linesToProcess.taxCalculated);

amountResponse.setTotalAmount(linesToProcess.lineAmount);
```

```
amountResponse.setTotalAmountWithTax(linesToProcess.lineAmount+linesToProcess.taxCalculated);


amountResponse.setExemptAmount(linesToProcess.exemptAmount);
                                lineItemResponse.setAmountDetails(amountResponse);

lineItemResponse.setIsTaxable(linesToProcess.isItemTaxable);
                         lineItemResponse.setProductCode(linesToProcess.itemCode);

                            lineItemResponse.setTaxCode(linesToProcess.taxCode);
                        lineItemResponse.setLineNumber(linesToProcess.lineNumber);

                            lineItemResponse.setQuantity(linesToProcess.quantity);
                            lineItemResponses.add(lineItemResponse);
                    }
                    response.setLineItems(lineItemResponses);
                    return response;
                }
                else
                {
                    JsonErrorParser jsonErrorParserClass =
JsonErrorParser.parse(responseString);
                    String message = null;
                    if(String.isNotBlank(jsonErrorParserClass.error.message))
                    {
                       message=jsonErrorParserClass.error.message;
                    }else{
                           String errorMessage = '';
                            for(JsonErrorParser.cls_details messageString :
jsonErrorParserClass.error.details)
                            {
                                if(String.isNotBlank(messageString.message) )
                                {
                                    errorMessage = messageString.message;
                                }
                            }
                            message = errorMessage;
                        }
                     return new
commercetax.ErrorResponse(commercetax.resultcode.TaxEngineError, '501', message);

                }
            }else return null;
        }
        catch (Exception e)
        {
            throw e;
        }
    }
}
```

- In the `HttpService` class, replace the `test` value in the endpoint variable with the name of the `TaxTypedNamedCredential` record. This class contains the credentials that are required to access your Avalara account through Salesforce.

```
public with sharing class HttpService
{
    // Attribute to implement singleton pattern for Order Product Service class
    private static HttpService httpServiceInstance;

    // VARIABLES

    private HttpResponse httpResponse;
    private Map<String,String> mapOfHeaderParameter = new Map<String,String>();
    private enum Method {GET, POST}

    /**
    * @name getInstance
    * @description get an Instance of Service class
    * @params NA
    * @return Http Service Class Instance
    */
    public static HttpService getInstance()
    {
        if (NULL == httpServiceInstance)
        {
            httpServiceInstance =  new HttpService();
        }
        return httpServiceInstance;
    }

    /**
    * @name get
    * @description Get Method to get a HTTP request
    */
    public void get(String endPoint)
    {
        send(newRequest(Method.GET, endPoint));
    }

    /**
    * @name post
    * @description Post Method to Post a HTTP request
    */
    public void post(String path, String requestBody)
    {
       String endPoint = 'callout:commerce.tax.TaxTypedNamedCredential:test'+path;
        send(newRequest(Method.POST, endPoint, requestBody));
    }

    /**
    * @name addHeader
    * @description addHeader Methods to add all the defualt Header's required fo
rthe request
```

```
    */
    public void addHeader(String name, String value)
    {
        mapOfHeaderParameter.put(name, value);
    }

    /**
    * @name setHeader
    * @description setHeader Methods to set setHeader for the request
    */
    private void setHeader(HttpRequest request)
    {
        for(String headerValue : mapOfHeaderParameter.keySet())
        {
            request.setHeader(headerValue, mapOfHeaderParameter.get(headerValue));
        }
    }
    /**
    * @name newRequest
    * @description newRequest Methods to make a new request
    */
    private HttpRequest newRequest(Method method, String endPoint)
    {
        return newRequest(method, endPoint, NULL);
    }

    /**
    * @name newRequest
    * @description newRequest Methods to make a new request
    */
   private HttpRequest newRequest(Method method, String endPoint, String requestBody)

    {
        HttpRequest request = new HttpRequest();
        request.setMethod(Method.name());
        setHeader(request);
        request.setEndpoint(endPoint);
        if (String.isNotBlank(requestBody))
        {
            request.setBody(requestBody);
        }
        request.setTimeout(120000);
        return request;
    }

    /**
    * @name send
    * @description send Methods to send a request
    */
    private void send(HttpRequest request)
    {
        try
        {
            Http http = new Http();
```

```
            httpResponse = http.send(request);
        }
        catch(System.CalloutException e)
        {
            system.debug('callout exception happened' + e.getMessage());
        }
        catch(Exception e)
        {
            system.debug('callout did not happen' + e.getMessage());
        }
    }

    /**
    * @name getResponse
    * @description getResponse Method to get the Response
    */
    public HttpResponse getResponse()
    {
        return httpResponse;
    }

    /**
    * @name getResponseToString
    * @description getResponse Method to get the Responses
    */
    public String getResponseToString()
    {
        return getResponse().toString();
    }
}
```

- Parse the `JsonSuccessParser` response object by using the `AvalaraJSONBuilder` class to build the response for your adapter.

  This example shows the `JsonSuccessParser` class.

```
global with sharing class JsonSuccessParser
{
  public static void consumeObject(JSONParser parser)
  {
    Integer depth = 0;
    do {
      JSONToken curr = parser.getCurrentToken();
      if (curr == JSONToken.START_OBJECT ||
        curr == JSONToken.START_ARRAY) {
        depth++;
      } else if (curr == JSONToken.END_OBJECT ||
        curr == JSONToken.END_ARRAY) {
        depth--;
      }
    } while (depth > 0 && parser.nextToken() != null);
  }

    public class Addresses {
        public String id {get;set;}
```

```
    public String transactionId {get;set;}
    public String boundaryLevel {get;set;}
    public String line1 {get;set;}
    public String city {get;set;}
    public String region {get;set;}
    public String postalCode {get;set;}
    public String country {get;set;}
    public Integer taxRegionId {get;set;}

    public Addresses(JSONParser parser) {
        while (parser.nextToken() != JSONToken.END_OBJECT) {
            if (parser.getCurrentToken() == JSONToken.FIELD_NAME) {
                String text = parser.getText();
                if (parser.nextToken() != JSONToken.VALUE_NULL) {
                    if (text == 'id') {
                        id = parser.getText();
                    } else if (text == 'transactionId') {
                        transactionId = parser.getText();
                    } else if (text == 'boundaryLevel') {
                        boundaryLevel = parser.getText();
                    } else if (text == 'line1') {
                        line1 = parser.getText();
                    } else if (text == 'city') {
                        city = parser.getText();
                    } else if (text == 'region') {
                        region = parser.getText();
                    } else if (text == 'postalCode') {
                        postalCode = parser.getText();
                    } else if (text == 'country') {
                        country = parser.getText();
                    } else if (text == 'taxRegionId') {
                        taxRegionId = parser.getIntegerValue();
                    } else {
                        consumeObject(parser);
                    }
                }
            }
        }
    }
}

public class Details {
    public String id {get;set;}
    public String transactionLineId {get;set;}
    public String transactionId {get;set;}
    public String country {get;set;}
    public String region {get;set;}
    public Integer exemptAmount {get;set;}
    public String jurisCode {get;set;}
    public String jurisName {get;set;}
    public String stateAssignedNo {get;set;}
    public String jurisType {get;set;}
    public Integer nonTaxableAmount {get;set;}
    public Double rate {get;set;}
```

```
        public Double tax {get;set;}
        public Integer taxableAmount {get;set;}
        public String taxType {get;set;}
        public String taxName {get;set;}
        public Integer taxAuthorityTypeId {get;set;}
        public Double taxCalculated {get;set;}
        public String rateType {get;set;}

        public Details(JSONParser parser) {
            while (parser.nextToken() != JSONToken.END_OBJECT) {
                if (parser.getCurrentToken() == JSONToken.FIELD_NAME) {
                    String text = parser.getText();
                    if (parser.nextToken() != JSONToken.VALUE_NULL) {
                        if (text == 'id') {
                            id = parser.getText();
                        } else if (text == 'transactionLineId') {
                            transactionLineId = parser.getText();
                        } else if (text == 'transactionId') {
                            transactionId = parser.getText();
                        } else if (text == 'country') {
                            country = parser.getText();
                        } else if (text == 'region') {
                            region = parser.getText();
                        } else if (text == 'exemptAmount') {
                            exemptAmount = parser.getIntegerValue();
                        } else if (text == 'jurisCode') {
                            jurisCode = parser.getText();
                        } else if (text == 'jurisName') {
                            jurisName = parser.getText();
                        } else if (text == 'stateAssignedNo') {
                            stateAssignedNo = parser.getText();
                        } else if (text == 'jurisType') {
                            jurisType = parser.getText();
                        } else if (text == 'nonTaxableAmount') {
                            nonTaxableAmount = parser.getIntegerValue();
                        } else if (text == 'rate') {
                            rate = parser.getDoubleValue();
                        } else if (text == 'tax') {
                            tax = parser.getDoubleValue();
                        } else if (text == 'taxableAmount') {
                            taxableAmount = parser.getIntegerValue();
                        } else if (text == 'taxType') {
                            taxType = parser.getText();
                        } else if (text == 'taxName') {
                            taxName = parser.getText();
                        } else if (text == 'taxAuthorityTypeId') {
                            taxAuthorityTypeId = parser.getIntegerValue();
                        } else if (text == 'taxCalculated') {
                            taxCalculated = parser.getDoubleValue();
                        } else if (text == 'rateType') {
                            rateType = parser.getText();
                        } else {
                            consumeObject(parser);
                        }
```

```
                }
            }
        }
    }
}

public class Messages {
    public String summary {get;set;}
    public String details {get;set;}
    public String refersTo {get;set;}
    public String severity {get;set;}
    public String source {get;set;}

    public Messages(JSONParser parser) {
        while (parser.nextToken() != JSONToken.END_OBJECT) {
            if (parser.getCurrentToken() == JSONToken.FIELD_NAME) {
                String text = parser.getText();
                if (parser.nextToken() != JSONToken.VALUE_NULL) {
                    if (text == 'summary') {
                        summary = parser.getText();
                    } else if (text == 'details') {
                        details = parser.getText();
                    } else if (text == 'refersTo') {
                        refersTo = parser.getText();
                    } else if (text == 'severity') {
                        severity = parser.getText();
                    } else if (text == 'source') {
                        source = parser.getText();
                    } else {
                        consumeObject(parser);
                    }
                }
            }
        }
    }
}

public String id {get;set;}
public String code {get;set;}
public String referenceCode {get;set;}
public Integer companyId {get;set;}
public String taxDate {get;set;}
public String transactionDate {get;set;}
public String status {get;set;}
public String type_Z {get;set;} // in json: type
public Boolean reconciled {get;set;}
public Integer totalAmount {get;set;}
public Integer totalExempt {get;set;}
public Double totalTax {get;set;}
public Integer totalTaxable {get;set;}
public Double totalTaxCalculated {get;set;}
public String adjustmentReason {get;set;}
public Boolean locked {get;set;}
public Integer version {get;set;}
```

```
public String modifiedDate {get;set;}
public Integer modifiedUserId {get;set;}
public List<Lines> lines {get;set;}
public List<Addresses> addresses {get;set;}
public List<Summary> summary {get;set;}
public List<Messages> messages {get;set;}

public JsonSuccessParser(JSONParser parser) {
    while (parser.nextToken() != JSONToken.END_OBJECT) {
        if (parser.getCurrentToken() == JSONToken.FIELD_NAME) {
            String text = parser.getText();
            if (parser.nextToken() != JSONToken.VALUE_NULL) {
                if (text == 'id') {
                    id = parser.getText();
                } else if (text == 'code') {
                    code = parser.getText();
                } else if (text == 'referenceCode'){
                    referenceCode = parser.getText();
                } else if (text == 'companyId') {
                    companyId = parser.getIntegerValue();
                } else if (text == 'taxDate') {
                    taxDate = parser.getText();
                } else if (text == 'date') {
                    transactionDate = parser.getText();
                } else if (text == 'status') {
                    status = parser.getText();
                } else if (text == 'type') {
                    type_Z = parser.getText();
                } else if (text == 'reconciled') {
                    reconciled = parser.getBooleanValue();
                } else if (text == 'totalAmount') {
                    totalAmount = parser.getIntegerValue();
                } else if (text == 'totalExempt') {
                    totalExempt = parser.getIntegerValue();
                } else if (text == 'totalTax') {
                    totalTax = parser.getDoubleValue();
                } else if (text == 'totalTaxable') {
                    totalTaxable = parser.getIntegerValue();
                } else if (text == 'totalTaxCalculated') {
                    totalTaxCalculated = parser.getDoubleValue();
                } else if (text == 'adjustmentReason') {
                    adjustmentReason = parser.getText();
                } else if (text == 'locked') {
                    locked = parser.getBooleanValue();
                } else if (text == 'version') {
                    version = parser.getIntegerValue();
                } else if (text == 'modifiedDate') {
                    modifiedDate = parser.getText();
                } else if (text == 'modifiedUserId') {
                    modifiedUserId = parser.getIntegerValue();
                } else if (text == 'lines') {
                    lines = new List<Lines>();
                    while (parser.nextToken() != JSONToken.END_ARRAY) {
                        lines.add(new Lines(parser));
```

```
                    }
                } else if (text == 'addresses') {
                    addresses = new List<Addresses>();
                    while (parser.nextToken() != JSONToken.END_ARRAY) {
                        addresses.add(new Addresses(parser));
                    }
                } else if (text == 'summary') {
                    summary = new List<Summary>();
                    while (parser.nextToken() != JSONToken.END_ARRAY) {
                        summary.add(new Summary(parser));
                    }
                } else if (text == 'messages') {
                    messages = new List<Messages>();
                    while (parser.nextToken() != JSONToken.END_ARRAY) {
                        messages.add(new Messages(parser));
                    }
                } else {
                    consumeObject(parser);
                }
            }
        }
    }
}

public class Summary {
    public String country {get;set;}
    public String region {get;set;}
    public String jurisType {get;set;}
    public String jurisCode {get;set;}
    public String jurisName {get;set;}
    public Integer taxAuthorityType {get;set;}
    public String stateAssignedNo {get;set;}
    public String taxType {get;set;}
    public String taxName {get;set;}
    public String taxGroup {get;set;}
    public String rateType {get;set;}
    public Integer taxable {get;set;}
    public Double rate {get;set;}
    public Double tax {get;set;}
    public Double taxCalculated {get;set;}
    public Integer nonTaxable {get;set;}
    public Integer exemption {get;set;}

    public Summary(JSONParser parser) {
        while (parser.nextToken() != JSONToken.END_OBJECT) {
            if (parser.getCurrentToken() == JSONToken.FIELD_NAME) {
                String text = parser.getText();
                if (parser.nextToken() != JSONToken.VALUE_NULL) {
                    if (text == 'country') {
                        country = parser.getText();
                    } else if (text == 'region') {
                        region = parser.getText();
                    } else if (text == 'jurisType') {
                        jurisType = parser.getText();
```

```
                        } else if (text == 'jurisCode') {
                            jurisCode = parser.getText();
                        } else if (text == 'jurisName') {
                            jurisName = parser.getText();
                        } else if (text == 'taxAuthorityType') {
                            taxAuthorityType = parser.getIntegerValue();
                        } else if (text == 'stateAssignedNo') {
                            stateAssignedNo = parser.getText();
                        } else if (text == 'taxType') {
                            taxType = parser.getText();
                        } else if (text == 'taxName') {
                            taxName = parser.getText();
                        } else if (text == 'taxGroup') {
                            taxGroup = parser.getText();
                        } else if (text == 'rateType') {
                            rateType = parser.getText();
                        } else if (text == 'taxable') {
                            taxable = parser.getIntegerValue();
                        } else if (text == 'rate') {
                            rate = parser.getDoubleValue();
                        } else if (text == 'tax') {
                            tax = parser.getDoubleValue();
                        } else if (text == 'taxCalculated') {
                            taxCalculated = parser.getDoubleValue();
                        } else if (text == 'nonTaxable') {
                            nonTaxable = parser.getIntegerValue();
                        } else if (text == 'exemption') {
                            exemption = parser.getIntegerValue();
                        } else {
                            consumeObject(parser);
                        }
                    }
                }
            }
        }
    }

    public class Lines {
        public String id {get;set;}
        public String transactionId {get;set;}
        public String lineNumber {get;set;}
        public Integer discountAmount {get;set;}
        public Integer exemptAmount {get;set;}
        public Integer exemptCertId {get;set;}
        public Boolean isItemTaxable {get;set;}
        public Integer lineAmount {get;set;}
        public Double quantity {get;set;}
        public String reportingDate {get;set;}
        public Double tax {get;set;}
        public Integer taxableAmount {get;set;}
        public Double taxCalculated {get;set;}
        public String taxCode {get;set;}
        public String taxDate {get;set;}
        public Boolean taxIncluded {get;set;}
```

```
        public List<Details> details {get;set;}
        public String itemCode {get;set;}
        public Lines(JSONParser parser) {
            while (parser.nextToken() != JSONToken.END_OBJECT) {
                if (parser.getCurrentToken() == JSONToken.FIELD_NAME) {
                    String text = parser.getText();
                    if (parser.nextToken() != JSONToken.VALUE_NULL) {
                        if (text == 'id') {
                            id = parser.getText();
                        } else if (text == 'transactionId') {
                            transactionId = parser.getText();
                        }else if (text == 'itemCode') {
                            itemCode = parser.getText();
                        }else if (text == 'lineNumber') {
                            lineNumber = parser.getText();
                        } else if (text == 'discountAmount') {
                            discountAmount = parser.getIntegerValue();
                        } else if (text == 'exemptAmount') {
                            exemptAmount = parser.getIntegerValue();
                        } else if (text == 'exemptCertId') {
                            exemptCertId = parser.getIntegerValue();
                        } else if (text == 'isItemTaxable') {
                            isItemTaxable = parser.getBooleanValue();
                        } else if (text == 'lineAmount') {
                            lineAmount = parser.getIntegerValue();
                        } else if (text == 'quantity') {
                            quantity = parser.getDoubleValue();
                        } else if (text == 'reportingDate') {
                            reportingDate = parser.getText();
                        } else if (text == 'tax') {
                            tax = parser.getDoubleValue();
                        } else if (text == 'taxableAmount') {
                            taxableAmount = parser.getIntegerValue();
                        } else if (text == 'taxCalculated') {
                            taxCalculated = parser.getDoubleValue();
                        } else if (text == 'taxCode') {
                            taxCode = parser.getText();
                        } else if (text == 'taxDate') {
                            taxDate = parser.getText();
                        } else if (text == 'taxIncluded') {
                            taxIncluded = parser.getBooleanValue();
                        } else if (text == 'details') {
                            details = new List<Details>();
                            while (parser.nextToken() != JSONToken.END_ARRAY) {
                                details.add(new Details(parser));
                            }
                        } else {
                            consumeObject(parser);
                        }
                    }
                }
            }
        }
    }
```

```
    public static JsonSuccessParser parse(String json)
    {
        return new JsonSuccessParser(System.JSON.createParser(json));
    }
}
```

Prepare your JSON request to call the Avalara endpoint by using the `AvalaraJSONBuilder` class.

```
public with sharing class AvalaraJSONBuilder
{
    private static AvalaraJSONBuilder avalaraJSONBuilderInstance;

    public static AvalaraJSONBuilder getInstance()
    {
        if (NULL == avalaraJSONBuilderInstance)
        {
            avalaraJSONBuilderInstance = new AvalaraJSONBuilder();
        }
        return avalaraJSONBuilderInstance;
    }

    public String frameJsonForGetTaxOrderItem(commercetax.CalculateTaxRequest
calculateTaxRequest)
    {
        try
        {
            Id accountid  = null;
            if(calculateTaxRequest.CustomerDetails.AccountId != null &&
calculateTaxRequest.CustomerDetails.AccountId != '')
             accountid = Id.valueof(calculateTaxRequest.CustomerDetails.AccountId);

            JSONGenerator jsonGeneratorInstance = JSON.createGenerator(true);
            jsonGeneratorInstance.writeStartObject();
            String type = null;
            if(calculateTaxRequest.taxtype == commercetax.CalculateTaxType.Actual)
                type ='SalesInvoice';
                else type = 'SalesOrder';
            jsonGeneratorInstance.writeStringField('type', type);
            if(calculateTaxRequest.SellerDetails != null)
                jsonGeneratorInstance.writeStringField('companyCode',
calculateTaxRequest.SellerDetails.code);
            else
                jsonGeneratorInstance.writeStringField('companyCode', 'billing2');
            if(calculateTaxRequest.isCommit != null) {
                jsonGeneratorInstance.writeBooleanField('commit',
calculateTaxRequest.isCommit);
            }
            if(calculateTaxRequest.documentcode != null){
                jsonGeneratorInstance.writeStringField('code',
calculateTaxRequest.documentcode);
            }else if(calculateTaxRequest.referenceEntityId != null) {
                jsonGeneratorInstance.writeStringField('code',
calculateTaxRequest.referenceEntityId);
```

```
            }
          if(calculateTaxRequest.CustomerDetails.code == null && accountid !=null)
 {
                Account acc = [select id, name from account where id=:accountid];
                jsonGeneratorInstance.writeStringField('customerCode', acc.name);
            } else {
                jsonGeneratorInstance.writeStringField('customerCode',
calculateTaxRequest.CustomerDetails.code);
            }
            if(calculateTaxRequest.EffectiveDate == null)
                jsonGeneratorInstance.writeDateField('date', system.today());
            else
                jsonGeneratorInstance.writeDateTimeField('date',
calculateTaxRequest.EffectiveDate);

            jsonGeneratorInstance.writeFieldName('lines');
            jsonGeneratorInstance.writeStartArray();
            for(integer i=0;i<1;i++){
                for(Commercetax.TaxLineItemRequest lineItem :
calculateTaxRequest.LineItems)
                {
                    jsonGeneratorInstance.writeStartObject();
                    if(lineItem.linenumber != null){
                        jsonGeneratorInstance.writeStringField('number',
lineItem.linenumber);
                    }
                    jsonGeneratorInstance.writeNumberField('quantity',
lineItem.Quantity);
                    jsonGeneratorInstance.writeNumberField('amount',
(lineItem.Amount));

jsonGeneratorInstance.writeStringField('taxCode',lineItem.taxCode);

                    jsonGeneratorInstance.writeFieldName('addresses');
                    jsonGeneratorInstance.writeStartObject();
                    jsonGeneratorInstance.writeFieldName('ShipFrom');
                    jsonGeneratorInstance.writeStartObject();
                    jsonGeneratorInstance.writeStringField('line1',
lineItem.addresses.shipfrom.street);
                    jsonGeneratorInstance.writeStringField('line2',
lineItem.addresses.shipfrom.street);
                    jsonGeneratorInstance.writeStringField('city',
lineItem.addresses.shipfrom.city);
                    jsonGeneratorInstance.writeStringField('region',
lineItem.addresses.shipfrom.state);
                    jsonGeneratorInstance.writeStringField('country',
lineItem.addresses.shipfrom.country);

jsonGeneratorInstance.writeStringField('postalCode',lineItem.addresses.shipfrom.postalcode);

                    jsonGeneratorInstance.writeEndObject();

                    jsonGeneratorInstance.writeFieldName('ShipTo');
                    jsonGeneratorInstance.writeStartObject();
```

```
                    jsonGeneratorInstance.writeStringField('line1',
lineItem.addresses.shipto.street);
                    jsonGeneratorInstance.writeStringField('line2',
lineItem.addresses.shipto.street);
                    jsonGeneratorInstance.writeStringField('city',
lineItem.addresses.shipto.city);
                    jsonGeneratorInstance.writeStringField('region',
lineItem.addresses.shipto.state);
                    jsonGeneratorInstance.writeStringField('country',
lineItem.addresses.shipto.country);

jsonGeneratorInstance.writeStringField('postalCode',lineItem.addresses.shipto.postalcode);

                    jsonGeneratorInstance.writeEndObject();

                    jsonGeneratorInstance.writeFieldName('pointOfOrderOrigin');
                    jsonGeneratorInstance.writeStartObject();
                    jsonGeneratorInstance.writeStringField('line1',
lineItem.addresses.soldto.street);
                    jsonGeneratorInstance.writeStringField('line2',
lineItem.addresses.soldto.street);
                    jsonGeneratorInstance.writeStringField('city',
lineItem.addresses.soldto.city);
                    jsonGeneratorInstance.writeStringField('region',
lineItem.addresses.soldto.state);
                    jsonGeneratorInstance.writeStringField('country',
lineItem.addresses.soldto.country);

jsonGeneratorInstance.writeStringField('postalCode',lineItem.addresses.soldto.postalcode);

                    jsonGeneratorInstance.writeEndObject();


                    if(lineItem.effectiveDate != null)
                    {
                        jsonGeneratorInstance.writeFieldName('taxOverride');
                        jsonGeneratorInstance.writeStartObject();
                        jsonGeneratorInstance.writeDateTimeField('taxDate',
lineItem.effectiveDate);
                        jsonGeneratorInstance.writeEndObject();
                    }
                    jsonGeneratorInstance.writeEndObject();
                    jsonGeneratorInstance.writeEndObject();
                }
            }
                jsonGeneratorInstance.writeEndArray();
            jsonGeneratorInstance.writeEndObject();
            return jsonGeneratorInstance.getAsString();
        }
        catch (Exception e)
        {
             throw e;
        }
```

```
    }
}
```

- Use the `JsonErrorParser` class to extract the error details, if any.

```
global with sharing class JsonErrorParser
{
    public cls_error error;

    public class cls_error
    {
        public String code;
        public String message;
        public String target;
        public cls_details[] details;
    }

    public class cls_details
    {
        public String code;
        public String message;
        public String description;
        public String faultCode;
        public String helpLink;
        public String severity;
    }
    public static JsonErrorParser parse(String json)
    {
       return (JsonErrorParser) System.JSON.deserialize(json, JsonErrorParser.class);

    }
}
```

TaxEngineContext Class

Wrapper class that stores details about the type of a tax calculation request.

Namespace

CommerceTax

Example

At the beginning of a tax adapter, use `TaxEngineContext` class to pass the value of a request type to an instance of `RequestType`.

```
global virtual class MockAdapter implements commercetax.TaxEngineAdapter {

     global commercetax.TaxEngineResponse processRequest(commercetax.TaxEngineContext
taxEngineContext) {
         commercetax.RequestType requestType = taxEngineContext.getRequestType();
         commercetax.CalculateTaxRequest request =
(commercetax.CalculateTaxRequest)taxEngineContext.getRequest();
```

Build the rest of your adapter based on the type of request that you got from `TaxEngineContext` class.

```
if(requestType == commercetax.RequestType.CalculateTax){
              commercetax.calculatetaxtype type = request.taxtype;
              String docCode='';
              if(request.DocumentCode == 'simulateEmptyDocumentCode')
                  docCode = '';
              else if(request.DocumentCode != null)
                  docCode =request.DocumentCode;
            else if(request.ReferenceEntityId != null) docCode = request.ReferenceEntityId;

              else docCode =  String.valueOf(getRandomInteger(0,2147483647));
              commercetax.CalculateTaxResponse response = new
commercetax.CalculateTaxResponse();
              if(request.isCommit == true) {
                  response.setStatus(commercetax.TaxTransactionStatus.Committed);
              } else {
                  response.setStatus(commercetax.TaxTransactionStatus.Uncommitted);
              }
}
```

IN THIS SECTION:

TaxEngineContext Constructors

Learn more about the available constructors with the `TaxEngineContext` class.

TaxEngineContext Methods

Learn more about the available methods with the `TaxEngineContext` class.

TaxEngineContext Constructors

Learn more about the available constructors with the `TaxEngineContext` class.

The `TaxEngineContext` class includes these constructors.

IN THIS SECTION:

TaxEngineContext(request, requestType, namedUri)

Initializes the `TaxEngineContext` object. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

`TaxEngineContext(request, requestType, namedUri)`

Initializes the `TaxEngineContext` object. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Signature

```
TaxEngineContext(commercetax.TaxEngineRequest request, commercetax.RequestType
requestType, String namedUri)
```

Parameters

request
Type: TaxEngineRequest

Information about the request.

requestType
Type: RequestType

Whether the tax request is to calculate or estimate tax.

namedUri
Type: String

URI that was called as part of the tax calculation request.

TaxEngineContext Methods

Learn more about the available methods with the `TaxEngineContext` class.

The `TaxEngineContext` class includes these methods.

IN THIS SECTION:

getNamedUri()
Retrieves the value of the NamedUri field of the `TaxEngineContext` class.

getRequest()
Gets the value of the `TaxEngineContext`'s Request field.

getRequestType()
Gets the value of the RequestType field of the `TaxEngineContext` class.

`getNamedUri()`

Retrieves the value of the NamedUri field of the `TaxEngineContext` class.

Signature

```
global String getNamedUri()
```

Return Value

Type: String

`getRequest()`

Gets the value of the `TaxEngineContext`'s Request field.

Signature

```
global commercetax.TaxEngineRequest getRequest()
```

Return Value

Type: TaxEngineRequest

An implemented instance of an external tax engine's interface for processing requests. We've provided the `TaxEngineRequest` interface for you to test within mock adapters with classes that implement it, such as CalculateTaxRequest. However, don't use it outside of a testing context.

`getRequestType()`

Gets the value of the RequestType field of the `TaxEngineContext` class.

Signature

```
global commercetax.RequestType getRequestType()
```

Return Value

Type: RequestType

Indicates whether the calculation request was for actual or calculated tax.

TaxLineItemRequest Class

Contains line item details of a tax request.

Namespace

CommerceTax

IN THIS SECTION:

TaxLineItemRequest Constructors
Learn more about the constructors available with the `TaxLineItemRequest` class.

TaxLineItemRequest Properties
Learn more about the available properties with the `TaxLineItemRequest` class.

TaxLineItemRequest Methods
Learn more about the available methods with the `TaxLineItemRequest` class.

TaxLineItemRequest Constructors

Learn more about the constructors available with the `TaxLineItemRequest` class.

The `TaxLineItemRequest` class includes these constructors.

IN THIS SECTION:

TaxLineItemRequest(addresses, amount, description, productCode, quantity, lineNumber, taxCode, effectiveDate)
Initializes the request for the tax line item. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

`TaxLineItemRequest(addresses, amount, description, productCode, quantity, lineNumber, taxCode, effectiveDate)`

Initializes the request for the tax line item. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Signature

```
global TaxLineItemRequest(commercetax.LineTaxAddressesRequest addresses, Double amount, String description, String productCode, Double quantity, String lineNumber, String taxCode, Datetime effectiveDate)
```

Parameters

addresses
Type: LineTaxAddressesRequest

Information about the addresses applied to each line item in a tax calculation request.

amount
Type: Double

Total amount (in a given currency) represented by a line item sent for tax calculation.

description
Type: String

User-defined description for a tax line item.

productCode
Type: String

Catalog code for the product represented by the tax line item.

quantity
Type: Double

The number of units of a given product that the tax line item represents.

lineNumber
Type: String

Unique number used to identify a tax line item.

taxCode
Type: String

Code used to identify how tax is calculated for a tax line item.

effectiveDate
Type: Datetime

This is a user-defined date used for reporting only.

TaxLineItemRequest Properties

Learn more about the available properties with the `TaxLineItemRequest` class.

The `TaxLineItemRequest` class includes these properties.

IN THIS SECTION:

addresses
Contains the list of addresses of a line item.

amount
Total amount (in a given currency) represented by a line item sent for tax calculation.

description
User-defined description for a tax line item.

effectiveDate
The date that a tax transaction takes effect on a line item. This is a user-defined date used for reporting only.

lineNumber
Unique number used to identify a tax line item.

productCode
Catalog code for the product represented by the tax line item.

quantity
Number of units of a given product that the tax line item represents.

taxCode
Code used to identify how tax is calculated for a tax line item.

addresses

Contains the list of addresses of a line item.

Signature

```
public commercetax.LineTaxAddressesRequest addresses {get; set;}
```

Property Value

Type: commercetax.LineTaxAddressesRequest

amount

Total amount (in a given currency) represented by a line item sent for tax calculation.

Signature

```
global Double amount {get; set;}
```

Property Value

Type: Double

description

User-defined description for a tax line item.

Signature

```
global String description {get; set;}
```

Property Value

Type: String

`effectiveDate`

The date that a tax transaction takes effect on a line item. This is a user-defined date used for reporting only.

Signature

```
global Datetime effectiveDate {get; set;}
```

Property Value

Type: Datetime

`lineNumber`

Unique number used to identify a tax line item.

Signature

```
global String lineNumber {get; set;}
```

Property Value

Type: String

`productCode`

Catalog code for the product represented by the tax line item.

Signature

```
global String productCode {get; set;}
```

Property Value

Type: String

`quantity`

Number of units of a given product that the tax line item represents.

Signature

```
global Double quantity {get; set;}
```

Property Value

Type: Double

taxCode

Code used to identify how tax is calculated for a tax line item.

Signature

```
global String taxCode {get; set;}
```

Property Value

Type: String

TaxLineItemRequest Methods

Learn more about the available methods with the `TaxLineItemRequest` class.

The `TaxLineItemRequest` class includes these methods.

IN THIS SECTION:

equals(obj)
Maintains the integrity of lists of type `TaxLineItemRequest` by determining the equality of external objects in a list. This method is dynamic and is based on the `equals()` method in Java.

hashCode()
Maintains the integrity of lists of type `TaxLineItemRequest` by determining the uniqueness of the external object records in a list.

toString()
Converts a value to a string.

equals(obj)

Maintains the integrity of lists of type `TaxLineItemRequest` by determining the equality of external objects in a list. This method is dynamic and is based on the `equals()` method in Java.

Signature

```
global Boolean equals(Object obj)
```

Parameters

obj

Type: Object

External object whose key is to be validated.

Return Value

Type: Boolean

`hashCode()`

Maintains the integrity of lists of type `TaxLineItemRequest` by determining the uniqueness of the external object records in a list.

Signature

```
global Integer hashCode()
```

Return Value

Type: Integer

`toString()`

Converts a value to a string.

Signature

```
global String toString()
```

Return Value

Type: String

TaxSellerDetailsRequest Class

Contains tax code details used in the tax calculation request.

Namespace

CommerceTax

IN THIS SECTION:

TaxSellerDetailsRequest Constructors
Learn more about the available constructors with the `TaxSellerDetailsRequest` class.

TaxSellerDetailsRequest Properties
Learn more about the available properties with the `TaxSellerDetailsRequest` class.

TaxSellerDetailsRequest Methods
Learn more about the available methods with the `TaxSellerDetailsRequest` class.

TaxSellerDetailsRequest Constructors

Learn more about the available constructors with the `TaxSellerDetailsRequest` class.

The `TaxSellerDetailsRequest` class includes these constructors.

IN THIS SECTION:

TaxSellerDetailsRequest(code)
Initializes the request for the tax seller details. This constructor is intended for test usage and throws an exception if used outside of the Apex test context

`TaxSellerDetailsRequest(code)`

Initializes the request for the tax seller details. This constructor is intended for test usage and throws an exception if used outside of the Apex test context

Signature

```
global TaxSellerDetailsRequest(String code)
```

Parameters

code
Type: String

Tax code used for tax calculation.

TaxSellerDetailsRequest Properties

Learn more about the available properties with the `TaxSellerDetailsRequest` class.

The `TaxSellerDetailsRequest` class includes these properties.

IN THIS SECTION:

code
Tax code used for tax calculation.

`code`

Tax code used for tax calculation.

Signature

```
global String code {get; set;}
```

Property Value

Type: String

TaxSellerDetailsRequest Methods

Learn more about the available methods with the `TaxSellerDetailsRequest` class.

The `TaxSellerDetailsRequest` class includes these methods.

IN THIS SECTION:

equals(obj)
Maintains the integrity of lists of type `TaxSellerDetailsRequest` by determining the equality of the external objects in a list. This method is dynamic and based on the `equals()` method in Java.

hashCode()
Maintains the integrity of lists of type `TaxSellerDetailsRequest` by determining the uniqueness of the external objects in a list.

toString()
Converts a value to a string.

equals(obj)

Maintains the integrity of lists of type `TaxSellerDetailsRequest` by determining the equality of the external objects in a list. This method is dynamic and based on the `equals()` method in Java.

Signature

```
global Boolean equals(Object obj)
```

Parameters

obj
Type: Object

External object whose key is to be validated.

Return Value

Type: Boolean

hashCode()

Maintains the integrity of lists of type `TaxSellerDetailsRequest` by determining the uniqueness of the external objects in a list.

Signature

```
global Integer hashCode()
```

Return Value

Type: Integer

toString()

Converts a value to a string.

Signature

```
global String toString()
```

Return Value

Type: String

TaxTransactionRequest Class

Abstract class for storing customer details used in tax calculation and estimation requests.

Namespace

CommerceTax

Usage

Specify the `CommerceTax` namespace when creating an instance of this class. The constructor of this class takes no arguments. For example, let's say you create an instance of `CalculateTaxRequest` class, which extends the `TaxTransactionRequest` class.

IN THIS SECTION:

TaxTransactionRequest Constructors
Learn more about the available constructors with the `TaxTransactionRequest` class.

TaxTransactionRequest Properties
Learn more about the available properties with the `TaxTransactionRequest` class.

TaxTransactionRequest Methods

TaxTransactionRequest Constructors

Learn more about the available constructors with the `TaxTransactionRequest` class.

The `TaxTransactionRequest` class includes these constructors.

IN THIS SECTION:

TaxTransactionRequest(addresses, currencyIsoCode, customerDetails, description, documentCode, referenceDocumentCode, transactionDate, effectiveDate, lineItems, referenceEntityId, sellerDetails)
Initializes the request for the tax transaction. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

`TaxTransactionRequest(addresses, currencyIsoCode, customerDetails, description, documentCode, referenceDocumentCode, transactionDate, effectiveDate, lineItems, referenceEntityId, sellerDetails)`

Initializes the request for the tax transaction. This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Signature

```
global TaxTransactionRequest(commercetax.HeaderTaxAddressesRequest addresses, String
currencyIsoCode, commercetax.TaxCustomerDetailsRequest customerDetails, String
description, String documentCode, String referenceDocumentCode, Datetime transactionDate,
Datetime effectiveDate, List<commercetax.TaxLineItemRequest> lineItems, String
referenceEntityId, commercetax.TaxSellerDetailsRequest sellerDetails)
```

Parameters

addresses
Type: HeaderTaxAddressesRequest

Tax addresses, such as Ship To and Bill From.

currencyIsoCode
Type: String

Three-letter ISO 4217 currency code associated with the `TaxTransactionRequest`.

customerDetails
Type: TaxCustomerDetailsRequest

Customer information used in tax calculation.

description
Type: String

Optional user-defined description for providing more information about the tax transaction request.

documentCode
Type: String

Code for documents that are used to provide more information in the tax calculation process.

referenceDocumentCode
Type: String

Code for additional documents that are used in the tax calculation process.

transactionDate
Type: Datetime

The date that the tax transaction occurred.

effectiveDate
Type: Datetime

The date that the tax transaction takes effect. User-defined and used only for reporting purposes.

lineItems
Type: List<TaxLineItemRequest>

A list of line items on which tax is calculated.

referenceEntityId
Type: String

ID of an object related to the line items sent for tax calculation.

sellerDetails
Type: TaxSellerDetailsRequest

Contains tax code information used in a tax calculation request.

TaxTransactionRequest Properties

Learn more about the available properties with the `TaxTransactionRequest` class.

The `TaxTransactionRequest` class includes these properties.

IN THIS SECTION:

addresses
A list of addresses (such as Ship To and Sold To) used as part of the tax transaction.

currencyIsoCode
Three-letter ISO 4217 currency code associated with the `TaxTransactionRequest`.

customerDetails
Customer information used in tax calculation.

description
Optional user-defined description for providing more information about the tax transaction request.

documentCode
Code for documents used to provide more information in the tax calculation process.

effectiveDate
The date that the tax transaction takes effect. User-defined and used only for reporting purposes.

lineItems
A list of line items on which tax will be calculated.

referenceDocumentCode
Code for documents used to provide more information in the tax calculation process.

referenceEntityId
ID of an object related to the line items sent for tax calculation.

sellerDetails
Contains tax code information used in a tax calculation request.

transactionDate
The date that the tax transaction occurred.

`addresses`

A list of addresses (such as Ship To and Sold To) used as part of the tax transaction.

Signature

```
global commercetax.HeaderTaxAddressesRequest addresses {get; set;}
```

Property Value

Type: HeaderTaxAddressesRequest

`currencyIsoCode`

Three-letter ISO 4217 currency code associated with the `TaxTransactionRequest`.

Signature

```
global String currencyIsoCode {get; set;}
```

Property Value

Type: String

`customerDetails`

Customer information used in tax calculation.

Signature

```
global CommerceTax.TaxCustomerDetailsRequest customerDetails {get; set;}
```

Property Value

Type: TaxCustomerDetailsRequest

`description`

Optional user-defined description for providing more information about the tax transaction request.

Signature

```
global String description {get; set;}
```

Property Value

Type: String

`documentCode`

Code for documents used to provide more information in the tax calculation process.

Signature

```
global String documentCode {get; set;}
```

Property Value

Type: String

`effectiveDate`

The date that the tax transaction takes effect. User-defined and used only for reporting purposes.

Signature

```
global Datetime effectiveDate {get; set;}
```

Property Value

Type: Datetime

`lineItems`

A list of line items on which tax will be calculated.

Signature

```
global List<CommerceTax.TaxLineItemRequest> lineItems {get; set;}
```

Property Value

Type: List<TaxLineItemRequest>

`referenceDocumentCode`

Code for documents used to provide more information in the tax calculation process.

Signature

```
global String referenceDocumentCode {get; set;}
```

Property Value

Type: String

`referenceEntityId`

ID of an object related to the line items sent for tax calculation.

Signature

```
global String referenceEntityId {get; set;}
```

Property Value

Type: String

`sellerDetails`

Contains tax code information used in a tax calculation request.

Signature

```
global commercetax.TaxSellerDetailsRequest sellerDetails {get; set;}
```

Property Value

Type: TaxSellerDetailsRequest

transactionDate

The date that the tax transaction occurred.

Signature

```
global Datetime transactionDate {get; set;}
```

Property Value

Type: Datetime

TaxTransactionRequest Methods

The following are methods for `TaxTransactionRequest`.

IN THIS SECTION:

equals(obj)
Maintains the integrity of lists of type `TaxTransactionRequest` by determining the equality of external objects in a list. This method is dynamic and based on the `equals()` method in Java.

hashCode()
Maintains the integrity of lists of type `TaxTransactionRequest` by determining the uniqueness of the external object records in a list.

toString()
Converts a value to a string.

equals(obj)

Maintains the integrity of lists of type `TaxTransactionRequest` by determining the equality of external objects in a list. This method is dynamic and based on the `equals()` method in Java.

Signature

```
global Boolean equals(Object obj)
```

Parameters

obj
Type: Object

Return Value

Type: Boolean

hashCode()

Maintains the integrity of lists of type `TaxTransactionRequest` by determining the uniqueness of the external object records in a list.

Signature

```
global Integer hashCode()
```

Return Value

Type: Integer

`toString()`

Converts a value to a string.

Signature

```
global String toString()
```

Return Value

Type: String

TaxTransactionStatus Enum

Shows whether the tax transaction has been committed or uncommitted.

Usage

Used by the CalculateTaxResponse class method.

Enum Values

The `commercetax.TaxTransactionStatus` enum includes these values.

Value	Description
`Committed`	Tax has been calculated and committed.
`Uncommitted`	Tax has been calculated but hasn't been committed.

TaxTransactionType Enum

Shows whether the tax transaction is for a credit or debit transaction.

Usage

Used by the CalculateTaxResponse and CalculateTaxRequest class methods.

Enum Values

The `commercetax.TaxTransactionType` enum includes these values.

Value	Description
`Credit`	Represents a credit transaction.
`Debit`	Represents a debit transaction.

Compression Namespace

The Compression namespace provides classes and methods to create and extract zip files.

The following are the classes and enums in the `Compression` namespace.

IN THIS SECTION:

Level Enum
Specifies the compression level for creating a zip file.

Method Enum
Specifies the compression method for the zip entries.

ZipEntry Class
Contains methods to get and set information about a zip file entry.

ZipReader Class
Contains methods to get information about zip entries and to extract content for specified zip entries from the zip file.

ZipWriter Class
Contains methods to add zip entries, generate a zipped archive, and return the result as an Apex blob.

Compression Exceptions
The `Compression` namespace contains exception classes.

Level Enum

Specifies the compression level for creating a zip file.

Usage

Use `Level` enum with the `getLevel()` and `setLevel(value)` methods in the `ZipWriter` class.

Enum Values

The following are the values of the `Compression.Level` enum.

Value	Description
`BEST_COMPRESSION`	Compression level for best compression.
`BEST_SPEED`	Compression level for fastest compression.
`DEFAULT_LEVEL`	Default compression level.

Value	Description
NO_COMPRESSION	Compression level for no compression.

Method Enum

Specifies the compression method for the zip entries.

Usage

Use the `Method` enum with the `getMethod()` and `setMethod(method)` methods in the `ZipEntry` and `ZipWriter` classes.

Enum Values

The following are the values of the `Compression.Method` enum.

Value	Description
DEFLATED	Deflated compression method for compressed entries.
STORED	No compression method for zip entries.

ZipEntry Class

Contains methods to get and set information about a zip file entry.

Namespace

Compression

IN THIS SECTION:

ZipEntry Methods

ZipEntry Methods

The following are methods for `ZipEntry`.

IN THIS SECTION:

equals(obj)
Compares this object with the specified object and returns `true` if both objects are equal; otherwise, returns `false`.

hashcode()
Returns the hash code value for the zip entry.

getComment()
Gets the comment string for the zip entry.

getCompressedSize()
Gets the size in bytes of the compressed zip entry.

getContent()
Gets the content of the zip entry. This method doesn't work with the `ZipReader` class.

getCrc()
Gets the cyclic redundancy check (CRC) value for the zip entry.

getLastModifiedTime()
Gets the last modification timestamp of the zip entry.

getMethod()
Gets the compression method of the zip entry.

getName()
Gets the name of the zip entry.

getUncompressedSize()
Gets the uncompressed size in bytes of the zip entry content.

setComment(comment)
Sets the comment string for the zip entry that's written to the Zip archive. This method doesn't work with the `ZipReader` class.

setContent(blob)
Sets the content of the zip entry that's written to the Zip archive. This method doesn't work with the `ZipReader` class.

setLastModifiedTime(modTime)
Sets the last modification time of the zip entry that's written to the Zip archive. This method doesn't work with the `ZipReader` class.

setMethod(method)
Sets the compression method for the zip entry that's written to the zip archive. This method doesn't work with the `ZipReader` class.

toString()
Returns a string representation of the zip entry.

`equals(obj)`

Compares this object with the specified object and returns `true` if both objects are equal; otherwise, returns `false`.

Signature

```
public Boolean equals(Object obj)
```

Parameters

obj
Type: Object

Return Value

Type: Boolean

hashcode()

Returns the hash code value for the zip entry.

Signature

```
public Integer hashcode()
```

Return Value

Type: Integer

getComment()

Gets the comment string for the zip entry.

Signature

```
public string getComment()
```

Return Value

Type: string

getCompressedSize()

Gets the size in bytes of the compressed zip entry.

Signature

```
public long getCompressedSize()
```

Return Value

Type: long

getContent()

Gets the content of the zip entry. This method doesn't work with the `ZipReader` class.

Signature

```
public blob getContent()
```

Return Value

Type: blob

getCrc()

Gets the cyclic redundancy check (CRC) value for the zip entry.

Signature

```
public long getCrc()
```

Return Value

Type: long

`getLastModifiedTime()`

Gets the last modification timestamp of the zip entry.

Signature

```
public Datetime getLastModifiedTime()
```

Return Value

Type: Datetime

`getMethod()`

Gets the compression method of the zip entry.

Signature

```
public Compression.Method getMethod()
```

Return Value

Type: Compression.Method

Uses values from the `Method` enum and indicates whether the zip entry has *`DEFLATED`* or *`STORED`* method.

`getName()`

Gets the name of the zip entry.

Signature

```
public string getName()
```

Return Value

Type: string

`getUncompressedSize()`

Gets the uncompressed size in bytes of the zip entry content.

Signature

```
public long getUncompressedSize()
```

Return Value

Type: long

`setComment(comment)`

Sets the comment string for the zip entry that's written to the Zip archive. This method doesn't work with the `ZipReader` class.

Signature

```
public Compression.ZipEntry setComment(String comment)
```

Parameters

`comment`
Type: String

Return Value

Type: Compression.ZipEntry

`setContent(blob)`

Sets the content of the zip entry that's written to the Zip archive. This method doesn't work with the `ZipReader` class.

Signature

```
public Compression.ZipEntry setContent(Blob blob)
```

Parameters

`blob`
Type: Blob

Return Value

Type: Compression.ZipEntry

`setLastModifiedTime(modTime)`

Sets the last modification time of the zip entry that's written to the Zip archive. This method doesn't work with the `ZipReader` class.

Signature

```
public Compression.ZipEntry setLastModifiedTime(Datetime modTime)
```

Parameters

modTime
Type: Datetime

Return Value

Type: Compression.ZipEntry

setMethod(method)

Sets the compression method for the zip entry that's written to the zip archive. This method doesn't work with the `ZipReader` class.

Signature

```
public Compression.ZipEntry setMethod(Compression.Method method)
```

Parameters

method
Type: Compression.Method

Uses the `Method` enum values and sets the compression method as `DEFLATED` or `STORED`.

Return Value

Type: Compression.ZipEntry

toString()

Returns a string representation of the zip entry.

Signature

```
public string toString()
```

Return Value

Type: string

ZipReader Class

Contains methods to get information about zip entries and to extract content for specified zip entries from the zip file.

Namespace

Compression

IN THIS SECTION:

ZipReader Constructors

ZipReader Methods

ZipReader Constructors

The following are constructors for `ZipReader`.

IN THIS SECTION:

ZipReader(data)
Creates a new instance of the `ZipReader` class using the specified blob data.

`ZipReader(data)`

Creates a new instance of the `ZipReader` class using the specified blob data.

Signature

```
global ZipReader(Blob data)
```

Parameters

`data`
Type: Blob

Apex blob that contains the compressed content.

ZipReader Methods

The following are methods for `ZipReader`.

IN THIS SECTION:

extract(name)
Extracts the bytes for the specified zip entry name and decompresses the content.

extract(entry)
Extracts the bytes for the specified zip entry and decompresses the content.

getEntries()
Gets a list of all the entries from the zip file.

getEntriesMap()
Gets a map of names and the corresponding zip entries from the zip file.

getEntry(name)
Gets a zip entry for the specified name from the zip file.

getEntryNames()
Gets a list of all the zip entry names from the zip file.

`extract(name)`

Extracts the bytes for the specified zip entry name and decompresses the content.

Signature

```
public blob extract(string name)
```

Parameters

name
Type: string

Species the zip entry name to extract and decompress.

Return Value

Type: blob

Apex blob that contains the decompressed content.

extract(entry)

Extracts the bytes for the specified zip entry and decompresses the content.

Signature

```
public blob extract(Compression.ZipEntry entry)
```

Parameters

entry
Type: Compression.ZipEntry

Species the zip entry to extract and decompress.

Return Value

Type: blob

Apex blob that contains the decompressed content.

getEntries()

Gets a list of all the entries from the zip file.

Signature

```
public List<compression.ZipEntry> getEntries()
```

Return Value

Type: List<Compression.ZipEntry>

getEntriesMap()

Gets a map of names and the corresponding zip entries from the zip file.

Signature

```
public Map<String,Compression.ZipEntry> getEntriesMap()
```

Return Value

Type: Map<string,Compression.ZipEntry>

`getEntry(name)`

Gets a zip entry for the specified name from the zip file.

Signature

```
public compression.ZipEntry getEntry(string name)
```

Parameters

`name`
Type: string

Name of the zip entry.

Return Value

Type: Compression.ZipEntry

Throws a `ZipException` if the specified name isn't found.

`getEntryNames()`

Gets a list of all the zip entry names from the zip file.

Signature

```
public List<String> getEntryNames()
```

Return Value

Type: List<String>

ZipWriter Class

Contains methods to add zip entries, generate a zipped archive, and return the result as an Apex blob.

Namespace

Compression

Example

This sample code compresses email attachments into a single file.

```
Compression.ZipWriter writer = new Compression.ZipWriter();

List<id> contentDocumentIds = new List<id>();

// Add IDs of documents to be compressed to contentDocumentIds

for ( ContentVersion cv : [SELECT PathOnClient, Versiondata
                           FROM ContentVersion
                           WHERE ContentDocumentId IN :contentDocumentIds])
{
      writer.addEntry(cv.PathOnClient, cv.versiondata);
}

blob zipAttachment = writer.getArchive();

Messaging.EmailFileAttachment efa = new Messaging.EmailFileAttachment();
efa.setFileName('attachments.zip');
efa.setBody(zipAttachment);

List<Messaging.EmailFileAttachment> fileAttachments = new
List<Messaging.EmailFileAttachment>();
fileAttachments.add(efa);

Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();

// Set all the other email fields, such as addresses, subject, and body

email.setFileAttachments(fileAttachments);

Messaging.sendEmail(new Messaging.SingleEmailMessage[] { email });
```

IN THIS SECTION:

ZipWriter Constructors

ZipWriter Methods

ZipWriter Constructors

The following are constructors for `ZipWriter`.

IN THIS SECTION:

ZipWriter()
Creates a new instance of the `ZipWriter` class.

`ZipWriter()`

Creates a new instance of the `ZipWriter` class.

Signature

```
global ZipWriter()
```

ZipWriter Methods

The following are methods for `ZipWriter`.

IN THIS SECTION:

addEntry(name, data)
Adds an entry to the zip file with the specified name and content.

addEntry(prototype)
Adds a copy of the specified prototype entry to the zip file and includes details such as the zip entry name, comment, last modification time, and content.

addEntry(name, comment, modTime, method, data)
Adds an entry to the zip file with the specified name, comment, last modification time, compression method, and content.

getArchive()
Compresses the zip entries and generates a ZIP archive.

getEntries()
Gets a list of all the entries in the zip file.

getEntry(name)
Gets the entry with the specified name from the zip file.

getEntryNames()
Gets a set of all the zip entry names in the zip file.

getLevel()
Gets the compression level of the zip file.

getMethod()
Gets the compression method of the zip file.

removeEntry(name)
Removes the entry with the specified name from the zip file.

setLevel(level)
Sets the compression level of the zip file.

setMethod(method)
Sets the compression method for the zip file.

`addEntry(name, data)`

Adds an entry to the zip file with the specified name and content.

Signature

```
public Compression.ZipEntry addEntry(string name, blob data)
```

Parameters

name
Type: string

The name of the zip entry.

data
Type: blob

The content of the zip entry.

Return Value

Type: Compression.ZipEntry

Zip entry added to the zip file.

addEntry(prototype)

Adds a copy of the specified prototype entry to the zip file and includes details such as the zip entry name, comment, last modification time, and content.

Signature

```
public Compression.ZipEntry addEntry(compression.ZipEntry prototype)
```

Parameters

prototype
Type: Compression.ZipEntry

Details of the entry to be added to the zip file.

Return Value

Type: Compression.ZipEntry

addEntry(name, comment, modTime, method, data)

Adds an entry to the zip file with the specified name, comment, last modification time, compression method, and content.

Signature

```
public Compression.ZipEntry addEntry(String name, String comment, Datetime modTime,
Compression.Method method, Blob data)
```

Parameters

name
Type: String

The name of the zip entry.

comment
Type: String

The comment about the zip entry.

modTime
Type: Datetime

The last modification timestamp of the zip entry.

method
Type: Compression.Method

The compression method of the zip entry, which is either `DEFLATED` or `STORED`.

data
Type: Blob

The content of the zip entry.

Return Value

Type: Compression.ZipEntry

Zip entry added to the zip file.

getArchive()

Compresses the zip entries and generates a ZIP archive.

Signature

```
public blob getArchive()
```

Return Value

Type: blob

Apex blob that contains the bytes of the compression operation.

getEntries()

Gets a list of all the entries in the zip file.

Signature

```
public List<Compression.ZipEntry> getEntries()
```

Return Value

Type: List<Compression.ZipEntry>

getEntry(name)

Gets the entry with the specified name from the zip file.

Signature

```
public compression.ZipEntry getEntry(string name)
```

Parameters

name
Type: string

Name of the zip entry to be retrieved.

Return Value

Type: Compression.ZipEntry

`getEntryNames()`

Gets a set of all the zip entry names in the zip file.

Signature

```
public Set<String> getEntryNames()
```

Return Value

Type: Set<String> on page 3767

`getLevel()`

Gets the compression level of the zip file.

Signature

```
public Compression.Level getLevel()
```

Return Value

Type: Compression.Level

Uses the `Level` enum values to indicate the compression level as `BEST_COMPRESSION`, `BEST_SPEED`, `DEFAULT_LEVEL`, or `NO_COMPRESSION`.

`getMethod()`

Gets the compression method of the zip file.

Signature

```
public Compression.Method getMethod()
```

Return Value

Type: Compression.Method

Uses the `Method` enum values to indicate the compression method as `DEFLATED` or `STORED`.

`removeEntry(name)`

Removes the entry with the specified name from the zip file.

Signature

```
public Void removeEntry(string name)
```

Parameters

`name`
Type: string

Name of the zip entry to be removed. If an entry with this name isn't found, the method throws a `ZipException` exception.

Return Value

Type: Void

`setLevel(level)`

Sets the compression level of the zip file.

Signature

```
public Compression.ZipWriter setLevel(compression.Level value)
```

Parameters

`value`
Type: Compression.Level

Uses the `Level` enum to set the compression level.

Return Value

Type: Compression.ZipWriter

Returns the zip file set with the specified compression level.

`setMethod(method)`

Sets the compression method for the zip file.

Signature

```
public Compression.ZipWriter setMethod(compression.Method value)
```

Parameters

`value`
Type: Compression.Method

Uses the `Method` enum to set the compression method.

Return Value

Type: Compression.ZipWriter

Returns the zip file set with the specified compression method.

Compression Exceptions

The `Compression` namespace contains exception classes.

All exception classes support built-in methods for returning the error message and exception type. See Exception Class and Built-In Exceptions.

The `Compression` namespace contains this exception:

Exception	Description
`Compression.ZipException`	Any problem with the zip operations, such as a zip entry name isn't found.

`ConnectApi` Namespace

The `ConnectApi` namespace (also called Connect in Apex) provides classes for accessing the same data available in Connect REST API. Use Connect in Apex to create custom experiences in Salesforce.

For information about working with the `ConnectApi` classes, see Connect in Apex.

IN THIS SECTION:

ActionLinks Class
Create, delete, and get information about an action link group definition; get information about an action link group; get action link diagnostic information.

Announcements Class
Access information about announcements and post announcements.

BotVersionActivation Class
Access and update activation information of a bot version.

CdpCalculatedInsight Class
Create, delete, get, run, and update Data Cloud calculated insights.

CdpIdentityResolution Class
Create, delete, get, run, and update Data Cloud identity resolution rulesets.

CdpQuery Class
Get Data Cloud metadata and query data.

CdpSegment Class
Create, delete, get, publish, and update Data Cloud segments.

Chatter Class
Access information about followers and subscriptions for records.

ChatterFavorites Class
Chatter favorites give you easy access to topics, list views, and feed searches.

ChatterFeeds Class

Get, post, and delete feed elements, likes, comments, and bookmarks. You can also search feed elements, share feed elements, and vote on polls.

ChatterGroups Class

Information about groups, such as the group's members, photo, and the groups the specified user is a member of. Add members to a group, remove members, and change the group photo.

ChatterMessages Class

Get, send, search, and reply to private messages. You can also get and search private conversations, mark conversations as read, and get a count of unread private messages.

ChatterUsers Class

Access information about users, such as activity, followers, subscriptions, files, and groups.

Clm Class

Create and update Contract Lifecycle Management (CLM) contracts using object ID.

CommerceBuyerExperience Class

Create, delete, or get commerce addresses. Get order delivery group, order item, order shipments, shipment items, and order summaries. Get adjustments for order items and order summaries.

CommerceCart Class

Get, create, update, calculate, and delete carts. Get cart items, add items to carts, update and delete cart items.

CommerceCatalog Class

Get products, product categories, and product category paths.

CommerceCatalogManagement Class

Create or update a composite product. Create a variation product.

CommercePromotions Class

Evaluate promotions for Commerce orders. Get coupon code redemption usage.

CommerceSearch Class

Get sort rules for the live search index. Get product search suggestions. Search products.

CommerceSearchConnectFamily Class

Search products by search term or category in a webstore.

CommerceSearchSettings Class

Get indexes. Get index logs. Create an index of a product catalog.

CommerceStorePricing Class

Get product prices.

CommerceWishlist Class

Get, create, update, and delete wishlists. Add wishlists to carts. Get wishlist items, add items to wishlists, and delete wishlist items.

Communities Class

Get information about Experience Cloud sites in your org.

CommunityModeration Class

Get information about flagged feed items and comments in an Experience Cloud site. Add and remove flags from comments and feed items.

ContentHub Class

Access Files Connect repositories and their files and folders.

ConversationApplicationDefinition Class

Access information about a conversation application definition.

Datacloud Class

Purchase Data.com contact or company records, and retrieve purchase information.

EinsteinLLM Class

Get a list of prompt templates and generate LLM responses for prompt templates.

EmailMergeFieldService Class

Extract a list of merge fields for an object. A merge field is a field you can put in an email template, mail merge template, custom link, or formula to incorporate values from a record.

EmployeeProfiles Class

Get, set and crop, and delete employee banner photos and photos.

Exchanges Class

Preview and submit cart to exchange orders.

ExtendedCommerceDelivery Class

Access information about delivery estimation.

ExternalEmailServices Class

Access information about integration with external email services, such as sending email within Salesforce through an external email account.

ExternalManagedAccount Class

Get externally managed accounts.

FieldService Class

Preview and create shifts from a pattern or filter fields on recordset filter criteria.

FulfillmentOrder Class

Fulfill orders in Order Management.

Knowledge Class

Get information about trending articles in Experience Cloud sites.

LightningScheduler Class

Create and update service appointments.

ManagedContent Class

Clone managed content. Create and get managed content. Delete and replace variants. Get channels. Get a managed content space. Get targets that managed content space folders can be shared with. Get and update targets that managed content space folders are shared with. Publish and unpublish content.

ManagedContentChannels Class

Get managed content channels. Create, get, update, or delete a managed content channel.

ManagedContentDelivery Class

Get collection items. Get a managed content channel. Get managed content.

ManagedContentSpaces Class

Get channels in a managed content space. Add or remove channels from a managed content space.

ManagedTopics Class

Get managed topics in an Experience Cloud site. Create, delete, and reorder managed topics.

MarketingIntegration Class

Get, save, and submit a microsites marketing integration form for an Experience Cloud site.

Mentions Class

Access information about mentions. A mention is an "@" character followed by a user or group name. When a user or group is mentioned, they receive a notification.

Missions Class

Export and purge mission activity for users. Get a user's mission progress. Update mission activity counts for users.

NamedCredentials Class

Create, refresh, get, delete, replace, and update credentials. Create and get external credentials. Create and get named credentials. Create, get, delete, and update external auth identity providers. Get the URL for the OAuth token flow for an external credential.

NavigationMenu Class

Get navigation menu items for an Experience Cloud site.

NextBestAction Class

Execute recommendation strategies, get recommendations, manage recommendation reactions.

OmnichannelInventoryService Class

Route orders to inventory locations in Order Management.

OMSAnalytics Class

Get products with return rates, get text classified into different classifications using text analysis, and capture the return reasons from external sources based on the product ids.

Orchestration Class

Get orchestration instances.

OrderPaymentSummary Class

Work with payments in Order Management.

OrderSummary Class

Work with orders in Order Management.

OrderSummaryCreation Class

Create Order Summaries in Order Management.

Organization Class

Access information about an org.

PardotBusinessUnitContext Class

Get the Pardot business units the context user has access to.

Payments Class

Authorize a payment, capture an authorized payment, and refund an authorized payment.

Personalization Class

Get assigned personalization audiences that match the user context. Create, get, update, and delete an audience. Get personalization targets that match the user context, based on the assigned audiences that include the user. Create and update targets. Get and delete a target.

PickTicket Class

Create tickets to fulfill orders.

QuestionAndAnswers Class

Access question and answers suggestions.

Recommendations Class

Get and reject Chatter, custom, and static recommendations. Create, get, update, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

Records Class

Access information about record motifs, which are small icons used to distinguish record types in the Salesforce UI.

RegisterGuestBuyer Class

Register a guest buyer for a webstore using an account ID, enabling a guest buyer to order on behalf of another buyer.

Repricing Class

Perform functions related to repricing orders in Order Management.

ReturnOrder Class

Process ReturnOrders in Order Management, limited to 2,000 requests per hour.

Routing Class

Route orders to inventory locations in Order Management.

SalesforceInbox Class

Access information about Automated Activity Capture, which is available in Einstein and Salesforce Inbox.

Search Class

Search objects using keywords or a natural language query.

Sites Class

Search an Experience Cloud site.

SmartDataDiscovery Class

Get predictions on Salesforce objects.

SocialEngagement Class

Manage information about social accounts or fan pages for social networks.

Surveys Class

Send survey invitations by email.

TaxPlatform Class

Apply or cancel tax.

Topics Class

Access information about topics, such as their descriptions, the number of people talking about them, related topics, and information about groups contributing to the topic. Update a topic's name or description, merge topics, and add and remove topics from records and feed items.

UserProfiles Class

Access user profile data. The user profile data populates the profile page (also called the Chatter profile page). This data includes user information (such as address, manager, and phone number), some user capabilities (permissions), and a set of subtab apps, which are custom tabs on the profile page.

Zones Class

Access information about Chatter Answers zones in your organization. Zones organize questions into logical groups, with each zone having its own focus and unique questions.

ConnectApi Input Classes

Some `ConnectApi` methods take arguments that are instances of `ConnectApi` input classes.

ConnectApi Output Classes

Most `ConnectApi` methods return instances of `ConnectApi` output classes.

ConnectApi Enums

Enums specific to the `ConnectApi` namespace.

ConnectApi Exceptions

The `ConnectApi` namespace contains exception classes.

ConnectApi Utilities

The `ConnectApi` namespace contains a utility class.

ConnectApi Release Notes

Use the Salesforce Release Notes to learn about the most recent updates and changes to the ConnectApi namespace in Apex.

ActionLinks Class

Create, delete, and get information about an action link group definition; get information about an action link group; get action link diagnostic information.

Namespace

ConnectApi

Usage

An action link is a button on a feed element. Clicking an action link can take a user to a Web page, initiate a file download, or invoke an API call to Salesforce or to an external server. An action link includes a URL and an HTTP method, and can include a request body and header information, such as an OAuth token for authentication. Use action links to integrate Salesforce and third-party services into the feed so that users can drive productivity and accelerate innovation.

There are two views of an action link and an action link group: the definition, and the context user's view. The definition includes potentially sensitive information, such as authentication information. The context user's view is filtered by visibility options and the values reflect the state of the context user.

Action link definition can be sensitive to a third party (for example, OAuth bearer token headers). For this reason, only calls made from the Apex namespace that created the action link definition can read, modify, or delete the definition. In addition, the user making the call must have created the definition or have View All Data permission. Use these methods to operate on action link group definitions (which contain action link definitions).

- createActionLinkGroupDefinition(communityId, actionLinkGroup)
- deleteActionLinkGroupDefinition(communityId, actionLinkGroupId)
- getActionLinkGroupDefinition(communityId, actionLinkGroupId)

Use these methods to operate on a context user's view of an action link or an action link group.

- getActionLink(communityId, actionLinkId)
- getActionLinkGroup(communityId, actionLinkGroupId)
- getActionLinkDiagnosticInfo(communityId, actionLinkId)

For information about how to use action links, see Working with Action Links.

ActionLinks Methods

These methods are for `ActionLinks`. All methods are static.

IN THIS SECTION:

createActionLinkGroupDefinition(communityId, actionLinkGroup)

Create an action link group definition. To associate an action link group with a feed element, first create an action link group definition. Then post a feed element with an associated actions capability.

deleteActionLinkGroupDefinition(communityId, actionLinkGroupId)

Delete an action link group definition. Deleting an action link group definition removes all references to it from feed elements.

getActionLink(communityId, actionLinkId)

Get information about an action link, including state for the context user.

getActionLinkDiagnosticInfo(communityId, actionLinkId)

Get diagnostic information returned when an action link executes. Diagnostic information is given only for users who can access the action link.

getActionLinkGroup(communityId, actionLinkGroupId)

Get information about an action link group including state for the context user.

getActionLinkGroupDefinition(communityId, actionLinkGroupId)

Get information about an action link group definition.

`createActionLinkGroupDefinition(communityId, actionLinkGroup)`

Create an action link group definition. To associate an action link group with a feed element, first create an action link group definition. Then post a feed element with an associated actions capability.

API Version

33.0

Requires Chatter

No

Signature

```
public static ConnectApi.ActionLinkGroupDefinition createActionLinkGroupDefinition(String communityId, ConnectApi.ActionLinkGroupDefinitionInput actionLinkGroup)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`actionLinkGroup`
Type: `ConnectApi.ActionLinkGroupDefinitionInput`

A `ConnectApi.ActionLinkGroupDefinitionInput` object that defines the action link group.

Return Value

Type: `ConnectApi.ActionLinkGroupDefinition`

Usage

An action link is a button on a feed element. Clicking an action link can take a user to a Web page, initiate a file download, or invoke an API call to Salesforce or to an external server. An action link includes a URL and an HTTP method, and can include a request body and header information, such as an OAuth token for authentication. Use action links to integrate Salesforce and third-party services into the feed so that users can drive productivity and accelerate innovation.

All action links must belong to a group. Action links in a group are mutually exclusive and share some properties. Define standalone actions in their own action group.

Information in the action link group definition can be sensitive to a third party (for example, OAuth bearer token headers). For this reason, only calls made from the Apex namespace that created the action link group definition can read, modify, or delete the definition. In addition, the user making the call must have created the definition or have View All Data permission.

Note: Invoking `ApiAsync` action links from an app requires a call to set the status. However, there isn't currently a way to set the status of an action link using Apex. To set the status, use Connect REST API. See the Action Link resource in the Connect REST API Developer Guidefor more information.

Example for Defining an Action Link and Posting with a Feed Element

For more information about this example, see Define an Action Link and Post with a Feed Element.

```
ConnectApi.ActionLinkGroupDefinitionInput actionLinkGroupDefinitionInput = new
ConnectApi.ActionLinkGroupDefinitionInput();
ConnectApi.ActionLinkDefinitionInput actionLinkDefinitionInput = new
ConnectApi.ActionLinkDefinitionInput();
ConnectApi.RequestHeaderInput requestHeaderInput1 = new ConnectApi.RequestHeaderInput();
ConnectApi.RequestHeaderInput requestHeaderInput2 = new ConnectApi.RequestHeaderInput();

// Create the action link group definition.
actionLinkGroupDefinitionInput.actionLinks = New
List<ConnectApi.ActionLinkDefinitionInput>();
actionLinkGroupDefinitionInput.executionsAllowed =
ConnectApi.ActionLinkExecutionsAllowed.OncePerUser;
actionLinkGroupDefinitionInput.category = ConnectApi.PlatformActionGroupCategory.Primary;
// To Do: Verify that the date is in the future.
// Action link groups are removed from feed elements on the expiration date.
datetime myDate = datetime.newInstance(2016, 3, 1);
actionLinkGroupDefinitionInput.expirationDate = myDate;

// Create the action link definition.
actionLinkDefinitionInput.actionType = ConnectApi.ActionLinkType.Api;
actionLinkDefinitionInput.actionUrl = '/services/data/v33.0/chatter/feed-elements';
actionLinkDefinitionInput.headers = new List<ConnectApi.RequestHeaderInput>();
actionLinkDefinitionInput.labelKey = 'Post';
actionLinkDefinitionInput.method = ConnectApi.HttpRequestMethod.HttpPost;
actionLinkDefinitionInput.requestBody = '{\"subjectId\": \"me\",\"feedElementType\":
\"FeedItem\",\"body\": {\"messageSegments\": [{\"type\": \"Text\",\"text\": \"This is a
test post created via an API action link.\"}]}}';
actionLinkDefinitionInput.requiresConfirmation = true;
```

```
// To Do: Substitute an OAuth value for your Salesforce org.
requestHeaderInput1.name = 'Authorization';
requestHeaderInput1.value = 'OAuth
00DD00000007WNP!ARsAQCwoeV0zzAV847FTl4zF.85w.EwsPbUgXR4SAjsp';
actionLinkDefinitionInput.headers.add(requestHeaderInput1);

requestHeaderInput2.name = 'Content-Type';
requestHeaderInput2.value = 'application/json';
actionLinkDefinitionInput.headers.add(requestHeaderInput2);

// Add the action link definition to the action link group definition.
actionLinkGroupDefinitionInput.actionLinks.add(actionLinkDefinitionInput);

// Instantiate the action link group definition.
ConnectApi.ActionLinkGroupDefinition actionLinkGroupDefinition =
ConnectApi.ActionLinks.createActionLinkGroupDefinition(Network.getNetworkId(),
actionLinkGroupDefinitionInput);

ConnectApi.FeedItemInput feedItemInput = new ConnectApi.FeedItemInput();
ConnectApi.FeedElementCapabilitiesInput feedElementCapabilitiesInput = new
ConnectApi.FeedElementCapabilitiesInput();
ConnectApi.AssociatedActionsCapabilityInput associatedActionsCapabilityInput = new
ConnectApi.AssociatedActionsCapabilityInput();
ConnectApi.MessageBodyInput messageBodyInput = new ConnectApi.MessageBodyInput();
ConnectApi.TextSegmentInput textSegmentInput = new ConnectApi.TextSegmentInput();

// Set the properties of the feedItemInput object.
feedItemInput.body = messageBodyInput;
feedItemInput.capabilities = feedElementCapabilitiesInput;
feedItemInput.subjectId = 'me';

// Create the text for the post.
messageBodyInput.messageSegments = new List<ConnectApi.MessageSegmentInput>();
textSegmentInput.text = 'Click to post a feed item.';
messageBodyInput.messageSegments.add(textSegmentInput);


// The feedElementCapabilitiesInput object holds the capabilities of the feed item.
// Define an associated actions capability to hold the action link group.
// The action link group ID is returned from the call to create the action link group
definition.
feedElementCapabilitiesInput.associatedActions = associatedActionsCapabilityInput;
associatedActionsCapabilityInput.actionLinkGroupIds = new List<String>();
associatedActionsCapabilityInput.actionLinkGroupIds.add(actionLinkGroupDefinition.id);

// Post the feed item.
ConnectApi.FeedElement feedElement =
ConnectApi.ChatterFeeds.postFeedElement(Network.getNetworkId(), feedItemInput);
```

Example for Defining an Action Link in a Template and Posting with a Feed Element

For more information about this example, see Define an Action Link in a Template and Post with a Feed Element.

```
// Get the action link group template Id.
ActionLinkGroupTemplate template = [SELECT Id FROM ActionLinkGroupTemplate WHERE
DeveloperName='Doc_Example'];

// Add binding name-value pairs to a map.
// The names are defined in the action link template(s) associated with the action link
group template.
// Get them from Setup UI or SOQL.
Map<String, String> bindingMap = new Map<String, String>();
bindingMap.put('ApiVersion', 'v33.0');
bindingMap.put('Text', 'This post was created by an API action link.');
bindingMap.put('SubjectId', 'me');

// Create ActionLinkTemplateBindingInput objects from the map elements.
List<ConnectApi.ActionLinkTemplateBindingInput> bindingInputs = new
List<ConnectApi.ActionLinkTemplateBindingInput>();

for (String key : bindingMap.keySet()) {
    ConnectApi.ActionLinkTemplateBindingInput bindingInput = new
ConnectApi.ActionLinkTemplateBindingInput();
    bindingInput.key = key;
    bindingInput.value = bindingMap.get(key);
    bindingInputs.add(bindingInput);
}

// Set the template Id and template binding values in the action link group definition.
ConnectApi.ActionLinkGroupDefinitionInput actionLinkGroupDefinitionInput = new
ConnectApi.ActionLinkGroupDefinitionInput();
actionLinkGroupDefinitionInput.templateId = template.id;
actionLinkGroupDefinitionInput.templateBindings = bindingInputs;

// Instantiate the action link group definition.
ConnectApi.ActionLinkGroupDefinition actionLinkGroupDefinition =
 ConnectApi.ActionLinks.createActionLinkGroupDefinition(Network.getNetworkId(),
actionLinkGroupDefinitionInput);

ConnectApi.FeedItemInput feedItemInput = new ConnectApi.FeedItemInput();
ConnectApi.FeedElementCapabilitiesInput feedElementCapabilitiesInput = new
ConnectApi.FeedElementCapabilitiesInput();
ConnectApi.AssociatedActionsCapabilityInput associatedActionsCapabilityInput = new
ConnectApi.AssociatedActionsCapabilityInput();
ConnectApi.MessageBodyInput messageBodyInput = new ConnectApi.MessageBodyInput();
ConnectApi.TextSegmentInput textSegmentInput = new ConnectApi.TextSegmentInput();

// Define the FeedItemInput object to pass to postFeedElement
feedItemInput.body = messageBodyInput;
feedItemInput.capabilities = feedElementCapabilitiesInput;
feedItemInput.subjectId = 'me';

// The MessageBodyInput object holds the text in the post
messageBodyInput.messageSegments = new List<ConnectApi.MessageSegmentInput>();
```

```
textSegmentInput.text = 'Click to post a feed item.';
messageBodyInput.messageSegments.add(textSegmentInput);


// The FeedElementCapabilitiesInput object holds the capabilities of the feed item.
// For this feed item, we define an associated actions capability to hold the action link
 group.
// The action link group ID is returned from the call to create the action link group
definition.
feedElementCapabilitiesInput.associatedActions = associatedActionsCapabilityInput;
associatedActionsCapabilityInput.actionLinkGroupIds = new List<String>();
associatedActionsCapabilityInput.actionLinkGroupIds.add(actionLinkGroupDefinition.id);

// Post the feed item.
ConnectApi.FeedElement feedElement =
ConnectApi.ChatterFeeds.postFeedElement(Network.getNetworkId(), feedItemInput);
```

`deleteActionLinkGroupDefinition(communityId, actionLinkGroupId)`

Delete an action link group definition. Deleting an action link group definition removes all references to it from feed elements.

API Version

33.0

Requires Chatter

No

Signature

```
public static void deleteActionLinkGroupDefinition(String communityId, String
actionLinkGroupId)
```

Parameters

communityId
: Type: String

  ID for an Experience Cloud site, `internal`, or `null`.

actionLinkGroupId
: Type: String

  The ID of the action link group.

Return Value

Type: Void

Usage

Information in the action link group definition can be sensitive to a third party (for example, OAuth bearer token headers). For this reason, only calls made from the Apex namespace that created the action link group definition can read, modify, or delete the definition. In addition, the user making the call must have created the definition or have View All Data permission.

`getActionLink(communityId, actionLinkId)`

Get information about an action link, including state for the context user.

API Version

33.0

Requires Chatter

No

Signature

```
public static ConnectApi.PlatformAction getActionLink(String communityId, String actionLinkId)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`actionLinkId`
Type: String

The ID of the action link.

Return Value

Type: `ConnectApi.PlatformAction`

`getActionLinkDiagnosticInfo(communityId, actionLinkId)`

Get diagnostic information returned when an action link executes. Diagnostic information is given only for users who can access the action link.

API Version

33.0

Requires Chatter

No

Signature

```
public static ConnectApi.ActionLinkDiagnosticInfo getActionLinkDiagnosticInfo(String communityId, String actionLinkId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

actionLinkId
Type: String

The ID of the action link.

Return Value

Type: `ConnectApi.ActionLinkDiagnosticInfo`

getActionLinkGroup(communityId, actionLinkGroupId)

Get information about an action link group including state for the context user.

API Version

33.0

Requires Chatter

No

Signature

```
public static ConnectApi.PlatformActionGroup getActionLinkGroup(String communityId, String actionLinkGroupId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

actionLinkGroupId
Type: String

The ID of the action link group.

Return Value

Type: `ConnectApi.PlatformActionGroup`

Usage

All action links must belong to a group. Action links in a group are mutually exclusive and share some properties. Action link groups are accessible by clients, unlike action link group definitions.

`getActionLinkGroupDefinition(communityId, actionLinkGroupId)`

Get information about an action link group definition.

API Version

33.0

Requires Chatter

No

Signature

```
public static ConnectApi.ActionLinkGroupDefinition getActionLinkGroupDefinition(String communityId, String actionLinkGroupId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

actionLinkGroupId
Type: String

The ID of the action link group.

Return Value

Type: `ConnectApi.ActionLinkGroupDefinition`

Usage

Information in the action link group definition can be sensitive to a third party (for example, OAuth bearer token headers). For this reason, only calls made from the Apex namespace that created the action link group definition can read, modify, or delete the definition. In addition, the user making the call must have created the definition or have View All Data permission.

Announcements Class

Access information about announcements and post announcements.

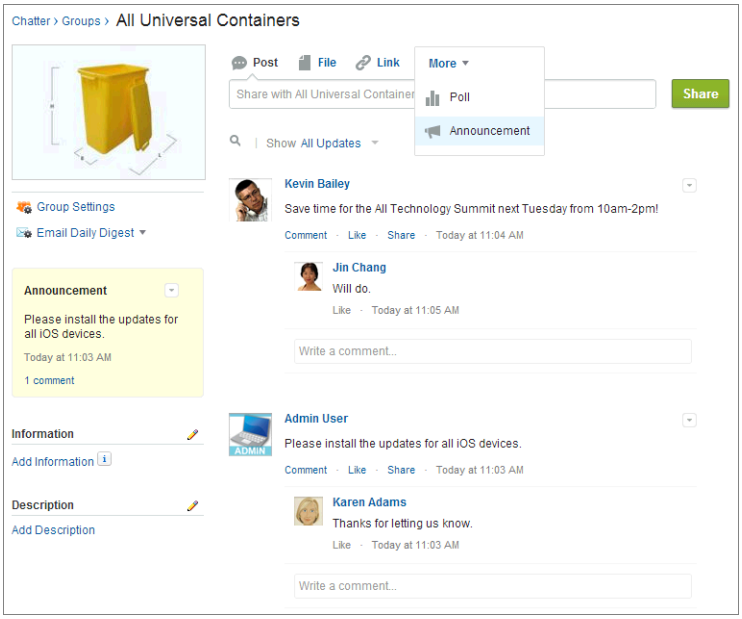
Namespace

ConnectApi

Usage

Use the `ConnectApi.Announcements` class to get, create, update, and delete announcements. Use an announcement to highlight information. Users can discuss, like, and post comments on announcements. Deleting the feed post deletes the announcement.

This image shows an announcement displayed in a group. Creating an announcement also creates a feed item with the announcement text.



An announcement displays in a designated location in the Salesforce UI until 11:59 p.m. on its expiration date, unless it's deleted or replaced by another announcement.

Announcements Methods

These methods are for `Announcements`. All methods are static.

All methods in this class require Chatter and are subject to the per user, per namespace, per hour rate limit.

IN THIS SECTION:

deleteAnnouncement(communityId, announcementId)
Delete an announcement.

getAnnouncement(communityId, announcementId)
Get an announcement.

getAnnouncements(communityId, parentId)
Get the first page of announcements.

getAnnouncements(communityId, parentId, pageParam, pageSize)
Get a page of announcements.

postAnnouncement(communityId, announcement)
Post an announcement.

updateAnnouncement(communityId, announcementId, expirationDate)
Update the expiration date of an announcement.

deleteAnnouncement(communityId, announcementId)

Delete an announcement.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static void deleteAnnouncement(String communityId, String announcementId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

announcementId
Type: String

An announcement ID, which has a prefix of 0BT.

Return Value

Type: Void

Usage

To get a list of announcements in a group, call `getAnnouncements(communityId, parentId)` or `getAnnouncements(communityId, parentId, pageParam, pageSize)`.

To post an announcement to a group, call `postAnnouncement(communityId, announcement)` .

getAnnouncement(communityId, announcementId)

Get an announcement.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Announcement getAnnouncement(String communityId, String
announcementId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

announcementId
Type: String

An announcement ID, which has a prefix of 0BT.

Return Value

Type: `ConnectApi.Announcement`

Usage

To get a list of announcements in a group, call `getAnnouncements(communityId, parentId)` or `getAnnouncements(communityId, parentId, pageParam, pageSize)`.

To post an announcement to a group, call `postAnnouncement(communityId, announcement)` .

`getAnnouncements(communityId, parentId)`

Get the first page of announcements.

API Version

36.0

Available to Guest Users

38.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.AnnouncementPage getAnnouncements(String communityId, String
parentId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

parentId
Type: String

ID of the parent entity for the announcement, that is, a group ID when the announcement appears in a group.

Return Value

Type: `ConnectApi.AnnouncementPage`

getAnnouncements(communityId, parentId, pageParam, pageSize)

Get a page of announcements.

API Version

36.0

Available to Guest Users

38.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.AnnouncementPage getAnnouncements(String communityId, String
parentId, Integer pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

parentId
Type: String

ID of the parent entity for the announcement, that is, a group ID when the announcement appears in a group.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of announcements per page.

Return Value

Type: ConnectApi.AnnouncementPage

postAnnouncement(communityId, announcement)

Post an announcement.

API Version

36.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Announcement postAnnouncement(String communityId,
ConnectApi.AnnouncementInput announcement)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

announcement
Type: ConnectApi.AnnouncementInput

A `ConnectApi.AnnouncementInput` object.

Return Value

Type: ConnectApi.Announcement

updateAnnouncement(communityId, announcementId, expirationDate)

Update the expiration date of an announcement.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Announcement updateAnnouncement(String communityId, String
announcementId, Datetime expirationDate)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

announcementId
Type: String

An announcement ID, which has a prefix of 0BT.

expirationDate
Type: Datetime

The Salesforce UI displays an announcement until 11:59 p.m. on this date unless another announcement is posted first. The Salesforce UI ignores the time value in the `expirationDate`. However, you can use the time value to create your own display logic in your own UI.

Return Value

Type: `ConnectApi.Announcement`

Usage

To get a list of announcements in a group, call `getAnnouncements(communityId, parentId)` or `getAnnouncements(communityId, parentId, pageParam, pageSize)`.

To post an announcement to a group, call `postAnnouncement(communityId, announcement)` .

BotVersionActivation Class

Access and update activation information of a bot version.

Namespace

ConnectApi

BotVersionActivation Methods

These methods are for `BotVersionActivation`. All methods are static.

IN THIS SECTION:

getVersionActivationInfo(botVersionId)
Get the active or inactive status of the bot version.

updateVersionStatus(botVersionId, status, postBody)
Update the status of the specified bot version.

`getVersionActivationInfo(botVersionId)`

Get the active or inactive status of the bot version.

API Version

50.0

Requires Chatter

No

Signature

```
public static ConnectApi.BotVersionActivationInfo getVersionActivationInfo(String
botVersionId)
```

Parameters

botVersionId
Type: String

ID of the bot version.

Return Value

Type: ConnectApi.BotVersionActivationInfo

Usage

To access this method, enable the bot feature, and the user must be an admin or have the Manage Bots or Manage Bots Training Data user permissions.

updateVersionStatus(botVersionId, status, postBody)

Update the status of the specified bot version.

API Version

50.0

Requires Chatter

No

Signature

```
public static ConnectApi.BotVersionActivationInfo updateVersionStatus(String
botVersionId, ConnectApi.BotVersionActivationStatus status,
ConnectApi.BotVersionActivationInput postBody)
```

Parameters

botVersionId
Type: String

ID of the bot version.

status
Type: ConnectApi.BotVersionActivationStatus

Activation status of the bot version. Values are:

- Active
- Inactive

Activation status must be specified in the *status* or *postBody* parameter.

postBody
Type: ConnectApi.BotVersionActivationInput

Parameters to update for the bot version. Activation status must be specified in the *status* or *postBody* parameter.

Return Value

Type: ConnectApi.BotVersionActivationInfo

Usage

To access this method, enable the bot feature, and the user must be an admin or have the Manage Bots or Manage Bots Training Data user permissions.

CdpCalculatedInsight Class

Create, delete, get, run, and update Data Cloud calculated insights.

Namespace

ConnectApi

CdpCalculatedInsight Methods

These methods are for `CdpCalculatedInsight`. All methods are static.

IN THIS SECTION:

createCalculatedInsight(input)
Create a calculated insight.

deleteCalculatedInsight(apiName)
Delete a calculated insight.

getCalculatedInsight(apiName)
Get a calculated insight.

getCalculatedInsights(definitionType, batchSize, offset, orderby, dataspace)
Get calculated insights.

getCalculatedInsights(definitionType, batchSize, offset, orderby, dataspace, pageToken)
Get a page of calculated insights.

runCalculatedInsight(apiName)
Run a calculated insight.

updateCalculatedInsight(apiName, input)
Update a calculated insight.

createCalculatedInsight(input)

Create a calculated insight.

API Version

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpCalculatedInsightOutput
createCalculatedInsight(ConnectApi.CdpCalculatedInsightInput input)
```

Parameters

input
Type: ConnectApi.CdpCalculatedInsightInput
Input representation for a calculated insight.

Return Value

Type: ConnectApi.CdpCalculatedInsightOutput

deleteCalculatedInsight(apiName)

Delete a calculated insight.

API Version

57.0

Requires Chatter

No

Signature

```
public static Void deleteCalculatedInsight(String apiName)
```

Parameters

apiName
Type: String

API name of the calculated insight to delete.

Return Value

Type: Void

getCalculatedInsight(apiName)

Get a calculated insight.

API Version

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpCalculatedInsightOutput getCalculatedInsight(String apiName)
```

Parameters

apiName
Type: String

API name of the calculated insight to get.

Return Value

Type: ConnectApi.CdpCalculatedInsightOutput

getCalculatedInsights(definitionType, batchSize, offset, orderby, dataspace)

Get calculated insights.

API Version

56.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpCalculatedInsightPage getCalculatedInsights(String
definitionType, Integer batchSize, Integer offset, String orderby, String dataspace)
```

Parameters

definitionType
Type: String

Definition type of the calculated insight. Values are:

- `CALCULATED_METRIC`
- `CALCULATED_METRIC`
- `CALCULATED_METRIC`

batchSize
Type: Integer

Number of items to return. Values are from 1–200. If unspecified, the default value is `25`.

offset
Type: Integer

Number of rows to skip before returning results. If unspecified, no rows are skipped.

orderby
Type: String

Sort order for the result set, such as `GenderId__c ASC,Occupation__c DESC`. If unspecified, items are returned in the order they are retrieved.

dataspace
Type: String

Name of the data space.

Return Value

Type: `ConnectApi.CdpCalculatedInsightPage`

`getCalculatedInsights(definitionType, batchSize, offset, orderby, dataspace, pageToken)`

Get a page of calculated insights.

API Version

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpCalculatedInsightPage getCalculatedInsights(String
definitionType, Integer batchSize, Integer offset, String orderby, String dataspace,
String pageToken)
```

Parameters

definitionType
Type: String

Definition type of the calculated insight. Values are:

- `CALCULATED_METRIC`
- `CALCULATED_METRIC`
- `CALCULATED_METRIC`

batchSize
Type: Integer

Number of items to return. Values are from 1–200. If unspecified, the default value is `25`.

offset
Type: Integer

Number of rows to skip before returning results. If unspecified, no rows are skipped.

orderby
Type: String

Sort order for the result set, such as `GenderId__c ASC,Occupation__c DESC`. If unspecified, items are returned in the order they are retrieved.

dataspace
Type: String

Name of the data space.

pageToken
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

Return Value

Type: `ConnectApi.CdpCalculatedInsightPage`

`runCalculatedInsight(apiName)`

Run a calculated insight.

API Version

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpCalculatedInsightStandardActionesponseRepresentation
runCalculatedInsight(String apiName)
```

Parameters

apiName
Type: String

API name of the calculated insight to run.

Return Value

Type: ConnectApi.CdpCalculatedInsightStandardActionResponseRepresentation

updateCalculatedInsight(apiName, input)

Update a calculated insight.

API Version

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpCalculatedInsightOutput updateCalculatedInsight(String apiName, ConnectApi.CdpCalculatedInsightInput input)
```

Parameters

apiName
Type: String

API name of the calculated insight to update.

input
Type: ConnectApi.CdpCalculatedInsightInput

Input representation for a calculated insight.

Return Value

Type: ConnectApi.CdpCalculatedInsightOutput

CdpIdentityResolution Class

Create, delete, get, run, and update Data Cloud identity resolution rulesets.

Namespace

ConnectApi

CdpIdentityResolution Methods

These methods are for `CdpIdentityResolution`. All methods are static.

IN THIS SECTION:

createIdentityResolution(input)
Create an identity resolution ruleset.

deleteIdentityResolution(identityResolution)
Delete an identity resolution ruleset.

getIdentityResolution(identityResolution)
Get an identity resolution ruleset.

getIdentityResolutions()
Get identity resolution rulesets.

runIdentityResolutionNow(identityResolution, input)
Trigger an immediate identity resolution ruleset job run.

updateIdentityResolution(identityResolution, input)
Update an identity resolution ruleset.

`createIdentityResolution(input)`

Create an identity resolution ruleset.

API Version

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpIdentityResolutionOutput
createIdentityResolution(ConnectApi.CdpIdentityResolutionConfigInput input)
```

Parameters

input
Type: `ConnectApi.CdpIdentityResolutionConfigInput`

Input representation for creating an identity resolution ruleset.

Return Value

Type: `ConnectApi.CdpIdentityResolutionOutput`

deleteIdentityResolution(identityResolution)

Delete an identity resolution ruleset.

API Version

57.0

Requires Chatter

No

Signature

```
public static Void deleteIdentityResolution(String identityResolution)
```

Parameters

identityResolution
Type: String

Developer name or ID of the ruleset.

Return Value

Type: Void

getIdentityResolution(identityResolution)

Get an identity resolution ruleset.

API Version

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpIdentityResolutionOutput getIdentityResolution(String
identityResolution)
```

Parameters

identityResolution
Type: String

Developer name or ID of the ruleset.

Return Value

Type: ConnectApi.CdpIdentityResolutionOutput

`getIdentityResolutions()`

Get identity resolution rulesets.

API Version

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpIdentityResolutionsOutput getIdentityResolutions()
```

Return Value

Type: ConnectApi.CdpIdentityResolutionsOutput

`runIdentityResolutionNow(identityResolution, input)`

Trigger an immediate identity resolution ruleset job run.

API Version

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpIdentityResolutionRunNowOutput
runIdentityResolutionNow(String identityResolution,
ConnectApi.CdpIdentityResolutionRunNowInput input)
```

Parameters

identityResolution
Type: String

Developer name of the ruleset.

input
Type: ConnectApi.CdpIdentityResolutionRunNowInput

Input representation for running an identity resolution ruleset job on demand.

Return Value

Type: `ConnectApi.CdpIdentityResolutionRunNowOutput`

`updateIdentityResolution(identityResolution, input)`

Update an identity resolution ruleset.

API Version

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpIdentityResolutionOutput updateIdentityResolution(String identityResolution, ConnectApi.CdpIdentityResolutionConfigPatchInput input)
```

Parameters

identityResolution
Type: String

Developer name or ID of the ruleset.

input
Type: `ConnectApi.CdpIdentityResolutionConfigPatchInput`

Input representation for updating an identity resolution ruleset.

Return Value

Type: `ConnectApi.CdpIdentityResolutionOutput`

CdpQuery Class

Get Data Cloud metadata and query data.

Namespace

ConnectApi

CdpQuery Methods

These methods are for `CdpQuery`. All methods are static.

IN THIS SECTION:

getAllMetadata()
Get all metadata, including Calculated Insights, Engagement, Profile, and other objects, as well as their relationships to other objects.

getAllMetadata(entityType, entityCategory, entityName)
Get all metadata, filtering for entity type, category, and name.

getInsightsMetadata()
Get Insight metadata, including Calculated Insight objects, their dimensions and measures.

getInsightsMetadata(ciName)
Get metadata for a Calculated Insight object. Metadata includes dimensions and measures.

getProfileMetadata()
Get metadata for data model objects in the profile category, including Individual, Contact Point Email, Unified Individual, and Contact Point Address objects. Metadata includes the objects, their fields, and category.

getProfileMetadata(dataModelName)
Get metadata for a data model object in the profile category, such as Individual, Contact Point Email, Unified Individual, and Contact Point Address. Metadata includes the list of fields, data types, and indexes available for lookup.

nextBatchAnsiSqlV2(nextBatchId)
Get the next batch of data across data model, lake, unified, and linked objects.

queryANSISql(input)
Synchronously query data across data model, lake, unified, and linked objects. This query returns up to 4,999 rows.

queryANSISql(input, batchSize, offset, orderby)
Synchronously query data across data model, lake, unified, and linked objects. Specify batch size, offset, and order of the results. This query returns up to 4,999 rows.

queryAnsiSqlV2(input)
Query up to 8 MB of data across data model, lake, unified, and linked objects.

queryCalculatedInsights(ciName, dimensions, measures, orderby, filters, batchSize, offset)
Query a Calculated Insight object.

queryCalculatedInsights(ciName, dimensions, measures, orderby, filters, batchSize, offset, timeGranularity)
Query a Calculated Insight object within a specified time range.

queryProfileApi(dataModelName, filters, fields, batchSize, offset, orderby)
Query a Profile data model object using filters.

queryProfileApi(dataModelName, id, searchKey, filters, fields, batchSize, offset, orderby)
Query a Profile data model object using filters and a search key.

queryProfileApi(dataModelName, id, childDataModelName, searchKey, filters, fields, batchSize, offset, orderby)
Query a Profile data model object and a child object using filters and a search key.

queryProfileApi(dataModelName, id, ciName, searchKey, dimensions, measures, filters, fields, batchSize, offset, orderby)
Query a Profile data model object and a Calculated Insight object using filters and a search key.

queryProfileApi(dataModelName, id, ciName, searchKey, dimensions, measures, filters, fields, batchSize, offset, orderby, timeGranularity)
Query a Profile data model object and a Calculated Insight object using filters, a search key, and a time range.

universalIdLookupBySourceId(entityName, dataSourceId, dataSourceObjectId, sourceRecordId)
Look up objects by source ID.

`getAllMetadata()`

Get all metadata, including Calculated Insights, Engagement, Profile, and other objects, as well as their relationships to other objects.

API Version

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpQueryMetadataOutput getAllMetadata()
```

Return Value

Type: `ConnectApi.CdpQueryMetadataOutput`

`getAllMetadata(entityType, entityCategory, entityName)`

Get all metadata, filtering for entity type, category, and name.

API Version

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpQueryMetadataOutput getAllMetadata(String entityType, String
entityCategory, String entityName)
```

Parameters

`entityType`
Type: String

Type of metadata entity requested. Valid values are `DataLakeObject`, `DataModelObject`, and `CalculatedInsight`. If unspecified, all types are returned.

`entityCategory`
Type: String

Category of the metadata entity. Valid values are `Profile`, `Engagement`, and `Related`. If unspecified, all category entities are returned.

`entityName`
Type: String

Metadata name of the entity, for example `UnifiedIndividual__dlm`. If unspecified, a complete list of entities is returned.

Return Value

Type: `ConnectApi.CdpQueryMetadataOutput`

`getInsightsMetadata()`

Get Insight metadata, including Calculated Insight objects, their dimensions and measures.

API Version

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpQueryMetadataOutput getInsightsMetadata()
```

Return Value

Type: `ConnectApi.CdpQueryMetadataOutput`

`getInsightsMetadata(ciName)`

Get metadata for a Calculated Insight object. Metadata includes dimensions and measures.

API Version

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpQueryMetadataOutput getInsightsMetadata(String ciName)
```

Parameters

`ciName`
Type: String

Name of the Calculated Insight object, for example, `IndividualChildrenCount__cio`.

Return Value

Type: `ConnectApi.CdpQueryMetadataOutput`

getProfileMetadata()

Get metadata for data model objects in the profile category, including Individual, Contact Point Email, Unified Individual, and Contact Point Address objects. Metadata includes the objects, their fields, and category.

API Version

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpQueryMetadataOutput getProfileMetadata()
```

Return Value

Type: `ConnectApi.CdpQueryMetadataOutput`

getProfileMetadata(dataModelName)

Get metadata for a data model object in the profile category, such as Individual, Contact Point Email, Unified Individual, and Contact Point Address. Metadata includes the list of fields, data types, and indexes available for lookup.

API Version

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpQueryMetadataOutput getProfileMetadata(String dataModelName)
```

Parameters

dataModelName
Type: String

Name of the data model object, for example, `UnifiedIndividual__dlm`.

Return Value

Type: `ConnectApi.CdpQueryMetadataOutput`

nextBatchAnsiSqlV2(nextBatchId)

Get the next batch of data across data model, lake, unified, and linked objects.

API Version

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpQueryOutputV2 nextBatchAnsiSqlV2(String nextBatchId)
```

Parameters

nextBatchId
Type: String

ID of the next batch. See the Usage section for more information.

Return Value

Type: `ConnectApi.CdpQueryOutputV2`

Usage

Initially, use the `queryAnsiSqlV2(input)` method to query up to 8 MB of data. Use the `nextBatchId` from the `ConnectApi.CdpQueryOutputV2` output class as the *nextBatchId* parameter in this method to get the next batch of data. You can continue using subsequent next batch IDs for up to an hour.

queryANSISql(input)

Synchronously query data across data model, lake, unified, and linked objects. This query returns up to 4,999 rows.

Note: A newer version of the Query API is available. We recommend using `queryAnsiSqlV2(input)` and `nextBatchAnsiSqlV2(nextBatchId)` to take advantage of subsequent requests and larger response sizes.

API Version

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpQueryOutput queryANSISql(ConnectApi.CdpQueryInput input)
```

Parameters

input
Type: `ConnectApi.CdpQueryInput`

A `ConnectApi.CdpQueryInput` body with the SQL query.

Return Value

Type: `ConnectApi.CdpQueryOutput`

`queryANSISql(input, batchSize, offset, orderby)`

Synchronously query data across data model, lake, unified, and linked objects. Specify batch size, offset, and order of the results. This query returns up to 4,999 rows.

Note: A newer version of the Query API is available. We recommend using `queryAnsiSqlV2(input)` and `nextBatchAnsiSqlV2(nextBatchId)` to take advantage of subsequent requests and larger response sizes.

API Version

53.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpQueryOutput queryANSISql(ConnectApi.CdpQueryInput input,
Integer batchSize, Integer offset, String orderby)
```

Parameters

input
Type: `ConnectApi.CdpQueryInput`

A `ConnectApi.CdpQueryInput` body with the SQL query.

batchSize
Type: Integer

Number of records to return. Values are from 1–4999. The default value is 4999.

offset
Type: Integer

Number of rows to skip before returning results. The sum of *offset* and *batchSize* must be less than 2147483647. The default value is 0.

orderby
Type: String

Comma-separated values to sort the results in ascending or descending order, for example, `GenderId__c ASC,Occupation__c DESC`.

Return Value

Type: `ConnectApi.CdpQueryOutput`

queryAnsiSqlV2(input)

Query up to 8 MB of data across data model, lake, unified, and linked objects.

API Version

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpQueryOutputV2 queryAnsiSqlV2(ConnectApi.CdpQueryInput input)
```

Parameters

input
Type: ConnectApi.CdpQueryInput

A ConnectApi.CdpQueryInput body with the SQL query.

Return Value

Type: ConnectApi.CdpQueryOutputV2

Usage

Use the nextBatchId in the ConnectApi.CdpQueryOutputV2 output class as the *nextBatchId* parameter in the nextBatchAnsiSqlV2(nextBatchId) method to continue getting batches of data for up to an hour.

queryCalculatedInsights(ciName, dimensions, measures, orderby, filters, batchSize, offset)

Query a Calculated Insight object.

API Version

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpQueryOutput queryCalculatedInsights(String ciName, String dimensions, String measures, String orderby, String filters, Integer batchSize, Integer offset)
```

Parameters

ciName
Type: String

Name of the Calculated Insight object, for example, `IndividualChildrenCount__cio`.

dimensions
Type: String

Comma-separated list of up to 10 dimensions, such as `GenderId__c`, to project. If unspecified, this parameter includes all of the available dimensions.

measures
Type: String

Comma-separated list of up to 5 measures, such as `TotalSales__c`, to project. If unspecified, this parameter includes all of the available measures.

orderby
Type: String

Sort order for the result set, such as `GenderId__c ASC,Occupation__c DESC`. If unspecified, items are returned in the order they are retrieved.

filters
Type: String

Filter the result set to a more narrow scope or specific type, such as `[GenderId__c=Male,FirstName__c=Angel]`.

batchSize
Type: Integer

Number of items to return. Values are from 1–4,999. If unspecified, the default value is `4999`.

offset
Type: Integer

Number of rows to skip before returning results. If unspecified, no rows are skipped.

Return Value

Type: `ConnectApi.CdpQueryOutput`

queryCalculatedInsights(ciName, dimensions, measures, orderby, filters, batchSize, offset, timeGranularity)

Query a Calculated Insight object within a specified time range.

API Version

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpQueryOutput queryCalculatedInsights(String ciName, String dimensions, String measures, String orderby, String filters, Integer batchSize, Integer offset, String timeGranularity)
```

Parameters

ciName
Type: String

Name of the Calculated Insight object, for example, `IndividualChildrenCount__cio`.

dimensions
Type: String

Comma-separated list of up to 10 dimensions, such as `GenderId__c`, to project. If unspecified, this parameter includes all of the available dimensions.

measures
Type: String

Comma-separated list of up to 5 measures, such as `TotalSales__c`, to project. If unspecified, this parameter includes all of the available measures.

orderby
Type: String

Sort order for the result set, such as `GenderId__c ASC,Occupation__c DESC`. If unspecified, items are returned in the order they are retrieved.

filters
Type: String

Filter the result set to a more narrow scope or specific type, such as `[GenderId__c=Male,FirstName__c=Angel]`.

batchSize
Type: Integer

Number of items to return. Values are from 1–4,999. If unspecified, the default value is `4999`.

offset
Type: Integer

Number of rows to skip before returning results. If unspecified, no rows are skipped.

timeGranularity
Type: String

Time range for the measures. Values are:

- `HOUR`
- `DAY`
- `MONTH`
- `QUARTER`
- `YEAR`

If unspecified, no time range is applied.

Return Value

Type: ConnectApi.CdpQueryOutput

queryProfileApi(dataModelName, filters, fields, batchSize, offset, orderby)

Query a Profile data model object using filters.

API Version

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpQueryOutput queryProfileApi(String dataModelName, String filters, String fields, Integer batchSize, Integer offset, String orderby)
```

Parameters

dataModelName
Type: String

Name of the data model object, for example, `UnifiedIndividual__dlm`.

filters
Type: String

Comma-separated list of equality expressions within square brackets, for example, `[FirstName__c=DON]`.

fields
Type: String

Comma-separated list of up to 50 field names that you want to include in the result, for example, `Id__c,FirstName__c, GenderId__c,Occupation__c`. If unspecified, `Id__c` is returned.

batchSize
Type: Integer

Number of items to return. Values are from 1–4,999. If unspecified, the default value is `100`.

offset
Type: Integer

Number of rows to skip before returning results. If unspecified, no rows are skipped.

orderby
Type: String

Sort order for the result set, such as `GenderId__c ASC,Occupation__c DESC`. If unspecified, items are returned in the order they are retrieved.

Return Value

Type: ConnectApi.CdpQueryOutput

queryProfileApi(dataModelName, id, searchKey, filters, fields, batchSize, offset, orderby)

Query a Profile data model object using filters and a search key.

API Version

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpQueryOutput queryProfileApi(String dataModelName, String id, String searchKey, String filters, String fields, Integer batchSize, Integer offset, String orderby)
```

Parameters

dataModelName
Type: String

Name of the data model object, for example, `UnifiedIndividual__dlm`.

id
Type: String

Value of the primary or secondary key field, for example, `John`. If unspecified, defaults to the value of the primary key field.

searchKey
Type: String

If a field other than the primary key is used, name of the key field, for example, `FirstName__c`.

filters
Type: String

Comma-separated list of equality expressions within square brackets, for example, `[FirstName__c=DON]`.

fields
Type: String

Comma-separated list of up to 50 field names that you want to include in the result, for example, `Id__c,FirstName__c, GenderId__c,Occupation__c`. If unspecified, `Id__c` is returned.

batchSize
Type: Integer

Number of items to return. Values are from 1–4,999. If unspecified, the default value is `100`.

offset
Type: Integer

Number of rows to skip before returning results. If unspecified, no rows are skipped.

orderby
Type: String

Sort order for the result set, such as `GenderId__c ASC,Occupation__c DESC`. If unspecified, items are returned in the order they are retrieved.

Return Value

Type: `ConnectApi.CdpQueryOutput`

queryProfileApi(dataModelName, id, childDataModelName, searchKey, filters, fields, batchSize, offset, orderby)

Query a Profile data model object and a child object using filters and a search key.

API Version

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpQueryOutput queryProfileApi(String dataModelName, String id, String childDataModelName, String searchKey, String filters, String fields, Integer batchSize, Integer offset, String orderby)
```

Parameters

dataModelName
Type: String

Name of the data model object, for example, `UnifiedIndividual__dlm`.

id
Type: String

Value of the primary or secondary key field, for example, `John`. If unspecified, defaults to the value of the primary key field.

childDataModelName
Type: String

Name of the child data model object, for example, `UnifiedContactPointEmail__dlm`.

searchKey
Type: String

If a field other than the primary key is used, name of the key field, for example, `FirstName__c`.

filters
Type: String

Comma-separated list of equality expressions within square brackets, for example, `[FirstName__c=DON]`. Filters are applied to the parent object only.

fields
Type: String

Comma-separated list of child object field names that you want to include in the result, for example, `Id__c,EmailAddress__c`. If unspecified, the first 10 alphabetically sorted fields are returned.

batchSize
Type: Integer

Number of items to return. Values are from 1–4,999. If unspecified, the default value is `100`.

offset
Type: Integer

Number of rows to skip before returning results. If unspecified, no rows are skipped.

orderby
Type: String

Sort order for the result set, such as `GenderId__c ASC,Occupation__c DESC`. If unspecified, items are returned in the order they are retrieved.

Return Value

Type: `ConnectApi.CdpQueryOutput`

queryProfileApi(dataModelName, id, ciName, searchKey, dimensions, measures, filters, fields, batchSize, offset, orderby)

Query a Profile data model object and a Calculated Insight object using filters and a search key.

API Version

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpQueryOutput queryProfileApi(String dataModelName, String id, String ciName, String searchKey, String dimensions, String measures, String filters, String fields, Integer batchSize, Integer offset, String orderby)
```

Parameters

dataModelName
Type: String

Name of the data model object, for example, `UnifiedIndividual__dlm`.

id
Type: String

Value of the primary or secondary key field, for example, `John`. If unspecified, defaults to the value of the primary key field.

ciName
Type: String

Name of the Calculated Insight object, for example, `IndividualChildrenCount__cio`.

searchKey
Type: String

If a field other than the primary key is used, name of the key field, for example, `FirstName__c`.

dimensions
Type: String

Comma-separated list of up to 10 dimensions, such as `GenderId__c`, to project. If unspecified, this parameter includes all of the available dimensions.

measures
Type: String

Comma-separated list of up to 5 measures, such as `TotalSales__c`, to project. If unspecified, this parameter includes all of the available measures.

filters
Type: String

Comma-separated list of equality expressions within square brackets, for example, `[FirstName__c=DON]`.

fields
Type: String

Comma-separated list of up to 50 field names that you want to include in the result, for example, `Id__c,FirstName__c, GenderId__c,Occupation__c`. If unspecified, `Id__c` is returned.

batchSize
Type: Integer

Number of items to return. Values are from 1–4,999. If unspecified, the default value is `100`.

offset
Type: Integer

Number of rows to skip before returning results. If unspecified, no rows are skipped.

orderby
Type: String

Sort order for the result set, such as `GenderId__c ASC,Occupation__c DESC`. If unspecified, items are returned in the order they are retrieved.

Return Value

Type: `ConnectApi.CdpQueryOutput`

`queryProfileApi(dataModelName, id, ciName, searchKey, dimensions, measures, filters, fields, batchSize, offset, orderby, timeGranularity)`

Query a Profile data model object and a Calculated Insight object using filters, a search key, and a time range.

API Version

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpQueryOutput queryProfileApi(String dataModelName, String id, String ciName, String searchKey, String dimensions, String measures, String filters, String fields, Integer batchSize, Integer offset, String orderby, String timeGranularity)
```

Parameters

dataModelName
Type: String

Name of the data model object, for example, `UnifiedIndividual__dlm`.

id
Type: String

Value of the primary or secondary key field, for example, `John`. If unspecified, defaults to the value of the primary key field.

ciName
Type: String

Name of the Calculated Insight object, for example, `IndividualChildrenCount__cio`.

searchKey
Type: String

If a field other than the primary key is used, name of the key field, for example, `FirstName__c`.

dimensions
Type: String

Comma-separated list of up to 10 dimensions, such as `GenderId__c`, to project. If unspecified, this parameter includes all of the available dimensions.

measures
Type: String

Comma-separated list of up to 5 measures, such as `TotalSales__c`, to project. If unspecified, this parameter includes all of the available measures.

filters
Type: String

Comma-separated list of equality expressions within square brackets, for example, `[FirstName__c=DON]`.

fields
Type: String

Comma-separated list of up to 50 field names that you want to include in the result, for example, `Id__c,FirstName__c, GenderId__c,Occupation__c`. If unspecified, `Id__c` is returned.

batchSize
Type: Integer

Number of items to return. Values are from 1–4,999. If unspecified, the default value is `100`.

offset
Type: Integer

Number of rows to skip before returning results. If unspecified, no rows are skipped.

orderby
Type: String

Sort order for the result set, such as `GenderId__c ASC,Occupation__c DESC`. If unspecified, items are returned in the order they are retrieved.

timeGranularity
Type: String

Time range for the measures. Values are:

- `HOUR`
- `DAY`
- `MONTH`
- `QUARTER`
- `YEAR`

If unspecified, no time range is applied.

Return Value

Type: `ConnectApi.CdpQueryOutput`

`universalIdLookupBySourceId(entityName, dataSourceId, dataSourceObjectId, sourceRecordId)`

Look up objects by source ID.

API Version

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpQueryDataOutput universalIdLookupBySourceId(String
entityName, String dataSourceId, String dataSourceObjectId, String sourceRecordId)
```

Parameters

entityName
Type: String

Entity name.

dataSourceId
Type: String

Data source ID.

dataSourceObjectId
Type: String

Data source object ID.

sourceRecordId
Type: String

Source record ID.

Return Value

Type: `ConnectApi.CdpQueryDataOutput`

CdpSegment Class

Create, delete, get, publish, and update Data Cloud segments.

Namespace

ConnectApi

CdpSegment Methods

These methods are for `CdpSegment`. All methods are static.

IN THIS SECTION:

createSegment(input)
Create a segment.

createSegment(input, dataspace)
Create a segment in a dataspace.

deactivateSegmentByApiName(segmentApiName)
Deactivate a segment by API name.

deactivateSegmentById(segmentId)
Deactivate a segment by ID.

deleteSegment(segmentApiName)
Delete a segment.

executePublishAdhoc(segmentId)
Publish a segment.

getSegment(segmentApiName)
Get a segment.

getSegments()
Get segments.

getSegmentsPaginated(batchSize, offset, orderBy)
Get an ordered batch of segments.

getSegmentsPaginated(batchSize, offset, orderBy, dataspace)

Get paginated segments in a dataspace.

updateSegment(segmentApiName, input)

Update a segment.

createSegment(input)

Create a segment.

API Version

55.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpSegmentOutput createSegment(ConnectApi.CdpSegmentInput
input)
```

Parameters

input

Type: ConnectApi.CdpSegmentInput

A ConnectApi.CdpSegmentInput class.

Return Value

Type: ConnectApi.CdpSegmentOutput

createSegment(input, dataspace)

Create a segment in a dataspace.

API Version

58.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpSegmentOutput createSegment(ConnectApi.CdpSegmentInput
input, String dataspace)
```

Parameters

input
Type: ConnectApi.CdpSegmentInput

A ConnectApi.CdpSegmentInput class.

dataspace
Type: String

Name of the dataspace in which to perform the action. User must have permission to this dataspace. Optional when bulk-listing segments. If unspecified, default is the default dataspace.

Return Value

Type: ConnectApi.CdpSegmentOutput

deactivateSegmentByApiName(segmentApiName)

Deactivate a segment by API name.

API Version

59.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpSegmentActionOutput deactivateSegmentByApiName(String segmentApiName)
```

Parameters

segmentApiName
Type: String

API name of the segment.

Return Value

Type: ConnectApi.CdpSegmentActionOutput

deactivateSegmentById(segmentId)

Deactivate a segment by ID.

API Version

59.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpSegmentActionOutput deactivateSegmentById(String segmentId)
```

Parameters

segmentId
Type: String

ID of the segment.

Return Value

Type: ConnectApi.CdpSegmentActionOutput

deleteSegment(segmentApiName)

Delete a segment.

API Version

56.0

Requires Chatter

No

Signature

```
public static Void deleteSegment(String segmentApiName)
```

Parameters

segmentApiName
Type: String

API name of the segment.

Return Value

Type: Void

executePublishAdhoc(segmentId)

Publish a segment.

API Version

56.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpSegmentActionOutput executePublishAdhoc(String segmentId)
```

Parameters

segmentId
Type: String

ID of the segment to publish.

Return Value

Type: ConnectApi.CdpSegmentActionOutput

getSegment(segmentApiName)

Get a segment.

API Version

56.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpSegmentContainerOutput getSegment(String segmentApiName)
```

Parameters

segmentApiName
Type: String

API name of the segment.

Return Value

Type: ConnectApi.CdpSegmentContainerOutput

getSegments()

Get segments.

API Version

55.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpSegmentContainerOutput getSegments()
```

Return Value

Type: ConnectApi.CdpSegmentContainerOutput

getSegmentsPaginated(batchSize, offset, orderBy)

Get an ordered batch of segments.

API Version

56.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpSegmentContainerOutput getSegmentsPaginated(Integer
batchSize, Integer offset, String orderBy)
```

Parameters

batchSize
Type: Integer

Number of items to return. Values are from 1–200. If unspecified, the default value is `20`.

offset
Type: Integer

Number of rows to skip before returning results. If unspecified, no rows are skipped.

orderBy
Type: String

Sort order for the result set, such as `Name ASC` or `MarketSegmentType DESC`. If unspecified, items are returned by ID in ascending order.

Return Value

Type: ConnectApi.CdpSegmentContainerOutput

getSegmentsPaginated(batchSize, offset, orderBy, dataspace)

Get paginated segments in a dataspace.

API Version

58.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpSegmentContainerOutput getSegmentsPaginated(Integer batchSize, Integer offset, String orderBy, String dataspace)
```

Parameters

batchSize
Type: Integer

Number of items to return. Values are from 1–200. If unspecified, the default value is `20`.

offset
Type: Integer

Number of rows to skip before returning results. If unspecified, no rows are skipped.

orderBy
Type: String

Sort order for the result set, such as `Name ASC` or `MarketSegmentType DESC`. If unspecified, items are returned by ID in ascending order.

dataspace
Type: String

Name of the dataspace in which to perform the action. User must have permission to this dataspace. Optional when bulk-listing segments. If unspecified, default is the `default` dataspace.

Return Value

Type: `ConnectApi.CdpSegmentContainerOutput`

updateSegment(segmentApiName, input)

Update a segment.

API Version

56.0

Requires Chatter

No

Signature

```
public static ConnectApi.CdpSegmentOutput updateSegment(String segmentApiName,
ConnectApi.CdpSegmentInput input)
```

Parameters

segmentApiName
Type: String

API name of the segment.

input
Type: ConnectApi.CdpSegmentInput

A ConnectApi.CdpSegmentInput class with the updates.

Return Value

Type: ConnectApi.CdpSegmentOutput

Chatter Class

Access information about followers and subscriptions for records.

Namespace

ConnectApi

Chatter Methods

These methods are for Chatter. All methods are static.

All methods in this class require Chatter and are subject to the per user, per namespace, per hour rate limit.

IN THIS SECTION:

deleteSubscription(communityId, subscriptionId)
Delete a subscription. Use this method to stop following a record, a user, or a file.

getFollowers(communityId, recordId)
Get the first page of followers for a record.

getFollowers(communityId, recordId, pageParam, pageSize)
Get a page of followers for a record.

getSubscription(communityId, subscriptionId)
Get information about a subscription.

submitDigestJob(period)
Submit a daily or weekly Chatter email digest job.

`deleteSubscription(communityId, subscriptionId)`

Delete a subscription. Use this method to stop following a record, a user, or a file.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static void deleteSubscription(String communityId, String subscriptionId)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`subscriptionId`
Type: String

The ID for a subscription.

Return Value

Type: Void

Usage

"Following" a user, group, or record is the same as "subscribing" to a user, group, or record. A "follower" is the user who followed the user, group, or record. A "subscription" is an object describing the relationship between the follower and the user, group, or record they followed.

To leave a group, call `deleteMember(communityId, membershipId)`.

Example

When you follow a user, the call to `ConnectApi.ChatterUsers.follow` returns a `ConnectApi.Subscription` object. To stop following the user, pass the `id` property of that object to this method.

```
ConnectApi.Chatter.deleteSubscription(null, '0E8RR0000004CnK0AU');
```

SEE ALSO:

Follow a Record

follow(communityId, userId, subjectId)

`getFollowers(communityId, recordId)`

Get the first page of followers for a record.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FollowerPage getFollowers(String communityId, String recordId)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`recordId`
Type: String

ID for a record or the keyword `me`.

Return Value

Type: `ConnectApi.FollowerPage`

Usage

"Following" a user, group, or record is the same as "subscribing" to a user, group, or record. A "follower" is the user who followed the user, group, or record. A "subscription" is an object describing the relationship between the follower and the user, group, or record they followed.

SEE ALSO:

Follow a Record

`getFollowers(communityId, recordId, pageParam, pageSize)`

Get a page of followers for a record.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FollowerPage getFollowers(String communityId, String recordId,
Integer pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

recordId
Type: String

ID for a record or the keyword `me`.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.FollowerPage`

Usage

"Following" a user, group, or record is the same as "subscribing" to a user, group, or record. A "follower" is the user who followed the user, group, or record. A "subscription" is an object describing the relationship between the follower and the user, group, or record they followed.

SEE ALSO:

Follow a Record

`getSubscription(communityId, subscriptionId)`

Get information about a subscription.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Subscription getSubscription(String communityId, String subscriptionId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subscriptionId
Type: String

The ID for a subscription.

Return Value

Type: `ConnectApi.Subscription`

Usage

"Following" a user, group, or record is the same as "subscribing" to a user, group, or record. A "follower" is the user who followed the user, group, or record. A "subscription" is an object describing the relationship between the follower and the user, group, or record they followed.

SEE ALSO:

Follow a Record

submitDigestJob(period)

Submit a daily or weekly Chatter email digest job.

API Version

37.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.DigestJobRepresentation submitDigestJob(ConnectApi.DigestPeriod period)
```

Parameters

period
Type: `ConnectApi.DigestPeriod`

Time period that's included in a Chatter email digest. Values are:

- `DailyDigest`—The email includes up to the 50 latest posts from the previous day.
- `WeeklyDigest`—The email includes up to the 50 latest posts from the previous week.

Return Value

Type: `ConnectApi.DigestJob`

Usage

The times when Chatter sends email digests are not configurable in the UI. To control when email digests are sent and to use this method, contact Salesforce to enable API-only Chatter Digests.

Warning: Enabling API-only Chatter Digests disables the scheduled digests for your org. You must call the API for your users to receive their digests.

We recommend scheduling digest jobs by implementing the Apex `Schedulable` interface with this method. To monitor your digest jobs from Setup, enter *`Background Jobs`* in the `Quick Find` box, then select **Background Jobs**.

Example

Schedule daily digests:

```
global class ExampleDigestJob1 implements Schedulable {
   global void execute(SchedulableContext context) {
      ConnectApi.Chatter.submitDigestJob(ConnectApi.DigestPeriod.DailyDigest);
   }
}
```

Schedule weekly digests:

```
global class ExampleDigestJob2 implements Schedulable {
   global void execute(SchedulableContext context) {
      ConnectApi.Chatter.submitDigestJob(ConnectApi.DigestPeriod.WeeklyDigest);
   }
}
```

SEE ALSO:

Apex Scheduler

ChatterFavorites Class

Chatter favorites give you easy access to topics, list views, and feed searches.

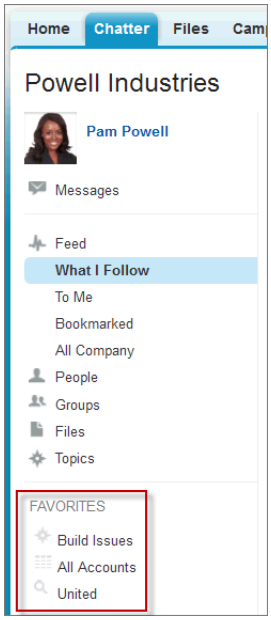
Namespace

ConnectApi

Usage

Use Connect in Apex to get and delete topics, list views, and feed searches that have been added as favorites. Add topics and feed searches as favorites, and update the last view date of a feed search or list view feed to the current system time.

In this image of Salesforce, “Build Issues” is a topic, “All Accounts” is a list view, and “United” is a feed search.



ChatterFavorites Methods

These methods are for `ChatterFavorites`. All methods are static.

All methods in this class require Chatter and are subject to the per user, per namespace, per hour rate limit.

IN THIS SECTION:

addFavorite(communityId, subjectId, searchText)
Add a feed search favorite for a user.

addRecordFavorite(communityId, subjectId, targetId)
Add a topic as a favorite.

deleteFavorite(communityId, subjectId, favoriteId)
Delete a favorite.

getFavorite(communityId, subjectId, favoriteId)
Get information about a favorite.

getFavorites(communityId, subjectId)
Get a list of favorites for a user.

getFeedElements(communityId, subjectId, favoriteId)
Get the first page of feed elements for a favorite.

getFeedElements(communityId, subjectId, favoriteId, pageParam, pageSize, sortParam)
Get a page of sorted feed elements for a favorite.

getFeedElements(communityId, subjectId, favoriteId, recentCommentCount, elementsPerBundle, pageParam, pageSize, sortParam)
Get a page of sorted feed elements for a favorite. Include no more than the specified number of comments per feed element.

updateFavorite(communityId, subjectId, favoriteId, updateLastViewDate)

Update the last view date of the saved search or list view feed to the current system time.

addFavorite(communityId, subjectId, searchText)

Add a feed search favorite for a user.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedFavorite addFavorite(String communityId, String subjectId,
String searchText)
```

Parameters

communityId

Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId

Type: String

ID of the context user or the alias `me`.

searchText

Type: String

Specify the text of the search to be saved as a favorite. This method can only create a feed search favorite, not a list view favorite or a topic.

Return Value

Type: `ConnectApi.FeedFavorite`

addRecordFavorite(communityId, subjectId, targetId)

Add a topic as a favorite.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedFavorite addRecordFavorite(String communityId, String subjectId, String targetId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

targetId
Type: String

The ID of the topic to add as a favorite.

Return Value

Type: `ConnectApi.FeedFavorite`

deleteFavorite(communityId, subjectId, favoriteId)

Delete a favorite.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static Void deleteFavorite(String communityId, String subjectId, String favoriteId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

favoriteId
Type: String

ID of a favorite.

Return Value

Type: Void

getFavorite(communityId, subjectId, favoriteId)

Get information about a favorite.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedFavorite getFavorite(String communityId, String subjectId,
String favoriteId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

favoriteId
Type: String

ID of a favorite.

Return Value

Type: `ConnectApi.FeedFavorite`

getFavorites(communityId, subjectId)

Get a list of favorites for a user.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedFavorites getFavorites(String communityId, String subjectId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

Return Value

Type: `ConnectApi.FeedFavorites`

getFeedElements(communityId, subjectId, favoriteId)

Get the first page of feed elements for a favorite.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElements(String communityId, String
subjectId, String favoriteId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

favoriteId
Type: String

ID of a favorite.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElements(communityId, subjectId, favoriteId, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElements(communityId, subjectId, favoriteId, pageParam, pageSize, sortParam)`

Get a page of sorted feed elements for a favorite.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElements(String communityId, String subjectId, String favoriteId, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`subjectId`
Type: String

ID of the context user or the alias `me`.

`favoriteId`
Type: String

ID of a favorite.

`pageParam`
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

`pageSize`
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

`sortParam`
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElements(communityId, subjectId, favoriteId, pageParam, pageSize, sortParam, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElements(communityId, subjectId, favoriteId, recentCommentCount, elementsPerBundle, pageParam, pageSize, sortParam)`

Get a page of sorted feed elements for a favorite. Include no more than the specified number of comments per feed element.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElements(String communityId, String
subjectId, String favoriteId, Integer recentCommentCount, Integer elementsPerBundle,
String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

favoriteId
Type: String

ID of a favorite.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElements(communityId, subjectId, favoriteId, recentCommentCount, elementsPerBundle, pageParam, pageSize, sortParam, result)

Apex Developer Guide: Testing ConnectApi Code

`updateFavorite(communityId, subjectId, favoriteId, updateLastViewDate)`

Update the last view date of the saved search or list view feed to the current system time.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedFavorite updateFavorite(String communityId, String
subjectId, String favoriteId, Boolean updateLastViewDate)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`subjectId`
Type: String

ID of the context user or the alias `me`.

`favoriteId`
Type: String

ID of a favorite.

`updateLastViewDate`
Type: Boolean

Specify whether to update the last view date of the specified favorite to the current system time (`true`) or not (`false`).

Return Value

Type: `ConnectApi.FeedFavorite`

ChatterFavorites Test Methods

These test methods are for `ChatterFavorites`. All methods are static.

For information about using these methods to test your `ConnectApi` code, see Testing ConnectApi Code.

IN THIS SECTION:

setTestGetFeedElements(communityId, subjectId, favoriteId, result)
Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElements` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetFeedElements(communityId, subjectId, favoriteId, pageParam, pageSize, sortParam, result)
Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElements` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetFeedElements(communityId, subjectId, favoriteId, recentCommentCount, elementsPerBundle, pageParam, pageSize, sortParam, result)
Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElements` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

`setTestGetFeedElements(communityId, subjectId, favoriteId, result)`

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElements` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

31.0

Signature

```
public static Void setTestGetFeedElements(String communityId, String subjectId, String favoriteId, ConnectApi.FeedElementPage result)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`subjectId`
Type: String

ID of the context user or the alias `me`.

`favoriteId`
Type: String

ID of a favorite.

result
Type: ConnectApi.FeedElementPage

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElements(communityId, subjectId, favoriteId)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedElements(communityId, subjectId, favoriteId, pageParam, pageSize, sortParam, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElements` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

31.0

Signature

```
public static Void setTestGetFeedElements(String communityId, String subjectId, String favoriteId, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

favoriteId
Type: String

ID of a favorite.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElements(communityId, subjectId, favoriteId, pageParam, pageSize, sortParam)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedElements(communityId, subjectId, favoriteId, recentCommentCount, elementsPerBundle, pageParam, pageSize, sortParam, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElements` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

31.0

Signature

```
public static Void setTestGetFeedElements(String communityId, String subjectId, String favoriteId, Integer recentCommentCount, Integer elementsPerBundle, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

favoriteId
Type: String

ID of a favorite.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElements(communityId, subjectId, favoriteId, recentCommentCount, elementsPerBundle, pageParam, pageSize, sortParam)

Apex Developer Guide: Testing ConnectApi Code

Retired ChatterFavorites Methods

These methods for `ChatterFavorites` are retired.

IN THIS SECTION:

getFeedItems(communityId, subjectId, favoriteId)
Get the first page of feed items for a favorite.

getFeedItems(communityId, subjectId, favoriteId, pageParam, pageSize, sortParam)
Get a page of sorted feed items for a favorite.

getFeedItems(communityId, subjectId, favoriteId, recentCommentCount, pageParam, pageSize, sortParam)
Get a page of sorted feed items for a favorite. Include no more than the specified number of comments per feed item.

setTestGetFeedItems(communityId, subjectId, favoriteId, result)
Register a `ConnectApi.FeedItemPage` object to be returned when `getFeedItems` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetFeedItems(communityId, subjectId, favoriteId, pageParam, pageSize, sortParam, result)
Register a `ConnectApi.FeedItemPage` object to be returned when `getFeedItems` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetFeedItems(communityId, subjectId, favoriteId, recentCommentCount, pageParam, pageSize, sortParam, result)
Register a `ConnectApi.FeedItemPage` object to be returned when `getFeedItems` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

`getFeedItems(communityId, subjectId, favoriteId)`

Get the first page of feed items for a favorite.

API Version

28.0–31.0

Important: In version 32.0 and later, use getFeedElements(communityId, subjectId, favoriteId).

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage getFeedItems(String communityId, String subjectId,
String favoriteId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

favoriteId
Type: String

ID of a favorite.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedItems(communityId, subjectId, favoriteId, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedItems(communityId, subjectId, favoriteId, pageParam, pageSize, sortParam)`

Get a page of sorted feed items for a favorite.

API Version

28.0–31.0

Important: In version 32.0 and later, use getFeedElements(communityId, subjectId, favoriteId, pageParam, pageSize, sortParam).

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage getFeedItems(String communityId, String subjectId,
String favoriteId, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder
sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

favoriteId
Type: String

ID of a favorite.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedItems(communityId, subjectId, favoriteId, pageParam, pageSize, sortParam, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedItems(communityId, subjectId, favoriteId, recentCommentCount, pageParam, pageSize, sortParam)`

Get a page of sorted feed items for a favorite. Include no more than the specified number of comments per feed item.

API Version

29.0–31.0

Important: In version 32.0 and later, use getFeedElements(communityId, subjectId, favoriteId, recentCommentCount, elementsPerBundle, pageParam, pageSize, sortParam).

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage getFeedItems(String communityId, String subjectId,
String favoriteId, Integer recentCommentCount, String pageParam, Integer pageSize,
FeedSortOrder sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

favoriteId
Type: String

ID of a favorite.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.

- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedItems(communityId, subjectId, favoriteId, recentCommentCount, pageParam, pageSize, sortParam, result)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetFeedItems(communityId, subjectId, favoriteId, result)`

Register a `ConnectApi.FeedItemPage` object to be returned when `getFeedItems` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

28.0–31.0

Signature

```
public static Void setTestGetFeedItems(String communityId, String subjectId, String favoriteId, ConnectApi.FeedItemPage result)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`subjectId`
Type: String

ID of the context user or the alias `me`.

`favoriteId`
Type: String

ID of a favorite.

`result`
Type: `ConnectApi.FeedItemPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedItems(communityId, subjectId, favoriteId)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedItems(communityId, subjectId, favoriteId, pageParam, pageSize, sortParam, result)

Register a `ConnectApi.FeedItemPage` object to be returned when `getFeedItems` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

28.0–31.0

Signature

```
public static Void setTestGetFeedItems(String communityId, String subjectId, String favoriteId, String pageParam, Integer pageSize, FeedSortOrder sortParam, ConnectApi.FeedItemPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

favoriteId
Type: String

ID of a favorite.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: ConnectApi.FeedSortOrder

Values are:

- CreatedDateAsc—Sorts by oldest creation date. This sort order is available only for DirectMessageModeration, Draft, Isolated, Moderation, and PendingReview feeds.
- CreatedDateDesc—Sorts by most recent creation date.
- LastModifiedDateDesc—Sorts by most recent activity.
- MostViewed—Sorts by most viewed content. This sort order is available only for Home feeds when the ConnectApi.FeedFilter is UnansweredQuestions.
- Relevance—Sorts by most relevant content. This sort order is available only for Company, Home, and Topics feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in null, the default value CreatedDateDesc is used.

result
Type: ConnectApi.FeedItemPage

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedItems(communityId, subjectId, favoriteId, pageParam, pageSize, sortParam)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedItems(communityId, subjectId, favoriteId, recentCommentCount, pageParam, pageSize, sortParam, result)

Register a ConnectApi.FeedItemPage object to be returned when getFeedItems is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

29.0–31.0

Signature

```
public static Void setTestGetFeedItems(String communityId, String subjectId, String favoriteId, Integer recentCommentCount, String pageParam, Integer pageSize, FeedSortOrder sortParam, ConnectApi.FeedItemPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, internal, or null.

subjectId
Type: String

ID of the context user or the alias `me`.

favoriteId
Type: String

ID of a favorite.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

result
Type: `ConnectApi.FeedItemPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedItems(communityId, subjectId, favoriteId, recentCommentCount, pageParam, pageSize, sortParam)

Apex Developer Guide: Testing ConnectApi Code

ChatterFeeds Class

Get, post, and delete feed elements, likes, comments, and bookmarks. You can also search feed elements, share feed elements, and vote on polls.

Namespace

ConnectApi

Usage

The Chatter feed is a container of feed elements. The abstract class `ConnectApi.FeedElement` is a parent class to the `ConnectApi.FeedItem` class, representing feed posts, and the `ConnectApi.GenericFeedElement` class, representing bundles and recommendations in the feed. For detailed information, see Working with Feeds and Feed Elements.

Important: Feed item methods aren't available in version 32.0. In version 32.0 and later, use feed element methods.

Message segments in a feed item are typed as `ConnectApi.MessageSegment`. Feed item capabilities are typed as `ConnectApi.FeedItemCapability`. Record fields are typed as `ConnectApi.AbstractRecordField`. These classes are all abstract and have several concrete subclasses. At runtime you can use `instanceof` to check the concrete types of these objects and then safely proceed with the corresponding downcast. When you downcast, you must have a default case that handles unknown subclasses.

Important: The composition of a feed can change between releases. Write your code to handle instances of unknown subclasses.

ChatterFeeds Methods

These methods are for `ChatterFeeds`. All methods are static.

All methods in this class require Chatter and are subject to the per user, per namespace, per hour rate limit.

IN THIS SECTION:

createStream(communityId, streamInput)
Create a Chatter feed stream.

deleteComment(communityId, commentId)
Delete a comment.

deleteFeedElement(communityId, feedElementId)
Delete a feed element.

deleteLike(communityId, likeId)
Delete a like on a comment or post.

deleteStream(communityId, streamId)
Delete a Chatter feed stream.

getComment(communityId, commentId)
Get a comment.

getCommentBatch(communityId, commentIds)
Get a list of comments.

getCommentInContext(communityId, commentId, pageSize)
Get a threaded comment in the context of its parent comments and post.

getCommentsForFeedElement(communityId, feedElementId)
Get comments for a feed element.

getCommentsForFeedElement(communityId, feedElementId, threadedCommentsCollapsed)
Get comments in a threaded style for a feed element.

getCommentsForFeedElement(communityId, feedElementId, pageParam, pageSize)
Get a page of comments for a feed element.

getCommentsForFeedElement(communityId, feedElementId, pageParam, pageSize, threadedCommentsCollapsed)
Get a page of comments in a threaded style for a feed element.

getCommentsForFeedElement(communityId, feedElementId, threadedCommentsCollapsed, sortParam)
Get sorted comments in a threaded style for a feed element.

getCommentsForFeedElement(communityId, feedElementId, pageParam, pageSize, threadedCommentsCollapsed, sortParam)
Get a page of sorted comments in a threaded style for a feed element.

getCommentsForFeedElement(communityId, feedElementId, sortParam)
Get sorted comments for a feed element.

getCommentsForFeedElement(communityId, feedElementId, sortParam, threadedCommentsCollapsed)
Get sorted comments in a threaded style for a feed element.

getExtensions(communityId, pageParam, pageSize)
Get extensions.

getFeed(communityId, feedType)
Get a feed.

getFeed(communityId, feedType, sortParam)
Get a sorted feed.

getFeed(communityId, feedType, subjectId)
Get a feed for a record or user.

getFeed(communityId, feedType, subjectId, sortParam)
Get a sorted feed for a record or user.

getFeedDirectory(String)
Get a list of all feeds available to the context user.

getFeedElement(communityId, feedElementId)
Get a feed element.

getFeedElement(communityId, feedElementId, commentSort)
Get a feed element with sorted comments.

getFeedElement(communityId, feedElementId, threadedCommentsCollapsed)
Get a feed element and its comments in a threaded style.

getFeedElement(communityId, feedElementId, threadedCommentsCollapsed, commentSort)
Get a feed element and its sorted comments in a threaded style.

getFeedElement(communityId, feedElementId, recentCommentCount, elementsPerBundle)
Get a feed element with the specified number of elements per bundle including no more than the specified number of comments per feed element.

getFeedElement(communityId, feedElementId, recentCommentCount, elementsPerBundle, threadedCommentsCollapsed)

Get a feed element with its comments in a threaded style with the specified number of elements per bundle and comments per feed element.

getFeedElement(communityId, feedElementId, recentCommentCount, elementsPerBundle, threadedCommentsCollapsed, commentSort)

Get a feed element with its sorted comments in a threaded style with the specified number of elements per bundle and comments per feed element.

getFeedElement(communityId, feedElementId, recentCommentCount, elementsPerBundle, commentSort)

Get a feed element with the specified number of elements per bundle including no more than the specified number of sorted comments per feed element.

getFeedElementBatch(communityId, feedElementIds)

Get a list of feed elements.

getFeedElementPoll(communityId, feedElementId)

Get the poll associated with a feed element.

getFeedElementsFromBundle(communityId, feedElementId)

Get feed elements from a bundle.

getFeedElementsFromBundle(communityId, feedElementId, pageParam, pageSize, elementsPerBundle, recentCommentCount)

Get a page of feed elements from a bundle. Specify the number of elements per bundle and include no more than the specified number of comments per feed element.

getFeedElementsFromFeed(communityId, feedType)

Get feed elements from the `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Moderation`, and `PendingReview` feeds.

getFeedElementsFromFeed(communityId, feedType, pageParam, pageSize, sortParam)

Get a page of sorted feed elements from the `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Moderation`, and `PendingReview` feeds.

getFeedElementsFromFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam)

Get a page of sorted feed elements from the `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Moderation`, and `PendingReview` feeds. Each feed element contains no more than the specified number of comments.

getFeedElementsFromFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, filter)

Get a page of sorted and filtered feed elements from the `Home` feed. Each feed element contains no more than the specified number of comments.

getFeedElementsFromFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, filter, threadedCommentsCollapsed)

Get a page of filtered and sorted feed elements with comments in a threaded style from the `Home` feed. Each feed element contains no more than the specified number of comments.

getFeedElementsFromFeed(communityId, feedType, subjectId)

Get feed elements from any feed other than `Company`, `DirectMessageModeration`, `DirectMessages`, `Filter`, `Home`, `Isolated`, `Landing`, `Moderation`, and `PendingReview` for a user or record.

getFeedElementsFromFeed(communityId, feedType, subjectId, pageParam, pageSize, sortParam)

Get a page of sorted feed elements from any feed other than `Company`, `DirectMessageModeration`, `DirectMessages`, `Filter`, `Home`, `Isolated`, `Landing`, `Moderation`, and `PendingReview`.

getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam)

Get a page of sorted feed elements from any feed other than `Company`, `DirectMessageModeration`, `DirectMessages`, `Filter`, `Home`, `Isolated`, `Landing`, `Moderation`, and `PendingReview`. Each feed element includes no more than the specified number of comments.

getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, showInternalOnly)

Get a page of sorted feed elements from a record feed. Each feed element includes no more than the specified number of comments. Specify whether to return feed elements posted by internal (non-Experience Cloud site) users only.

getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, filter)

Get a page of sorted and filtered feed elements from the `UserProfile` feed.

getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, filter, threadedCommentsCollapsed)

Get a page of feed elements with comments in a threaded style from the `UserProfile` feed.

getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, customFilter)

Get a page of sorted and filtered feed elements from the case feed.

getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly)

Get a page of sorted feed elements from a record feed. Specify the number of elements per bundle and include no more than the specified number of comments per feed element. Specify whether to return feed elements posted by internal (non-Experience Cloud site) users only.

getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, filter)

Get a page of sorted and filtered feed elements from a record feed. Specify the number of elements per bundle and include no more than the specified number of comments per feed element. Specify whether to return feed elements posted by internal (non-Experience Cloud site) users only.

getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, filter, threadedCommentsCollapsed)

Get a page of sorted and filtered feed elements with comments in a threaded style for a record feed. Specify the number of elements per bundle and include no more than the specified number of comments per feed element. Specify whether to return feed elements posted by internal (non-Experience Cloud site) users only.

getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, customFilter)

Get a page of sorted and filtered feed elements from a case feed.

getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, customFilter, threadedCommentsCollapsed)

Get a page of filtered and sorted feed elements with comments in a threaded style from a case feed.

getFeedElementsFromFilterFeed(communityId, subjectId, keyPrefix)

Get feed elements from a feed filtered by a key prefix for a user.

getFeedElementsFromFilterFeed(communityId, subjectId, keyPrefix, pageParam, pageSize, sortParam)

Get a page of sorted feed elements from a feed filtered by a key prefix for a user.

getFeedElementsFromFilterFeed(communityId, subjectId, keyPrefix, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam)

Get a page of sorted feed elements from a feed filtered by a key prefix for a user. Each feed element contains no more than the specified number of comments.

getFeedElementsFromFilterFeedUpdatedSince(communityId, subjectId, keyPrefix, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince)

Get a page of feed elements from a feed filtered by a key prefix for a user. Include only feed elements that have been updated since the time specified in the `updatedSince` parameter.

getFeedElementsUpdatedSince(communityId, feedType, recentCommentCount, density, pageParam, pageSize, updatedSince)

Get a page of feed elements from the `Company`, `DirectMessageModeration`, `Home`, and `Moderation` feeds. Include only feed elements that have been updated since the time specified in the `updatedSince` parameter. Each feed element contains no more than the specified number of comments.

getFeedElementsUpdatedSince(communityId, feedType, recentCommentCount, density, pageParam, pageSize, updatedSince, filter)

Get a page of filtered feed elements from the `Home` feed. Include only feed elements that have been updated since the time specified in the `updatedSince` parameter. Each feed element contains no more than the specified number of comments.

getFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, updatedSince)

Get a page of feed elements from the `Files`, `Groups`, `News`, `People`, and `Record` feeds. Include only feed elements that have been updated since the time specified in the `updatedSince` parameter. Each feed element contains no more than the specified number of comments.

getFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, updatedSince, showInternalOnly)

Get a page of feed elements from a record feed. Include only feed elements that have been updated since the time specified in the `updatedSince` parameter. Specify whether to return feed elements posted by internal (non-Experience Cloud site) users only.

getFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, filter)

Get a page of filtered feed elements from a `UserProfile` feed. Include only feed elements that have been updated since the time specified in the `updatedSince` parameter.

getFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, customFilter)

Get a page of filtered feed elements from a case feed. Include only feed elements that have been updated since the time specified in the `updatedSince` parameter.

getFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, showInternalOnly)

Get a page of feed elements from a record feed. Include only feed elements that have been updated since the time specified in the `updatedSince` parameter. Specify the maximum number of feed elements in a bundle and whether to return feed elements posted by internal (non-Experience Cloud site) users only.

getFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, showInternalOnly, filter)

Get a page of filtered feed elements from a record feed. Include only feed elements that have been updated since the time specified in the `updatedSince` parameter. Specify the maximum number of feed elements in a bundle and whether to return feed elements posted by internal (non-Experience Cloud site) users only.

getFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, showInternalOnly, customFilter)

Get a page of filtered feed elements from a case feed. Include only feed elements that have been updated since the time specified in the `updatedSince` parameter.

getFeedWithFeedElements(communityId, feedType, pageSize)

Get information about a feed and a page of feed elements from the feed.

getFeedWithFeedElements(communityId, feedType, pageSize, recentCommentCount)

Get a page of information about the feed and the feed elements with the specified number of comments per feed element from the feed.

getFilterFeed(communityId, subjectId, keyPrefix)

Get a feed filtered by a key prefix for a user.

getFilterFeed(communityId, subjectId, keyPrefix, sortParam)

Get a sorted feed filtered by a key prefix for a user.

getFilterFeedDirectory(communityId, subjectId)

Get a feed directory of filter feeds available to the context user.

getLike(communityId, likeId)

Get a like on a post or comment.

getLikesForComment(communityId, commentId)

Get likes for a comment.

getLikesForComment(communityId, commentId, pageParam, pageSize)

Get a page of likes for a comment.

getLikesForFeedElement(communityId, feedElementId)

Get likes for a feed element.

getLikesForFeedElement(communityId, feedElementId, pageParam, pageSize)

Get a page of likes for a feed element.

getLinkMetadata(communityId, urls)

Get link metadata for URLs.

getPinnedFeedElementsFromFeed(communityId, feedType, subjectId)

Get pinned feed elements from a group or topic feed.

getReadByForFeedElement(communityId, feedElementId)

Get information about who read a feed element and when.

getReadByForFeedElement(communityId, feedElementId, pageParam, pageSize)

Get a page of information about who read a feed element and when.

getRelatedPosts(communityId, feedElementId, filter, maxResults)

Get posts related to the context feed element.

getStream(communityId, streamId)

Get information about a Chatter feed stream.

getStream(communityId, streamId, globalScope)

Get information about a Chatter feed stream, regardless of Experience Cloud site.

getStreams(communityId)

Get the Chatter feed streams for the context user.

getStreams(communityId, sortParam)
Get and sort the Chatter feed streams for the context user.

getStreams(communityId, pageParam, pageSize)
Get a page of Chatter feed streams for the context user.

getStreams(communityId, pageParam, pageSize, sortParam)
Get a sorted page of Chatter feed streams for the context user.

getStreams(communityId, pageParam, pageSize, sortParam, globalScope)
Get a sorted page of Chatter feed streams from all Enterprise Cloud sites for the context user.

getSupportedEmojis()
Get supported emojis for the org.

getThreadsForFeedComment(communityId, commentId)
Get threaded comments for a comment.

getThreadsForFeedComment(communityId, commentId, pageParam, pageSize)
Get a page of threaded comments for a comment.

getThreadsForFeedComment(communityId, commentId, threadedCommentsCollapsed)
Access the comments capability for a comment.

getTopUnansweredQuestions(communityId) (Pilot)
Get top unanswered questions for the context user in aExperience Cloud site.

getTopUnansweredQuestions(communityId, filter) (Pilot)
Get filtered top unanswered questions for the context user in an Experience Cloud site.

getTopUnansweredQuestions(communityId, pageSize) (Pilot)
Get a page of top unanswered questions for the context user in an Experience Cloud site.

getTopUnansweredQuestions(communityId, filter, pageSize) (Pilot)
Get a page of filtered top unanswered questions for the context user in an Experience Cloud site.

getVotesForComment(communityId, commentId, vote)
Get the first page of users who upvoted or downvoted a comment.

getVotesForComment(communityId, commentId, vote, pageParam, pageSize)
Get a page of users who upvoted or downvoted a comment.

getVotesForFeedElement(communityId, feedElementId, vote)
Get the first page of users who upvoted or downvoted a feed element.

getVotesForFeedElement(communityId, feedElementId, vote, pageParam, pageSize)
Get a page of users who upvoted or downvoted a feed element.

isCommentEditableByMe(communityId, commentId)
Discover whether the context user can edit a comment.

isFeedElementEditableByMe(communityId, feedElementId)
Discover whether the context user can edit a feed element.

isModified(communityId, feedType, subjectId, since)
Discover whether a news feed has been updated or changed. Use this method to poll a news feed for updates.

likeComment(communityId, commentId)
Like a comment for the context user.

likeFeedElement(communityId, feedElementId)

Like a feed element.

postCommentToFeedElement(communityId, feedElementId, text)

Post a plain-text comment to a feed element.

postCommentToFeedElement(communityId, feedElementId, comment, feedElementFileUpload)

Post a rich-text comment to a feed element. Use this method to include mentions and to attach a file.

postFeedElement(communityId, subjectId, feedElementType, text)

Post a plain-text feed element.

postFeedElement(communityId, feedElement)

Post a rich-text feed element. Include mentions and hashtag topics, attach already uploaded files to a feed element, and associate action link groups with a feed element. You can also use this method to share a feed element and add a comment.

postFeedElementBatch(communityId, feedElements)

Post a list of feed elements.

publishDraftFeedElement(communityId, feedElementId, feedElement)

Publish a draft feed element.

searchFeedElements(communityId, q)

Get the first page of feed elements that match the search criteria.

searchFeedElements(communityId, q, sortParam)

Get the first page of sorted feed elements that match the search criteria.

searchFeedElements(communityId, q, threadedCommentsCollapsed)

Get the feed elements and comments that match the search criteria.

searchFeedElements(communityId, q, pageParam, pageSize)

Get a page of feed elements that match the search criteria.

searchFeedElements(communityId, q, pageParam, pageSize, sortParam)

Get a page of sorted feed elements that match the search criteria.

searchFeedElements(communityId, q, pageParam, pageSize, threadedCommentsCollapsed)

Get a page of feed elements with comments in a threaded style that match the search criteria.

searchFeedElements(communityId, q, recentCommentCount, pageParam, pageSize, sortParam)

Get a page of sorted feed elements that match the search criteria. Each feed element includes no more than the specified number of comments.

searchFeedElementsInFeed(communityId, feedType, q)

Get the feed elements from the `Company`, `DirectMessageModeration`, `Home`, `Isolated`, `Moderation`, and `PendingReview` feeds that match the search criteria.

searchFeedElementsInFeed(communityId, feedType, pageParam, pageSize, sortParam, q)

Get a page of sorted feed elements from the `Company`, `DirectMessageModeration`, `Home`, `Isolated`, `Moderation`, and `PendingReview` feeds that match the search criteria.

searchFeedElementsInFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, q)

Get a page of sorted feed elements from the `Company`, `DirectMessageModeration`, `Home`, `Isolated`, `Moderation`, and `PendingReview` feeds that match the search criteria. Each feed element includes no more than the specified number of comments.

searchFeedElementsInFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, q, filter)

Get a page of sorted and filtered feed elements from the `Home` feed that match the search criteria. Each feed element includes no more than the specified number of comments.

searchFeedElementsInFeed(communityId, feedType, subjectId, q)

Search up to 5,000 of the most recent feed elements in a feed for a subject ID that match the search string. Feed elements are returned in order of most recent activity.

searchFeedElementsInFeed(communityId, feedType, subjectId, pageParam, pageSize, sortParam, q)

Get a page of sorted feed elements from a feed for a record or user that match the search criteria.

searchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q)

Get a page of sorted feed elements from a feed that match the search criteria. Each feed element includes no more than the specified number of comments.

searchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, filter)

Get a page of sorted and filtered feed elements from a `UserProfile` feed that match the search criteria.

searchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, customFilter)

Get a page of sorted and filtered feed elements from a case feed that match the search criteria.

searchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, showInternalOnly)

Get a page of sorted feed elements from a feed for a record or user that match the search criteria. Each feed element includes no more than the specified number of comments. Specify whether to return feed elements posted by internal (non-Experience Cloud site) users only.

searchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, showInternalOnly, filter)

Get a page of sorted and filtered feed elements from a feed for a record or user that match the search criteria. Each feed element includes no more than the specified number of comments. Specify whether to return feed elements posted by internal (non-Experience Cloud site) users only.

searchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, showInternalOnly, customFilter)

Get a page of sorted and filtered feed elements from a case feed that match the search criteria.

searchFeedElementsInFilterFeed(communityId, subjectId, keyPrefix, q)

Get the feed elements from a feed filtered by a key prefix that match the search criteria.

searchFeedElementsInFilterFeed(communityId, subjectId, keyPrefix, pageParam, pageSize, sortParam, q)

Get a page of sorted feed elements from a feed filtered by a key prefix that match the search criteria.

searchFeedElementsInFilterFeed(communityId, subjectId, keyPrefix, recentCommentCount, density, pageParam, pageSize, sortParam, q)

Get a page of sorted feed elements from a feed filtered by a key prefix that match the search criteria. Each feed element includes no more than the specified number of comments.

searchStreams(communityId, q)

Search the Chatter feed streams for the context user.

searchStreams(communityId, q, sortParam)

Search and sort the Chatter feed streams for the context user.

searchStreams(communityId, q, pageParam, pageSize)
Search the Chatter feed streams for the context user and return a page of results.

searchStreams(communityId, q, pageParam, pageSize, sortParam)
Search the Chatter feed streams for the context user and return a sorted page of results.

searchStreams(communityId, q, pageParam, pageSize, sortParam, globalScope)
Search the Chatter feed streams from all Experience Cloud sites for the context user and return a sorted page of results.

setCommentIsVerified(communityId, commentId, isVerified)
Mark a comment as verified or unverified.

setCommentIsVerifiedByAnonymized(communityId, commentId, isVerified, isVerifiedByAnonymized)
Mark a comment as verified by an anonymous user.

setCommentVote(communityId, commentId, upDownVote)
Upvote or downvote a comment.

setFeedCommentStatus(communityId, commentId, status)
Set the status of a comment.

setFeedElementIsClosed(communityId, feedElementId, isClosed)
Set a feed element to closed.

setFeedElementVote(communityId, feedElementId, upDownVote)
Upvote or downvote a feed element.

setFeedEntityStatus(communityId, feedElementId, status)
Set the status of a feed post.

setIsMutedByMe(communityId, feedElementId, isMutedByMe)
Mute or unmute a feed element.

setIsReadByMe(communityId, feedElementId, readBy)
Mark a feed element as read for the context user using an input class.

setIsReadByMe(communityId, feedElementId, isReadByMe)
Mark a feed element as read for the context user.

updateComment(communityId, commentId, comment)
Edit a comment.

updateDirectMessage(communityId, feedElementId, directMessage)
Update the members of a direct message.

updateFeedElement(communityId, feedElementId, feedElement)
Edit a feed element.

updateFeedElementBookmarks(communityId, feedElementId, bookmarks)
Bookmark a feed element or remove a bookmark from a feed element using an input class.

updateFeedElementBookmarks(communityId, feedElementId, isBookmarkedByCurrentUser)
Bookmark a feed element or remove a bookmark from a feed element.

updateFeedElementReadByCapabilityBatch(communityId, feedElementIds, readBy)
Mark multiple feed elements as read by the context user at the same time using an input class.

updateFeedElementReadByCapabilityBatch(communityId, feedElementIds, isReadByMe)
Mark multiple feed elements as read by the context user at the same time.

updateLikeForComment(communityId, commentId, isLikedByCurrentUser)
Like or unlike a comment.

updateLikeForFeedElement(communityId, feedElementId, isLikedByCurrentUser)
Like or unlike a feed element.

updatePinnedFeedElements(communityId, feedType, subjectId, pin)
Pin or unpin feed elements to a group or topic feed.

updateStream(communityId, streamId, streamInput)
Update a Chatter feed stream.

voteOnFeedElementPoll(communityId, feedElementId, myChoiceId)
Vote on a poll or change your vote on a poll.

createStream(communityId, streamInput)

Create a Chatter feed stream.

API Version

39.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterStream createStream(String communityId,
ConnectApi.ChatterStreamInput streamInput)
```

Parameters

communityId
Type: String
ID for an Experience Cloud site, `internal`, or `null`.

streamInput
Type: `ConnectApi.ChatterStreamInput`
A `ConnectApi.ChatterStreamInput` body.

Return Value

Type: `ConnectApi.ChatterStream`

deleteComment(communityId, commentId)

Delete a comment.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static Void deleteComment(String communityId, String commentId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

commentId
Type: String

ID for a comment.

Return Value

Type: Void

deleteFeedElement(communityId, feedElementId)

Delete a feed element.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static deleteFeedElement(String communityId, String feedElementId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

Return Value

Type: Void

deleteLike(communityId, likeId)

Delete a like on a comment or post.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static Void deleteLike(String communityId, String likeId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

likeId
Type: String

ID for a like.

Return Value

Type: Void

deleteStream(communityId, streamId)

Delete a Chatter feed stream.

API Version

39.0

Requires Chatter

Yes

Signature

```
public static Void deleteStream(String communityId, String streamId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

streamId
Type: String

ID of the Chatter feed stream.

Return Value

Type: Void

getComment(communityId, commentId)

Get a comment.

API Version

28.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Comment getComment(String communityId, String commentId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

commentId
Type: String

ID for a comment.

Return Value

Type: `ConnectApi.Comment`

getCommentBatch(communityId, commentIds)

Get a list of comments.

API Version

42.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.BatchResult[] getCommentBatch(String communityId, List<String>
commentIds)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

commentIds
Type: List<String>

A list of up to 100 comment IDs.

Return Value

Type: `ConnectApi.BatchResult`[]

The `ConnectApi.BatchResult.getResult()` method returns a `ConnectApi.Comment` object and errors for comments that didn't load.

getCommentInContext(communityId, commentId, pageSize)

Get a threaded comment in the context of its parent comments and post.

API Version

44.0

Available to Guest Users

44.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElement getCommentInContext(String communityId, String
commentId, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

commentId
Type: String

ID of the comment.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you don't specify a value, the default size is 25.

Return Value

Type: `ConnectApi.FeedElement`

If the comment doesn't support the `comments` capability, the return value is `ConnectApi.NotFoundException`.

getCommentsForFeedElement(communityId, feedElementId)

Get comments for a feed element.

API Version

32.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.CommentPage getCommentsForFeedElement(String communityId,
String feedElementId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

Return Value

Type: `ConnectApi.CommentPage`

If the feed element doesn't support the `Comments` capability, the return value is `ConnectApi.NotFoundException`.

getCommentsForFeedElement(communityId, feedElementId, threadedCommentsCollapsed)

Get comments in a threaded style for a feed element.

API Version

44.0

Available to Guest Users

44.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.CommentPage getCommentsForFeedElement(String communityId,
String feedElementId, Boolean threadedCommentsCollapsed)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

threadedCommentsCollapsed
Type: Boolean

Specifies whether to return threaded comments in a collapsed style (`true`) or not (`false`). If you pass in `null`, the default is `false`.

Return Value

Type: `ConnectApi.CommentPage`

If the feed element doesn't support the `Comments` capability, the return value is `ConnectApi.NotFoundException`.

getCommentsForFeedElement(communityId, feedElementId, pageParam, pageSize)

Get a page of comments for a feed element.

API Version

32.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.CommentPage getCommentsForFeedElement(String communityId,
String feedElementId, String pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of comments per page. Valid values are from 1 through 100. If you pass `null`, the default size is 25.

Return Value

Type: `ConnectApi.CommentPage`

If the feed element doesn't support the `Comments` capability, the return value is `ConnectApi.NotFoundException`.

`getCommentsForFeedElement(communityId, feedElementId, pageParam, pageSize, threadedCommentsCollapsed)`

Get a page of comments in a threaded style for a feed element.

API Version

44.0

Available to Guest Users

44.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.CommentPage getCommentsForFeedElement(String communityId,
String feedElementId, String pageParam, Integer pageSize, Boolean
threadedCommentsCollapsed)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of comments per page. Valid values are from 1 through 100. If you pass `null`, the default size is 25.

threadedCommentsCollapsed
Type: Boolean

Specifies whether to return threaded comments in a collapsed style (`true`) or not (`false`). If you pass in `null`, the default is `false`.

Return Value

Type: `ConnectApi.CommentPage`

If the feed element doesn't support the `Comments` capability, the return value is `ConnectApi.NotFoundException`.

`getCommentsForFeedElement(communityId, feedElementId, threadedCommentsCollapsed, sortParam)`

Get sorted comments in a threaded style for a feed element.

API Version

44.0

Available to Guest Users

44.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.CommentsCapability getCommentsForFeedElement(String communityId, String feedElementId, Boolean threadedCommentsCollapsed, ConnectApi.FeedCommentSortOrder sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

threadedCommentsCollapsed
Type: Boolean

Specifies whether to return threaded comments in a collapsed style (`true`) or not (`false`). If you pass in `null`, the default is `false`.

sortParam
Type: `ConnectApi.FeedCommentSortOrder`

Order of comments. Values are:

- `CreatedDateLatestAsc`—Sorts by most recently created comments in ascending order.
- `CreatedDateOldestAsc`—Sorts by oldest comments in ascending order.
- `Relevance`—Sorts by most relevant content.

Sorting in descending order isn't supported.

Return Value

Type: `ConnectApi.CommentPage`

If the feed element doesn't support the `Comments` capability, the return value is `ConnectApi.NotFoundException`.

`getCommentsForFeedElement(communityId, feedElementId, pageParam, pageSize, threadedCommentsCollapsed, sortParam)`

Get a page of sorted comments in a threaded style for a feed element.

API Version

44.0

Available to Guest Users

44.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.CommentPage getCommentsForFeedElement(String communityId,
String feedElementId, String pageParam, Integer pageSize, Boolean
threadedCommentsCollapsed, ConnectApi.FeedCommentSortOrder sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of comments per page. Valid values are from 1 through 100. If you pass `null`, the default size is 25.

threadedCommentsCollapsed
Type: Boolean

Specifies whether to return threaded comments in a collapsed style (`true`) or not (`false`). If you pass in `null`, the default is `false`.

sortParam
Type: `ConnectApi.FeedCommentSortOrder`

Order of comments. Values are:

- `CreatedDateLatestAsc`—Sorts by most recently created comments in ascending order.
- `CreatedDateOldestAsc`—Sorts by oldest comments in ascending order.
- `Relevance`—Sorts by most relevant content.

Sorting in descending order isn't supported.

Return Value

Type: `ConnectApi.CommentPage`

If the feed element doesn't support the `Comments` capability, the return value is `ConnectApi.NotFoundException`.

`getCommentsForFeedElement(communityId, feedElementId, sortParam)`

Get sorted comments for a feed element.

API Version

41.0

Available to Guest Users

41.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.CommentsCapability getCommentsForFeedElement(String communityId,
String feedElementId, ConnectApi.FeedCommentSortOrder sortParam)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedElementId`
Type: String

ID of the feed element.

`sortParam`
Type: `ConnectApi.FeedCommentSortOrder`

Order of comments. Values are:

- `CreatedDateLatestAsc`—Sorts by most recently created comments in ascending order.
- `CreatedDateOldestAsc`—Sorts by oldest comments in ascending order.
- `Relevance`—Sorts by most relevant content.

Sorting in descending order isn't supported.

Return Value

Type: `ConnectApi.CommentsCapability`

If the feed element doesn't support the `Comments` capability, the return value is `ConnectApi.NotFoundException`.

`getCommentsForFeedElement(communityId, feedElementId, sortParam, threadedCommentsCollapsed)`

Get sorted comments in a threaded style for a feed element.

API Version

44.0

Available to Guest Users

44.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.CommentsCapability getCommentsForFeedElement(String communityId, String feedElementId, ConnectApi.FeedCommentSortOrder sortParam, Boolean threadedCommentsCollapsed)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedElementId`
Type: String

ID of the feed element.

`sortParam`
Type: `ConnectApi.FeedCommentSortOrder`

Order of comments. Values are:

- `CreatedDateLatestAsc`—Sorts by most recently created comments in ascending order.
- `CreatedDateOldestAsc`—Sorts by oldest comments in ascending order.
- `Relevance`—Sorts by most relevant content.

Sorting in descending order isn't supported.

`threadedCommentsCollapsed`
Type: Boolean

Specifies whether to return threaded comments in a collapsed style (`true`) or not (`false`). If you pass in `null`, the default is `false`.

Return Value

Type: `ConnectApi.CommentsCapability`

If the feed element doesn't support the `Comments` capability, the return value is `ConnectApi.NotFoundException`.

getExtensions(communityId, pageParam, pageSize)

Get extensions.

API Version

40.0

Available to Guest Users

40.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ExtensionDefinitions getExtensions(String communityId, String pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

pageParam
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. The default size is 15.

Return Value

Type: `ConnectApi.ExtensionDefinitions`

getFeed(communityId, feedType)

Get a feed.

API Version

28.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Feed getFeed(String communityId, ConnectApi.FeedType feedType)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

Return Value

Type: `ConnectApi.Feed`

`getFeed(communityId, feedType, sortParam)`

Get a sorted feed.

API Version

28.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Feed getFeed(String communityId, ConnectApi.FeedType feedType,
ConnectApi.FeedSortOrder sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

If *feedType* is `DirectMessages`, *sortParam* must be `LastModifiedDateDesc`.

Return Value

Type: `ConnectApi.Feed`

getFeed(communityId, feedType, subjectId)

Get a feed for a record or user.

API Version

28.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Feed getFeed(String communityId, ConnectApi.FeedType feedType,
String subjectId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessageModeration`, `DirectMessages`, `Filter`, `Home`, `Isolated`, `Landing`, `Moderation`, and `PendingReview`.

subjectId
Type: String

If *feedType* is `Record`, *subjectId* can be any record ID, including a group ID. If *feedType* is `Streams`, *subjectId* must be a stream ID. If *feedType* is `Topics`, *subjectId* must be a topic ID. If *feedType* is `UserProfile`, *subjectId* can be any user ID. If the *feedType* is any other value, *subjectId* must be the ID of the context user or the alias `me`.

Return Value

Type: `ConnectApi.Feed`

getFeed(communityId, feedType, subjectId, sortParam)

Get a sorted feed for a record or user.

API Version

28.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Feed getFeed(String communityId, ConnectApi.FeedType feedType,
String subjectId, ConnectApi.FeedSortOrder sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessageModeration`, `DirectMessages`, `Filter`, `Home`, `Isolated`, `Landing`, `Moderation`, and `PendingReview`.

`subjectId`
Type: String

If *`feedType`* is `Record`, *`subjectId`* can be any record ID, including a group ID. If *`feedType`* is `Streams`, *`subjectId`* must be a stream ID. If *`feedType`* is `Topics`, *`subjectId`* must be a topic ID. If *`feedType`* is `UserProfile`, *`subjectId`* can be any user ID. If the *`feedType`* is any other value, *`subjectId`* must be the ID of the context user or the alias `me`.

`sortParam`
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

Return Value

Type: `ConnectApi.Feed`

`getFeedDirectory(String)`

Get a list of all feeds available to the context user.

API Version

30.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedDirectory getFeedDirectory(String communityId)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

Return Value

Type: ConnectApi.FeedDirectory

getFeedElement(communityId, feedElementId)

Get a feed element.

API Version

31.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElement getFeedElement(String communityId, String feedElementId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

Return Value

Type: ConnectApi.FeedElement

getFeedElement(communityId, feedElementId, commentSort)

Get a feed element with sorted comments.

API Version

41.0

Available to Guest Users

41.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElement getFeedElement(String communityId, String
feedElementId, ConnectApi.FeedCommentSortOrder commentSort)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

commentSort
Type: `ConnectApi.FeedCommentSortOrder`

Order of comments.

- `CreatedDateLatestAsc`—Sorts by most recently created comments in ascending order.
- `CreatedDateOldestAsc`—Sorts by oldest comments in ascending order.
- `Relevance`—Sorts by most relevant content.

The default value is `CreatedDateLatestAsc`.

Sorting in descending order isn't supported.

Return Value

Type: `ConnectApi.FeedElement`

`getFeedElement(communityId, feedElementId, threadedCommentsCollapsed)`

Get a feed element and its comments in a threaded style.

API Version

44.0

Available to Guest Users

44.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElement getFeedElement(String communityId, String
feedElementId, Boolean threadedCommentsCollapsed)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

threadedCommentsCollapsed
Type: Boolean

Specifies whether to return threaded comments in a collapsed style (`true`) or not (`false`). If you pass in `null`, the default is `false`.

Return Value

Type: `ConnectApi.FeedElement`

`getFeedElement(communityId, feedElementId, threadedCommentsCollapsed, commentSort)`

Get a feed element and its sorted comments in a threaded style.

API Version

44.0

Available to Guest Users

44.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElement getFeedElement(String communityId, String
feedElementId, Boolean threadedCommentsCollapsed, ConnectApi.FeedCommentSortOrder
commentSort)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

threadedCommentsCollapsed
Type: Boolean

Specifies whether to return threaded comments in a collapsed style (`true`) or not (`false`). If you pass in `null`, the default is `false`.

commentSort
Type: `ConnectApi.FeedCommentSortOrder`

Order of comments.

- `CreatedDateLatestAsc`—Sorts by most recently created comments in ascending order.
- `CreatedDateOldestAsc`—Sorts by oldest comments in ascending order.
- `Relevance`—Sorts by most relevant content.

Sorting in descending order isn't supported.

Return Value

Type: `ConnectApi.FeedElement`

getFeedElement(communityId, feedElementId, recentCommentCount, elementsPerBundle)

Get a feed element with the specified number of elements per bundle including no more than the specified number of comments per feed element.

API Version

31.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElement getFeedElement(String communityId, String
feedElementId, Integer recentCommentCount, Integer elementsPerBundle)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

Return Value

Type: `ConnectApi.FeedElement`

getFeedElement(communityId, feedElementId, recentCommentCount, elementsPerBundle, threadedCommentsCollapsed)

Get a feed element with its comments in a threaded style with the specified number of elements per bundle and comments per feed element.

API Version

44.0

Available to Guest Users

44.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElement getFeedElement(String communityId, String feedElementId, Integer recentCommentCount, Integer elementsPerBundle, Boolean threadedCommentsCollapsed)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

threadedCommentsCollapsed
Type: Boolean

Specifies whether to return threaded comments in a collapsed style (`true`) or not (`false`). If you pass in `null`, the default is `false`.

Return Value

Type: `ConnectApi.FeedElement`

getFeedElement(communityId, feedElementId, recentCommentCount, elementsPerBundle, threadedCommentsCollapsed, commentSort)

Get a feed element with its sorted comments in a threaded style with the specified number of elements per bundle and comments per feed element.

API Version

44.0

Available to Guest Users

44.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElement getFeedElement(String communityId, String
feedElementId, Integer recentCommentCount, Integer elementsPerBundle, Boolean
threadedCommentsCollapsed, ConnectApi.FeedCommentSortOrder commentSort)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

threadedCommentsCollapsed
Type: Boolean

Specifies whether to return threaded comments in a collapsed style (`true`) or not (`false`). If you pass in `null`, the default is `false`.

commentSort
Type: `ConnectApi.FeedCommentSortOrder`

Order of comments.

- `CreatedDateLatestAsc`—Sorts by most recently created comments in ascending order.
- `CreatedDateOldestAsc`—Sorts by oldest comments in ascending order.
- `Relevance`—Sorts by most relevant content.

Sorting in descending order isn't supported.

Return Value

Type: `ConnectApi.FeedElement`

getFeedElement(communityId, feedElementId, recentCommentCount, elementsPerBundle, commentSort)

Get a feed element with the specified number of elements per bundle including no more than the specified number of sorted comments per feed element.

API Version

41.0

Available to Guest Users

41.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElement getFeedElement(String communityId, String
feedElementId, Integer recentCommentCount, Integer elementsPerBundle,
ConnectApi.FeedCommentSortOrder commentSort)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

commentSort
Type: `ConnectApi.FeedCommentSortOrder`

Order of comments.

- `CreatedDateLatestAsc`—Sorts by most recently created comments in ascending order.
- `CreatedDateOldestAsc`—Sorts by oldest comments in ascending order.
- `Relevance`—Sorts by most relevant content.

The default value is `CreatedDateLatestAsc`.

Sorting in descending order isn't supported.

Return Value

Type: `ConnectApi.FeedElement`

`getFeedElementBatch(communityId, feedElementIds)`

Get a list of feed elements.

API Version

31.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.BatchResult[] getFeedElementBatch(String communityId,
List<String> feedElementIds)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementIds
Type: List<String>

A list of up to 500 feed element IDs.

Return Value

Type: `ConnectApi.BatchResult`[]

The `ConnectApi.BatchResult.getResult()` method returns a `ConnectApi.FeedElement` object and errors for feed elements that didn't load.

getFeedElementPoll(communityId, feedElementId)

Get the poll associated with a feed element.

API Version

32.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.PollCapability getFeedElementPoll(String communityId, String
feedElementId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

Return Value

Type: `ConnectApi.PollCapability`

If the feed element doesn't support this capability, the return value is `ConnectApi.NotFoundException`.

Note: Triggers on FeedItem objects run before their attachment and capabilities information is saved, which means that `ConnectApi.FeedItem.attachment` information and `ConnectApi.FeedElement.capabilities` information may not be available in the trigger.

`getFeedElementsFromBundle(communityId, feedElementId)`

Get feed elements from a bundle.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsFromBundle(String communityId,
String feedElementId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

Return Value

Type: `ConnectApi.FeedElementPage`

`getFeedElementsFromBundle(communityId, feedElementId, pageParam, pageSize, elementsPerBundle, recentCommentCount)`

Get a page of feed elements from a bundle. Specify the number of elements per bundle and include no more than the specified number of comments per feed element.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsFromBundle(String communityId,
String feedElementId, String pageParam, Integer pageSize, Integer elementsPerBundle,
Integer recentCommentCount)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

pageParam
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

Return Value

Type: `ConnectApi.FeedElementPage`

getFeedElementsFromFeed(communityId, feedType)

Get feed elements from the `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Moderation`, and `PendingReview` feeds.

API Version

31.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsFromFeed(String communityId,
ConnectApi.FeedType feedType)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsFromFeed(communityId, feedType, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElementsFromFeed(communityId, feedType, pageParam, pageSize, sortParam)`

Get a page of sorted feed elements from the `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Moderation`, and `PendingReview` feeds.

API Version

31.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsFromFeed(String communityId,
ConnectApi.FeedType feedType, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

If *feedType* is `DirectMessages`, *sortParam* must be `LastModifiedDateDesc`.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsFromFeed(communityId, feedType, pageParam, pageSize, sortParam, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElementsFromFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam)`

Get a page of sorted feed elements from the `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Moderation`, and `PendingReview` feeds. Each feed element contains no more than the specified number of comments.

API Version

31.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsFromFeed(String communityId, ConnectApi.FeedType feedType, Integer recentCommentCount, ConnectApi.FeedDensity density, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedType`
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

`recentCommentCount`
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

`density`
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

If *feedType* is `DirectMessages`, *sortParam* must be `LastModifiedDateDesc`.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsFromFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElementsFromFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, filter)`

Get a page of sorted and filtered feed elements from the `Home` feed. Each feed element contains no more than the specified number of comments.

API Version

32.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsFromFeed(String communityId,
ConnectApi.FeedType feedType, Integer recentCommentCount, ConnectApi.FeedDensity density,
String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam,
ConnectApi.FeedFilter filter)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

The type of feed. The only valid value is `Home`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

When the *sortParam* is `MostViewed`, you must pass in `null` for the *pageParam*.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

When the *sortParam* is `MostViewed`, the *pageSize* must be a value from 1 to 25.

sortParam

Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

filter

Type: `ConnectApi.FeedFilter`

Specifies the feed filters.

- `AllQuestions`—Feed elements that are questions.
- `AuthoredBy`—Feed elements authored by the user profile owner. This value is valid only for the `UserProfile` feed.
- `CommunityScoped`—Feed elements that are scoped to Experience Cloud sites. Currently, these feed elements have a User or a Group parent record. However, other parent record types could be scoped to sites in the future. Feed elements that are always visible in all sites are filtered out. This value is valid only for the `UserProfile` feed.
- `QuestionsWithCandidateAnswers`—Feed elements that are questions that have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `QuestionsWithCandidateAnswersReviewedPublished`—Feed elements that are questions that have candidate answers that have been reviewed or published. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Read`—Feed elements that are older than 30 days or are marked as read for the context user. Includes existing feed elements when the context user joined the group. This value is valid only for the `Record` feed of a group.
- `SolvedQuestions`—Feed elements that are questions and that have a best answer.
- `UnansweredQuestions`—Feed elements that are questions and that don't have any answers.
- `UnansweredQuestionsWithCandidateAnswers`—Feed elements that are questions that don't have answers but have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Unread`—Feed elements that are created in the past 30 days and aren't marked as read for the context user. This value is valid only for the `Record` feed of a group.
- `UnsolvedQuestions`—Feed elements that are questions and that don't have a best answer.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsFromFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, filter, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElementsFromFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, filter, threadedCommentsCollapsed)`

Get a page of filtered and sorted feed elements with comments in a threaded style from the `Home` feed. Each feed element contains no more than the specified number of comments.

API Version

44.0

Available to Guest Users

44.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsFromFeed(String communityId,
ConnectApi.FeedType feedType, Integer recentCommentCount, ConnectApi.FeedDensity density,
String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam,
ConnectApi.FeedFilter filter, Boolean threadedCommentsCollapsed)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedType`
Type: `ConnectApi.FeedType`

The type of feed. The only valid value is `Home`.

`recentCommentCount`
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: ConnectApi.FeedDensity

Specify the amount of content in a feed.

- AllUpdates—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- FewerUpdates—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, currentPageToken or nextPageToken. If you pass in null, the first page is returned.

When the *sortParam* is MostViewed, you must pass in null for the *pageParam*.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in null, the default size is 25.

When the *sortParam* is MostViewed, the *pageSize* must be a value from 1 to 25.

sortParam
Type: ConnectApi.FeedSortOrder

Values are:

- CreatedDateAsc—Sorts by oldest creation date. This sort order is available only for DirectMessageModeration, Draft, Isolated, Moderation, and PendingReview feeds.
- CreatedDateDesc—Sorts by most recent creation date.
- LastModifiedDateDesc—Sorts by most recent activity.
- MostViewed—Sorts by most viewed content. This sort order is available only for Home feeds when the ConnectApi.FeedFilter is UnansweredQuestions.
- Relevance—Sorts by most relevant content. This sort order is available only for Company, Home, and Topics feeds.

If you pass in null, the default value CreatedDateDesc is used.

filter
Type: ConnectApi.FeedFilter

Specifies the feed filters.

- AllQuestions—Feed elements that are questions.
- AuthoredBy—Feed elements authored by the user profile owner. This value is valid only for the UserProfile feed.
- CommunityScoped—Feed elements that are scoped to Experience Cloud sites. Currently, these feed elements have a User or a Group parent record. However, other parent record types could be scoped to sites in the future. Feed elements that are always visible in all sites are filtered out. This value is valid only for the UserProfile feed.
- QuestionsWithCandidateAnswers—Feed elements that are questions that have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- QuestionsWithCandidateAnswersReviewedPublished—Feed elements that are questions that have candidate answers that have been reviewed or published. This value is valid only for users with the Access Einstein-Generated Answers permission.

- `Read`—Feed elements that are older than 30 days or are marked as read for the context user. Includes existing feed elements when the context user joined the group. This value is valid only for the `Record` feed of a group.
- `SolvedQuestions`—Feed elements that are questions and that have a best answer.
- `UnansweredQuestions`—Feed elements that are questions and that don't have any answers.
- `UnansweredQuestionsWithCandidateAnswers`—Feed elements that are questions that don't have answers but have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Unread`—Feed elements that are created in the past 30 days and aren't marked as read for the context user. This value is valid only for the `Record` feed of a group.
- `UnsolvedQuestions`—Feed elements that are questions and that don't have a best answer.

`threadedCommentsCollapsed`
Type: Boolean

Specifies whether to return threaded comments in a collapsed style (`true`) or not (`false`). If you pass in `null`, the default is `false`.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsFromFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, filter, threadedCommentsCollapsed, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElementsFromFeed(communityId, feedType, subjectId)`

Get feed elements from any feed other than `Company`, `DirectMessageModeration`, `DirectMessages`, `Filter`, `Home`, `Isolated`, `Landing`, `Moderation`, and `PendingReview` for a user or record.

API Version

31.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsFromFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessageModeration`, `DirectMessages`, `Filter`, `Home`, `Isolated`, `Landing`, `Moderation`, and `PendingReview`.

subjectId
Type: String

If *feedType* is `Record`, *subjectId* can be any record ID, including a group ID. If *feedType* is `Streams`, *subjectId* must be a stream ID. If *feedType* is `Topics`, *subjectId* must be a topic ID. If *feedType* is `UserProfile`, *subjectId* can be any user ID. If the *feedType* is any other value, *subjectId* must be the ID of the context user or the alias `me`.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

Example for Getting the Context User's News Feed

```
ConnectApi.FeedElementPage fep =
ConnectApi.ChatterFeeds.getFeedElementsFromFeed(Network.getNetworkId(),
ConnectApi.FeedType.News, 'me');
```

Example for Getting Another User's Profile Feed

```
ConnectApi.FeedElementPage fep =
ConnectApi.ChatterFeeds.getFeedElementsFromFeed(Network.getNetworkId(),
ConnectApi.FeedType.UserProfile, '005R0000000HwMA');
```

Example for Getting Another User's Record Feed

```
ConnectApi.FeedElementPage fep =
ConnectApi.ChatterFeeds.getFeedElementsFromFeed(Network.getNetworkId(),
ConnectApi.FeedType.Record, '005R0000000HwMA');
```

SEE ALSO:

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, result)

Apex Developer Guide: Testing ConnectApi Code

getFeedElementsFromFeed(communityId, feedType, subjectId, pageParam, pageSize, sortParam)

Get a page of sorted feed elements from any feed other than `Company`, `DirectMessageModeration`, `DirectMessages`, `Filter`, `Home`, `Isolated`, `Landing`, `Moderation`, and `PendingReview`.

API Version

31.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsFromFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedType`
Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessageModeration`, `DirectMessages`, `Filter`, `Home`, `Isolated`, `Landing`, `Moderation`, and `PendingReview`.

`subjectId`
Type: String

If *`feedType`* is `Record`, *`subjectId`* can be any record ID, including a group ID. If *`feedType`* is `Streams`, *`subjectId`* must be a stream ID. If *`feedType`* is `Topics`, *`subjectId`* must be a topic ID. If *`feedType`* is `UserProfile`, *`subjectId`* can be any user ID. If the *`feedType`* is any other value, *`subjectId`* must be the ID of the context user or the alias `me`.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

The number of feed elements per page.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, pageParam, pageSize, sortParam, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam)`

Get a page of sorted feed elements from any feed other than `Company`, `DirectMessageModeration`, `DirectMessages`, `Filter`, `Home`, `Isolated`, `Landing`, `Moderation`, and `PendingReview`. Each feed element includes no more than the specified number of comments.

API Version

31.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsFromFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessageModeration`, `DirectMessages`, `Filter`, `Home`, `Isolated`, `Landing`, `Moderation`, and `PendingReview`.

subjectId
Type: String

If *feedType* is `Record`, *subjectId* can be any record ID, including a group ID. If *feedType* is `Streams`, *subjectId* must be a stream ID. If *feedType* is `Topics`, *subjectId* must be a topic ID. If *feedType* is `UserProfile`, *subjectId* can be any user ID. If the *feedType* is any other value, *subjectId* must be the ID of the context user or the alias `me`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam

Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, showInternalOnly)`

Get a page of sorted feed elements from a record feed. Each feed element includes no more than the specified number of comments. Specify whether to return feed elements posted by internal (non-Experience Cloud site) users only.

API Version

31.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsFromFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, Boolean showInternalOnly)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

Any record ID, including a group ID.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.

- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

`showInternalOnly`
Type: Boolean

Specifies whether to show only feed items from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, showInternalOnly, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, filter)`

Get a page of sorted and filtered feed elements from the `UserProfile` feed.

API Version

35.0

Available to Guest Users

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsFromFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, ConnectApi.FeedFilter filter)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.UserProfile`.

subjectId
Type: String

ID of any user. To specify the context user, use the user ID or the alias `me`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

filter
Type: `ConnectApi.FeedFilter`

Value must be `ConnectApi.FeedFilter.CommunityScoped` or `ConnectApi.FeedFilter.AuthoredBy`.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

Example

This example gets only community-specific feed elements.

```
ConnectApi.FeedElementPage fep =
ConnectApi.ChatterFeeds.getFeedElementsFromFeed(Network.getNetworkId(),
ConnectApi.FeedType.UserProfile, 'me', 3, ConnectApi.FeedDensity.FewerUpdates, null, null,
 ConnectApi.FeedSortOrder.LastModifiedDateDesc, ConnectApi.FeedFilter.CommunityScoped);
```

SEE ALSO:

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, filter, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, filter, threadedCommentsCollapsed)`

Get a page of feed elements with comments in a threaded style from the `UserProfile` feed.

API Version

44.0

Available to Guest Users

44.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsFromFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, ConnectApi.FeedFilter filter, Boolean
threadedCommentsCollapsed)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.UserProfile`.

subjectId
Type: String

ID of any user. To specify the context user, use the user ID or the alias `me`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

filter
Type: `ConnectApi.FeedFilter`

Value must be `ConnectApi.FeedFilter.CommunityScoped` or `ConnectApi.FeedFilter.AuthoredBy`.

`threadedCommentsCollapsed`
Type: Boolean

Specifies whether to return threaded comments in a collapsed style (`true`) or not (`false`). If you pass in `null`, the default is `false`.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, filter, threadedCommentsCollapsed, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, customFilter)`

Get a page of sorted and filtered feed elements from the case feed.

API Version

40.0

Available to Guest Users

40.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsFromFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, String customFilter)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

The ID of a case.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

customFilter
Type: String

Custom filter that applies only to the case feed. See customFeedFilter in the *Metadata API Developer Guide* for supported values.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, customFilter, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly)`

Get a page of sorted feed elements from a record feed. Specify the number of elements per bundle and include no more than the specified number of comments per feed element. Specify whether to return feed elements posted by internal (non-Experience Cloud site) users only.

API Version

31.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsFromFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount, Integer
elementsPerBundle, ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, Boolean showInternalOnly)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedType`
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

Any record ID, including a group ID.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

showInternalOnly
Type: Boolean

Specifies whether to show only feed items from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, filter)`

Get a page of sorted and filtered feed elements from a record feed. Specify the number of elements per bundle and include no more than the specified number of comments per feed element. Specify whether to return feed elements posted by internal (non-Experience Cloud site) users only.

API Version

32.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsFromFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount, Integer
elementsPerBundle, ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, Boolean showInternalOnly, ConnectApi.FeedFilter
filter)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

Any record ID, including a group ID.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

showInternalOnly
Type: Boolean

Specifies whether to show only feed items from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

filter
Type: `ConnectApi.FeedFilter`

Specifies the feed filters.

- `AllQuestions`—Feed elements that are questions.
- `AuthoredBy`—Feed elements authored by the user profile owner. This value is valid only for the `UserProfile` feed.
- `CommunityScoped`—Feed elements that are scoped to Experience Cloud sites. Currently, these feed elements have a User or a Group parent record. However, other parent record types could be scoped to sites in the future. Feed elements that are always visible in all sites are filtered out. This value is valid only for the `UserProfile` feed.
- `QuestionsWithCandidateAnswers`—Feed elements that are questions that have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `QuestionsWithCandidateAnswersReviewedPublished`—Feed elements that are questions that have candidate answers that have been reviewed or published. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Read`—Feed elements that are older than 30 days or are marked as read for the context user. Includes existing feed elements when the context user joined the group. This value is valid only for the `Record` feed of a group.
- `SolvedQuestions`—Feed elements that are questions and that have a best answer.
- `UnansweredQuestions`—Feed elements that are questions and that don't have any answers.
- `UnansweredQuestionsWithCandidateAnswers`—Feed elements that are questions that don't have answers but have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Unread`—Feed elements that are created in the past 30 days and aren't marked as read for the context user. This value is valid only for the `Record` feed of a group.
- `UnsolvedQuestions`—Feed elements that are questions and that don't have a best answer.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, filter, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, filter, threadedCommentsCollapsed)`

Get a page of sorted and filtered feed elements with comments in a threaded style for a record feed. Specify the number of elements per bundle and include no more than the specified number of comments per feed element. Specify whether to return feed elements posted by internal (non-Experience Cloud site) users only.

API Version

44.0

Available to Guest Users

44.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsFromFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount, Integer
elementsPerBundle, ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, Boolean showInternalOnly, ConnectApi.FeedFilter
filter, Boolean threadedCommentsCollapsed)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

Any record ID, including a group ID.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

showInternalOnly
Type: Boolean

Specifies whether to show only feed items from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

filter
Type: `ConnectApi.FeedFilter`

Specifies the feed filters.

- `AllQuestions`—Feed elements that are questions.
- `AuthoredBy`—Feed elements authored by the user profile owner. This value is valid only for the `UserProfile` feed.
- `CommunityScoped`—Feed elements that are scoped to Experience Cloud sites. Currently, these feed elements have a User or a Group parent record. However, other parent record types could be scoped to sites in the future. Feed elements that are always visible in all sites are filtered out. This value is valid only for the `UserProfile` feed.
- `QuestionsWithCandidateAnswers`—Feed elements that are questions that have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `QuestionsWithCandidateAnswersReviewedPublished`—Feed elements that are questions that have candidate answers that have been reviewed or published. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Read`—Feed elements that are older than 30 days or are marked as read for the context user. Includes existing feed elements when the context user joined the group. This value is valid only for the `Record` feed of a group.
- `SolvedQuestions`—Feed elements that are questions and that have a best answer.
- `UnansweredQuestions`—Feed elements that are questions and that don't have any answers.
- `UnansweredQuestionsWithCandidateAnswers`—Feed elements that are questions that don't have answers but have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Unread`—Feed elements that are created in the past 30 days and aren't marked as read for the context user. This value is valid only for the `Record` feed of a group.

- `UnsolvedQuestions`—Feed elements that are questions and that don't have a best answer.

`threadedCommentsCollapsed`
Type: Boolean

Specifies whether to return threaded comments in a collapsed style (`true`) or not (`false`). If you pass in `null`, the default is `false`.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, filter, threadedCommentsCollapsed, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, customFilter)`

Get a page of sorted and filtered feed elements from a case feed.

API Version

40.0

Available to Guest Users

40.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsFromFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount, Integer
elementsPerBundle, ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, Boolean showInternalOnly, String customFilter)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

The ID of a case.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

`showInternalOnly`
Type: Boolean

Specifies whether to show only feed items from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

`customFilter`
Type: String

Custom filter that applies only to the case feed. See customFeedFilter in the *Metadata API Developer Guide* for supported values.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, customFilter, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, customFilter, threadedCommentsCollapsed)`

Get a page of filtered and sorted feed elements with comments in a threaded style from a case feed.

API Version

44.0

Available to Guest Users

44.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsFromFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount, Integer
elementsPerBundle, ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, Boolean showInternalOnly, String customFilter,
Boolean threadedCommentsCollapsed)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

The ID of a case.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

showInternalOnly
Type: Boolean

Specifies whether to show only feed items from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

customFilter
Type: String

Custom filter that applies only to the case feed. See customFeedFilter in the *Metadata API Developer Guide* for supported values.

threadedCommentsCollapsed
Type: Boolean

Specifies whether to return threaded comments in a collapsed style (`true`) or not (`false`). If you pass in `null`, the default is `false`.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, customFilter, threadedCommentsCollapsed, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElementsFromFilterFeed(communityId, subjectId, keyPrefix)`

Get feed elements from a feed filtered by a key prefix for a user.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsFromFilterFeed(String
communityId, String subjectId, String keyPrefix)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsFromFilterFeed(communityId, subjectId, keyPrefix, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElementsFromFilterFeed(communityId, subjectId, keyPrefix, pageParam, pageSize, sortParam)`

Get a page of sorted feed elements from a feed filtered by a key prefix for a user.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsFromFilterFeed(String
communityId, String subjectId, String keyPrefix, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsFromFilterFeed(communityId, subjectId, keyPrefix, pageParam, pageSize, sortParam, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElementsFromFilterFeed(communityId, subjectId, keyPrefix, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam)`

Get a page of sorted feed elements from a feed filtered by a key prefix for a user. Each feed element contains no more than the specified number of comments.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsFromFilterFeed(String communityId, String subjectId, String keyPrefix, Integer recentCommentCount, Integer elementsPerBundle, ConnectApi.FeedDensity density, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.

- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

`pageParam`
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

`pageSize`
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

`sortParam`
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsFromFilterFeed(communityId, subjectId, keyPrefix, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElementsFromFilterFeedUpdatedSince(communityId, subjectId, keyPrefix, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince)`

Get a page of feed elements from a feed filtered by a key prefix for a user. Include only feed elements that have been updated since the time specified in the *`updatedSince`* parameter.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsFromFilterFeedUpdatedSince(String communityId, String subjectId, String keyPrefix, Integer recentCommentCount, Integer elementsPerBundle, ConnectApi.FeedDensity density, String pageParam, Integer pageSize, String updatedSince)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

updatedSince
Type: String

Opaque token defining the modification timestamp of the feed and the sort order.

The *updatedSince* parameter doesn't return feed elements that are created in the same second as the call.

Return Value

Type: ConnectApi.FeedElementPage

Usage

To test code that uses this method, use the matching set test method (prefix the method name with setTest). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsFromFilterFeedUpdatedSince(communityId, subjectId, keyPrefix, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, result)

Apex Developer Guide: Testing ConnectApi Code

getFeedElementsUpdatedSince(communityId, feedType, recentCommentCount, density, pageParam, pageSize, updatedSince)

Get a page of feed elements from the Company, DirectMessageModeration, Home, and Moderation feeds. Include only feed elements that have been updated since the time specified in the *updatedSince* parameter. Each feed element contains no more than the specified number of comments.

API Version

31.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsUpdatedSince(String communityId,
ConnectApi.FeedType feedType, Integer recentCommentCount, ConnectApi.FeedDensity density,
String pageParam, Integer pageSize, String updatedSince)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `Home`, and `Moderation`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

updatedSince
Type: String

An opaque token containing information about the last modified date of the feed. Do not construct this token. Retrieve this token from the `updatesToken` property of the `ConnectApi.FeedElementPage` response body.

The *updatedSince* parameter doesn't return feed elements that are created in the same second as the call.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsUpdatedSince(communityId, feedType, recentCommentCount, density, pageParam, pageSize, updatedSince, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElementsUpdatedSince(communityId, feedType, recentCommentCount, density, pageParam, pageSize, updatedSince, filter)`

Get a page of filtered feed elements from the `Home` feed. Include only feed elements that have been updated since the time specified in the *`updatedSince`* parameter. Each feed element contains no more than the specified number of comments.

API Version

32.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsUpdatedSince(String communityId,
ConnectApi.FeedType feedType, Integer recentCommentCount, ConnectApi.FeedDensity density,
String pageParam, Integer pageSize, String updatedSince, ConnectApi.FeedFilter filter)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedType`
Type: `ConnectApi.FeedType`

The type of feed. The only valid value is `Home`.

`recentCommentCount`
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

`density`
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

`pageParam`
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

updatedSince
Type: String

An opaque token containing information about the last modified date of the feed. Do not construct this token. Retrieve this token from the `updatesToken` property of the `ConnectApi.FeedElementPage` response body.

The *updatedSince* parameter doesn't return feed elements that are created in the same second as the call.

filter
Type: `ConnectApi.FeedFilter`

Specifies the feed filters.

- `AllQuestions`—Feed elements that are questions.
- `AuthoredBy`—Feed elements authored by the user profile owner. This value is valid only for the `UserProfile` feed.
- `CommunityScoped`—Feed elements that are scoped to Experience Cloud sites. Currently, these feed elements have a User or a Group parent record. However, other parent record types could be scoped to sites in the future. Feed elements that are always visible in all sites are filtered out. This value is valid only for the `UserProfile` feed.
- `QuestionsWithCandidateAnswers`—Feed elements that are questions that have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `QuestionsWithCandidateAnswersReviewedPublished`—Feed elements that are questions that have candidate answers that have been reviewed or published. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Read`—Feed elements that are older than 30 days or are marked as read for the context user. Includes existing feed elements when the context user joined the group. This value is valid only for the `Record` feed of a group.
- `SolvedQuestions`—Feed elements that are questions and that have a best answer.
- `UnansweredQuestions`—Feed elements that are questions and that don't have any answers.
- `UnansweredQuestionsWithCandidateAnswers`—Feed elements that are questions that don't have answers but have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Unread`—Feed elements that are created in the past 30 days and aren't marked as read for the context user. This value is valid only for the `Record` feed of a group.
- `UnsolvedQuestions`—Feed elements that are questions and that don't have a best answer.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsUpdatedSince(communityId, feedType, recentCommentCount, density, pageParam, pageSize, updatedSince, filter, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, updatedSince)`

Get a page of feed elements from the `Files`, `Groups`, `News`, `People`, and `Record` feeds. Include only feed elements that have been updated since the time specified in the *`updatedSince`* parameter. Each feed element contains no more than the specified number of comments.

API Version

31.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsUpdatedSince(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize, String updatedSince)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedType`
Type: `ConnectApi.FeedType`

One of these values:

- `Files`
- `Groups`
- `News`
- `People`

- `Record`

`subjectId`
Type: String

If *`feedType`* is `ConnectApi.Record`, *`subjectId`* can be any record ID, including a group ID. Otherwise, it must be the context user or the alias `me`.

`recentCommentCount`
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

`density`
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

`pageParam`
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

`pageSize`
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

`updatedSince`
Type: String

An opaque token containing information about the last modified date of the feed. Do not construct this token. Retrieve this token from the `updatesToken` property of the `ConnectApi.FeedElementPage` response body.

The *`updatedSince`* parameter doesn't return feed elements that are created in the same second as the call.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, updatedSince, result)

Apex Developer Guide: Testing ConnectApi Code

getFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, updatedSince, showInternalOnly)

Get a page of feed elements from a record feed. Include only feed elements that have been updated since the time specified in the *updatedSince* parameter. Specify whether to return feed elements posted by internal (non-Experience Cloud site) users only.

API Version

31.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsUpdatedSince(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize, String updatedSince,
Boolean showInternalOnly)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

Any record ID, including a group ID.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.

- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

`pageParam`
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

`pageSize`
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

`updatedSince`
Type: String

An opaque token containing information about the last modified date of the feed. Do not construct this token. Retrieve this token from the `updatesToken` property of the `ConnectApi.FeedElementPage` response body.

The *`updatedSince`* parameter doesn't return feed elements that are created in the same second as the call.

`showInternalOnly`
Type: Boolean

Specifies whether to show only feed elements from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, updatedSince, showInternalOnly, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, filter)`

Get a page of filtered feed elements from a `UserProfile` feed. Include only feed elements that have been updated since the time specified in the *`updatedSince`* parameter.

API Version

35.0

Available to Guest Users

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsUpdatedSince(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount, Integer
elementsPerBundle, ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
String updatedSince, ConnectApi.FeedFilter filter)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.UserProfile`.

subjectId
Type: String

ID of any user. To specify the context user, use the user ID or the alias `me`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

`updatedSince`
Type: String

Opaque token defining the modification timestamp of the feed and the sort order.

The *`updatedSince`* parameter doesn't return feed elements that are created in the same second as the call.

`filter`
Type: `ConnectApi.FeedFilter`

Value must be `ConnectApi.FeedFilter.CommunityScoped`. Filters the feed to include only feed elements that are scoped to Experience Cloud sites. Feed elements that are always visible in all sites are filtered out.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, filter, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, customFilter)`

Get a page of filtered feed elements from a case feed. Include only feed elements that have been updated since the time specified in the *`updatedSince`* parameter.

API Version

40.0

Available to Guest Users

40.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsUpdatedSince(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount, Integer
```

```
elementsPerBundle, ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
String updatedSince, String customFilter)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

The ID of a case.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

updatedSince
Type: String

Opaque token defining the modification timestamp of the feed and the sort order.

The *updatedSince* parameter doesn't return feed elements that are created in the same second as the call.

customFilter
Type: String

Custom filter that applies only to the case feed. See customFeedFilter in the *Metadata API Developer Guide* for supported values.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, customFilter, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, showInternalOnly)`

Get a page of feed elements from a record feed. Include only feed elements that have been updated since the time specified in the *`updatedSince`* parameter. Specify the maximum number of feed elements in a bundle and whether to return feed elements posted by internal (non-Experience Cloud site) users only.

API Version

31.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsUpdatedSince(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount, Integer
elementsPerBundle, ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
String updatedSince, Boolean showInternalOnly)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedType`
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

Any record ID, including a group ID.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

updatedSince
Type: String

An opaque token containing information about the last modified date of the feed. Do not construct this token. Retrieve this token from the `updatesToken` property of the `ConnectApi.FeedElementPage` response body.

The *updatedSince* parameter doesn't return feed elements that are created in the same second as the call.

showInternalOnly
Type: Boolean

Specifies whether to show only feed elements from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, showInternalOnly, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, showInternalOnly, filter)`

Get a page of filtered feed elements from a record feed. Include only feed elements that have been updated since the time specified in the *`updatedSince`* parameter. Specify the maximum number of feed elements in a bundle and whether to return feed elements posted by internal (non-Experience Cloud site) users only.

API Version

32.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsUpdatedSince(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount, Integer
elementsPerBundle, ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
String updatedSince, Boolean showInternalOnly, ConnectApi.FeedFilter filter)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedType`
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

`subjectId`
Type: String

Any record ID, including a group ID.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

updatedSince
Type: String

An opaque token containing information about the last modified date of the feed. Do not construct this token. Retrieve this token from the `updatesToken` property of the `ConnectApi.FeedElementPage` response body.

The *updatedSince* parameter doesn't return feed elements that are created in the same second as the call.

showInternalOnly
Type: Boolean

Specifies whether to show only feed elements from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

filter
Type: `ConnectApi.FeedFilter`

Specifies the feed filters.

- `AllQuestions`—Feed elements that are questions.
- `AuthoredBy`—Feed elements authored by the user profile owner. This value is valid only for the `UserProfile` feed.
- `CommunityScoped`—Feed elements that are scoped to Experience Cloud sites. Currently, these feed elements have a User or a Group parent record. However, other parent record types could be scoped to sites in the future. Feed elements that are always visible in all sites are filtered out. This value is valid only for the `UserProfile` feed.
- `QuestionsWithCandidateAnswers`—Feed elements that are questions that have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.

- `QuestionsWithCandidateAnswersReviewedPublished`—Feed elements that are questions that have candidate answers that have been reviewed or published. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Read`—Feed elements that are older than 30 days or are marked as read for the context user. Includes existing feed elements when the context user joined the group. This value is valid only for the `Record` feed of a group.
- `SolvedQuestions`—Feed elements that are questions and that have a best answer.
- `UnansweredQuestions`—Feed elements that are questions and that don't have any answers.
- `UnansweredQuestionsWithCandidateAnswers`—Feed elements that are questions that don't have answers but have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Unread`—Feed elements that are created in the past 30 days and aren't marked as read for the context user. This value is valid only for the `Record` feed of a group.
- `UnsolvedQuestions`—Feed elements that are questions and that don't have a best answer.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, showInternalOnly, filter, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, showInternalOnly, customFilter)`

Get a page of filtered feed elements from a case feed. Include only feed elements that have been updated since the time specified in the *`updatedSince`* parameter.

API Version

40.0

Available to Guest Users

40.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getFeedElementsUpdatedSince(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount, Integer
elementsPerBundle, ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
String updatedSince, Boolean showInternalOnly, String customFilter)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

The ID of a case.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

updatedSince
Type: String

An opaque token containing information about the last modified date of the feed. Do not construct this token. Retrieve this token from the `updatesToken` property of the `ConnectApi.FeedElementPage` response body.

The *updatedSince* parameter doesn't return feed elements that are created in the same second as the call.

showInternalOnly
Type: Boolean

Specifies whether to show only feed elements from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

customFilter
Type: String

Custom filter that applies only to the case feed. See customFeedFilter in the *Metadata API Developer Guide* for supported values.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, showInternalOnly, customFilter, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedWithFeedElements(communityId, feedType, pageSize)`

Get information about a feed and a page of feed elements from the feed.

API Version

40.0

Available to Guest Users

40.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Feed getFeedWithFeedElements(String communityId,
ConnectApi.FeedType feedType, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: ConnectApi.FeedType

The type of feed. Valid values are `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Landing`, `Moderation`, and `PendingReview`. `Landing` is valid only when *communityId* is `internal`.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in 0, feed elements aren't returned with the feed.

Return Value

Type: ConnectApi.Feed

getFeedWithFeedElements(communityId, feedType, pageSize, recentCommentCount)

Get a page of information about the feed and the feed elements with the specified number of comments per feed element from the feed.

API Version

40.0

Available to Guest Users

40.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Feed getFeedWithFeedElements(String communityId,
ConnectApi.FeedType feedType, Integer pageSize, Integer recentCommentCount)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: ConnectApi.FeedType

The type of feed. Valid values are `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Landing`, `Moderation`, and `PendingReview`. `Landing` is valid only when *communityId* is `internal`.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in 0, feed elements aren't returned with the feed.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

Return Value

Type: `ConnectApi.Feed`

getFilterFeed(communityId, subjectId, keyPrefix)

Get a feed filtered by a key prefix for a user.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Feed getFilterFeed(String communityId, String subjectId, String keyPrefix)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A *key prefix* is the first three characters of a record ID, which specifies the object type.

Return Value

Type: `ConnectApi.Feed`

getFilterFeed(communityId, subjectId, keyPrefix, sortParam)

Get a sorted feed filtered by a key prefix for a user.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Feed getFilterFeed(String communityId, String subjectId, String keyPrefix, ConnectApi.FeedType sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

sortParam
Type: `ConnectApi.FeedType`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

Return Value

Type: `ConnectApi.Feed`

getFilterFeedDirectory(communityId, subjectId)

Get a feed directory of filter feeds available to the context user.

API Version

30.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedDirectory getFilterFeedDirectory(String communityId, String subjectId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

Return Value

Type: `ConnectApi.FeedDirectory`

This feed directory contains a list of filter feeds, which are the news feed filtered to include feed items whose parent is a specific entity type.

Usage

Call this method to return a directory containing a list of `ConnectApi.FeedDirectoryItem` objects. Each object contains a key prefix associated with an entity type the context user is following. A *key prefix* is the first three characters of a record ID, which specifies the object type.

Use key prefixes to filter the news feed so that it contains only feed items whose parent is the entity type associated with the key prefix. For example, get all the feed items whose parent is an Account. To get the feed items, pass a key prefix to the `ConnectApi.getFeedItemsFromFilterFeed` method.

The information about filter feeds never contains the key prefixes for users (`005`) or groups (`0F9`), but all users can use those key prefixes as filters.

The `ConnectApi.FeedDirectory.favorites` property is always empty when returned by a call to `getFilterFeedDirectory` because you can't filter a news feed by favorites.

Example

This example calls `getFilterFeedDirectory` and loops through the returned `FeedDirectoryItem` objects to find the key prefixes the context user can use to filter their news feed. It then copies each `keyPrefix` value to a list. Finally, it passes one of

the key prefixes from the list to the `getFeedItemsFromFilterFeed` method. The returned feed items include every feed item from the news feed whose parent is the entity type specified by the passed key prefix.

```
String communityId = null;
String subjectId = 'me';

// Create a list to populate with key prefixes.
List<String> keyPrefixList = new List<String>();

// Prepopulate with User and Group record types
// which are available to all users.
keyPrefixList.add('005');
keyPrefixList.add('0F9');

System.debug(keyPrefixList);

// Get the key prefixes available to the context user.
ConnectApi.FeedDirectory myFeedDirectory =
   ConnectApi.ChatterFeeds.getFilterFeedDirectory(null, 'me');

// Loop through the returned feeds list.
for (ConnectApi.FeedDirectoryItem i : myFeedDirectory.feeds) {

   // Grab each key prefix and add it to the list.
   keyPrefixList.add(i.keyPrefix);
}
System.debug(keyPrefixList);

// Use a key prefix from the list to filter the feed items in the news feed.
ConnectApi.FeedItemPage myFeedItemPage =
   ConnectApi.ChatterFeeds.getFeedItemsFromFilterFeed(communityId, subjectId,
keyPrefixList[0]);
System.debug(myFeedItemPage);
```

`getLike(communityId, likeId)`

Get a like on a post or comment.

API Version

28.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterLike getLike(String communityId, String likeId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

likeId
Type: String

ID for a like.

Return Value

Type: `ConnectApi.ChatterLike`

getLikesForComment(communityId, commentId)

Get likes for a comment.

API Version

28.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterLikePage getLikesForComment(String communityId, String commentId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

commentId
Type: String

ID for a comment.

Return Value

Type: `ConnectApi.ChatterLikePage`

getLikesForComment(communityId, commentId, pageParam, pageSize)

Get a page of likes for a comment.

API Version

28.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterLikePage getLikesForComment(String communityId, String commentId, Integer pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

commentId
Type: String

ID for a comment.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.ChatterLikePage`

getLikesForFeedElement(communityId, feedElementId)

Get likes for a feed element.

API Version

32.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterLikePage getLikesForFeedElement(String communityId,
String feedElementId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

Return Value

Type: `ConnectApi.ChatterLikePage`

If the feed element doesn't support the `ChatterLikes` capability, the return value is `ConnectApi.NotFoundException`.

getLikesForFeedElement(communityId, feedElementId, pageParam, pageSize)

Get a page of likes for a feed element.

API Version

32.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterLikePage getLikesForFeedElement(String communityId,
String feedElementId, Integer pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.ChatterLikePage`

If the feed element doesn't support the `ChatterLikes` capability, the return value is `ConnectApi.NotFoundException`.

`getLinkMetadata(communityId, urls)`

Get link metadata for URLs.

API Version

42.0

Available to Guest Users

42.0

Requires Chatter

No

Signature

```
public static ConnectApi.LinkMetadataCollection getLinkMetadata(String communityId,
String urls)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

urls
Type: String

Comma-separated list of URL-encoded URLs.

Return Value

Type: ConnectApi.LinkMetadataCollection

getPinnedFeedElementsFromFeed(communityId, feedType, subjectId)

Get pinned feed elements from a group or topic feed.

API Version

41.0

Available to Guest Users

41.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.PinnedFeedElements getPinnedFeedElementsFromFeed(String
communityId, ConnectApi.FeedType feedType, String subjectId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: ConnectApi.FeedType

The type of feed. Valid values are `Record` and `Topics`.

subjectId
Type: String

If *feedType* is `Record`, *subjectId* must be a group ID. If *feedType* is `Topics`, *subjectId* must be a topic ID.

Return Value

Type: ConnectApi.PinnedFeedElements

If the feed doesn't support this capability, the return value is ConnectApi.NotFoundException.

Usage

In the UI, pinned feed elements don't show all auxiliary information, such as comments, likes, interaction counts, or read by information. As a result, the `ConnectApi.PinnedFeedElements` output class doesn't include all the information for these capabilities.

`getReadByForFeedElement(communityId, feedElementId)`

Get information about who read a feed element and when.

API Version

40.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ReadByPage getReadByForFeedElement(String communityId, String feedElementId)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedElementId`
Type: String

ID of the feed element.

Return Value

Type: `ConnectApi.ReadByPage`

If the feed element doesn't support this capability, the return value is `ConnectApi.NotFoundException`.

`getReadByForFeedElement(communityId, feedElementId, pageParam, pageSize)`

Get a page of information about who read a feed element and when.

API Version

40.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ReadByPage getReadByForFeedElement(String communityId, String feedElementId, Integer pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

pageParam
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.ReadByPage`

If the feed element doesn't support this capability, the return value is `ConnectApi.NotFoundException`.

`getRelatedPosts(communityId, feedElementId, filter, maxResults)`

Get posts related to the context feed element.

API Version

37.0

Available to Guest Users

37.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.RelatedFeedPosts getRelatedPosts(String communityId, String feedElementId, ConnectApi.RelatedFeedPostType filter, Integer maxResults)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element. The feed element must be a question.

filter
Type: `ConnectApi.RelatedFeedPostType`

Specifies the type of related post. Values are:

- `Answered`—Related questions that have at least one answer.
- `BestAnswer`—Related questions that have a best answer.
- `Generic`—All types of related questions, including answered, with a best answer, and unanswered.
- `Unanswered`—Related questions that don't have answers.

`Generic` is the default value.

maxResults
Type: Integer

The maximum number of results to return. You can return up to 25 results; 5 is the default.

Return Value

Type: `ConnectApi.RelatedFeedPosts`

In version 37.0 and later, related feed posts are questions.

Each related feed post has a score indicating how closely it's related to the context feed post. We return related feed posts sorted by score, with the highest score first.

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

`getStream(communityId, streamId)`

Get information about a Chatter feed stream.

API Version

39.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterStream getStream(String communityId, String streamId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

streamId
Type: String

ID of the Chatter feed stream.

Return Value

Type: `ConnectApi.ChatterStream`

getStream(communityId, streamId, globalScope)

Get information about a Chatter feed stream, regardless of Experience Cloud site.

API Version

41.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterStream getStream(String communityId, String streamId,
Boolean globalScope)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

streamId
Type: String

ID of the Chatter feed stream.

globalScope
Type: Boolean

Specifies whether to get streams from all the context user's Experience Cloud sites, regardless of the *communityId* value.

Tip: If you know the *communityId* for the stream, we recommend setting *globalScope* to `false`.

Return Value

Type: ConnectApi.ChatterStream

getStreams(communityId)

Get the Chatter feed streams for the context user.

API Version

39.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterStreamPage getStreams(String communityId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

Return Value

Type: ConnectApi.ChatterStreamPage

getStreams(communityId, sortParam)

Get and sort the Chatter feed streams for the context user.

API Version

40.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterStreamPage getStreams(String communityId,
ConnectApi.SortOrder sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

sortParam
Type: `ConnectApi.SortOrder`

Specifies the sort order. Values are:

- `Ascending`—Items are in ascending alphabetical order (A-Z).
- `Descending`—Items are in descending alphabetical order (Z-A).
- `MostRecentlyViewed`—Items are in descending chronological order by view. This sort order is valid only for Chatter feed streams.

If not specified, default value is `Ascending`.

Return Value

Type: `ConnectApi.ChatterStreamPage`

getStreams(communityId, pageParam, pageSize)

Get a page of Chatter feed streams for the context user.

API Version

39.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterStreamPage getStreams(String communityId, Integer
pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 to 250. The default size is 25.

Return Value

Type: `ConnectApi.ChatterStreamPage`

getStreams(communityId, pageParam, pageSize, sortParam)

Get a sorted page of Chatter feed streams for the context user.

API Version

40.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterStreamPage getStreams(String communityId, Integer pageParam, Integer pageSize, ConnectApi.SortOrder sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 to 250. The default size is 25.

sortParam
Type: `ConnectApi.SortOrder`

Specifies the sort order. Values are:

- `Ascending`—Items are in ascending alphabetical order (A-Z).
- `Descending`—Items are in descending alphabetical order (Z-A).
- `MostRecentlyViewed`—Items are in descending chronological order by view. This sort order is valid only for Chatter feed streams.

If not specified, default value is `Ascending`.

Return Value

Type: `ConnectApi.ChatterStreamPage`

getStreams(communityId, pageParam, pageSize, sortParam, globalScope)

Get a sorted page of Chatter feed streams from all Enterprise Cloud sites for the context user.

API Version

41.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterStreamPage getStreams(String communityId, Integer
pageParam, Integer pageSize, ConnectApi.SortOrder sortParam, Boolean globalScope)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 to 250. The default size is 25.

sortParam
Type: `ConnectApi.SortOrder`

Specifies the sort order. Values are:

- `Ascending`—Items are in ascending alphabetical order (A-Z).
- `Descending`—Items are in descending alphabetical order (Z-A).
- `MostRecentlyViewed`—Items are in descending chronological order by view. This sort order is valid only for Chatter feed streams.

If not specified, default value is `Ascending`.

globalScope
Type: Boolean

Specifies whether to get streams from all the context user's Experience Cloud sites, regardless of the *communityId* value.

Tip: If you know the *communityId* for the streams, we recommend setting *globalScope* to `false`.

Return Value

Type: `ConnectApi.ChatterStreamPage`

`getSupportedEmojis()`

Get supported emojis for the org.

API Version

39.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.SupportedEmojis getSupportedEmojis()
```

Return Value

Type: `ConnectApi.SupportedEmojis`

Usage

To get the list, emojis must be enabled in your org.

`getThreadsForFeedComment(communityId, commentId)`

Get threaded comments for a comment.

API Version

44.0

Available to Guest Users

44.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.CommentPage getThreadsForFeedComment(String communityId, String commentId)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`commentId`
Type: String

ID of the comment.

Return Value

Type: `ConnectApi.CommentPage`

If the comment doesn't support the `comments` capability, the return value is `ConnectApi.NotFoundException`.

`getThreadsForFeedComment(communityId, commentId, pageParam, pageSize)`

Get a page of threaded comments for a comment.

API Version

44.0

Available to Guest Users

44.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.CommentPage getThreadsForFeedComment(String communityId, String commentId, String pageParam, Integer pageSize)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`commentId`
Type: String

ID of the comment.

`pageParam`
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

`pageSize`
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.CommentPage`

If the comment doesn't support the `comments` capability, the return value is `ConnectApi.NotFoundException`.

`getThreadsForFeedComment(communityId, commentId, threadedCommentsCollapsed)`

Access the comments capability for a comment.

API Version

44.0

Available to Guest Users

44.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.CommentsCapability getThreadsForFeedComment(String communityId,
String commentId, Boolean threadedCommentsCollapsed)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

commentId
Type: String

ID of the comment.

threadedCommentsCollapsed
Type: Boolean

Specifies whether to return threaded comments in a collapsed style (`true`) or not (`false`). If you pass in `null`, the default is `false`.

Return Value

Type: `ConnectApi.CommentsCapability`

If the comment doesn't support the `comments` capability, the return value is `ConnectApi.NotFoundException`.

`getTopUnansweredQuestions(communityId)` (Pilot)

Get top unanswered questions for the context user in aExperience Cloud site.

Note: We provided top-five unanswered questions to selected customers through a pilot program that required agreement to specific terms and conditions. This pilot program is closed and not accepting new participants.

API Version

42.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getTopUnansweredQuestions(String communityId)
```

Parameters

communityId
Type: String

ID of the Experience Cloud site.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetTopUnansweredQuestions(communityId, result) (Pilot)

Apex Developer Guide: Testing ConnectApi Code

`getTopUnansweredQuestions(communityId, filter)` (Pilot)

Get filtered top unanswered questions for the context user in an Experience Cloud site.

Note: We provided top-five unanswered questions to selected customers through a pilot program that required agreement to specific terms and conditions. This pilot program is closed and not accepting new participants.

API Version

42.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getTopUnansweredQuestions(String communityId,
ConnectApi.TopUnansweredQuestionsFilterType filter)
```

Parameters

communityId
Type: String

ID of the Experience Cloud site.

filter
Type: `ConnectApi.FeedFilter`

Specifies the filter for the feed. `UnansweredQuestionsWithCandidateAnswers` is the only valid value.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetTopUnansweredQuestions(communityId, filter, result) (Pilot)

Apex Developer Guide: Testing ConnectApi Code

`getTopUnansweredQuestions(communityId, pageSize)` (Pilot)

Get a page of top unanswered questions for the context user in an Experience Cloud site.

Note: We provided top-five unanswered questions to selected customers through a pilot program that required agreement to specific terms and conditions. This pilot program is closed and not accepting new participants.

API Version

42.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getTopUnansweredQuestions(String communityId,
Integer pageSize)
```

Parameters

communityId
Type: String

ID of the Experience Cloud site.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 0 through 10. If you pass in `null`, the default size is 5.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetTopUnansweredQuestions(communityId, pageSize, result) (Pilot)

Apex Developer Guide: Testing ConnectApi Code

`getTopUnansweredQuestions(communityId, filter, pageSize)` (Pilot)

Get a page of filtered top unanswered questions for the context user in an Experience Cloud site.

Note: We provided top-five unanswered questions to selected customers through a pilot program that required agreement to specific terms and conditions. This pilot program is closed and not accepting new participants.

API Version

42.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage getTopUnansweredQuestions(String communityId,
ConnectApi.FeedFilter filter, Integer pageSize)
```

Parameters

communityId

Type: String

ID of the Experience Cloud site.

filter

Type: `ConnectApi.FeedFilter`

Specifies the filter for the feed. `UnansweredQuestionsWithCandidateAnswers` is the only valid value.

pageSize

Type: Integer

Specifies the number of items per page. Valid values are from 0 through 10. If you pass in `null`, the default size is 5.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetTopUnansweredQuestions(communityId, filter, pageSize, result) (Pilot)

Apex Developer Guide: Testing ConnectApi Code

`getVotesForComment(communityId, commentId, vote)`

Get the first page of users who upvoted or downvoted a comment.

API Version

42.0

Available to Guest Users

42.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.VotePage getVotesForComment(String communityId, String
commentId, ConnectApi.UpDownVoteValue vote)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`commentId`
Type: String

ID of the comment.

`vote`
Type: `ConnectApi.UpDownVoteValue`

Specifies the value of the vote for the feed element. Values are:

- `Down`
- `Up`

You can't specify `None`.

Return Value

Type: `ConnectApi.VotePage`

If the comment doesn't support this capability, the return value is `ConnectApi.NotFoundException`.

`getVotesForComment(communityId, commentId, vote, pageParam, pageSize)`

Get a page of users who upvoted or downvoted a comment.

API Version

42.0

Available to Guest Users

42.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.VotePage getVotesForComment(String communityId, String
commentId, ConnectApi.UpDownVoteValue vote, Integer pageParam, Integer pageSize)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`commentId`
Type: String

ID of the comment.

`vote`
Type: `ConnectApi.UpDownVoteValue`

Specifies the value of the vote for the feed element. Values are:

- `Down`
- `Up`

You can't specify `None`.

`pageParam`
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.VotePage`

If the comment doesn't support this capability, the return value is `ConnectApi.NotFoundException`.

`getVotesForFeedElement(communityId, feedElementId, vote)`

Get the first page of users who upvoted or downvoted a feed element.

API Version

42.0

Available to Guest Users

42.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.VotePage getVotesForFeedElement(String communityId, String feedElementId, ConnectApi.UpDownVoteValue vote)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

vote
Type: `ConnectApi.UpDownVoteValue`

Specifies the value of the vote for the feed element. Values are:

- `Down`
- `Up`

You can't specify `None`.

Return Value

Type: `ConnectApi.VotePage`

If the feed element doesn't support this capability, the return value is `ConnectApi.NotFoundException`.

`getVotesForFeedElement(communityId, feedElementId, vote, pageParam, pageSize)`

Get a page of users who upvoted or downvoted a feed element.

API Version

42.0

Available to Guest Users

42.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.VotePage getVotesForFeedElement(String communityId, String
feedElementId, ConnectApi.UpDownVoteValue vote, Integer pageParam, Integer pageSize)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedElementId`
Type: String

ID of the feed element.

`vote`
Type: `ConnectApi.UpDownVoteValue`

Specifies the value of the vote for the feed element. Values are:

- `Down`
- `Up`

You can't specify `None`.

`pageParam`
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

`pageSize`
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: ConnectApi.VotePage

If the feed element doesn't support this capability, the return value is ConnectApi.NotFoundException.

isCommentEditableByMe(communityId, commentId)

Discover whether the context user can edit a comment.

API Version

34.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedEntityIsEditable isCommentEditableByMe(String communityId,
String commentId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

commentId
Type: String

ID of the comment.

Return Value

Type: ConnectApi.FeedEntityIsEditable

If the comment doesn't support the `edit` capability, the return value is ConnectApi.NotFoundException.

SEE ALSO:

Edit a Comment

isFeedElementEditableByMe(communityId, feedElementId)

Discover whether the context user can edit a feed element.

API Version

34.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedEntityIsEditable isFeedElementEditableByMe(String
communityId, String feedElementId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element. Feed items are the only type of feed element that can be edited.

Return Value

Type: `ConnectApi.FeedEntityIsEditable`

If the feed element doesn't support the `edit` capability, the return value is `ConnectApi.NotFoundException`.

SEE ALSO:

Edit a Feed Element

Edit a Question Title and Post

isModified(communityId, feedType, subjectId, since)

Discover whether a news feed has been updated or changed. Use this method to poll a news feed for updates.

Important: This feature is available through a Feed Polling pilot program. This pilot program is closed and not accepting new participants.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedModifiedInfo isModified(String communityId,
ConnectApi.FeedType feedType, String subjectId, String since)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Specifies the type of feed. The only supported type is `News`

subjectId
Type: String

ID of the context user or the alias `me`.

since
Type: String

An opaque token containing information about the last modified date of the feed. Retrieve this token from the `FeedElementPage.isModifiedToken` property.

Return Value

Type: `ConnectApi.FeedModifiedInfo`

`likeComment(communityId, commentId)`

Like a comment for the context user.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterLike likeComment(String communityId, String commentId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

commentId
Type: String

ID for a comment.

Return Value

Type: ConnectApi.ChatterLike

If the context user has already liked the comment, this method is a non-operation and returns the existing like.

likeFeedElement(communityId, feedElementId)

Like a feed element.

API Version

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterLike likeFeedElement(String communityId, String feedElementId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

Return Value

Type: ConnectApi.ChatterLike

If the feed element doesn't support the `ChatterLikes` capability, the return value is ConnectApi.NotFoundException.

Example

```
ConnectApi.ChatterLike chatterLike = ConnectApi.ChatterFeeds.likeFeedElement(null, '0D5D0000000KuGh');
```

postCommentToFeedElement(communityId, feedElementId, text)

Post a plain-text comment to a feed element.

API Version

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Comment postCommentToFeedElement(String communityId, String feedElementId, String text)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

text
Type: String

Text of the comment. A comment can contain up to 10,000 characters.

Return Value

Type: `ConnectApi.Comment`

If the feed element doesn't support the `Comments` capability, the return value is `ConnectApi.NotFoundException`.

Example

```
ConnectApi.Comment comment = ConnectApi.ChatterFeeds.postCommentToFeedElement(null, '0D5D0000000KuGh', 'I agree with the proposal.' );
```

postCommentToFeedElement(communityId, feedElementId, comment, feedElementFileUpload)

Post a rich-text comment to a feed element. Use this method to include mentions and to attach a file.

API Version

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Comment postCommentToFeedElement(String communityId, String
feedElementId, ConnectApi.CommentInput comment, ConnectApi.BinaryInput
feedElementFileUpload)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

comment
Type: `ConnectApi.CommentInput`

The comment body, including text and mentions, and capabilities, such as information about an attached file. A comment can contain up to 10,000 characters.

feedElementFileUpload
Type: `ConnectApi.BinaryInput`

A new binary file to attach to the comment, or `null`. If you specify a binary file, specify the title and description of the file in the *comment* parameter.

Return Value

Type: `ConnectApi.Comment`

If the feed element doesn't support the `Comments` capability, the return value is `ConnectApi.NotFoundException`.

Example for Posting a Comment with Mentions

You can post comments with mentions two ways. Use the ConnectApiHelper repository on GitHub to write a single line of code, or use this method example.

```
String communityId = null;
String feedElementId = '0D5D0000000KtW3';

ConnectApi.CommentInput commentInput = new ConnectApi.CommentInput();
ConnectApi.MentionSegmentInput mentionSegmentInput = new ConnectApi.MentionSegmentInput();
ConnectApi.MessageBodyInput messageBodyInput = new ConnectApi.MessageBodyInput();
ConnectApi.TextSegmentInput textSegmentInput = new ConnectApi.TextSegmentInput();

messageBodyInput.messageSegments = new List<ConnectApi.MessageSegmentInput>();

textSegmentInput.text = 'Does anyone in this group have an idea? ';
messageBodyInput.messageSegments.add(textSegmentInput);

mentionSegmentInput.id = '005D00000000oOT';
messageBodyInput.messageSegments.add(mentionSegmentInput);
```

```
commentInput.body = messageBodyInput;

ConnectApi.Comment commentRep = ConnectApi.ChatterFeeds.postCommentToFeedElement(communityId,
 feedElementId, commentInput, null);
```

Example for Posting a Comment with an Existing File

```
String feedElementId = '0D5D0000000KtW3';

ConnectApi.CommentInput commentInput = new ConnectApi.CommentInput();

ConnectApi.MessageBodyInput messageBodyInput = new ConnectApi.MessageBodyInput();
ConnectApi.TextSegmentInput textSegmentInput = new ConnectApi.TextSegmentInput();

textSegmentInput.text = 'I attached this file from Salesforce Files.';

messageBodyInput.messageSegments = new List<ConnectApi.MessageSegmentInput>();
messageBodyInput.messageSegments.add(textSegmentInput);
commentInput.body = messageBodyInput;

ConnectApi.CommentCapabilitiesInput commentCapabilitiesInput = new
ConnectApi.CommentCapabilitiesInput();
ConnectApi.ContentCapabilityInput contentCapabilityInput = new
ConnectApi.ContentCapabilityInput();

commentCapabilitiesInput.content = contentCapabilityInput;
contentCapabilityInput.contentDocumentId = '069D00000001rNJ';

commentInput.capabilities = commentCapabilitiesInput;

ConnectApi.Comment commentRep =
ConnectApi.ChatterFeeds.postCommentToFeedElement(Network.getNetworkId(), feedElementId,
commentInput, null);
```

Example for Posting a Comment with a New File

```
String feedElementId = '0D5D0000000KtW3';

ConnectApi.CommentInput commentInput = new ConnectApi.CommentInput();

ConnectApi.MessageBodyInput messageBodyInput = new ConnectApi.MessageBodyInput();
ConnectApi.TextSegmentInput textSegmentInput = new ConnectApi.TextSegmentInput();

textSegmentInput.text = 'Enjoy this new file.';

messageBodyInput.messageSegments = new List<ConnectApi.MessageSegmentInput>();
messageBodyInput.messageSegments.add(textSegmentInput);
commentInput.body = messageBodyInput;

ConnectApi.CommentCapabilitiesInput commentCapabilitiesInput = new
ConnectApi.CommentCapabilitiesInput();
ConnectApi.ContentCapabilityInput contentCapabilityInput = new
```

```
ConnectApi.ContentCapabilityInput();

commentCapabilitiesInput.content = contentCapabilityInput;
contentCapabilityInput.title = 'Title';

commentInput.capabilities = commentCapabilitiesInput;

String text = 'These are the contents of the new file.';
Blob myBlob = Blob.valueOf(text);
ConnectApi.BinaryInput binInput = new ConnectApi.BinaryInput(myBlob, 'text/plain',
'fileName');

ConnectApi.Comment commentRep =
ConnectApi.ChatterFeeds.postCommentToFeedElement(Network.getNetworkId(), feedElementId,
commentInput, binInput);
```

Example for Posting a Rich-Text Comment with an Inline Image

You can post rich-text comments with inline images and mentions two ways. Use the ConnectApiHelper repository on GitHub to write a single line of code, or use this method example. In this example, the image file is existing content that has already been uploaded to Salesforce.

```
String communityId = null;
String feedElementId = '0D5R0000000SBEr';
String imageId = '069R00000000IgQ';
String mentionedUserId = '005R0000000DiMz';

ConnectApi.CommentInput input = new ConnectApi.CommentInput();
ConnectApi.MessageBodyInput messageInput = new ConnectApi.MessageBodyInput();
ConnectApi.TextSegmentInput textSegment;
ConnectApi.MentionSegmentInput mentionSegment;
ConnectApi.MarkupBeginSegmentInput markupBeginSegment;
ConnectApi.MarkupEndSegmentInput markupEndSegment;
ConnectApi.InlineImageSegmentInput inlineImageSegment;

messageInput.messageSegments = new List<ConnectApi.MessageSegmentInput>();


markupBeginSegment = new ConnectApi.MarkupBeginSegmentInput();
markupBeginSegment.markupType = ConnectApi.MarkupType.Bold;
messageInput.messageSegments.add(markupBeginSegment);

textSegment = new ConnectApi.TextSegmentInput();
textSegment.text = 'Hello ';
messageInput.messageSegments.add(textSegment);

mentionSegment = new ConnectApi.MentionSegmentInput();
mentionSegment.id = mentionedUserId;
messageInput.messageSegments.add(mentionSegment);

textSegment = new ConnectApi.TextSegmentInput();
textSegment.text = '!';
messageInput.messageSegments.add(textSegment);
```

```
markupEndSegment = new ConnectApi.MarkupEndSegmentInput();
markupEndSegment.markupType = ConnectApi.MarkupType.Bold;
messageInput.messageSegments.add(markupEndSegment);

inlineImageSegment = new ConnectApi.InlineImageSegmentInput();
inlineImageSegment.altText = 'image one';
inlineImageSegment.fileId = imageId;
messageInput.messageSegments.add(inlineImageSegment);

input.body = messageInput;

ConnectApi.ChatterFeeds.postCommentToFeedElement(communityId, feedElementId, input, null);
```

Example for Posting a Rich-Text Comment with a Code Block

```
String communityId = null;
String feedElementId = '0D5R0000000SBEr';
String codeSnippet = '<html>\n\t<body>\n\t\tHello, world!\n\t</body>\n</html>';

ConnectApi.CommentInput input = new ConnectApi.CommentInput();
ConnectApi.MessageBodyInput messageInput = new ConnectApi.MessageBodyInput();
ConnectApi.TextSegmentInput textSegment;
ConnectApi.MarkupBeginSegmentInput markupBeginSegment;
ConnectApi.MarkupEndSegmentInput markupEndSegment;

messageInput.messageSegments = new List<ConnectApi.MessageSegmentInput>();

markupBeginSegment = new ConnectApi.MarkupBeginSegmentInput();
markupBeginSegment.markupType = ConnectApi.MarkupType.Code;
messageInput.messageSegments.add(markupBeginSegment);

textSegment = new ConnectApi.TextSegmentInput();
textSegment.text = codeSnippet;
messageInput.messageSegments.add(textSegment);

markupEndSegment = new ConnectApi.MarkupEndSegmentInput();
markupEndSegment.markupType = ConnectApi.MarkupType.Code;
messageInput.messageSegments.add(markupEndSegment);

input.body = messageInput;

ConnectApi.ChatterFeeds.postCommentToFeedElement(communityId, feedElementId, input, null);
```

postFeedElement(communityId, subjectId, feedElementType, text)

Post a plain-text feed element.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElement postFeedElement(String communityId, String
subjectId, ConnectApi.FeedElementType feedElementType, String text)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

The ID of the parent this feed element is being posted to. This value can be the ID of a user, group, or record, or the string `me` to indicate the context user.

feedElementType
Type: `ConnectApi.FeedElementType`

The only possible value is `FeedItem`.

text
Type: String

The text of the feed element. A feed element can contain up to 10,000 characters.

Return Value

Type: `ConnectApi.FeedElement`

Example

```
ConnectApi.FeedElement feedElement =
ConnectApi.ChatterFeeds.postFeedElement(Network.getNetworkId(), '0F9d0000000TreH',
ConnectApi.FeedElementType.FeedItem, 'On vacation this week.');
```

postFeedElement(communityId, feedElement)

Post a rich-text feed element. Include mentions and hashtag topics, attach already uploaded files to a feed element, and associate action link groups with a feed element. You can also use this method to share a feed element and add a comment.

API Version

36.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElement postFeedElement(String communityId,
ConnectApi.FeedElementInput feedElement)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElement
Type: `ConnectApi.FeedElementInput`

Specify rich text, including mentions. Optionally, specify a link, a poll, or up to 10 existing files.

Return Value

Type: `ConnectApi.FeedElement`

Example for Posting a Feed Element with a Mention

You can post feed elements with mentions two ways. Use the ConnectApiHelper repository on GitHub to write a single line of code, or use this method example.

```
ConnectApi.FeedItemInput feedItemInput = new ConnectApi.FeedItemInput();
ConnectApi.MentionSegmentInput mentionSegmentInput = new ConnectApi.MentionSegmentInput();
ConnectApi.MessageBodyInput messageBodyInput = new ConnectApi.MessageBodyInput();
ConnectApi.TextSegmentInput textSegmentInput = new ConnectApi.TextSegmentInput();

messageBodyInput.messageSegments = new List<ConnectApi.MessageSegmentInput>();

mentionSegmentInput.id = '005RR000000Dme9';
messageBodyInput.messageSegments.add(mentionSegmentInput);

textSegmentInput.text = 'Could you take a look?';
messageBodyInput.messageSegments.add(textSegmentInput);

feedItemInput.body = messageBodyInput;
feedItemInput.feedElementType = ConnectApi.FeedElementType.FeedItem;
feedItemInput.subjectId = '0F9RR0000004CPw';

ConnectApi.FeedElement feedElement =
ConnectApi.ChatterFeeds.postFeedElement(Network.getNetworkId(), feedItemInput);
```

Example for Posting a Feed Element with Existing Content

```
// Define the FeedItemInput object to pass to postFeedElement
ConnectApi.FeedItemInput feedItemInput = new ConnectApi.FeedItemInput();
feedItemInput.subjectId = 'me';

ConnectApi.TextSegmentInput textSegmentInput = new ConnectApi.TextSegmentInput();
textSegmentInput.text = 'Would you please review these docs?';
```

```
// The MessageBodyInput object holds the text in the post
ConnectApi.MessageBodyInput messageBodyInput = new ConnectApi.MessageBodyInput();
messageBodyInput.messageSegments = new List<ConnectApi.MessageSegmentInput>();
messageBodyInput.messageSegments.add(textSegmentInput);
feedItemInput.body = messageBodyInput;

// The FeedElementCapabilitiesInput object holds the capabilities of the feed item.
// For this feed item, we define a files capability to hold the file(s).

List<String> fileIds = new List<String>();
fileIds.add('069xx00000000QO');
fileIds.add('069xx00000000QT');
fileIds.add('069xx00000000Qn');
fileIds.add('069xx00000000Qi');
fileIds.add('069xx00000000Qd');

ConnectApi.FilesCapabilityInput filesInput = new ConnectApi.FilesCapabilityInput();
filesInput.items = new List<ConnectApi.FileIdInput>();

for (String fileId : fileIds) {
    ConnectApi.FileIdInput idInput = new ConnectApi.FileIdInput();
    idInput.id = fileId;
    filesInput.items.add(idInput);
}

ConnectApi.FeedElementCapabilitiesInput feedElementCapabilitiesInput = new
ConnectApi.FeedElementCapabilitiesInput();
feedElementCapabilitiesInput.files = filesInput;

feedItemInput.capabilities = feedElementCapabilitiesInput;

// Post the feed item.
ConnectApi.FeedElement feedElement =
ConnectApi.ChatterFeeds.postFeedElement(Network.getNetworkId(), feedItemInput);
```

Example for Posting a Rich-Text Feed Element with an Inline Image

You can post rich-text feed elements with inline images and mentions two ways. Use the ConnectApiHelper repository on GitHub to write a single line of code, or use this method example. In this example, the image file is existing content that has already been uploaded to Salesforce. The post also includes text and a mention.

```
String communityId = null;
String imageId = '069D00000001INA';
String mentionedUserId = '005D0000001QNpr';
String targetUserOrGroupOrRecordId  = '005D0000001Gif0';
ConnectApi.FeedItemInput input = new ConnectApi.FeedItemInput();
input.subjectId = targetUserOrGroupOrRecordId;
input.feedElementType = ConnectApi.FeedElementType.FeedItem;

ConnectApi.MessageBodyInput messageInput = new ConnectApi.MessageBodyInput();
ConnectApi.TextSegmentInput textSegment;
ConnectApi.MentionSegmentInput mentionSegment;
ConnectApi.MarkupBeginSegmentInput markupBeginSegment;
ConnectApi.MarkupEndSegmentInput markupEndSegment;
```

```
ConnectApi.InlineImageSegmentInput inlineImageSegment;

messageInput.messageSegments = new List<ConnectApi.MessageSegmentInput>();

markupBeginSegment = new ConnectApi.MarkupBeginSegmentInput();
markupBeginSegment.markupType = ConnectApi.MarkupType.Bold;
messageInput.messageSegments.add(markupBeginSegment);

textSegment = new ConnectApi.TextSegmentInput();
textSegment.text = 'Hello ';
messageInput.messageSegments.add(textSegment);

mentionSegment = new ConnectApi.MentionSegmentInput();
mentionSegment.id = mentionedUserId;
messageInput.messageSegments.add(mentionSegment);

textSegment = new ConnectApi.TextSegmentInput();
textSegment.text = '!';
messageInput.messageSegments.add(textSegment);

markupEndSegment = new ConnectApi.MarkupEndSegmentInput();
markupEndSegment.markupType = ConnectApi.MarkupType.Bold;
messageInput.messageSegments.add(markupEndSegment);

inlineImageSegment = new ConnectApi.InlineImageSegmentInput();
inlineImageSegment.altText = 'image one';
inlineImageSegment.fileId = imageId;
messageInput.messageSegments.add(inlineImageSegment);

input.body = messageInput;

ConnectApi.ChatterFeeds.postFeedElement(communityId, input);
```

Example for Posting a Rich-Text Feed Element with a Code Block

```
String communityId = null;
String targetUserOrGroupOrRecordId  = 'me';
String codeSnippet = '<html>\n\t<body>\n\t\tHello, world!\n\t</body>\n</html>';
ConnectApi.FeedItemInput input = new ConnectApi.FeedItemInput();
input.subjectId = targetUserOrGroupOrRecordId;
input.feedElementType = ConnectApi.FeedElementType.FeedItem;

ConnectApi.MessageBodyInput messageInput = new ConnectApi.MessageBodyInput();
ConnectApi.TextSegmentInput textSegment;
ConnectApi.MarkupBeginSegmentInput markupBeginSegment;
ConnectApi.MarkupEndSegmentInput markupEndSegment;

messageInput.messageSegments = new List<ConnectApi.MessageSegmentInput>();

markupBeginSegment = new ConnectApi.MarkupBeginSegmentInput();
markupBeginSegment.markupType = ConnectApi.MarkupType.Code;
messageInput.messageSegments.add(markupBeginSegment);

textSegment = new ConnectApi.TextSegmentInput();
```

```
textSegment.text = codeSnippet;
messageInput.messageSegments.add(textSegment);

markupEndSegment = new ConnectApi.MarkupEndSegmentInput();
markupEndSegment.markupType = ConnectApi.MarkupType.Code;
messageInput.messageSegments.add(markupEndSegment);

input.body = messageInput;

ConnectApi.ChatterFeeds.postFeedElement(communityId, input);
```

Example for Sharing a Feed Element (in Version 39.0 and Later)

```
// Define the FeedItemInput object to pass to postFeedElement
ConnectApi.FeedItemInput feedItemInput = new ConnectApi.FeedItemInput();
feedItemInput.subjectId = 'me';
ConnectApi.TextSegmentInput textSegmentInput = new ConnectApi.TextSegmentInput();
textSegmentInput.text = 'Look at this post I'm sharing.';
// The MessageBodyInput object holds the text in the post
ConnectApi.MessageBodyInput messageBodyInput = new ConnectApi.MessageBodyInput();
messageBodyInput.messageSegments = new List<ConnectApi.MessageSegmentInput>();
messageBodyInput.messageSegments.add(textSegmentInput);
feedItemInput.body = messageBodyInput;

ConnectApi.FeedEntityShareCapabilityInput shareInput = new
ConnectApi.FeedEntityShareCapabilityInput();
shareInput.feedEntityId = '0D5R0000000SEbc';
ConnectApi.FeedElementCapabilitiesInput feedElementCapabilitiesInput = new
ConnectApi.FeedElementCapabilitiesInput();
feedElementCapabilitiesInput.feedEntityShare = shareInput;
feedItemInput.capabilities = feedElementCapabilitiesInput;
// Post the feed item.
ConnectApi.FeedElement feedElement =
ConnectApi.ChatterFeeds.postFeedElement(Network.getNetworkId(), feedItemInput);
```

Example for Sending a Direct Message

```
// Define the FeedItemInput object to pass to postFeedElement
ConnectApi.FeedItemInput feedItemInput = new ConnectApi.FeedItemInput();

ConnectApi.TextSegmentInput textSegmentInput = new ConnectApi.TextSegmentInput();
textSegmentInput.text = 'Thanks for attending my presentation test run this morning. Send
 me any feedback.';

// The MessageBodyInput object holds the text in the post
ConnectApi.MessageBodyInput messageBodyInput = new ConnectApi.MessageBodyInput();
messageBodyInput.messageSegments = new List<ConnectApi.MessageSegmentInput>();
messageBodyInput.messageSegments.add(textSegmentInput);
feedItemInput.body = messageBodyInput;

// The FeedElementCapabilitiesInput object holds the capabilities of the feed item.
// For this feed item, we define a direct message capability to hold the member(s) and the
 subject.
```

```
List<String> memberIds = new List<String>();
memberIds.add('005B00000016OUQ');
memberIds.add('005B0000001rIN6');

ConnectApi.DirectMessageCapabilityInput dmInput = new
ConnectApi.DirectMessageCapabilityInput();
dmInput.subject = 'Thank you!';
dmInput.membersToAdd = memberIds;

ConnectApi.FeedElementCapabilitiesInput feedElementCapabilitiesInput = new
ConnectApi.FeedElementCapabilitiesInput();
feedElementCapabilitiesInput.directMessage = dmInput;

feedItemInput.capabilities = feedElementCapabilitiesInput;

// Post the feed item.
ConnectApi.FeedElement feedElement =
ConnectApi.ChatterFeeds.postFeedElement(Network.getNetworkId(), feedItemInput);
```

postFeedElementBatch(communityId, feedElements)

Post a list of feed elements.

API Version

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.BatchResult[] postFeedElementBatch(String communityId,
List<ConnectApi.BatchInput> feedElements)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElements
Type: List<`ConnectApi.BatchInput`>

The list can contain up to 500 `ConnectApi.BatchInput` objects. In the `ConnectApi.BatchInput` constructor, the input object must be a concrete instance of the abstract `ConnectApi.FeedElementInput` class.

Return Value

Type: `ConnectApi.BatchResult`[]

The `ConnectApi.BatchResult.getResult()` method returns a `ConnectApi.FeedElement` object.

The returned objects correspond to each of the input objects and are returned in the same order as the input objects.

The method call fails only if an error occurs that affects the entire operation (such as a parsing failure). If an individual object causes an error, the error is embedded within the `ConnectApi.BatchResult` list.

Usage

Use this method to post a list of feed elements efficiently. Create a list containing up to 500 objects and insert them all for the cost of one DML statement.

In version 36.0 and later, you can attach only one already uploaded file to each post. The `ConnectApi.BatchInput` has three constructors, but the `postFeedElementBatch` method supports only `ConnectApi.BatchInput(Object input)` in version 35.0 and later. This constructor doesn't support a binary input.

In version 32.0–35.0, this method supports both `ConnectApi.BatchInput(Object input)` and `ConnectApi.BatchInput(Object input, ConnectApi.BinaryInput binary)` constructors. The `ConnectApi.BatchInput(Object input, ConnectApi.BinaryInput binary)` constructor allows for a single binary input.

In each constructor, the input object must be an instance of `ConnectApi.FeedElementInput`.

Example

This trigger bulk posts to the feeds of newly inserted accounts.

```
trigger postFeedItemToAccount on Account (after insert) {
    Account[] accounts = Trigger.new;

    // Bulk post to the account feeds.

    List<ConnectApi.BatchInput> batchInputs = new List<ConnectApi.BatchInput>();

    for (Account a : accounts) {
        ConnectApi.FeedItemInput input = new ConnectApi.FeedItemInput();

        input.subjectId = a.id;

        ConnectApi.MessageBodyInput body = new ConnectApi.MessageBodyInput();
        body.messageSegments = new List<ConnectApi.MessageSegmentInput>();

        ConnectApi.TextSegmentInput textSegment = new ConnectApi.TextSegmentInput();
        textSegment.text = 'Let\'s win the ' + a.name + ' account.';

        body.messageSegments.add(textSegment);
        input.body = body;

        ConnectApi.BatchInput batchInput = new ConnectApi.BatchInput(input);
        batchInputs.add(batchInput);
    }

    ConnectApi.ChatterFeeds.postFeedElementBatch(Network.getNetworkId(), batchInputs);
}
```

`publishDraftFeedElement(communityId, feedElementId, feedElement)`

Publish a draft feed element.

API Version

44.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElement publishDraftFeedElement(String communityId, String feedElementId, ConnectApi.FeedElementInput feedElement)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element to publish.

feedElement
Type: `ConnectApi.FeedElementInput`

Use this optional parameter to edit your draft before publishing.

Return Value

Type: `ConnectApi.FeedElement`

The published feed element has a new ID.

`searchFeedElements(communityId, q)`

Get the first page of feed elements that match the search criteria.

API Version

31.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage searchFeedElements(String communityId, String
q)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedElements(communityId, q, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedElements(communityId, q, sortParam)`

Get the first page of sorted feed elements that match the search criteria.

API Version

31.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage searchFeedElements(String communityId, String
q, ConnectApi.FeedSortOrder sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedElements(communityId, q, sortParam, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedElements(communityId, q, threadedCommentsCollapsed)`

Get the feed elements and comments that match the search criteria.

API Version

44.0

Available to Guest Users

44.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage searchFeedElements(String communityId, String
q, Boolean threadedCommentsCollapsed)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

threadedCommentsCollapsed
Type: Boolean

Specifies whether to return threaded comments in a collapsed style (`true`) or not (`false`). If you pass in `null`, the default is `false`.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedElements(communityId, q, threadedCommentsCollapsed, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedElements(communityId, q, pageParam, pageSize)`

Get a page of feed elements that match the search criteria.

API Version

31.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage searchFeedElements(String communityId, String q, String pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedElements(communityId, q, pageParam, pageSize, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedElements(communityId, q, pageParam, pageSize, sortParam)`

Get a page of sorted feed elements that match the search criteria.

API Version

31.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage searchFeedElements(String communityId, String q, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

- setTestSearchFeedElements(communityId, q, pageParam, pageSize, sortParam, result)
- *Apex Developer Guide*: Testing ConnectApi Code

`searchFeedElements(communityId, q, pageParam, pageSize, threadedCommentsCollapsed)`

Get a page of feed elements with comments in a threaded style that match the search criteria.

API Version

44.0

Available to Guest Users

44.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage searchFeedElements(String communityId, String q, String pageParam, Integer pageSize, Boolean threadedCommentsCollapsed)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`q`
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

`pageParam`
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

`pageSize`
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

threadedCommentsCollapsed
Type: Boolean

Specifies whether to return threaded comments in a collapsed style (`true`) or not (`false`). If you pass in `null`, the default is `false`.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedElements(communityId, q, pageParam, pageSize, threadedCommentsCollapsed, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedElements(communityId, q, recentCommentCount, pageParam, pageSize, sortParam)`

Get a page of sorted feed elements that match the search criteria. Each feed element includes no more than the specified number of comments.

API Version

31.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage searchFeedElements(String communityId, String
q, Integer recentCommentCount, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedElements(communityId, q, recentCommentCount, pageParam, pageSize, sortParam, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedElementsInFeed(communityId, feedType, q)`

Get the feed elements from the `Company`, `DirectMessageModeration`, `Home`, `Isolated`, `Moderation`, and `PendingReview` feeds that match the search criteria.

API Version

31.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage searchFeedElementsInFeed(String communityId,
ConnectApi.FeedType feedType, String q)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedElementsInFeed(communityId, feedType, q, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedElementsInFeed(communityId, feedType, pageParam, pageSize, sortParam, q)`

Get a page of sorted feed elements from the `Company`, `DirectMessageModeration`, `Home`, `Isolated`, `Moderation`, and `PendingReview` feeds that match the search criteria.

API Version

31.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage searchFeedElementsInFeed(String communityId,
ConnectApi.FeedType feedType, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, String q)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedType`
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

`pageParam`
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

`pageSize`
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

`sortParam`
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.

- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

q

Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedElementsInFeed(communityId, feedType, pageParam, pageSize, sortParam, q, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedElementsInFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, q)`

Get a page of sorted feed elements from the `Company`, `DirectMessageModeration`, `Home`, `Isolated`, `Moderation`, and `PendingReview` feeds that match the search criteria. Each feed element includes no more than the specified number of comments.

API Version

31.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage searchFeedElementsInFeed(String communityId,
ConnectApi.FeedType feedType, Integer recentCommentCount, ConnectApi.FeedDensity density,
String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, String q)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedElementsInFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, q, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedElementsInFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, q, filter)`

Get a page of sorted and filtered feed elements from the `Home` feed that match the search criteria. Each feed element includes no more than the specified number of comments.

API Version

32.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage searchFeedElementsInFeed(String communityId,
ConnectApi.FeedType feedType, Integer recentCommentCount, ConnectApi.FeedDensity density,
String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, String q,
ConnectApi.FeedFilter filter)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedType`
Type: `ConnectApi.FeedType`

The type of feed. The only valid value is `Home`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

When the *sortParam* is `MostViewed`, you must pass in `null` for the *pageParam*.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

When the *sortParam* is `MostViewed`, the *pageSize* must be a value from 1 to 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

filter
Type: `ConnectApi.FeedFilter`

Specifies the feed filters.

- `AllQuestions`—Feed elements that are questions.
- `AuthoredBy`—Feed elements authored by the user profile owner. This value is valid only for the `UserProfile` feed.

- `CommunityScoped`—Feed elements that are scoped to Experience Cloud sites. Currently, these feed elements have a User or a Group parent record. However, other parent record types could be scoped to sites in the future. Feed elements that are always visible in all sites are filtered out. This value is valid only for the `UserProfile` feed.
- `QuestionsWithCandidateAnswers`—Feed elements that are questions that have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `QuestionsWithCandidateAnswersReviewedPublished`—Feed elements that are questions that have candidate answers that have been reviewed or published. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Read`—Feed elements that are older than 30 days or are marked as read for the context user. Includes existing feed elements when the context user joined the group. This value is valid only for the `Record` feed of a group.
- `SolvedQuestions`—Feed elements that are questions and that have a best answer.
- `UnansweredQuestions`—Feed elements that are questions and that don't have any answers.
- `UnansweredQuestionsWithCandidateAnswers`—Feed elements that are questions that don't have answers but have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Unread`—Feed elements that are created in the past 30 days and aren't marked as read for the context user. This value is valid only for the `Record` feed of a group.
- `UnsolvedQuestions`—Feed elements that are questions and that don't have a best answer.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedElementsInFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, q, filter, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedElementsInFeed(communityId, feedType, subjectId, q)`

Search up to 5,000 of the most recent feed elements in a feed for a subject ID that match the search string. Feed elements are returned in order of most recent activity.

API Version

31.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage searchFeedElementsInFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, String q)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessages`, `Filter`, `Landing`, `Streams`, and `Topics`.

subjectId
Type: String

If *`feedType`* is `Record`, *`subjectId`* can be any record ID, including a group ID. If *`feedType`* is `UserProfile`, *`subjectId`* can be any user ID. If the *`feedType`* is any other value, *`subjectId`* must be the ID of the context user or the alias `me`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedElementsInFeed(communityId, feedType, subjectId, q, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedElementsInFeed(communityId, feedType, subjectId, pageParam, pageSize, sortParam, q)`

Get a page of sorted feed elements from a feed for a record or user that match the search criteria.

API Version

31.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage searchFeedElementsInFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, String q)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessages`, `Filter`, `Landing`, `Streams`, and `Topics`.

subjectId
Type: String

If *`feedType`* is `Record`, *`subjectId`* can be any record ID, including a group ID. If *`feedType`* is `UserProfile`, *`subjectId`* can be any user ID. If the *`feedType`* is any other value, *`subjectId`* must be the ID of the context user or the alias `me`.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Order of feed items in the feed.

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.

- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

q

Type: String

Search term. Searches keywords in the user or group name. A minimum of one character is required. This parameter doesn't support wildcards. This parameter is required.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedElementsInFeed(communityId, feedType, subjectId, pageParam, pageSize, sortParam, q, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q)`

Get a page of sorted feed elements from a feed that match the search criteria. Each feed element includes no more than the specified number of comments.

API Version

31.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage searchFeedElementsInFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, String q)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessages`, `Filter`, `Landing`, `Streams`, and `Topics`.

subjectId
Type: String

If *feedType* is `Record`, *subjectId* can be any record ID, including a group ID. If *feedType* is `UserProfile`, *subjectId* can be any user ID. If the *feedType* is any other value, *subjectId* must be the ID of the context user or the alias `me`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

q

Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, filter)`

Get a page of sorted and filtered feed elements from a `UserProfile` feed that match the search criteria.

API Version

35.0

Available to Guest Users

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage searchFeedElementsInFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, String q, ConnectApi.FeedFilter filter)
```

Parameters

communityId

Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: ConnectApi.FeedType

Value must be `ConnectApi.FeedType.UserProfile`.

subjectId
Type: String

ID of any user. To specify the context user, use the user ID or the alias `me`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: ConnectApi.FeedDensity

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: ConnectApi.FeedSortOrder

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

q
Type: String

One or more keywords to search for in the feed elements visible to the context user. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.

filter
Type: ConnectApi.FeedFilter

Value must be `ConnectApi.FeedFilter.CommunityScoped`. Filters the feed to include only feed elements that are scoped to Experience Cloud sites. Feed elements that are always visible in all sites are filtered out.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, filter, result)

Apex Developer Guide: Testing ConnectApi Code

searchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, customFilter)

Get a page of sorted and filtered feed elements from a case feed that match the search criteria.

API Version

40.0

Available to Guest Users

40.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage searchFeedElementsInFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, String q, String customFilter)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

The ID of a case.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

q
Type: String

One or more keywords to search for in the feed elements visible to the context user. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.

customFilter
Type: String

Custom filter that applies only to the case feed. See customFeedFilter in the *Metadata API Developer Guide* for supported values.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, customFilter, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, showInternalOnly)`

Get a page of sorted feed elements from a feed for a record or user that match the search criteria. Each feed element includes no more than the specified number of comments. Specify whether to return feed elements posted by internal (non-Experience Cloud site) users only.

API Version

31.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage searchFeedElementsInFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, String q, Boolean showInternalOnly)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedType`
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

Any record ID, including a group ID.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

showInternalOnly
Type: Boolean

Specifies whether to show only feed elements from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, showInternalOnly, result)

Apex Developer Guide: Testing ConnectApi Code

searchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, showInternalOnly, filter)

Get a page of sorted and filtered feed elements from a feed for a record or user that match the search criteria. Each feed element includes no more than the specified number of comments. Specify whether to return feed elements posted by internal (non-Experience Cloud site) users only.

API Version

32.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage searchFeedElementsInFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, String q, Boolean showInternalOnly,
ConnectApi.FeedFilter filter)
```

Parameters

communityId

Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType

Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

Any record ID, including a group ID.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

showInternalOnly
Type: Boolean

Specifies whether to show only feed elements from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

filter
Type: `ConnectApi.FeedFilter`

Specifies the feed filters.

- `AllQuestions`—Feed elements that are questions.
- `AuthoredBy`—Feed elements authored by the user profile owner. This value is valid only for the `UserProfile` feed.
- `CommunityScoped`—Feed elements that are scoped to Experience Cloud sites. Currently, these feed elements have a User or a Group parent record. However, other parent record types could be scoped to sites in the future. Feed elements that are always visible in all sites are filtered out. This value is valid only for the `UserProfile` feed.
- `QuestionsWithCandidateAnswers`—Feed elements that are questions that have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `QuestionsWithCandidateAnswersReviewedPublished`—Feed elements that are questions that have candidate answers that have been reviewed or published. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Read`—Feed elements that are older than 30 days or are marked as read for the context user. Includes existing feed elements when the context user joined the group. This value is valid only for the `Record` feed of a group.
- `SolvedQuestions`—Feed elements that are questions and that have a best answer.
- `UnansweredQuestions`—Feed elements that are questions and that don't have any answers.
- `UnansweredQuestionsWithCandidateAnswers`—Feed elements that are questions that don't have answers but have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Unread`—Feed elements that are created in the past 30 days and aren't marked as read for the context user. This value is valid only for the `Record` feed of a group.
- `UnsolvedQuestions`—Feed elements that are questions and that don't have a best answer.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, showInternalOnly, filter, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, showInternalOnly, customFilter)`

Get a page of sorted and filtered feed elements from a case feed that match the search criteria.

API Version

40.0

Available to Guest Users

40.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage searchFeedElementsInFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, String q, Boolean showInternalOnly, String
customFilter)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

The ID of a case.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

showInternalOnly
Type: Boolean

Specifies whether to show only feed elements from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

filter
Type: String

Custom filter that applies only to the case feed. See customFeedFilter in the *Metadata API Developer Guide* for supported values.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, showInternalOnly, customFilter, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedElementsInFilterFeed(communityId, subjectId, keyPrefix, q)`

Get the feed elements from a feed filtered by a key prefix that match the search criteria.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage searchFeedElementsInFilterFeed(String
communityId, String subjectId, String keyPrefix, String q)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedElementsInFilterFeed(communityId, subjectId, keyPrefix, q, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedElementsInFilterFeed(communityId, subjectId, keyPrefix, pageParam, pageSize, sortParam, q)`

Get a page of sorted feed elements from a feed filtered by a key prefix that match the search criteria.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage searchFeedElementsInFilterFeed(String communityId, String subjectId, String keyPrefix, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, String q)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedElementsInFilterFeed(communityId, subjectId, keyPrefix, pageParam, pageSize, sortParam, q, result)

Apex Developer Guide: Testing ConnectApi Code

searchFeedElementsInFilterFeed(communityId, subjectId, keyPrefix, recentCommentCount, density, pageParam, pageSize, sortParam, q)

Get a page of sorted feed elements from a feed filtered by a key prefix that match the search criteria. Each feed element includes no more than the specified number of comments.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElementPage searchFeedElementsInFilterFeed(String
communityId, String subjectId, String keyPrefix, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, String q)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.FeedElementPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedElementsInFilterFeed(communityId, subjectId, keyPrefix, recentCommentCount, density, pageParam, pageSize, sortParam, q, result)

Apex Developer Guide: Testing ConnectApi Code

`searchStreams(communityId, q)`

Search the Chatter feed streams for the context user.

API Version

40.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterStreamPage searchStreams(String communityId, String q)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.ChatterStreamPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchStreams(communityId, q, result)

Apex Developer Guide: Testing ConnectApi Code

`searchStreams(communityId, q, sortParam)`

Search and sort the Chatter feed streams for the context user.

API Version

40.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterStreamPage searchStreams(String communityId, String q,
ConnectApi.SortOrder sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

sortParam
Type: `ConnectApi.SortOrder`

Specifies the sort order. Values are:

- `Ascending`—Items are in ascending alphabetical order (A-Z).
- `Descending`—Items are in descending alphabetical order (Z-A).
- `MostRecentlyViewed`—Items are in descending chronological order by view. This sort order is valid only for Chatter feed streams.

If not specified, default value is `Ascending`.

Return Value

Type: `ConnectApi.ChatterStreamPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchStreams(communityId, q, sortParam, result)

Apex Developer Guide: Testing ConnectApi Code

`searchStreams(communityId, q, pageParam, pageSize)`

Search the Chatter feed streams for the context user and return a page of results.

API Version

40.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterStreamPage searchStreams(String communityId, String q,
Integer pageParam, Integer pageSize)
```

Parameters

communityId
: Type: String

  ID for an Experience Cloud site, `internal`, or `null`.

q
: Type: String

  Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

pageParam
: Type: Integer

  Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
: Type: Integer

  Specifies the number of items per page. Valid values are from 1 to 250. The default size is 25.

Return Value

Type: `ConnectApi.ChatterStreamPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

- setTestSearchStreams(communityId, q, pageParam, pageSize, result)
- *Apex Developer Guide*: Testing ConnectApi Code

`searchStreams(communityId, q, pageParam, pageSize, sortParam)`

Search the Chatter feed streams for the context user and return a sorted page of results.

API Version

40.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterStreamPage searchStreams(String communityId, String q,
Integer pageParam, Integer pageSize, ConnectApi.SortOrder sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 to 250. The default size is 25.

sortParam
Type: `ConnectApi.SortOrder`

Specifies the sort order. Values are:

- `Ascending`—Items are in ascending alphabetical order (A-Z).
- `Descending`—Items are in descending alphabetical order (Z-A).
- `MostRecentlyViewed`—Items are in descending chronological order by view. This sort order is valid only for Chatter feed streams.

If not specified, default value is `Ascending`.

Return Value

Type: `ConnectApi.ChatterStreamPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchStreams(communityId, q, pageParam, pageSize, sortParam, result)

Apex Developer Guide: Testing ConnectApi Code

`searchStreams(communityId, q, pageParam, pageSize, sortParam, globalScope)`

Search the Chatter feed streams from all Experience Cloud sites for the context user and return a sorted page of results.

API Version

41.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterStreamPage searchStreams(String communityId, String q,
Integer pageParam, Integer pageSize, ConnectApi.SortOrder sortParam, Boolean globalScope)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`q`
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

`pageParam`
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

`pageSize`
Type: Integer

Specifies the number of items per page. Valid values are from 1 to 250. The default size is 25.

`sortParam`
Type: `ConnectApi.SortOrder`

Specifies the sort order. Values are:

- `Ascending`—Items are in ascending alphabetical order (A-Z).
- `Descending`—Items are in descending alphabetical order (Z-A).

- `MostRecentlyViewed`—Items are in descending chronological order by view. This sort order is valid only for Chatter feed streams.

If not specified, default value is `Ascending`.

`globalScope`
Type: Boolean

Specifies whether to get streams from all the context user's Experience Cloud sites, regardless of the *`communityId`* value.

Tip: If you know the *`communityId`* for the streams, we recommend setting *`globalScope`* to `false`.

Return Value

Type: `ConnectApi.ChatterStreamPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

`setCommentIsVerified(communityId, commentId, isVerified)`

Mark a comment as verified or unverified.

API Version

41.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.VerifiedCapability setCommentIsVerified(String communityId,
String commentId, Boolean isVerified)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`commentId`
Type: String

ID of the comment on a question post. Only one comment on a question post can be marked as verified.

`isVerified`
Type: Boolean

Specifies whether to mark the comment as verified (`true`) or unverified (`false`).

Only verified comments can be marked as unverified, and only unverified comments can be marked as verified.

Return Value

Type: `ConnectApi.VerifiedCapability`

If the comment doesn't support this capability, the return value is `ConnectApi.NotFoundException`.

setCommentIsVerifiedByAnonymized(communityId, commentId, isVerified, isVerifiedByAnonymized)

Mark a comment as verified by an anonymous user.

API Version

43.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.VerifiedCapability setCommentIsVerifiedByAnonymized(String communityId, String commentId, Boolean isVerified, Boolean isVerifiedByAnonymized)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

commentId
Type: String

ID of the comment on a question post. Only one comment on a question post can be marked as verified.

isVerified
Type: Boolean

Specifies whether to mark the comment as verified (`true`) or unverified (`false`).

Only verified comments can be marked as unverified, and only unverified comments can be marked as verified.

isVerifiedByAnonymized
Type: Boolean

Specifies whether to mark the comment as verified by an anonymous user (`true`).

If a user previously verified a comment and then requested the activity to be deleted, use `isVerifiedByAnonymized` to maintain the verification and anonymize the value of `lastVerifiedByUser`.

You can't set `isVerified` and `isVerifiedByAnonymized` to `true` at the same time. `isVerifiedByAnonymized` can be set to `true` only if `isVerified` is already set to `true`.

You can't set `isVerifiedByAnonymized` to `false`. After `isVerifiedByAnonymized` is set to `true`, it can be undone only when another user marks the comment as unverified and then reverifies the comment.

Return Value

Type: ConnectApi.VerifiedCapability

If the comment doesn't support this capability, the return value is ConnectApi.NotFoundException.

setCommentVote(communityId, commentId, upDownVote)

Upvote or downvote a comment.

API Version

41.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UpDownVoteCapability setCommentVote(String communityId, String commentId, ConnectApi.UpDownVoteCapabilityInput upDownVote)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

commentId
Type: String

ID of the comment.

upDownVote
Type: ConnectApi.UpDownVoteCapabilityInput

A `ConnectApi.UpDownVoteCapabilityInput` object that includes your vote.

Return Value

Type: ConnectApi.UpDownVoteCapability

If the comment doesn't support this capability, the return value is ConnectApi.NotFoundException.

setFeedCommentStatus(communityId, commentId, status)

Set the status of a comment.

API Version

38.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.StatusCapability setFeedCommentStatus(String communityId,
String commentId, ConnectApi.StatusCapabilityInput status)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

commentId
Type: String

ID of the comment.

status
Type: `ConnectApi.StatusCapabilityInput`

A `ConnectApi.StatusCapabilityInput` object that includes the status you want to set.

Return Value

Type: `ConnectApi.StatusCapability`

If the comment doesn't support this capability, the return value is `ConnectApi.NotFoundException`.

Usage

Only users with the Can Approve Feed Post and Comment permission can set the status of a feed post or comment.

setFeedElementIsClosed(communityId, feedElementId, isClosed)

Set a feed element to closed.

Users can't edit (specifically the feed item body or title), comment on, or delete a closed feed element. If the closed feed element is a poll, users can't vote on it. Users can't edit (specifically the comment body) or delete a comment on a closed feed element or select or remove it as best answer.

Admins and moderators can edit and delete closed feed elements and comments on closed feed elements. Admins and moderators can select or remove the best answer status on comments on closed feed elements.

API Version

43.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.CloseCapability setFeedElementIsClosed(String communityId,
String feedElementId, Boolean isClosed)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

isClosed
Type: Boolean

Specifies whether to set the feed element to closed (`true`) or not (`false`).

Return Value

Type: `ConnectApi.CloseCapability`

If the feed element doesn't support this capability, the return value is `ConnectApi.NotFoundException`.

`setFeedElementVote(communityId, feedElementId, upDownVote)`

Upvote or downvote a feed element.

API Version

41.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UpDownVoteCapability setFeedElementVote(String communityId,
String feedElementId, ConnectApi.UpDownVoteCapabilityInput upDownVote)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

upDownVote
Type: ConnectApi.UpDownVoteCapabilityInput

A ConnectApi.UpDownVoteCapabilityInput object that includes your vote.

Return Value

Type: ConnectApi.UpDownVoteCapability

If the feed element doesn't support this capability, the return value is ConnectApi.NotFoundException.

setFeedEntityStatus(communityId, feedElementId, status)

Set the status of a feed post.

API Version

37.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.StatusCapability setFeedEntityStatus(String communityId, String feedElementId, ConnectApi.StatusCapabilityInput status)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, internal, or null.

feedElementId
Type: String

ID of the feed element.

status
Type: ConnectApi.StatusCapabilityInput

A ConnectApi.StatusCapabilityInput object that includes the status you want to set.

Return Value

Type: ConnectApi.StatusCapability

If the feed element doesn't support this capability, the return value is ConnectApi.NotFoundException.

Usage

Only users with the Can Approve Feed Post and Comment permission can set the status of a feed post or comment.

`setIsMutedByMe(communityId, feedElementId, isMutedByMe)`

Mute or unmute a feed element.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.MuteCapability setIsMutedByMe(String communityId, String feedElementId, Boolean isMutedByMe)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

isMutedByMe
Type: Boolean

Indicates whether the feed element is muted for the context user. Default value is `false`.

Return Value

Type: `ConnectApi.MuteCapability`

If the feed element doesn't support this capability, the return value is `ConnectApi.NotFoundException`.

`setIsReadByMe(communityId, feedElementId, readBy)`

Mark a feed element as read for the context user using an input class.

API Version

40.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ReadByCapability setIsReadByMe(String communityId, String
feedElementId, ConnectApi.ReadByCapabilityInput readBy)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element to mark as read.

readBy
Type: `ConnectApi.ReadByCapabilityInput`

A `ConnectApi.ReadByCapabilityInput` body indicating to mark the feed elements as read.

Return Value

Type: `ConnectApi.ReadByCapability`

If the feed element doesn't support this capability, the return value is `ConnectApi.NotFoundException`.

`setIsReadByMe(communityId, feedElementId, isReadByMe)`

Mark a feed element as read for the context user.

API Version

40.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ReadByCapability setIsReadByMe(String communityId, String
feedElementId, Boolean isReadByMe)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element to mark as read.

isReadByMe
Type: Boolean

Specifies to mark the feed element as read (`true`) for the context user.

Return Value

Type: `ConnectApi.ReadByCapability`

If the feed element doesn't support this capability, the return value is `ConnectApi.NotFoundException`.

updateComment(communityId, commentId, comment)

Edit a comment.

API Version

34.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Comment updateComment(String communityId, String commentId,
ConnectApi.CommentInput comment)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

commentId
Type: String

ID of the comment to be edited.

comment
Type: `ConnectApi.CommentInput`

Information about the comment to be edited.

Return Value

Type: `ConnectApi.Comment`

If the comment doesn't support the `edit` capability, the return value is `ConnectApi.NotFoundException`.

Example

```
String commentId;
String communityId = Network.getNetworkId();
```

```
// Get the last feed item created by the context user.
List<FeedItem> feedItems = [SELECT Id FROM FeedItem WHERE CreatedById = :UserInfo.getUserId()
 ORDER BY CreatedDate DESC];
if (feedItems.isEmpty()) {
    // Return null within anonymous apex.
    return null;
}
String feedElementId = feedItems[0].id;

ConnectApi.CommentPage commentPage =
ConnectApi.ChatterFeeds.getCommentsForFeedElement(communityId, feedElementId);
if (commentPage.items.isEmpty()) {
    // Return null within anonymous apex.
    return null;
}
commentId = commentPage.items[0].id;

ConnectApi.FeedEntityIsEditable isEditable =
ConnectApi.ChatterFeeds.isCommentEditableByMe(communityId, commentId);

if (isEditable.isEditableByMe == true){
    ConnectApi.CommentInput commentInput = new ConnectApi.CommentInput();
    ConnectApi.MessageBodyInput messageBodyInput = new ConnectApi.MessageBodyInput();
    ConnectApi.TextSegmentInput textSegmentInput = new ConnectApi.TextSegmentInput();

    messageBodyInput.messageSegments = new List<ConnectApi.MessageSegmentInput>();

    textSegmentInput.text = 'This is my edited comment.';
    messageBodyInput.messageSegments.add(textSegmentInput);

    commentInput.body = messageBodyInput;

    ConnectApi.Comment editedComment = ConnectApi.ChatterFeeds.updateComment(communityId,
 commentId, commentInput);
}
```

updateDirectMessage(communityId, feedElementId, directMessage)

Update the members of a direct message.

API Version

40.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.DirectMessageCapability updateDirectMessage(String communityId,
String feedElementId, ConnectApi.DirectMessageCapabilityInput directMessage)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

directMessage
Type: `ConnectApi.DirectMessageCapabilityInput`

A `ConnectApi.DirectMessageCapabilityInput` body that includes the members to add and remove.

Return Value

Type: `ConnectApi.DirectMessageCapability`

If the feed element doesn't support this capability, the return value is `ConnectApi.NotFoundException`.

updateFeedElement(communityId, feedElementId, feedElement)

Edit a feed element.

API Version

34.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElement updateFeedElement(String communityId, String
feedElementId, ConnectApi.FeedElementInput feedElement)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element to be edited. Feed items are the only type of feed element that can be edited.

feedElement
Type: `ConnectApi.FeedElementInput`

Information about the feed item to be edited.

Return Value

Type: `ConnectApi.FeedElement`

If the feed element doesn't support the `edit` capability, the return value is `ConnectApi.NotFoundException`.

Example for Editing a Feed Post

```
String communityId = Network.getNetworkId();

// Get the last feed item created by the context user.
List<FeedItem> feedItems = [SELECT Id FROM FeedItem WHERE CreatedById = :UserInfo.getUserId()
 ORDER BY CreatedDate DESC];
if (feedItems.isEmpty()) {
    // Return null within anonymous apex.
    return null;
}
String feedElementId = feedItems[0].id;

ConnectApi.FeedEntityIsEditable isEditable =
ConnectApi.ChatterFeeds.isFeedElementEditableByMe(communityId, feedElementId);

if (isEditable.isEditableByMe == true){
    ConnectApi.FeedItemInput feedItemInput = new ConnectApi.FeedItemInput();
    ConnectApi.MessageBodyInput messageBodyInput = new ConnectApi.MessageBodyInput();
    ConnectApi.TextSegmentInput textSegmentInput = new ConnectApi.TextSegmentInput();

    messageBodyInput.messageSegments = new List<ConnectApi.MessageSegmentInput>();

    textSegmentInput.text = 'This is my edited post.';
    messageBodyInput.messageSegments.add(textSegmentInput);

    feedItemInput.body = messageBodyInput;

    ConnectApi.FeedElement editedFeedElement =
ConnectApi.ChatterFeeds.updateFeedElement(communityId, feedElementId, feedItemInput);
}
```

Example for Editing a Question Title and Post

```
String communityId = Network.getNetworkId();

// Get the last feed item created by the context user.
List<FeedItem> feedItems = [SELECT Id FROM FeedItem WHERE CreatedById = :UserInfo.getUserId()
 ORDER BY CreatedDate DESC];
if (feedItems.isEmpty()) {
    // Return null within anonymous apex.
    return null;
}
String feedElementId = feedItems[0].id;

ConnectApi.FeedEntityIsEditable isEditable =
ConnectApi.ChatterFeeds.isFeedElementEditableByMe(communityId, feedElementId);
```

```
if (isEditable.isEditableByMe == true){

    ConnectApi.FeedItemInput feedItemInput = new ConnectApi.FeedItemInput();
    ConnectApi.FeedElementCapabilitiesInput feedElementCapabilitiesInput = new
ConnectApi.FeedElementCapabilitiesInput();
    ConnectApi.QuestionAndAnswersCapabilityInput questionAndAnswersCapabilityInput = new
ConnectApi.QuestionAndAnswersCapabilityInput();
    ConnectApi.MessageBodyInput messageBodyInput = new ConnectApi.MessageBodyInput();
    ConnectApi.TextSegmentInput textSegmentInput = new ConnectApi.TextSegmentInput();

    messageBodyInput.messageSegments = new List<ConnectApi.MessageSegmentInput>();

    textSegmentInput.text = 'This is my edited question.';
    messageBodyInput.messageSegments.add(textSegmentInput);

    feedItemInput.body = messageBodyInput;
    feedItemInput.capabilities = feedElementCapabilitiesInput;

    feedElementCapabilitiesInput.questionAndAnswers = questionAndAnswersCapabilityInput;
    questionAndAnswersCapabilityInput.questionTitle = 'Where is my edited question?';

    ConnectApi.FeedElement editedFeedElement =
ConnectApi.ChatterFeeds.updateFeedElement(communityId, feedElementId, feedItemInput);
}
```

updateFeedElementBookmarks(communityId, feedElementId, bookmarks)

Bookmark a feed element or remove a bookmark from a feed element using an input class.

API Version

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.BookmarksCapability updateFeedElementBookmarks(String
communityId, String feedElementId, ConnectApi.BookmarksCapabilityInput bookmarks)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

bookmarks
Type: ConnectApi.BookmarksCapabilityInput

Information about a bookmark.

Return Value

Type: ConnectApi.BookmarksCapability

If the feed element doesn't support this capability, the return value is ConnectApi.NotFoundException.

updateFeedElementBookmarks(communityId, feedElementId, isBookmarkedByCurrentUser)

Bookmark a feed element or remove a bookmark from a feed element.

API Version

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.BookmarksCapability updateFeedElementBookmarks(String communityId, String feedElementId, Boolean isBookmarkedByCurrentUser)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

isBookmarkedByCurrentUser
Type: Boolean

Specify whether to bookmark the feed element (`true`) or not (`false`).

Return Value

Type: ConnectApi.BookmarksCapability

If the feed element doesn't support this capability, the return value is ConnectApi.NotFoundException.

Example

```
ConnectApi.BookmarksCapability bookmark =
ConnectApi.ChatterFeeds.updateFeedElementBookmarks(null, '0D5D0000000KuGh', true);
```

updateFeedElementReadByCapabilityBatch(communityId, feedElementIds, readBy)

Mark multiple feed elements as read by the context user at the same time using an input class.

API Version

40.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.BatchResult[] updateFeedElementReadByCapabilityBatch(String
communityId, List<String> feedElementIds, ConnectApi.ReadByCapabilityInput readBy)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementIds
Type: List<String>

Up to 500 feed element IDs to mark as read.

readBy
Type: `ConnectApi.ReadByCapabilityInput`

A `ConnectApi.ReadByCapabilityInput` body indicating to mark the feed elements as read.

Return Value

Type: `ConnectApi.BatchResult`[]

If the feed element doesn't support this capability, the return value is `ConnectApi.NotFoundException`.

The returned objects correspond to each of the input objects and are returned in the same order as the input objects.

The method call fails only if an error occurs that affects the entire operation (such as a parsing failure). If an individual object causes an error, the error is embedded within the `ConnectApi.BatchResult` list.

updateFeedElementReadByCapabilityBatch(communityId, feedElementIds, isReadByMe)

Mark multiple feed elements as read by the context user at the same time.

API Version

40.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.BatchResult[] updateFeedElementReadByCapabilityBatch(String communityId, List<String> feedElementIds, Boolean isReadByMe)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementIds
Type: List<String>

Up to 500 feed element IDs to mark as read.

isReadByMe
Type: Boolean

Specifies to mark the feed element as read (`true`) for the context user.

Return Value

Type: `ConnectApi.BatchResult`[]

If the feed element doesn't support this capability, the return value is `ConnectApi.NotFoundException`.

updateLikeForComment(communityId, commentId, isLikedByCurrentUser)

Like or unlike a comment.

API Version

39.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterLikePage updateLikeForComment(String communityId, String commentId, Boolean isLikedByCurrentUser)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

commentId
Type: String

ID of the comment.

isLikedByCurrentUser
Type: Boolean

Specifies if the context user likes (`true`) or unlikes (`false`) the comment.

Return Value

Type: `ConnectApi.ChatterLikePage`

updateLikeForFeedElement(communityId, feedElementId, isLikedByCurrentUser)

Like or unlike a feed element.

API Version

39.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterLikePage updateLikeForFeedElement(String communityId,
String feedElementId, Boolean isLikedByCurrentUser)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

isLikedByCurrentUser
Type: Boolean

Specifies if the context user likes (`true`) or unlikes (`false`) the feed element.

Return Value

Type: ConnectApi.ChatterLikePage

If the feed element doesn't support the `ChatterLikes` capability, the return value is ConnectApi.NotFoundException.

updatePinnedFeedElements(communityId, feedType, subjectId, pin)

Pin or unpin feed elements to a group or topic feed.

API Version

41.0

Available to Guest Users

41.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.PinCapability updatePinnedFeedElements(String communityId,
ConnectApi.FeedType feedType, String subjectId, ConnectApi.PinCapabilityInput pin)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: ConnectApi.FeedType

The type of feed. Valid values are `Record` and `Topics`.

subjectId
Type: String

If *feedType* is `Record`, *subjectId* must be a group ID. If *feedType* is `Topics`, *subjectId* must be a topic ID.

pin
Type: ConnectApi.PinCapabilityInput

A `ConnectApi.PinCapabilityInput` object indicating the feed element to pin or unpin.

Return Value

Type: ConnectApi.PinCapability

If the feed doesn't support this capability, the return value is ConnectApi.NotFoundException.

updateStream(communityId, streamId, streamInput)

Update a Chatter feed stream.

API Version

39.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterStream updateStream(String communityId, String streamId,
ConnectApi.ChatterStreamInput streamInput)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

streamId
Type: String

ID of the Chatter feed stream.

streamInput
Type: `ConnectApi.ChatterStreamInput`

A `ConnectApi.ChatterStreamInput` object.

Return Value

Type: `ConnectApi.ChatterStream`

voteOnFeedElementPoll(communityId, feedElementId, myChoiceId)

Vote on a poll or change your vote on a poll.

API Version

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.PollCapability voteOnFeedElementPoll(String communityId, String
feedElementId, String myChoiceId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

myChoiceId
Type: String

ID of the poll item you're voting for. The key prefix for poll items is 09A.

Return Value

Type: `ConnectApi.PollCapability`

If the feed element doesn't support this capability, the return value is `ConnectApi.NotFoundException`.

Example

```
ConnectApi.PollCapability poll = ConnectApi.ChatterFeeds.voteOnFeedElementPoll(null,
'0D5D0000000XZaUKAW', '09AD000000000TKMAY');
```

ChatterFeeds Test Methods

These test methods are for `ChatterFeeds`. All methods are static.

For information about using these methods to test your `ConnectApi` code, see Testing ConnectApi Code.

IN THIS SECTION:

setTestGetFeedElementsFromFeed(communityId, feedType, result)
Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

setTestGetFeedElementsFromFeed(communityId, feedType, pageParam, pageSize, sortParam, result)
Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

setTestGetFeedElementsFromFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, result)
Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

setTestGetFeedElementsFromFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, filter, result)
Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

setTestGetFeedElementsFromFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, filter, threadedCommentsCollapsed, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.getFeedElementsFromFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, pageParam, pageSize, sortParam, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, showInternalOnly, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, filter, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, filter, threadedCommentsCollapsed, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.getFeedElementsFromFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, customFilter, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, filter, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, filter, threadedCommentsCollapsed, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, customFilter, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, customFilter, threadedCommentsCollapsed, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.getFeedElementsFromFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestGetFeedElementsFromFilterFeed(communityId, subjectId, keyPrefix, result)

a `ConnectApi.FeedElementPage` object to be returned when the matching `getFeedElementsFromFilterFeed` method is called in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetFeedElementsFromFilterFeed(communityId, subjectId, keyPrefix, pageParam, pageSize, sortParam, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `getFeedElements FromFilterFeed` method is called in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetFeedElementsFromFilterFeed(communityId, subjectId, keyPrefix, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `getFeedElements FromFilterFeed` method is called in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetFeedElementsFromFilterFeedUpdatedSince(communityId, subjectId, keyPrefix, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the `getFeedElementsFromFilterFeedUpdatedSince` method is called in a test context.

setTestGetFeedElementsUpdatedSince(communityId, feedType, recentCommentCount, density, pageParam, pageSize, updatedSince, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsUpdatedSince` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetFeedElementsUpdatedSince(communityId, feedType, recentCommentCount, density, pageParam, pageSize, updatedSince, filter, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsUpdatedSince` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, updatedSince, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsUpdatedSince` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, updatedSince, showInternalOnly, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsUpdatedSince` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, filter, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsUpdatedSince` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, customFilter, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsUpdatedSince` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, showInternalOnly, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsUpdatedSince` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, showInternalOnly, filter, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsUpdatedSince` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, showInternalOnly, customFilter, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsUpdatedSince` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetRelatedPosts(communityId, feedElementId, filter, maxResults, result)

Register a `ConnectApi.RelatedFeedPosts` object to be returned when the matching `ConnectApi.getRelatedPosts(communityId, feedElementId, filter, maxResults)` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestGetTopUnansweredQuestions(communityId, result) (Pilot)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.getTopUnansweredQuestions` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestGetTopUnansweredQuestions(communityId, filter, result) (Pilot)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.getTopUnansweredQuestions` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestGetTopUnansweredQuestions(communityId, pageSize, result) (Pilot)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.getTopUnansweredQuestions` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestGetTopUnansweredQuestions(communityId, filter, pageSize, result) (Pilot)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.getTopUnansweredQuestions` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedElements(communityId, q, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElements` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedElements(communityId, q, sortParam, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElements` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedElements(communityId, q, threadedCommentsCollapsed, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElements` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedElements(communityId, q, pageParam, pageSize, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElements` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedElements(communityId, q, pageParam, pageSize, sortParam, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElements` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedElements(communityId, q, pageParam, pageSize, threadedCommentsCollapsed, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElements` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedElements(communityId, q, recentCommentCount, pageParam, pageSize, sortParam, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElements` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedElementsInFeed(communityId, feedType, q, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedElementsInFeed(communityId, feedType, pageParam, pageSize, sortParam, q, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedElementsInFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, q, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedElementsInFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, q, filter, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedElementsInFeed(communityId, feedType, subjectId, q, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedElementsInFeed(communityId, feedType, subjectId, pageParam, pageSize, sortParam, q, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, filter, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `searchFeedElementsInFeed` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

setTestSearchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, customFilter, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, showInternalOnly, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, showInternalOnly, filter, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, showInternalOnly, customFilter, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedElementsInFilterFeed(communityId, subjectId, keyPrefix, q, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFilterFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedElementsInFilterFeed(communityId, subjectId, keyPrefix, pageParam, pageSize, sortParam, q, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFilterFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedElementsInFilterFeed(communityId, subjectId, keyPrefix, recentCommentCount, density, pageParam, pageSize, sortParam, q, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFilterFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchStreams(communityId, q, result)

Register a `ConnectApi.ChatterStreamPage` object to be returned when the matching `ConnectApi.searchStream(communityId, q)` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchStreams(communityId, q, sortParam, result)

Register a `ConnectApi.ChatterStreamPage` object to be returned when the matching `ConnectApi.searchStream(communityId, q, sortParam)` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchStreams(communityId, q, pageParam, pageSize, result)

Register a `ConnectApi.ChatterStreamPage` object to be returned when the matching `ConnectApi.searchStreams(communityId, q, pageParam, pageSize)` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchStreams(communityId, q, pageParam, pageSize, sortParam, result)

Register a `ConnectApi.ChatterStreamPage` object to be returned when the matching `ConnectApi.searchStreams(communityId, q, pageParam, pageSize, sortParam)` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchStreams(communityId, q, pageParam, pageSize, sortParam, globalScope, result)

Register a `ConnectApi.ChatterStreamPage` object to be returned when the matching `ConnectApi.searchStreams(communityId, q, pageParam, pageSize, sortParam, globalScope)` method is called in a test context. Use the method with the same parameters or you receive an exception.

`setTestGetFeedElementsFromFeed(communityId, feedType, result)`

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

API Version

31.0

Signature

```
public static Void setTestGetFeedElementsFromFeed(String communityId, ConnectApi.FeedType feedType, ConnectApi.FeedElementPage result)
```

Parameters

communityId

Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: ConnectApi.FeedType

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

result
Type: ConnectApi.FeedElementPage

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElementsFromFeed(communityId, feedType)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedElementsFromFeed(communityId, feedType, pageParam, pageSize, sortParam, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

API Version

31.0

Signature

```
public static Void setTestGetFeedElementsFromFeed(String communityId, ConnectApi.FeedType feedType, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: ConnectApi.FeedType

The only valid value for this parameter is `Company`.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElementsFromFeed(communityId, feedType, pageParam, pageSize, sortParam)
Apex Developer Guide: Testing ConnectApi Code

`setTestGetFeedElementsFromFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, result)`

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

API Version

31.0

Signature

```
public static Void setTestGetFeedElementsFromFeed(String communityId, ConnectApi.FeedType feedType, Integer recentCommentCount, ConnectApi.FeedDensity density, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElementsFromFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedElementsFromFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, filter, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

API Version

32.0

Signature

```
public static Void setTestGetFeedElementsFromFeed(String communityId, ConnectApi.FeedType feedType, Integer recentCommentCount, ConnectApi.FeedDensity density, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, ConnectApi.FeedFilter filter, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

The type of feed. The only valid value is `Home`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

filter
Type: `ConnectApi.FeedFilter`

Specifies the feed filters.

- `AllQuestions`—Feed elements that are questions.
- `AuthoredBy`—Feed elements authored by the user profile owner. This value is valid only for the `UserProfile` feed.
- `CommunityScoped`—Feed elements that are scoped to Experience Cloud sites. Currently, these feed elements have a User or a Group parent record. However, other parent record types could be scoped to sites in the future. Feed elements that are always visible in all sites are filtered out. This value is valid only for the `UserProfile` feed.
- `QuestionsWithCandidateAnswers`—Feed elements that are questions that have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `QuestionsWithCandidateAnswersReviewedPublished`—Feed elements that are questions that have candidate answers that have been reviewed or published. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Read`—Feed elements that are older than 30 days or are marked as read for the context user. Includes existing feed elements when the context user joined the group. This value is valid only for the `Record` feed of a group.
- `SolvedQuestions`—Feed elements that are questions and that have a best answer.
- `UnansweredQuestions`—Feed elements that are questions and that don't have any answers.
- `UnansweredQuestionsWithCandidateAnswers`—Feed elements that are questions that don't have answers but have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Unread`—Feed elements that are created in the past 30 days and aren't marked as read for the context user. This value is valid only for the `Record` feed of a group.
- `UnsolvedQuestions`—Feed elements that are questions and that don't have a best answer.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElementsFromFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, filter)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedElementsFromFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, filter, threadedCommentsCollapsed, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.getFeedElementsFromFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

44.0

Signature

```
public static Void setTestGetFeedElementsFromFeed(String communityId, ConnectApi.FeedType feedType, Integer recentCommentCount, ConnectApi.FeedDensity density, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, ConnectApi.FeedFilter filter, Boolean threadedCommentsCollapsed, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

The type of feed. The only valid value is `Home`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

filter
Type: `ConnectApi.FeedFilter`

Specifies the feed filters.

- `AllQuestions`—Feed elements that are questions.
- `AuthoredBy`—Feed elements authored by the user profile owner. This value is valid only for the `UserProfile` feed.
- `CommunityScoped`—Feed elements that are scoped to Experience Cloud sites. Currently, these feed elements have a User or a Group parent record. However, other parent record types could be scoped to sites in the future. Feed elements that are always visible in all sites are filtered out. This value is valid only for the `UserProfile` feed.
- `QuestionsWithCandidateAnswers`—Feed elements that are questions that have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `QuestionsWithCandidateAnswersReviewedPublished`—Feed elements that are questions that have candidate answers that have been reviewed or published. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Read`—Feed elements that are older than 30 days or are marked as read for the context user. Includes existing feed elements when the context user joined the group. This value is valid only for the `Record` feed of a group.
- `SolvedQuestions`—Feed elements that are questions and that have a best answer.
- `UnansweredQuestions`—Feed elements that are questions and that don't have any answers.
- `UnansweredQuestionsWithCandidateAnswers`—Feed elements that are questions that don't have answers but have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Unread`—Feed elements that are created in the past 30 days and aren't marked as read for the context user. This value is valid only for the `Record` feed of a group.
- `UnsolvedQuestions`—Feed elements that are questions and that don't have a best answer.

threadedCommentsCollapsed
Type: Boolean

Specifies whether to return threaded comments in a collapsed style (`true`) or not (`false`). If you pass in `null`, the default is `false`.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

Apex Developer Guide: Testing ConnectApi Code

getFeedElementsFromFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, filter, threadedCommentsCollapsed)

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

API Version

31.0

Signature

```
public static Void setTestGetFeedElementsFromFeed(String communityId, ConnectApi.FeedType feedType, String subjectId, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

The feed type.

subjectId
Type: String

ID of the context user or the alias `me`.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElementsFromFeed(communityId, feedType, subjectId)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, pageParam, pageSize, sortParam, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

API Version

31.0

Signature

```
public static Void setTestGetFeedElementsFromFeed(String communityId, ConnectApi.FeedType feedType, String subjectId, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessageModeration`, `DirectMessages`, `Filter`, `Home`, `Isolated`, `Landing`, `Moderation`, and `PendingReview`.

subjectId
Type: String

If *feedType* is `Record`, *subjectId* can be any record ID, including a group ID. If *feedType* is `Streams`, *subjectId* must be a stream ID. If *feedType* is `Topics`, *subjectId* must be a topic ID. If *feedType* is `UserProfile`, *subjectId* can be any user ID. If the *feedType* is any other value, *subjectId* must be the ID of the context user or the alias `me`.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: ConnectApi.FeedSortOrder

Values are:

- CreatedDateAsc—Sorts by oldest creation date. This sort order is available only for DirectMessageModeration, Draft, Isolated, Moderation, and PendingReview feeds.
- CreatedDateDesc—Sorts by most recent creation date.
- LastModifiedDateDesc—Sorts by most recent activity.
- MostViewed—Sorts by most viewed content. This sort order is available only for Home feeds when the ConnectApi.FeedFilter is UnansweredQuestions.
- Relevance—Sorts by most relevant content. This sort order is available only for Company, Home, and Topics feeds.

If you pass in null, the default value CreatedDateDesc is used.

result
Type: ConnectApi.FeedElementPage

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElementsFromFeed(communityId, feedType, subjectId, pageParam, pageSize, sortParam)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, result)

Register a ConnectApi.FeedElementPage object to be returned when getFeedElementsFromFeed is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

API Version

31.0

Signature

```
public static Void setTestGetFeedElementsFromFeed(String communityId, ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount, ConnectApi.FeedDensity density, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, internal, or null.

feedType
Type: ConnectApi.FeedType

Type of feed. Valid values include every ConnectApi.FeedType except Company, DirectMessageModeration, DirectMessages, Filter, Home, Isolated, Landing, Moderation, and PendingReview.

subjectId
Type: String

If *feedType* is Record, *subjectId* can be any record ID, including a group ID. If *feedType* is Streams, *subjectId* must be a stream ID. If *feedType* is Topics, *subjectId* must be a topic ID. If *feedType* is UserProfile, *subjectId* can be any user ID. If the *feedType* is any other value, *subjectId* must be the ID of the context user or the alias me.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: ConnectApi.FeedDensity

Specify the amount of content in a feed.

- AllUpdates—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- FewerUpdates—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, currentPageToken or nextPageToken. If you pass in null, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in null, the default size is 25.

sortParam
Type: ConnectApi.FeedSortOrder

Values are:

- CreatedDateAsc—Sorts by oldest creation date. This sort order is available only for DirectMessageModeration, Draft, Isolated, Moderation, and PendingReview feeds.
- CreatedDateDesc—Sorts by most recent creation date.
- LastModifiedDateDesc—Sorts by most recent activity.
- MostViewed—Sorts by most viewed content. This sort order is available only for Home feeds when the ConnectApi.FeedFilter is UnansweredQuestions.
- Relevance—Sorts by most relevant content. This sort order is available only for Company, Home, and Topics feeds.

If you pass in null, the default value CreatedDateDesc is used.

result
Type: ConnectApi.FeedElementPage

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, showInternalOnly, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

API Version

31.0

Signature

```
public static Void setTestGetFeedElementsFromFeed(String communityId, ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount, ConnectApi.FeedDensity density, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, Boolean showInternalOnly, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

Any record ID, including a group ID.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.

- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

showInternalOnly
Type: Boolean

Specifies whether to show only feed items from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, showInternalOnly)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, filter, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

35.0

Signature

```
public static Void setTestGetFeedElementsFromFeed(String communityId, ConnectApi.FeedType
feedType, String subjectId, Integer recentCommentCount, ConnectApi.FeedDensity density,
String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam,
ConnectApi.FeedFilter filter, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.UserProfile`.

subjectId
Type: String

ID of any user. To specify the context user, use the user ID or the alias `me`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.

- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

`filter`
Type: `ConnectApi.FeedFilter`

Value must be `ConnectApi.FeedFilter.CommunityScoped`. Filters the feed to include only feed elements that are scoped to Experience Cloud sites. Feed elements that are always visible in all sites are filtered out. Currently, feed elements scoped to sites have a User or a Group parent record. However, other parent record types could be scoped to sites in the future.

`result`
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, filter)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, filter, threadedCommentsCollapsed, result)`

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.getFeedElementsFromFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

44.0

Signature

```
public static Void setTestGetFeedElementsFromFeed(String communityId, ConnectApi.FeedType
feedType, String subjectId, Integer recentCommentCount, ConnectApi.FeedDensity density,
String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam,
ConnectApi.FeedFilter filter, Boolean threadedCommentsCollapsed,
ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.UserProfile`.

subjectId
Type: String

ID of any user. To specify the context user, use the user ID or the alias `me`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

filter
Type: `ConnectApi.FeedFilter`

Value must be `ConnectApi.FeedFilter.CommunityScoped`. Filters the feed to include only feed elements that are scoped to Experience Cloud sites. Feed elements that are always visible in all sites are filtered out. Currently, feed elements scoped to sites have a User or a Group parent record. However, other parent record types could be scoped to sites in the future.

threadedCommentsCollapsed
Type: Boolean

Specifies whether to return threaded comments in a collapsed style (`true`) or not (`false`). If you pass in `null`, the default is `false`.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

Apex Developer Guide: Testing ConnectApi Code

getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, filter, threadedCommentsCollapsed)

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, customFilter, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

API Version

40.0

Signature

```
public static Void setTestGetFeedElementsFromFeed(String communityId, ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount, ConnectApi.FeedDensity density, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, String customFilter, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

The ID of a case.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

customFilter
Type: String

Custom filter that applies only to the case feed. See customFeedFilter in the *Metadata API Developer Guide* for supported values.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, customFilter)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

API Version

31.0

Signature

```
public static Void setTestGetFeedElementsFromFeed(String communityId, ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount, Integer elementsPerBundle, ConnectApi.FeedDensity density, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, Boolean showInternalOnly, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

Any record ID, including a group ID.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

showInternalOnly
Type: Boolean

Specifies whether to show only feed items from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, filter, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

API Version

32.0

Signature

```
public static Void setTestGetFeedElementsFromFeed(String communityId, ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount, Integer elementsPerBundle, ConnectApi.FeedDensity density, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, Boolean showInternalOnly, ConnectApi.FeedFilter filter, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

Any record ID, including a group ID.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

showInternalOnly
Type: Boolean

Specifies whether to show only feed items from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

filter
Type: `ConnectApi.FeedFilter`

Specifies the feed filters.

- `AllQuestions`—Feed elements that are questions.
- `AuthoredBy`—Feed elements authored by the user profile owner. This value is valid only for the `UserProfile` feed.
- `CommunityScoped`—Feed elements that are scoped to Experience Cloud sites. Currently, these feed elements have a User or a Group parent record. However, other parent record types could be scoped to sites in the future. Feed elements that are always visible in all sites are filtered out. This value is valid only for the `UserProfile` feed.
- `QuestionsWithCandidateAnswers`—Feed elements that are questions that have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `QuestionsWithCandidateAnswersReviewedPublished`—Feed elements that are questions that have candidate answers that have been reviewed or published. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Read`—Feed elements that are older than 30 days or are marked as read for the context user. Includes existing feed elements when the context user joined the group. This value is valid only for the `Record` feed of a group.
- `SolvedQuestions`—Feed elements that are questions and that have a best answer.
- `UnansweredQuestions`—Feed elements that are questions and that don't have any answers.
- `UnansweredQuestionsWithCandidateAnswers`—Feed elements that are questions that don't have answers but have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Unread`—Feed elements that are created in the past 30 days and aren't marked as read for the context user. This value is valid only for the `Record` feed of a group.

- `UnsolvedQuestions`—Feed elements that are questions and that don't have a best answer.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, filter)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, filter, threadedCommentsCollapsed, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

API Version

44.0

Signature

```
public static Void setTestGetFeedElementsFromFeed(String communityId, ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount, Integer elementsPerBundle, ConnectApi.FeedDensity density, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, Boolean showInternalOnly, ConnectApi.FeedFilter filter, Boolean threadedCommentsCollapsed, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

Any record ID, including a group ID.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

showInternalOnly
Type: Boolean

Specifies whether to show only feed items from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

filter
Type: `ConnectApi.FeedFilter`

Specifies the feed filters.

- `AllQuestions`—Feed elements that are questions.
- `AuthoredBy`—Feed elements authored by the user profile owner. This value is valid only for the `UserProfile` feed.
- `CommunityScoped`—Feed elements that are scoped to Experience Cloud sites. Currently, these feed elements have a User or a Group parent record. However, other parent record types could be scoped to sites in the future. Feed elements that are always visible in all sites are filtered out. This value is valid only for the `UserProfile` feed.

- `QuestionsWithCandidateAnswers`—Feed elements that are questions that have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `QuestionsWithCandidateAnswersReviewedPublished`—Feed elements that are questions that have candidate answers that have been reviewed or published. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Read`—Feed elements that are older than 30 days or are marked as read for the context user. Includes existing feed elements when the context user joined the group. This value is valid only for the `Record` feed of a group.
- `SolvedQuestions`—Feed elements that are questions and that have a best answer.
- `UnansweredQuestions`—Feed elements that are questions and that don't have any answers.
- `UnansweredQuestionsWithCandidateAnswers`—Feed elements that are questions that don't have answers but have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Unread`—Feed elements that are created in the past 30 days and aren't marked as read for the context user. This value is valid only for the `Record` feed of a group.
- `UnsolvedQuestions`—Feed elements that are questions and that don't have a best answer.

`threadedCommentsCollapsed`
Type: Boolean

Specifies whether to return threaded comments in a collapsed style (`true`) or not (`false`). If you pass in `null`, the default is `false`.

`result`
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, filter, threadedCommentsCollapsed)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, customFilter, result)`

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

API Version

40.0

Signature

```
public static Void setTestGetFeedElementsFromFeed(String communityId, ConnectApi.FeedType
feedType, String subjectId, Integer recentCommentCount, Integer elementsPerBundle,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, Boolean showInternalOnly, String customFilter,
ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

The ID of a case.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

`showInternalOnly`
Type: Boolean

Specifies whether to show only feed items from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

`customFilter`
Type: String

Custom filter that applies only to the case feed. See customFeedFilter in the *Metadata API Developer Guide* for supported values.

`result`
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, customFilter)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, customFilter, threadedCommentsCollapsed, result)`

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.getFeedElementsFromFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

44.0

Signature

```
public static Void setTestGetFeedElementsFromFeed(String communityId, ConnectApi.FeedType
feedType, String subjectId, Integer recentCommentCount, Integer elementsPerBundle,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
```

```
ConnectApi.FeedSortOrder sortParam, Boolean showInternalOnly, String customFilter,
Boolean threadedCommentsCollapsed, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

The ID of a case.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.

- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

`showInternalOnly`
Type: Boolean

Specifies whether to show only feed items from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

`customFilter`
Type: String

Custom filter that applies only to the case feed. See customFeedFilter in the *Metadata API Developer Guide* for supported values.

`threadedCommentsCollapsed`
Type: Boolean

Specifies whether to return threaded comments in a collapsed style (`true`) or not (`false`). If you pass in `null`, the default is `false`.

`result`
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

Apex Developer Guide: Testing ConnectApi Code

getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, showInternalOnly, customFilter, threadedCommentsCollapsed)

`setTestGetFeedElementsFromFilterFeed(communityId, subjectId, keyPrefix, result)`

a `ConnectApi.FeedElementPage` object to be returned when the matching `getFeedElementsFromFilterFeed` method is called in a test context. Use the method with the same parameters or the code throws an exception.

API Version

31.0

Signature

```
public static Void setTestGetFeedElementsFromFilterFeed(String communityId, String subjectId, String keyPrefix, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElementsFromFilterFeed(communityId, subjectId, keyPrefix)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetFeedElementsFromFilterFeed(communityId, subjectId, keyPrefix, pageParam, pageSize, sortParam, result)`

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `getFeedElements FromFilterFeed` method is called in a test context. Use the method with the same parameters or the code throws an exception.

API Version

31.0

Signature

```
public static Void setTestGetFeedElementsFromFilterFeed(String communityId, String subjectId, String keyPrefix, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElementsFromFilterFeed(communityId, subjectId, keyPrefix, pageParam, pageSize, sortParam)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedElementsFromFilterFeed(communityId, subjectId, keyPrefix, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `getFeedElements FromFilterFeed` method is called in a test context. Use the method with the same parameters or the code throws an exception.

API Version

31.0

Signature

```
public static Void setTestGetFeedElementsFromFilterFeed(String communityId, String subjectId, String keyPrefix, Integer recentCommentCount, Integer elementsPerBundle, ConnectApi.FeedDensity density, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElementsFromFilterFeed(communityId, subjectId, keyPrefix, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedElementsFromFilterFeedUpdatedSince(communityId, subjectId, keyPrefix, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the `getFeedElementsFromFilterFeedUpdatedSince` method is called in a test context.

API Version

31.0

Signature

```
public static Void setTestGetFeedElementsFromFilterFeedUpdatedSince(String communityId,
String subjectId, String keyPrefix, Integer recentCommentCount, Integer
```

```
elementsPerBundle, ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
String updatedSince, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

updatedSince
Type: String

Opaque token defining the modification timestamp of the feed and the sort order.

The *updatedSince* parameter doesn't return feed elements that are created in the same second as the call.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

- getFeedElementsFromFilterFeedUpdatedSince(communityId, subjectId, keyPrefix, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince)
- *Apex Developer Guide*: Testing ConnectApi Code

setTestGetFeedElementsUpdatedSince(communityId, feedType, recentCommentCount, density, pageParam, pageSize, updatedSince, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsUpdatedSince` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

31.0

Signature

```
public static Void setTestGetFeedElementsUpdatedSince(String communityId,
ConnectApi.FeedType feedType, Integer recentCommentCount, ConnectApi.FeedDensity density,
String pageParam, Integer pageSize, String updatedSince, ConnectApi.FeedElementPage
result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

updatedSince
Type: String

An opaque token containing information about the last modified date of the feed. Do not construct this token. Retrieve this token from the `updatesToken` property of the `ConnectApi.FeedElementPage` response body.

The *updatedSince* parameter doesn't return feed elements that are created in the same second as the call.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElementsUpdatedSince(communityId, feedType, recentCommentCount, density, pageParam, pageSize, updatedSince)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedElementsUpdatedSince(communityId, feedType, recentCommentCount, density, pageParam, pageSize, updatedSince, filter, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsUpdatedSince` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

32.0

Signature

```
public static Void setTestGetFeedElementsUpdatedSince(String communityId,
ConnectApi.FeedType feedType, Integer recentCommentCount, ConnectApi.FeedDensity density,
String pageParam, Integer pageSize, String updatedSince, ConnectApi.FeedFilter filter,
ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

updatedSince
Type: String

An opaque token containing information about the last modified date of the feed. Do not construct this token. Retrieve this token from the `updatesToken` property of the `ConnectApi.FeedElementPage` response body.

The *updatedSince* parameter doesn't return feed elements that are created in the same second as the call.

filter
Type: `ConnectApi.FeedFilter`

Specifies the feed filters.

- `AllQuestions`—Feed elements that are questions.
- `AuthoredBy`—Feed elements authored by the user profile owner. This value is valid only for the `UserProfile` feed.
- `CommunityScoped`—Feed elements that are scoped to Experience Cloud sites. Currently, these feed elements have a User or a Group parent record. However, other parent record types could be scoped to sites in the future. Feed elements that are always visible in all sites are filtered out. This value is valid only for the `UserProfile` feed.
- `QuestionsWithCandidateAnswers`—Feed elements that are questions that have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `QuestionsWithCandidateAnswersReviewedPublished`—Feed elements that are questions that have candidate answers that have been reviewed or published. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Read`—Feed elements that are older than 30 days or are marked as read for the context user. Includes existing feed elements when the context user joined the group. This value is valid only for the `Record` feed of a group.

- `SolvedQuestions`—Feed elements that are questions and that have a best answer.
- `UnansweredQuestions`—Feed elements that are questions and that don't have any answers.
- `UnansweredQuestionsWithCandidateAnswers`—Feed elements that are questions that don't have answers but have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Unread`—Feed elements that are created in the past 30 days and aren't marked as read for the context user. This value is valid only for the `Record` feed of a group.
- `UnsolvedQuestions`—Feed elements that are questions and that don't have a best answer.

`result`
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElementsUpdatedSince(communityId, feedType, recentCommentCount, density, pageParam, pageSize, updatedSince, filter)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, updatedSince, result)`

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsUpdatedSince` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

31.0

Signature

```
public static Void setTestGetFeedElementsUpdatedSince(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize, String updatedSince,
ConnectApi.FeedElementPage result)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedType`
Type: `ConnectApi.FeedType`

One of these values:

- `Files`

- `Groups`
- `News`
- `People`
- `Record`

`subjectId`
Type: String

If *`feedType`* is `ConnectApi.Record`, *`subjectId`* can be any record ID, including a group ID. Otherwise, it must be the context user or the alias `me`.

`recentCommentCount`
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

`density`
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

`pageParam`
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

`pageSize`
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

`updatedSince`
Type: String

An opaque token containing information about the last modified date of the feed. Do not construct this token. Retrieve this token from the `updatesToken` property of the `ConnectApi.FeedElementPage` response body.

The *`updatedSince`* parameter doesn't return feed elements that are created in the same second as the call.

`result`
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, updatedSince)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, updatedSince, showInternalOnly, result)`

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsUpdatedSince` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

31.0

Signature

```
public static Void setTestGetFeedElementsUpdatedSince(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize, String updatedSince,
Boolean showInternalOnly, ConnectApi.FeedElementPage result)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedType`
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

`subjectId`
Type: String

Any record ID, including a group ID.

`recentCommentCount`
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

`density`
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

`pageParam`
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

`pageSize`
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

updatedSince
Type: String

An opaque token containing information about the last modified date of the feed. Do not construct this token. Retrieve this token from the `updatesToken` property of the `ConnectApi.FeedElementPage` response body.

The *updatedSince* parameter doesn't return feed elements that are created in the same second as the call.

showInternalOnly
Type: Boolean

Specifies whether to show only feed elements from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, updatedSince, showInternalOnly)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, filter, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsUpdatedSince` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

35.0

Signature

```
public static Void setTestGetFeedElementsUpdatedSince(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount, Integer
elementsPerBundle, ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
String updatedSince, ConnectApi.FeedFilter filter, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: ConnectApi.FeedType

Value must be `ConnectApi.FeedType.UserProfile`.

subjectId
Type: String

ID of any user. To specify the context user, use the user ID or the alias `me`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

density
Type: ConnectApi.FeedDensity

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

updatedSince
Type: String

Opaque token defining the modification timestamp of the feed and the sort order.

The *updatedSince* parameter doesn't return feed elements that are created in the same second as the call.

filter
Type: ConnectApi.FeedFilter

Value must be `ConnectApi.FeedFilter.CommunityScoped`. Filters the feed to include only feed elements that are scoped to Experience Cloud sites. Feed elements that are always visible in all sites are filtered out. Currently, feed elements scoped to sites have a User or a Group parent record. However, other parent record types could be scoped to sites in the future.

result
Type: ConnectApi.FeedElementPage

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, filter)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, customFilter, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsUpdatedSince` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

40.0

Signature

```
public static Void setTestGetFeedElementsUpdatedSince(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount, Integer
elementsPerBundle, ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
String updatedSince, String customFilter, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

The ID of a case.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

updatedSince
Type: String

Opaque token defining the modification timestamp of the feed and the sort order.

The *updatedSince* parameter doesn't return feed elements that are created in the same second as the call.

customFilter
Type: String

Custom filter that applies only to the case feed. See customFeedFilter in the *Metadata API Developer Guide* for supported values.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, customFilter)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, showInternalOnly, result)`

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsUpdatedSince` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

31.0

Signature

```
public static Void setTestGetFeedElementsUpdatedSince(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount, Integer
elementsPerBundle, ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
String updatedSince, Boolean showInternalOnly, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

Any record ID, including a group ID.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

updatedSince
Type: String

An opaque token containing information about the last modified date of the feed. Do not construct this token. Retrieve this token from the `updatesToken` property of the `ConnectApi.FeedElementPage` response body.

The *updatedSince* parameter doesn't return feed elements that are created in the same second as the call.

showInternalOnly
Type: Boolean

Specifies whether to show only feed elements from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, showInternalOnly)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, showInternalOnly, filter, result)

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsUpdatedSince` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

32.0

Signature

```
public static Void setTestGetFeedElementsUpdatedSince(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount, Integer
elementsPerBundle, ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
String updatedSince, Boolean showInternalOnly, ConnectApi.FeedFilter filter,
ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

Any record ID, including a group ID.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

updatedSince
Type: String

An opaque token containing information about the last modified date of the feed. Do not construct this token. Retrieve this token from the `updatesToken` property of the `ConnectApi.FeedElementPage` response body.

The *updatedSince* parameter doesn't return feed elements that are created in the same second as the call.

showInternalOnly
Type: Boolean

Specifies whether to show only feed elements from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

filter
Type: `ConnectApi.FeedFilter`

Specifies the feed filters.

- `AllQuestions`—Feed elements that are questions.
- `AuthoredBy`—Feed elements authored by the user profile owner. This value is valid only for the `UserProfile` feed.
- `CommunityScoped`—Feed elements that are scoped to Experience Cloud sites. Currently, these feed elements have a User or a Group parent record. However, other parent record types could be scoped to sites in the future. Feed elements that are always visible in all sites are filtered out. This value is valid only for the `UserProfile` feed.
- `QuestionsWithCandidateAnswers`—Feed elements that are questions that have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.

- `QuestionsWithCandidateAnswersReviewedPublished`—Feed elements that are questions that have candidate answers that have been reviewed or published. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Read`—Feed elements that are older than 30 days or are marked as read for the context user. Includes existing feed elements when the context user joined the group. This value is valid only for the `Record` feed of a group.
- `SolvedQuestions`—Feed elements that are questions and that have a best answer.
- `UnansweredQuestions`—Feed elements that are questions and that don't have any answers.
- `UnansweredQuestionsWithCandidateAnswers`—Feed elements that are questions that don't have answers but have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Unread`—Feed elements that are created in the past 30 days and aren't marked as read for the context user. This value is valid only for the `Record` feed of a group.
- `UnsolvedQuestions`—Feed elements that are questions and that don't have a best answer.

`result`

Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, showInternalOnly, filter)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, showInternalOnly, customFilter, result)`

Register a `ConnectApi.FeedElementPage` object to be returned when `getFeedElementsUpdatedSince` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

40.0

Signature

```
public static Void setTestGetFeedElementsUpdatedSince(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount, Integer
elementsPerBundle, ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
String updatedSince, Boolean showInternalOnly, String customFilter,
ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

The ID of a case.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

elementsPerBundle
Type: Integer

Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

updatedSince
Type: String

An opaque token containing information about the last modified date of the feed. Do not construct this token. Retrieve this token from the `updatesToken` property of the `ConnectApi.FeedElementPage` response body.

The *updatedSince* parameter doesn't return feed elements that are created in the same second as the call.

showInternalOnly
Type: Boolean

Specifies whether to show only feed elements from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

customFilter
Type: String

Custom filter that applies only to the case feed. See customFeedFilter in the *Metadata API Developer Guide* for supported values.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince, showInternalOnly, customFilter)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetRelatedPosts(communityId, feedElementId, filter, maxResults, result)`

Register a `ConnectApi.RelatedFeedPosts` object to be returned when the matching `ConnectApi.getRelatedPosts(communityId, feedElementId, filter, maxResults)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

37.0

Signature

```
public static Void setTestGetRelatedPosts(String communityId, String feedElementId,
ConnectApi.RelatedFeedPostType filter, Integer maxResults, ConnectApi.RelatedFeedPosts
result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element. The feed element must be a question.

filter
Type: `ConnectApi.RelatedFeedPostType`

Specifies the type of related feed post. Values are:

- `Answered`—Related questions that have at least one answer.
- `BestAnswer`—Related questions that have a best answer.
- `Generic`—All types of related questions, including answered, with a best answer, and unanswered.
- `Unanswered`—Related questions that don't have answers.

`Generic` is the default value.

maxResults
Type: Integer

The maximum number of results to return. You can return up to 25 results; 5 is the default.

result
Type: `ConnectApi.RelatedFeedPosts`

Object containing test data.

In version 37.0 and later, related feed posts are questions.

Return Value

Type: Void

`setTestGetTopUnansweredQuestions(communityId, result)` (Pilot)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.getTopUnansweredQuestions` method is called in a test context. Use the method with the same parameters or you receive an exception.

Note: We provided top-five unanswered questions to selected customers through a pilot program that required agreement to specific terms and conditions. This pilot program is closed and not accepting new participants.

API Version

42.0

Signature

```
public static Void setTestGetTopUnansweredQuestions(String communityId,
ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID of the Experience Cloud site.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getTopUnansweredQuestions(communityId) (Pilot)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetTopUnansweredQuestions(communityId, filter, result)` (Pilot)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.getTopUnansweredQuestions` method is called in a test context. Use the method with the same parameters or you receive an exception.

Note: We provided top-five unanswered questions to selected customers through a pilot program that required agreement to specific terms and conditions. This pilot program is closed and not accepting new participants.

API Version

42.0

Signature

```
public static Void setTestGetTopUnansweredQuestions(String communityId,
ConnectApi.TopUnansweredQuestionsFilterType filter, ConnectApi.FeedElementPage result)
```

Parameters

`communityId`
Type: String

ID of the Experience Cloud site.

`filter`
Type: `ConnectApi.FeedFilter`

Specifies the filter for the feed. `UnansweredQuestionsWithCandidateAnswers` is the only valid value.

`result`
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getTopUnansweredQuestions(communityId, filter) (Pilot)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetTopUnansweredQuestions(communityId, pageSize, result)` (Pilot)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.getTopUnansweredQuestions` method is called in a test context. Use the method with the same parameters or you receive an exception.

Note: We provided top-five unanswered questions to selected customers through a pilot program that required agreement to specific terms and conditions. This pilot program is closed and not accepting new participants.

API Version

42.0

Signature

```
public static Void setTestGetTopUnansweredQuestions(String communityId, Integer pageSize,
ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID of the Experience Cloud site.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 0 through 10. If you pass in `null`, the default size is 5.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getTopUnansweredQuestions(communityId, pageSize) (Pilot)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetTopUnansweredQuestions(communityId, filter, pageSize, result)` (Pilot)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.getTopUnansweredQuestions` method is called in a test context. Use the method with the same parameters or you receive an exception.

Note: We provided top-five unanswered questions to selected customers through a pilot program that required agreement to specific terms and conditions. This pilot program is closed and not accepting new participants.

API Version

42.0

Signature

```
public static Void setTestGetTopUnansweredQuestions(String communityId,
ConnectApi.FeedFilter filter, Integer pageSize, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID of the Experience Cloud site.

filter
Type: `ConnectApi.FeedFilter`

Specifies the filter for the feed. `UnansweredQuestionsWithCandidateAnswers` is the only valid value.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 0 through 10. If you pass in `null`, the default size is 5.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getTopUnansweredQuestions(communityId, filter, pageSize) (Pilot)

Apex Developer Guide: Testing ConnectApi Code

`setTestSearchFeedElements(communityId, q, result)`

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElements` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

31.0

Signature

```
public static Void setTestSearchFeedElements(String communityId, String q,
ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedElements(communityId, q)

Apex Developer Guide: Testing ConnectApi Code

setTestSearchFeedElements(communityId, q, sortParam, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElements` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

31.0

Signature

```
public static Void setTestSearchFeedElements(String communityId, String q,
ConnectApi.FeedSortOrder sortParam, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.

- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedElements(communityId, q, sortParam)

Apex Developer Guide: Testing ConnectApi Code

setTestSearchFeedElements(communityId, q, threadedCommentsCollapsed, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElements` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

44.0

Signature

```
public static Void setTestSearchFeedElements(String communityId, String q, Boolean threadedCommentsCollapsed, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

threadedCommentsCollapsed
Type: Boolean

Specifies whether to return threaded comments in a collapsed style (`true`) or not (`false`). If you pass in `null`, the default is `false`.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedElements(communityId, q, threadedCommentsCollapsed)

Apex Developer Guide: Testing ConnectApi Code

setTestSearchFeedElements(communityId, q, pageParam, pageSize, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElements` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

31.0

Signature

```
public static Void setTestSearchFeedElements(String communityId, String q, String pageParam, Integer pageSize, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.D

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedElements(communityId, q, pageParam, pageSize)

Apex Developer Guide: Testing ConnectApi Code

setTestSearchFeedElements(communityId, q, pageParam, pageSize, sortParam, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElements` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

31.0

Signature

```
public static Void setTestSearchFeedElements(String communityId, String q, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, ConnectApi.FeedElementPage result)
```

Parameters

communityId

Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q

Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

pageParam

Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize

Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam

Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.

- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedElements(communityId, q, pageParam, pageSize, sortParam)

Apex Developer Guide: Testing ConnectApi Code

setTestSearchFeedElements(communityId, q, pageParam, pageSize, threadedCommentsCollapsed, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElements` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

44.0

Signature

```
public static Void setTestSearchFeedElements(String communityId, String q, String
pageParam, Integer pageSize, Boolean threadedCommentsCollapsed,
ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

threadedCommentsCollapsed
Type: Boolean

Specifies whether to return threaded comments in a collapsed style (`true`) or not (`false`). If you pass in `null`, the default is `false`.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedElements(communityId, q, pageParam, pageSize, threadedCommentsCollapsed)

Apex Developer Guide: Testing ConnectApi Code

`setTestSearchFeedElements(communityId, q, recentCommentCount, pageParam, pageSize, sortParam, result)`

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElements` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

31.0

Signature

```
public static Void setTestSearchFeedElements(String communityId, String q, Integer recentCommentCount, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedElements(communityId, q, recentCommentCount, pageParam, pageSize, sortParam)

Apex Developer Guide: Testing ConnectApi Code

`setTestSearchFeedElementsInFeed(communityId, feedType, q, result)`

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

31.0

Signature

```
public static Void setTestSearchFeedElementsInFeed(String communityId,
ConnectApi.FeedType feedType, String q, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedElementsInFeed(communityId, feedType, q)

Apex Developer Guide: Testing ConnectApi Code

`setTestSearchFeedElementsInFeed(communityId, feedType, pageParam, pageSize, sortParam, q, result)`

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

31.0

Signature

```
public static Void setTestSearchFeedElementsInFeed(String communityId,
ConnectApi.FeedType feedType, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, String q, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedElementsInFeed(communityId, feedType, pageParam, pageSize, sortParam, q)

Apex Developer Guide: Testing ConnectApi Code

`setTestSearchFeedElementsInFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, q, result)`

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

31.0

Signature

```
public static Void setTestSearchFeedElementsInFeed(String communityId,
ConnectApi.FeedType feedType, Integer recentCommentCount, ConnectApi.FeedDensity density,
String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, String q,
ConnectApi.FeedElementPage result)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedType`
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

`recentCommentCount`
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

`density`
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

`pageParam`
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

`pageSize`
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: ConnectApi.FeedSortOrder

Values are:

- CreatedDateAsc—Sorts by oldest creation date. This sort order is available only for DirectMessageModeration, Draft, Isolated, Moderation, and PendingReview feeds.
- CreatedDateDesc—Sorts by most recent creation date.
- LastModifiedDateDesc—Sorts by most recent activity.
- MostViewed—Sorts by most viewed content. This sort order is available only for Home feeds when the ConnectApi.FeedFilter is UnansweredQuestions.
- Relevance—Sorts by most relevant content. This sort order is available only for Company, Home, and Topics feeds.

If you pass in null, the default value CreatedDateDesc is used.

q
Type: String

Required and can't be null. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

result
Type: ConnectApi.FeedElementPage

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedElementsInFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, q)

Apex Developer Guide: Testing ConnectApi Code

setTestSearchFeedElementsInFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, q, filter, result)

Register a ConnectApi.FeedElementPage object to be returned when the matching ConnectApi.searchFeedElementsInFeed method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

32.0

Signature

```
public static Void setTestSearchFeedElementsInFeed(String communityId,
ConnectApi.FeedType feedType, Integer recentCommentCount, ConnectApi.FeedDensity density,
String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, String q,
ConnectApi.FeedFilter filter, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

The type of feed. The only valid value is `Home`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

filter
Type: `ConnectApi.FeedFilter`

Specifies the feed filters.

- `AllQuestions`—Feed elements that are questions.
- `AuthoredBy`—Feed elements authored by the user profile owner. This value is valid only for the `UserProfile` feed.
- `CommunityScoped`—Feed elements that are scoped to Experience Cloud sites. Currently, these feed elements have a User or a Group parent record. However, other parent record types could be scoped to sites in the future. Feed elements that are always visible in all sites are filtered out. This value is valid only for the `UserProfile` feed.
- `QuestionsWithCandidateAnswers`—Feed elements that are questions that have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `QuestionsWithCandidateAnswersReviewedPublished`—Feed elements that are questions that have candidate answers that have been reviewed or published. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Read`—Feed elements that are older than 30 days or are marked as read for the context user. Includes existing feed elements when the context user joined the group. This value is valid only for the `Record` feed of a group.
- `SolvedQuestions`—Feed elements that are questions and that have a best answer.
- `UnansweredQuestions`—Feed elements that are questions and that don't have any answers.
- `UnansweredQuestionsWithCandidateAnswers`—Feed elements that are questions that don't have answers but have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Unread`—Feed elements that are created in the past 30 days and aren't marked as read for the context user. This value is valid only for the `Record` feed of a group.
- `UnsolvedQuestions`—Feed elements that are questions and that don't have a best answer.

`result`

Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedElementsInFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, q, filter)

Apex Developer Guide: Testing ConnectApi Code

`setTestSearchFeedElementsInFeed(communityId, feedType, subjectId, q, result)`

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

31.0

Signature

```
public static Void setTestSearchFeedElementsInFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, String q, ConnectApi.FeedElementPage
result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessages`, `Filter`, `Landing`, `Streams`, and `Topics`.

subjectId
Type: String

If *feedType* is `Record`, *subjectId* can be any record ID, including a group ID. If feed type is `UserProfile`, *subjectId* can be any user ID. If *feedType* is any other value, *subjectId* must be the ID of the context user or the alias `me`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedElementsInFeed(communityId, feedType, subjectId, q)

Apex Developer Guide: Testing ConnectApi Code

setTestSearchFeedElementsInFeed(communityId, feedType, subjectId, pageParam, pageSize, sortParam, q, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

31.0

Signature

```
public static Void setTestSearchFeedElementsInFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, String q, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessages`, `Filter`, `Landing`, `Streams`, and `Topics`.

subjectId
Type: String

If *feedType* is `Record`, *subjectId* can be any record ID, including a group ID. If *feedType* is `UserProfile`, *subjectId* can be any user ID. If the *feedType* is any other value, *subjectId* must be the ID of the context user or the alias `me`.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Order of feed items in the feed.

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

q
Type: String

Search term. Searches keywords in the user or group name. A minimum of one character is required. This parameter doesn't support wildcards. This parameter is required.

result
Type: ConnectApi.FeedElementPage

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedElementsInFeed(communityId, feedType, subjectId, pageParam, pageSize, sortParam, q)

Apex Developer Guide: Testing ConnectApi Code

setTestSearchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

31.0

Signature

```
public static Void setTestSearchFeedElementsInFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, String q, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: ConnectApi.FeedType

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessages`, `Filter`, `Landing`, `Streams`, and `Topics`.

subjectId
Type: String

If *feedType* is `Record`, *subjectId* can be any record ID, including a group ID. If *feedType* is `UserProfile`, *subjectId* can be any user ID. If the *feedType* is any other value, *subjectId* must be the ID of the context user or the alias `me`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q)

Apex Developer Guide: Testing ConnectApi Code

`setTestSearchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, filter, result)`

Register a `ConnectApi.FeedElementPage` object to be returned when `searchFeedElementsInFeed` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

35.0

Signature

```
public static Void setTestSearchFeedElementsInFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, String q, ConnectApi.FeedFilter filter,
ConnectApi.FeedElementPage result)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedType`
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.UserProfile`.

`subjectId`
Type: String

ID of any user. To specify the context user, use the user ID or the alias `me`.

`recentCommentCount`
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

`density`
Type: `ConnectApi.FeedDensity`

The amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

`pageParam`
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

q
Type: String

One or more keywords to search for in the feed elements visible to the context user. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.

filter
Type: `ConnectApi.FeedFilter`

Value must be `ConnectApi.FeedFilter.CommunityScoped`. Filters the feed to include only feed elements that are scoped to Experience Cloud sites. Feed elements that are always visible in all sites are filtered out. Currently, feed elements scoped to sites have a User or a Group parent record. However, other parent record types could be scoped to sites in the future.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, filter)

Apex Developer Guide: Testing ConnectApi Code

setTestSearchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, customFilter, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestSearchFeedElementsInFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, String q, String customFilter,
ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

The ID of a case.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

The amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

q
Type: String

One or more keywords to search for in the feed elements visible to the context user. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.

customFilter
Type: String

Custom filter that applies only to the case feed. See customFeedFilter in the *Metadata API Developer Guide* for supported values.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, customFilter)

Apex Developer Guide: Testing ConnectApi Code

setTestSearchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, showInternalOnly, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

31.0

Signature

```
public static Void setTestSearchFeedElementsInFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
```

```
ConnectApi.FeedSortOrder sortParam, String q, Boolean showInternalOnly,
ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

Any record ID, including a group ID.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

showInternalOnly
Type: Boolean

Specifies whether to show only feed elements from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, showInternalOnly)

Apex Developer Guide: Testing ConnectApi Code

setTestSearchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, showInternalOnly, filter, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

32.0

Signature

```
public static Void setTestSearchFeedElementsInFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, String q, Boolean showInternalOnly,
ConnectApi.FeedFilter filter, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: ConnectApi.FeedType

Value must be ConnectApi.FeedType.Record.

subjectId
Type: String

Any record ID, including a group ID.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: ConnectApi.FeedDensity

Specify the amount of content in a feed.

- AllUpdates—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- FewerUpdates—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, currentPageToken or nextPageToken. If you pass in null, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in null, the default size is 25.

sortParam
Type: ConnectApi.FeedSortOrder

Values are:

- CreatedDateAsc—Sorts by oldest creation date. This sort order is available only for DirectMessageModeration, Draft, Isolated, Moderation, and PendingReview feeds.
- CreatedDateDesc—Sorts by most recent creation date.
- LastModifiedDateDesc—Sorts by most recent activity.
- MostViewed—Sorts by most viewed content. This sort order is available only for Home feeds when the ConnectApi.FeedFilter is UnansweredQuestions.
- Relevance—Sorts by most relevant content. This sort order is available only for Company, Home, and Topics feeds.

If you pass in null, the default value CreatedDateDesc is used.

q
Type: String

Required and can't be null. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

showInternalOnly
Type: Boolean

Specifies whether to show only feed elements from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

`filter`
Type: `ConnectApi.FeedFilter`

Specifies the feed filters.

- `AllQuestions`—Feed elements that are questions.
- `AuthoredBy`—Feed elements authored by the user profile owner. This value is valid only for the `UserProfile` feed.
- `CommunityScoped`—Feed elements that are scoped to Experience Cloud sites. Currently, these feed elements have a User or a Group parent record. However, other parent record types could be scoped to sites in the future. Feed elements that are always visible in all sites are filtered out. This value is valid only for the `UserProfile` feed.
- `QuestionsWithCandidateAnswers`—Feed elements that are questions that have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `QuestionsWithCandidateAnswersReviewedPublished`—Feed elements that are questions that have candidate answers that have been reviewed or published. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Read`—Feed elements that are older than 30 days or are marked as read for the context user. Includes existing feed elements when the context user joined the group. This value is valid only for the `Record` feed of a group.
- `SolvedQuestions`—Feed elements that are questions and that have a best answer.
- `UnansweredQuestions`—Feed elements that are questions and that don't have any answers.
- `UnansweredQuestionsWithCandidateAnswers`—Feed elements that are questions that don't have answers but have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission.
- `Unread`—Feed elements that are created in the past 30 days and aren't marked as read for the context user. This value is valid only for the `Record` feed of a group.
- `UnsolvedQuestions`—Feed elements that are questions and that don't have a best answer.

`result`
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, showInternalOnly, filter)

Apex Developer Guide: Testing ConnectApi Code

setTestSearchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, showInternalOnly, customFilter, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestSearchFeedElementsInFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, String q, Boolean showInternalOnly, String
customFilter, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

The ID of a case.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

showInternalOnly
Type: Boolean

Specifies whether to show only feed elements from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

cusotmFilter
Type: String

Custom filter that applies only to the case feed. See customFeedFilter in the *Metadata API Developer Guide* for supported values.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, showInternalOnly, customFilter)

Apex Developer Guide: Testing ConnectApi Code

`setTestSearchFeedElementsInFilterFeed(communityId, subjectId, keyPrefix, q, result)`

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFilterFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

31.0

Signature

```
public static Void setTestSearchFeedElementsInFilterFeed(String communityId, String subjectId, String keyPrefix, String q, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedElementsInFilterFeed(communityId, subjectId, keyPrefix, q)

Apex Developer Guide: Testing ConnectApi Code

`setTestSearchFeedElementsInFilterFeed(communityId, subjectId, keyPrefix, pageParam, pageSize, sortParam, q, result)`

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFilterFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

31.0

Signature

```
public static Void setTestSearchFeedElementsInFilterFeed(String communityId, String subjectId, String keyPrefix, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, String q, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.

- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedElementsInFilterFeed(communityId, subjectId, keyPrefix, pageParam, pageSize, sortParam, q)

Apex Developer Guide: Testing ConnectApi Code

setTestSearchFeedElementsInFilterFeed(communityId, subjectId, keyPrefix, recentCommentCount, density, pageParam, pageSize, sortParam, q, result)

Register a `ConnectApi.FeedElementPage` object to be returned when the matching `ConnectApi.searchFeedElementsInFilterFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

31.0

Signature

```
public static Void setTestSearchFeedElementsInFilterFeed(String communityId, String subjectId, String keyPrefix, Integer recentCommentCount, ConnectApi.FeedDensity density, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, String q, ConnectApi.FeedElementPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed element. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

If you pass in `null`, the default value `CreatedDateDesc` is used.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

result
Type: `ConnectApi.FeedElementPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedElementsInFilterFeed(communityId, subjectId, keyPrefix, recentCommentCount, density, pageParam, pageSize, sortParam, q)

Apex Developer Guide: Testing ConnectApi Code

`setTestSearchStreams(communityId, q, result)`

Register a `ConnectApi.ChatterStreamPage` object to be returned when the matching `ConnectApi.searchStream(communityId, q)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestSearchStreams(String communityId, String q,
ConnectApi.ChatterStreamPage result)
```

Parameters

communityId

Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q

Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

result

Type: `ConnectApi.ChatterStreamPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchStreams(communityId, q)

Apex Developer Guide: Testing ConnectApi Code

`setTestSearchStreams(communityId, q, sortParam, result)`

Register a `ConnectApi.ChatterStreamPage` object to be returned when the matching `ConnectApi.searchStream(communityId, q, sortParam)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestSearchStreams(String communityId, String q,
ConnectApi.SortOrder sortParam, ConnectApi.ChatterStreamPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

sortParam
Type: `ConnectApi.SortOrder`

Specifies the sort order. Values are:

- `Ascending`—Items are in ascending alphabetical order (A-Z).
- `Descending`—Items are in descending alphabetical order (Z-A).
- `MostRecentlyViewed`—Items are in descending chronological order by view. This sort order is valid only for Chatter feed streams.

If not specified, default value is `Ascending`.

result
Type: `ConnectApi.ChatterStreamPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchStreams(communityId, q, sortParam)

Apex Developer Guide: Testing ConnectApi Code

`setTestSearchStreams(communityId, q, pageParam, pageSize, result)`

Register a `ConnectApi.ChatterStreamPage` object to be returned when the matching `ConnectApi.searchStreams(communityId, q, pageParam, pageSize)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestSearchStreams(String communityId, String q, Integer pageParam,
Integer pageSize, ConnectApi.ChatterStreamPage result)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`q`
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

`pageParam`
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

`pageSize`
Type: Integer

Specifies the number of items per page. Valid values are from 1 to 250. The default size is 25.

`result`
Type: `ConnectApi.ChatterStreamPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchStreams(communityId, q, pageParam, pageSize)

Apex Developer Guide: Testing ConnectApi Code

`setTestSearchStreams(communityId, q, pageParam, pageSize, sortParam, result)`

Register a `ConnectApi.ChatterStreamPage` object to be returned when the matching `ConnectApi.searchStreams(communityId, q, pageParam, pageSize, sortParam)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestSearchStreams(String communityId, String q, Integer pageParam,
Integer pageSize, ConnectApi.SortOrder sortParam, ConnectApi.ChatterStreamPage result)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`q`
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

`pageParam`
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

`pageSize`
Type: Integer

Specifies the number of items per page. Valid values are from 1 to 250. The default size is 25.

`sortParam`
Type: `ConnectApi.SortOrder`

Specifies the sort order. Values are:

- `Ascending`—Items are in ascending alphabetical order (A-Z).
- `Descending`—Items are in descending alphabetical order (Z-A).
- `MostRecentlyViewed`—Items are in descending chronological order by view. This sort order is valid only for Chatter feed streams.

If not specified, default value is `Ascending`.

`result`
Type: `ConnectApi.ChatterStreamPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchStreams(communityId, q, pageParam, pageSize, sortParam)

Apex Developer Guide: Testing ConnectApi Code

setTestSearchStreams(communityId, q, pageParam, pageSize, sortParam, globalScope, result)

Register a `ConnectApi.ChatterStreamPage` object to be returned when the matching `ConnectApi.searchStreams(communityId, q, pageParam, pageSize, sortParam, globalScope)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

41.0

Signature

```
public static Void setTestSearchStreams(String communityId, String q, Integer pageParam,
Integer pageSize, ConnectApi.SortOrder sortParam, Boolean globalScope,
ConnectApi.ChatterStreamPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 to 250. The default size is 25.

sortParam
Type: `ConnectApi.SortOrder`

Specifies the sort order. Values are:

- `Ascending`—Items are in ascending alphabetical order (A-Z).
- `Descending`—Items are in descending alphabetical order (Z-A).

- `MostRecentlyViewed`—Items are in descending chronological order by view. This sort order is valid only for Chatter feed streams.

If not specified, default value is `Ascending`.

`globalScope`
Type: Boolean

Specifies whether to get streams from all the context user's Experience Cloud sites, regardless of the *`communityId`* value.

`result`
Type: `ConnectApi.ChatterStreamPage`

Object containing test data.

Return Value

Type: Void

Retired ChatterFeeds Methods

These methods for `ChatterFeeds` are retired.

IN THIS SECTION:

deleteFeedItem(communityId, feedItemId)
Delete a feed item.

getCommentsForFeedItem(communityId, feedItemId)
Get comments for a feed item.

getCommentsForFeedItem(communityId, feedItemId, pageParam, pageSize)
Get a page of comments for a feed item.

getFeedItem(communityId, feedItemId)
Get a feed item.

getFeedItemBatch(communityId, feedItemIds)
Get a list of feed items.

getFeedItemsFromFeed(communityId, feedType)
Get feed items from the `Company`, `Home`, and `Moderation` feeds.

getFeedItemsFromFeed(communityId, feedType, pageParam, pageSize, sortParam)
Get a page of sorted feed items from the `Company`, `Home`, and `Moderation` feeds.

getFeedItemsFromFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam)
Get a page of sorted feed items from the `Company`, `Home`, and `Moderation` feeds. Each feed item contains no more than the specified number of comments.

getFeedItemsFromFeed(communityId, feedType, subjectId)
Get feed items from a feed for a user or record.

getFeedItemsFromFeed(communityId, feedType, subjectId, pageParam, pageSize, sortParam)
Get a page of sorted feed items from a feed for a user or record.

getFeedItemsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam)

Get a page of sorted feed items from a feed for a user or record. Each feed item includes no more than the specified number of comments.

getFeedItemsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, showInternalOnly)

Get a page of sorted feed items from a record feed for a user or record. Each feed item includes no more than the specified number of comments. Specify whether to return feed items posted by internal (non-Experience Cloud site) users only.

getFeedItemsFromFilterFeed(communityId, subjectId, keyPrefix)

Get feed items from a feed filtered by a key prefix for a user.

getFeedItemsFromFilterFeed(communityId, subjectId, keyPrefix, pageParam, pageSize, sortParam)

Get a page of sorted feed items from a feed filtered by a key prefix for a user.

getFeedItemsFromFilterFeed(communityId, subjectId, keyPrefix, recentCommentCount, density, pageParam, pageSize, sortParam)

Get a page of sorted feed items from a feed filtered by a key prefix for a user. Each feed item contains no more than the specified number of comments.

getFeedItemsFromFilterFeedUpdatedSince(communityId, subjectId, keyPrefix, recentCommentCount, density, pageParam, pageSize, updatedSince)

Get a page of feed items from a feed filtered by a key prefix for a user. Include only feed items that have been updated since the time specified in the *`updatedSince`* parameter.

getFeedItemsUpdatedSince(communityId, feedType, recentCommentCount, density, pageParam, pageSize, updatedSince)

Get a page of feed items from the `Company`, `Home`, and `Moderation` feeds. Include only feed items that have been updated since the time specified in the *`updatedSince`* parameter. Each feed item contains no more than the specified number of comments.

getFeedItemsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, updatedSince)

Get a page of feed items from the `Files`, `Groups`, `News`, `People`, and `Record` feeds. Include only feed items that have been updated since the time specified in the *`updatedSince`* parameter. Each feed item contains no more than the specified number of comments.

getFeedItemsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, updatedSince, showInternalOnly)

Get a page of feed items from a record feed. Include only feed items that have been updated since the time specified in the *`updatedSince`* parameter. Specify whether to return feed items posted by internal (non-Experience Cloud site) users only.

getFeedPoll(communityId, feedItemId)

Get the poll associated with a feed item.

getLikesForFeedItem(communityId, feedItemId)

Get likes for a feed item.

getLikesForFeedItem(communityId, feedItemId, pageParam, pageSize)

Get a page of likes for a feed item.

likeFeedItem(communityId, feedItemId)

Like a feed item for the context user.

postComment(communityId, feedItemId, text)

Post a plain-text comment to a feed item.

postComment(communityId, feedItemId, comment, feedItemFileUpload)

Post a rich-text comment to a feed item. Use this method to include mentions and to attach a file to a comment.

postFeedElement(communityId, feedElement, feedElementFileUpload)

Post a rich-text feed element. Include mentions and hashtag topics, attach a file to a feed element, and associate action link groups with a feed element. You can also use this method to share a feed element and add a comment.

postFeedItem(communityId, feedType, subjectId, text)

Post a plain-text feed item.

postFeedItem(communityId, feedType, subjectId, feedItemInput, feedItemFileUpload)

Post a rich-text feed item to a feed. Use this method to include mentions and hashtag topics and to attach a file to a feed item. You can also use this method to share a feed item and add a comment.

searchFeedItems(communityId, q)

Get the feed items that match the search criteria.

searchFeedItems(communityId, q, sortParam)

Get the sorted feed items that match the search criteria.

searchFeedItems(communityId, q, pageParam, pageSize)

Get a page of feed items that match the search criteria.

searchFeedItems(communityId, q, pageParam, pageSize, sortParam)

Get a page of sorted feed items that match the search criteria.

searchFeedItems(communityId, q, recentCommentCount, pageParam, pageSize, sortParam)

Get a page of sorted feed items that match the search criteria.

searchFeedItemsInFeed(communityId, feedType, q)

Get the feed items from the `Company`, `Home`, and `Moderation` feeds that match the search criteria.

searchFeedItemsInFeed(communityId, feedType, pageParam, pageSize, sortParam, q)

Get a page of sorted feed items from the `Company`, `Home`, and `Moderation` feeds that match the search criteria.

searchFeedItemsInFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, q)

Get a page of sorted feed items from the `Company`, `Home`, and `Moderation` feeds that match the search criteria. Each feed item includes no more than the specified number of comments.

searchFeedItemsInFeed(communityId, feedType, subjectId, q)

Get the feed items from a feed that match the search criteria.

searchFeedItemsInFeed(communityId, feedType, subjectId, pageParam, pageSize, sortParam, q)

Get a page of sorted feed items from a feed for a user or record that match the search criteria.

searchFeedItemsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q)

Get a page of sorted feed items from a feed that match the search criteria. Each feed item includes no more than the specified number of comments.

searchFeedItemsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, showInternalOnly)

Get a page of sorted feed items from a feed for a user or record that match the search criteria. Each feed item includes no more than the specified number of comments. Specify whether to return feed items posted by internal (non-Experience Cloud site) users only.

searchFeedItemsInFilterFeed(communityId, subjectId, keyPrefix, q)

Get the feed items that match the search criteria from a feed filtered by a key prefix for a user.

searchFeedItemsInFilterFeed(communityId, subjectId, keyPrefix, pageParam, pageSize, sortParam, q)

Get a page of sorted feed items that match the search criteria from a feed filtered by a key prefix for a user.

searchFeedItemsInFilterFeed(communityId, subjectId, keyPrefix, recentCommentCount, density, pageParam, pageSize, sortParam, q)

Get a page of sorted feed items that match the search criteria from a feed filtered by a key prefix for a user. Each feed item includes no more than the specified number of comments.

shareFeedElement(communityId, subjectId, feedElementType, originalFeedElementId)

Share the *originalFeedElementId* as the context user.

shareFeedItem(communityId, feedType, subjectId, originalFeedItemId)

Share the *originalFeedItemId* to the feed specified by the *feedType*.

updateBookmark(communityId, feedItemId, isBookmarkedByCurrentUser)

Bookmark a feed item or remove a bookmark from a feed item.

voteOnFeedPoll(communityId, feedItemId, myChoiceId)

Vote or change your vote on a feed poll.

setTestGetFeedItemsFromFeed(communityId, feedType, result)

Register a `ConnectApi.FeedItemPage` object to be returned when `getFeedItemsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

setTestGetFeedItemsFromFeed(communityId, feedType, pageParam, pageSize, sortParam, result)

Register a `ConnectApi.FeedItemPage` object to be returned when `getFeedItemsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

setTestGetFeedItemsFromFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, result)

Register a `ConnectApi.FeedItemPage` object to be returned when `getFeedItemsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

setTestGetFeedItemsFromFeed(communityId, feedType, subjectId, result)

Register a `ConnectApi.FeedItemPage` object to be returned when `getFeedItemsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

setTestGetFeedItemsFromFeed(communityId, feedType, subjectId, pageParam, pageSize, sortParam, result)

Register a `ConnectApi.FeedItemPage` object to be returned when `getFeedItemsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

setTestGetFeedItemsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, result)

Register a `ConnectApi.FeedItemPage` object to be returned when `getFeedItemsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

setTestGetFeedItemsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, showInternalOnly, result)

Register a `ConnectApi.FeedItemPage` object to be returned when `getFeedItemsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

setTestGetFeedItemsFromFilterFeed(communityId, subjectId, keyPrefix, result)

Register a `ConnectApi.FeedItemPage` object to be returned when the matching `getFeedItemsFromFilterFeed` method is called in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetFeedItemsFromFilterFeed(communityId, subjectId, keyPrefix, pageParam, pageSize, sortParam, result)

Register a `ConnectApi.FeedItemPage` object to be returned when the matching `getFeedItemsFromFilterFeed` method is called in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetFeedItemsFromFilterFeed(communityId, subjectId, keyPrefix, recentCommentCount, density, pageParam, pageSize, sortParam, result)

Register a `ConnectApi.FeedItemPage` object to be returned when the matching `getFeedItemsFromFilterFeed` method is called in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetFeedItemsFromFilterFeedUpdatedSince(communityId, subjectId, keyPrefix, recentCommentCount, density, pageParam, pageSize, sortParam, updatedSince, result)

Register a `ConnectApi.FeedItemPage` object to be returned when the `getFeedItemsFromFilterFeedUpdatedSince` method is called in a test context.

setTestGetFeedItemsUpdatedSince(communityId, feedType, recentCommentCount, density, pageParam, pageSize, updatedSince, ConnectApi.FeedItemPage, results)

Register a `ConnectApi.FeedItemPage` object to be returned when `getFeedItemsUpdatedSince` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetFeedItemsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, updatedSince, result)

Register a `ConnectApi.FeedItemPage` object to be returned when `getFeedItemsUpdatedSince` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetFeedItemsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, updatedSince, showInternalOnly, result)

Register a `ConnectApi.FeedItemPage` object to be returned when `getFeedItemsUpdatedSince` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

setTestSearchFeedItems(communityId, q, result)

Register a test feed item page to be returned when `searchFeedItems(communityId, q)` is called during a test.

setTestSearchFeedItems(communityId, q, sortParam, result)

Register a test feed item page to be returned when `searchFeedItems(String, String, ConnectApi.FeedSortOrder)` is called during a test.

setTestSearchFeedItems(communityId, q, pageParam, pageSize, result)

Register a test feed item page to be returned when `searchFeedItems(String, String, String, Integer)` is called during a test.

setTestSearchFeedItems(communityId, q, pageParam, pageSize, sortParam, result)

Register a test feed item page to be returned when `searchFeedItems(String, String, String, Integer, ConnectApi.FeedSortOrder)` is called during a test.

setTestSearchFeedItems(communityId, q, recentCommentCount, pageParam, pageSize, sortParam, result)

Register a test feed item page to be returned when `searchFeedItems(communityId, q, recentCommentCount, pageParam, pageSize, sortParam)` is called during a test.

setTestSearchFeedItemsInFeed(communityId, feedType, q, result)

Register a `ConnectApi.FeedItemPage` object to be returned when the matching `ConnectApi.searchFeedItemsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedItemsInFeed(communityId, feedType, pageParam, pageSize, sortParam, q, result)

Register a `ConnectApi.FeedItemPage` object to be returned when the matching `ConnectApi.searchFeedItemsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedItemsInFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, q, result)

Register a `ConnectApi.FeedItemPage` object to be returned when the matching `ConnectApi.searchFeedItemsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedItemsInFeed(communityId, feedType, subjectId, q, result)

Register a `ConnectApi.FeedItemPage` object to be returned when the matching `ConnectApi.searchFeedItemsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedItemsInFeed(communityId, feedType, subjectId, pageParam, pageSize, sortParam, q, result)

Register a `ConnectApi.FeedItemPage` object to be returned when the matching `ConnectApi.searchFeedItemsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedItemsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, result)

Register a `ConnectApi.FeedItemPage` object to be returned when the matching `ConnectApi.searchFeedItemsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedItemsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, showInternalOnly, result)

Register a `ConnectApi.FeedItemPage` object to be returned when the matching `ConnectApi.searchFeedItemsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedItemsInFilterFeed(communityId, subjectId, keyPrefix, q, result)

Register a `ConnectApi.FeedItemPage` object to be returned when the matching `ConnectApi.searchFeedItemsInFilterFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedItemsInFilterFeed(communityId, feedType, subjectId, keyPrefix, pageParam, pageSize, sortParam, q, result)

Register a `ConnectApi.FeedItemPage` object to be returned when the matching `ConnectApi.searchFeedItemsInFilterFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchFeedItemsInFilterFeed(communityId, feedType, subjectId, keyPrefix, recentCommentCount, density, pageParam, pageSize, sortParam, q, result)

Register a `ConnectApi.FeedItemPage` object to be returned when the matching `ConnectApi.searchFeedItemsInFilterFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

`deleteFeedItem(communityId, feedItemId)`

Delete a feed item.

API Version

28.0–31.0

Important: In version 32.0 and later, use deleteFeedElement(communityId, feedElementId).

Requires Chatter

Yes

Signature

```
public static Void deleteFeedItem(String communityId, String feedItemId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedItemId
Type: String

ID for a feed item.

Return Value

Type: Void

getCommentsForFeedItem(communityId, feedItemId)

Get comments for a feed item.

API Version

28.0–31.0

Important: In version 32.0 and later, use getCommentsForFeedElement(communityId, feedElementId).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.CommentPage getCommentsForFeedItem(String communityId, String
feedItemId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedItemId
Type: String

ID for a feed item.

Return Value

Type: `ConnectApi.CommentPage`

getCommentsForFeedItem(communityId, feedItemId, pageParam, pageSize)

Get a page of comments for a feed item.

API Version

28.0–31.0

Important: In version 32.0 and later, use getCommentsForFeedElement(communityId, feedElementId, pageParam, pageSize).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.CommentPage getCommentsForFeedItem(String communityId, String feedItemId, String pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedItemId
Type: String

ID for a feed item.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.CommentPage`

getFeedItem(communityId, feedItemId)

Get a feed item.

API Version

28.0–31.0

Important: In version 32.0 and later, use getFeedElement(communityId, feedElementId).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItem getFeedItem(String communityId, String feedItemId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedItemId
Type: String

ID for a feed item.

Return Value

Type: `ConnectApi.FeedItem`

Note: Triggers on FeedItem objects run before their attachment and capabilities information is saved, which means that `ConnectApi.FeedItem.attachment` information and `ConnectApi.FeedElement.capabilities` information may not be available in the trigger.

getFeedItemBatch(communityId, feedItemIds)

Get a list of feed items.

API Version

31.0–31.0

Important: In version 32.0 and later, use getFeedElementBatch(communityId, feedElementIds).

Requires Chatter

Yes

Signature

```
public static ConnectApi.BatchResult[] getFeedItemBatch(String communityId, List<String>
feedItemIds)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedItemIds
Type: List<String>

A list of up to 500 feed item IDs.

Return Value

Type: `ConnectApi.BatchResult`[]

The `ConnectApi.BatchResult.getResult()` method returns a `ConnectApi.FeedItem` object and errors for feed items that didn't load.

Example

```
// Create a list of feed items.
ConnectApi.FeedItemPage feedItemPage = ConnectApi.ChatterFeeds.getFeedItemsFromFeed(null,
 ConnectApi.FeedType.Company);
System.debug(feedItemPage);

// Create a list of feed item IDs.
List<String> feedItemIds = new List<String>();
for (ConnectApi.FeedItem aFeedItem : feedItemPage.items){
    feedItemIds.add(aFeedItem.id);
}

// Get info about the feed items in the list.
ConnectApi.BatchResult[] batchResults = ConnectApi.ChatterFeeds.getFeedItemBatch(null,
feedItemIds);

for (ConnectApi.BatchResult batchResult : batchResults) {
    if (batchResult.isSuccess()) {
        // Operation was successful.
        // Print the header for each feed item.
        ConnectApi.FeedItem aFeedItem;
        if(batchResult.getResult() instanceof ConnectApi.FeedItem) {
           aFeedItem = (ConnectApi.FeedItem) batchResult.getResult();
```

```
        }
        System.debug('SUCCESS');
        System.debug(aFeedItem.header.text);
    }
    else {
        // Operation failed. Print errors.
        System.debug('FAILURE');
        System.debug(batchResult.getErrorMessage());
    }
}
```

getFeedItemsFromFeed(communityId, feedType)

Get feed items from the `Company`, `Home`, and `Moderation` feeds.

API Version

28.0–31.0

Important: In version 32.0 and later, use getFeedElementsFromFeed(communityId, feedType).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage getFeedItemsFromFeed(String communityId,
ConnectApi.FeedType feedType)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedItemsFromFeed(communityId, feedType, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedItemsFromFeed(communityId, feedType, pageParam, pageSize, sortParam)`

Get a page of sorted feed items from the `Company`, `Home`, and `Moderation` feeds.

API Version

28.0–31.0

Important: In version 32.0 and later, use getFeedElementsFromFeed(communityId, feedType, pageParam, pageSize, sortParam).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage getFeedItemsFromFeed(String communityId,
ConnectApi.FeedType feedType, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedType`
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

`pageParam`
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

`pageSize`
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedItemsFromFeed(communityId, feedType, pageParam, pageSize, sortParam, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedItemsFromFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam)`

Get a page of sorted feed items from the `Company`, `Home`, and `Moderation` feeds. Each feed item contains no more than the specified number of comments.

API Version

29.0–31.0

Important: In version 32.0 and later, use getFeedElementsFromFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage getFeedItemsFromFeed(String communityId,
ConnectApi.FeedType feedType, Integer recentCommentCount, ConnectApi.FeedDensity density,
String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedItemsFromFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedItemsFromFeed(communityId, feedType, subjectId)`

Get feed items from a feed for a user or record.

API Version

28.0–31.0

Important: In version 32.0 and later, use getFeedElementsFromFeed(communityId, feedType, subjectId).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage getFeedItemsFromFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId)
```

Parameters

`communityId`

Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedType`

Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessageModeration`, `DirectMessages`, `Filter`, `Home`, `Isolated`, `Landing`, `Moderation`, and `PendingReview`.

subjectId
Type: String

If *feedType* is `Record`, *subjectId* can be any record ID, including a group ID. If *feedType* is `Streams`, *subjectId* must be a stream ID. If *feedType* is `Topics`, *subjectId* must be a topic ID. If *feedType* is `UserProfile`, *subjectId* can be any user ID. If the *feedType* is any other value, *subjectId* must be the ID of the context user or the alias `me`.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedItemsFromFeed(communityId, feedType, subjectId, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedItemsFromFeed(communityId, feedType, subjectId, pageParam, pageSize, sortParam)`

Get a page of sorted feed items from a feed for a user or record.

API Version

28.0–31.0

Important: In version 32.0 and later, use getFeedElementsFromFeed(communityId, feedType, subjectId, pageParam, pageSize, sortParam).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage getFeedItemsFromFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedType`
Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessageModeration`, `DirectMessages`, `Filter`, `Home`, `Isolated`, `Landing`, `Moderation`, and `PendingReview`.

`subjectId`
Type: String

If *`feedType`* is `Record`, *`subjectId`* can be any record ID, including a group ID. If *`feedType`* is `Streams`, *`subjectId`* must be a stream ID. If *`feedType`* is `Topics`, *`subjectId`* must be a topic ID. If *`feedType`* is `UserProfile`, *`subjectId`* can be any user ID. If the *`feedType`* is any other value, *`subjectId`* must be the ID of the context user or the alias `me`.

`pageParam`
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

`pageSize`
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

`sortParam`
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedItemsFromFeed(communityId, feedType, subjectId, pageParam, pageSize, sortParam, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedItemsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam)`

Get a page of sorted feed items from a feed for a user or record. Each feed item includes no more than the specified number of comments.

API Version

29.0–31.0

Important: In version 32.0 and later, use getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage getFeedItemsFromFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessageModeration`, `DirectMessages`, `Filter`, `Home`, `Isolated`, `Landing`, `Moderation`, and `PendingReview`.

subjectId
Type: String

If *feedType* is `Record`, *subjectId* can be any record ID, including a group ID. If *feedType* is `Streams`, *subjectId* must be a stream ID. If *feedType* is `Topics`, *subjectId* must be a topic ID. If *feedType* is `UserProfile`, *subjectId* can be any user ID. If the *feedType* is any other value, *subjectId* must be the ID of the context user or the alias `me`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedItemsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedItemsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, showInternalOnly)`

Get a page of sorted feed items from a record feed for a user or record. Each feed item includes no more than the specified number of comments. Specify whether to return feed items posted by internal (non-Experience Cloud site) users only.

API Version

30.0–31.0

Important: In version 32.0 and later, use getFeedElementsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, showInternalOnly).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage getFeedItemsFromFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, Boolean showInternalOnly)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

Any record ID, including a group ID.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

`pageSize`
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

`sortParam`
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

`showInternalOnly`
Type: Boolean

Specifies whether to show only feed items from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedItemsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, showInternalOnly, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedItemsFromFilterFeed(communityId, subjectId, keyPrefix)`

Get feed items from a feed filtered by a key prefix for a user.

API Version

28.0–31.0

Important: In version 32.0 and later, use getFeedElementsFromFilterFeed(communityId, subjectId, keyPrefix).

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage getFeedItemsFromFilterFeed(String communityId,
String subjectId, String keyPrefix)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedItemsFromFilterFeed(communityId, subjectId, keyPrefix, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedItemsFromFilterFeed(communityId, subjectId, keyPrefix, pageParam, pageSize, sortParam)`

Get a page of sorted feed items from a feed filtered by a key prefix for a user.

API Version

28.0–31.0

Important: In version 32.0 and later, use getFeedElementsFromFilterFeed(communityId, subjectId, keyPrefix, pageParam, pageSize, sortParam).

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage getFeedItemsFromFilterFeed(String communityId,
String subjectId, String keyPrefix, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedItemsFromFilterFeed(communityId, subjectId, keyPrefix, pageParam, pageSize, sortParam, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedItemsFromFilterFeed(communityId, subjectId, keyPrefix, recentCommentCount, density, pageParam, pageSize, sortParam)`

Get a page of sorted feed items from a feed filtered by a key prefix for a user. Each feed item contains no more than the specified number of comments.

API Version

29.0–31.0

Important: In version 32.0 and later, use getFeedElementsFromFilterFeed(communityId, subjectId, keyPrefix, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, sortParam).

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage getFeedItemsFromFilterFeed(String communityId, String subjectId, String keyPrefix, Integer recentCommentCount, ConnectApi.FeedDensity density, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`subjectId`
Type: String

ID of the context user or the alias `me`.

`keyPrefix`
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

`recentCommentCount`
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedItemsFromFilterFeed(communityId, subjectId, keyPrefix, recentCommentCount, density, pageParam, pageSize, sortParam, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedItemsFromFilterFeedUpdatedSince(communityId, subjectId, keyPrefix, recentCommentCount, density, pageParam, pageSize, updatedSince)`

Get a page of feed items from a feed filtered by a key prefix for a user. Include only feed items that have been updated since the time specified in the *updatedSince* parameter.

API Version

30.0–31.0

Important: In version 32.0 and later, use getFeedElementsFromFilterFeedUpdatedSince(communityId, subjectId, keyPrefix, recentCommentCount, elementsPerBundle, density, pageParam, pageSize, updatedSince).

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage getFeedItemsFromFilterFeedUpdatedSince(String communityId, String subjectId, String keyPrefix, Integer recentCommentCount, ConnectApi.FeedDensity density, String pageParam, Integer pageSize, String updatedSince)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

updatedSince
Type: String

Opaque token containing information about the last modified date of the feed. Do not construct this token. To retrieve this token, call `getFeedItemsFromFilterFeed` and take the value from the `updatesToken` property of the `ConnectApi.FeedItemPage` response body.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

This method returns only feed items that have been updated since the time specified in the *updatedSince* argument. A feed item is considered to be updated if it was created since the last feed request, or if `sort=LastModifiedDateDesc` and a comment was added to the feed item since the last feed request. Adding likes and topics doesn't update a feed item.

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFeedItemsFromFilterFeedUpdatedSince(communityId, subjectId, keyPrefix, recentCommentCount, density, pageParam, pageSize, sortParam, updatedSince, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedItemsUpdatedSince(communityId, feedType, recentCommentCount, density, pageParam, pageSize, updatedSince)`

Get a page of feed items from the `Company`, `Home`, and `Moderation` feeds. Include only feed items that have been updated since the time specified in the *updatedSince* parameter. Each feed item contains no more than the specified number of comments.

API Version

30.0–31.0

Important: In version 32.0 and later, use getFeedElementsUpdatedSince(communityId, feedType, recentCommentCount, density, pageParam, pageSize, updatedSince).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage getFeedItemsUpdatedSince(String communityId,
ConnectApi.FeedType feedType, Integer recentCommentCount, ConnectApi.FeedDensity density,
String pageParam, Integer pageSize, String updatedSince)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

updatedSince
Type: String

An opaque token containing information about the last modified date of the feed. Do not construct this token. Retrieve this token from the `updatesToken` property of the `ConnectApi.FeedItemPage` response body.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

This method returns only feed items that have been updated since the time specified in the *updatedSince* argument. A feed item is considered to be updated if it was created since the last feed request, or if `sort=LastModifiedDateDesc` and a comment was added to the feed item since the last feed request. Adding likes and topics doesn't update a feed item.

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

Example

This example gets the feed items in the company feed and grabs the `updatesToken` property from the returned object. It then passes the value of `updatesToken` to the `getFeedItemsUpdatedSince` method to get the feed items updated since the first call:

```
// Get the feed items in the company feed and return the updatesToken
String communityId = null;

// Get the feed and extract the update token
ConnectApi.FeedItemPage page = ConnectApi.ChatterFeeds.getFeedItemsFromFeed(communityId,
ConnectApi.FeedType.Company);

// page.updatesToken is opaque and has a value like '2:1384549034000'

// Get the feed items that changed since the provided updatesToken
ConnectApi.FeedItemPage feedItems= ConnectApi.ChatterFeeds.getFeedItemsUpdatedSince
  (communityId, ConnectApi.FeedType.Company, 1, ConnectApi.FeedDensity.AllUpdates, null,
1, page.updatesToken);
```

SEE ALSO:

setTestGetFeedItemsUpdatedSince(communityId, feedType, recentCommentCount, density, pageParam, pageSize, updatedSince, ConnectApi.FeedItemPage, results)

Apex Developer Guide: Testing ConnectApi Code

`getFeedItemsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, updatedSince)`

Get a page of feed items from the `Files`, `Groups`, `News`, `People`, and `Record` feeds. Include only feed items that have been updated since the time specified in the *updatedSince* parameter. Each feed item contains no more than the specified number of comments.

API Version

30.0–31.0

Important: In version 32.0 and later, use getFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, updatedSince).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage getFeedItemsUpdatedSince(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize, String updatedSince)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

One of these values:

- `Files`
- `Groups`
- `News`
- `People`
- `Record`

subjectId
Type: String

If *feedType* is `ConnectApi.Record`, *subjectId* can be any record ID, including a group ID. Otherwise, it must be the context user or the alias `me`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

updatedSince
Type: String

An opaque token containing information about the last modified date of the feed. Do not construct this token. Retrieve this token from the `updatesToken` property of the `ConnectApi.FeedItemPage` response body.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

This method returns only feed items that have been updated since the time specified in the *updatedSince* argument. A feed item is considered to be updated if it was created since the last feed request, or if `sort=LastModifiedDateDesc` and a comment was added to the feed item since the last feed request. Adding likes and topics doesn't update a feed item.

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

Example

This example gets the feed items in the news feed and grabs the `updatesToken` property from the returned object. It then passes the value of `updatesToken` to the `getFeedItemsUpdatedSince` method to get the feed items updated since the first call:

```
// Get the feed items in the news feed and return the updatesToken
String communityId = null;
String subjectId = 'me';

// Get the feed and extract the update token
ConnectApi.FeedItemPage page = ConnectApi.ChatterFeeds.getFeedItemsFromFeed(communityId,
ConnectApi.FeedType.News, subjectId);

// page.updatesToken is opaque and has a value like '2:1384549034000'

// Get the feed items that changed since the provided updatesToken
ConnectApi.FeedItemPage feedItems= ConnectApi.ChatterFeeds.getFeedItemsUpdatedSince
  (communityId, ConnectApi.FeedType.News, subjectId, 1, ConnectApi.FeedDensity.AllUpdates,
 null, 1, page.updatesToken);
```

SEE ALSO:

setTestGetFeedItemsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, updatedSince, result)

Apex Developer Guide: Testing ConnectApi Code

getFeedItemsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, updatedSince, showInternalOnly)

Get a page of feed items from a record feed. Include only feed items that have been updated since the time specified in the *updatedSince* parameter. Specify whether to return feed items posted by internal (non-Experience Cloud site) users only.

API Version

30.0–31.0

Important: In version 32.0 and later, use getFeedElementsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, updatedSince, showInternalOnly).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage getFeedItemsUpdatedSince(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize, String updatedSince,
Boolean showInternalOnly)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

Any record ID, including a group ID.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

`pageSize`
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

`updatedSince`
Type: String

An opaque token containing information about the last modified date of the feed. Do not construct this token. Retrieve this token from the `updatesToken` property of the `ConnectApi.FeedItemPage` response body.

`showInternalOnly`
Type: Boolean

Specifies whether to show only feed items from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

This method returns only feed items that have been updated since the time specified in the *`updatedSince`* argument. A feed item is considered to be updated if it was created since the last feed request, or if `sort=LastModifiedDateDesc` and a comment was added to the feed item since the last feed request. Adding likes and topics doesn't update a feed item.

If *`showInternalOnly`* is `true` and digital experiences is enabled, feed items from Experience Cloud sites are included. Otherwise, only feed items from the internal site are included.

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

Example

This example gets the feed items in the news feed and grabs the `updatesToken` property from the returned object. It then passes the value of `updatesToken` to the `getFeedItemsUpdatedSince` method to get the feed items updated since the first call:

```
// Get the feed items in the news feed and return the updatesToken
String communityId = null;
String subjectId = 'me';

// Get the feed and extract the update token
ConnectApi.FeedItemPage page = ConnectApi.ChatterFeeds.getFeedItemsFromFeed(communityId,
ConnectApi.FeedType.News, subjectId);

// page.updatesToken is opaque and has a value like '2:1384549034000'

// Get the feed items that changed since the provided updatesToken
ConnectApi.FeedItemPage feedItems= ConnectApi.ChatterFeeds.getFeedItemsUpdatedSince
```

```
  (communityId, ConnectApi.FeedType.News, subjectId, 1, ConnectApi.FeedDensity.AllUpdates,
null, 1, page.updatesToken, true);
```

SEE ALSO:

setTestGetFeedItemsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, updatedSince, showInternalOnly, result)

Apex Developer Guide: Testing ConnectApi Code

`getFeedPoll(communityId, feedItemId)`

Get the poll associated with a feed item.

API Version

28.0–31.0

Important: In version 32.0 and later, use getFeedElementPoll(communityId, feedElementId).

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedPoll getFeedPoll(String communityId, String feedItemId)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedItemId`
Type: String

ID for a feed item.

Return Value

Type: `ConnectApi.FeedPoll`

Note: Triggers on FeedItem objects run before their attachment and capabilities information is saved, which means that `ConnectApi.FeedItem.attachment` information and `ConnectApi.FeedElement.capabilities` information may not be available in the trigger.

`getLikesForFeedItem(communityId, feedItemId)`

Get likes for a feed item.

API Version

28.0–31.0

Important: In version 32.0 and later, use getLikesForFeedElement(communityId, feedElementId).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterLikePage getLikesForFeedItem(String communityId, String feedItemId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedItemId
Type: String

ID for a feed item.

Return Value

Type: `ConnectApi.ChatterLikePage`

getLikesForFeedItem(communityId, feedItemId, pageParam, pageSize)

Get a page of likes for a feed item.

API Version

28.0–31.0

Important: In version 32.0 and later, use getLikesForFeedElement(communityId, feedElementId, pageParam, pageSize).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterLikePage getLikesForFeedItem(String communityId, String feedItemId, Integer pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedItemId
Type: String

ID for a feed item.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.ChatterLikePage`

likeFeedItem(communityId, feedItemId)

Like a feed item for the context user.

API Version

28.0–31.0

Important: In version 32.0 and later, use likeFeedElement(communityId, feedElementId).

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterLike likeFeedItem(String communityId, String feedItemId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedItemId
Type: String

ID for a feed item.

Return Value

Type: `ConnectApi.ChatterLike`

If the context user already liked the feed item, this method is a non-operation and returns the existing like.

postComment(communityId, feedItemId, text)

Post a plain-text comment to a feed item.

API Version

28.0–31.0

Important: In version 32.0 and later, use postCommentToFeedElement(communityId, feedElementId, text).

Requires Chatter

Yes

Signature

```
public static ConnectApi.Comment postComment(String communityId, String feedItemId,
String text)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedItemId
Type: String

ID for a feed item.

text
Type: String

The text of the comment. Mentions are downgraded to plain text. To include a mention that links to a user, call `postComment(communityId, feedItemId, comment, feedItemFileUpload)` and pass the mention in a `ConnectApi.CommentInput` object.

Return Value

Type: `ConnectApi.Comment`

Usage

If hashtags or links are detected in *text*, they're included in the comment as hashtag and link segments. Mentions aren't detected in *text* and aren't separated out of the text.

Feed items and comments can contain up to 10,000 characters.

postComment(communityId, feedItemId, comment, feedItemFileUpload)

Post a rich-text comment to a feed item. Use this method to include mentions and to attach a file to a comment.

API Version

28.0–31.0

Important: In version 32.0 and later, use postCommentToFeedElement(communityId, feedElementId, comment, feedElementFileUpload).

Requires Chatter

Yes

Signature

```
public static ConnectApi.Comment postComment(String communityId, String feedItemId,
ConnectApi.CommentInput comment, ConnectApi.BinaryInput feedItemFileUpload)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedItemId
Type: String

ID for a feed item.

comment
Type: `ConnectApi.CommentInput`

In the `CommentInput` object, specify rich text, including @mentions. Optionally, in the `CommentInput.attachment` property, specify an existing file or a new file

feedItemFileUpload
Type: `ConnectApi.BinaryInput`

If you specify a `NewFileAttachmentInput` object in the `CommentInput.attachment` property, specify the new binary file to attach in this argument. Otherwise, do not specify a value.

Return Value

Type: `ConnectApi.Comment`

Usage

Feed items and comments can contain up to 10,000 characters.

Sample: Posting a Comment with a New File Attachment

To post a comment and upload and attach a new file to the comment, create a `ConnectApi.CommentInput` object and a `ConnectApi.BinaryInput` object to pass to the `ConnectApi.ChatterFeeds.postComment` method.

```
String communityId = null;
String feedItemId = '0D5D0000000Kcd1';

ConnectApi.CommentInput input = new ConnectApi.CommentInput();
ConnectApi.MessageBodyInput messageInput = new ConnectApi.MessageBodyInput();
ConnectApi.TextSegmentInput textSegment;

textSegment = new ConnectApi.TextSegmentInput();
textSegment.text = 'Comment Text Body';

messageInput.messageSegments = new List<ConnectApi.MessageSegmentInput>();
messageInput.messageSegments.add(textSegment);

input.body = messageInput;

ConnectApi.NewFileAttachmentInput attachmentInput = new ConnectApi.NewFileAttachmentInput();
attachmentInput.description = 'The description of the file';
attachmentInput.title = 'contentFile.txt';
input.attachment = attachmentInput;

String fileContents = 'This is the content of the file.';
Blob fileBlob = Blob.valueOf(fileContents);
ConnectApi.BinaryInput binaryInput = new ConnectApi.BinaryInput(fileBlob, 'text/plain',
'contentFile.txt');

ConnectApi.Comment commentRep = ConnectApi.ChatterFeeds.postComment(communityId, feedItemId,
 input, binaryInput);
```

`postFeedElement(communityId, feedElement, feedElementFileUpload)`

Post a rich-text feed element. Include mentions and hashtag topics, attach a file to a feed element, and associate action link groups with a feed element. You can also use this method to share a feed element and add a comment.

API Version

31.0–35.0

Important: In version 36.0 and later, this method is no longer available because you can't create a feed post and upload a binary file in the same call. Upload files to Salesforce first, and then use `postFeedElement(communityId, feedElement)` to create the feed post and attach the files.

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElement postFeedElement(String communityId,
ConnectApi.FeedElementInput feedElement, ConnectApi.BinaryInput feedElementFileUpload)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElement
Type: `ConnectApi.FeedElementInput`

Specify rich text, including mentions. Optionally, specify a link, a poll, an existing file, or a new file.

feedElementFileUpload
Type: `ConnectApi.BinaryInput`

Specify the new binary file to attach to the post only if you also specify a `NewFileAttachmentInput` object in the *feedElement* parameter. Otherwise, pass `null`.

Return Value

Type: `ConnectApi.FeedElement`

Example for Posting a Feed Element with a New (Binary) File

```
ConnectApi.FeedItemInput input = new ConnectApi.FeedItemInput();
input.subjectId = 'me';

ConnectApi.ContentCapabilityInput contentInput = new ConnectApi.ContentCapabilityInput();
contentInput.title = 'Title';

ConnectApi.FeedElementCapabilitiesInput capabilities = new
ConnectApi.FeedElementCapabilitiesInput();
capabilities.content = contentInput;

input.capabilities = capabilities;

String text = 'These are the contents of the new file.';
Blob myBlob = Blob.valueOf(text);
ConnectApi.BinaryInput binInput = new ConnectApi.BinaryInput(myBlob, 'text/plain',
'fileName');

ConnectApi.ChatterFeeds.postFeedElement(Network.getNetworkId(), input, binInput);
```

postFeedItem(communityId, feedType, subjectId, text)

Post a plain-text feed item.

API Version

28.0–31.0

Important: In version 32.0 and later, use postFeedElement(communityId, subjectId, feedElementType, text).

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItem postFeedItem(String communityId, ConnectApi.FeedType feedType, String subjectId, String text)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

One of the following:

- `News`
- `Record`
- `UserProfile`

Use `Record` to post to a group.

subjectId
Type: String

The value depends on the *feedType*:

- `News`—ID of the context user or the keyword `me`.
- `Record`—ID of any record with a feed, including groups.
- `UserProfile`—ID of any user.

text
Type: String

Text of the feed item. Mentions are downgraded to plain text. To include a mention that links to the user, call the `postFeedItem(communityId, feedType, subjectId, feedItemInput, feedItemFileUpload)` method and pass the mention in a `ConnectApi.FeedItemInput` object.

Return Value

Type: `ConnectApi.FeedItem`

Note: Triggers on FeedItem objects run before their attachment and capabilities information is saved, which means that `ConnectApi.FeedItem.attachment` information and `ConnectApi.FeedElement.capabilities` information may not be available in the trigger.

Usage

Feed items and comments can contain up to 10,000 characters.

Posts to `ConnectApi.FeedType.UserProfile` in API versions 23.0 and 24.0 created user status updates, not feed items. For posts to the User Profile Feed in those API versions, the character limit is 1,000 characters.

`postFeedItem(communityId, feedType, subjectId, feedItemInput, feedItemFileUpload)`

Post a rich-text feed item to a feed. Use this method to include mentions and hashtag topics and to attach a file to a feed item. You can also use this method to share a feed item and add a comment.

API Version

28.0–31.0

Important: In version 32.0 and later, use postFeedElement(communityId, feedElement, feedElementFileUpload).

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItem postFeedItem(String communityId, ConnectApi.FeedType
feedType, String subjectId, ConnectApi.FeedItemInput feedItemInput,
ConnectApi.BinaryInput feedItemFileUpload)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedType`
Type: `ConnectApi.FeedType`

One of the following:

- `News`
- `Record`
- `UserProfile`

To post a feed item to a group, use `Record` and use a group ID as the *`subjectId`*.

`subjectId`
Type: String

If *`feedType`* is `Record`, *`subjectId`* can be any record ID, including a group ID. If *`feedType`* is `Streams`, *`subjectId`* must be a stream ID. If *`feedType`* is `Topics`, *`subjectId`* must be a topic ID. If *`feedType`* is `UserProfile`, *`subjectId`* can be any user ID. If the *`feedType`* is any other value, *`subjectId`* must be the ID of the context user or the alias `me`.

`feedItemInput`
Type: `ConnectApi.FeedItemInput`

In the `FeedItemInput` object, specify rich text. Optionally, in the `FeedItemInput.attachment` property, specify a link, a poll, an existing file, or a new file.

`feedItemFileUpload`
Type: `ConnectApi.BinaryInput`

If you specify a `NewFileAttachmentInput` object in the `FeedItemInput.attachment` property, specify the new binary file to attach in this argument. Otherwise, do not specify a value.

Return Value

Type: `ConnectApi.FeedItem`

Note: Triggers on FeedItem objects run before their attachment and capabilities information is saved, which means that `ConnectApi.FeedItem.attachment` information and `ConnectApi.FeedElement.capabilities` information may not be available in the trigger.

Usage

Feed items and comments can contain up to 10,000 characters. Posts to `ConnectApi.FeedType.UserProfile` in API versions 23.0 and 24.0 created user status updates, not feed items. For posts to the User Profile Feed in those API versions, the character limit is 1,000 characters.

Example for Sharing a Feed Item and Adding a Comment

To share a feed item and add a comment, create a `ConnectApi.FeedItemInput` object containing the comment and the feed item to share. Then pass the object to `ConnectApi.ChatterFeeds.postFeeditem` in the *`feedItemInput`* argument. The message segments in the message body input are used as the comment.

```
ConnectApi.FeedItemInput input = new ConnectApi.FeedItemInput();
input.originalFeedItemId = '0D5D0000000JuAG';

ConnectApi.MessageBodyInput body = new ConnectApi.MessageBodyInput();
List<ConnectApi.MessageSegmentInput> segmentList = new
List<ConnectApi.MessageSegmentInput>();
ConnectApi.TextSegmentInput textSegment = new ConnectApi.TextSegmentInput();
textSegment.text = 'I hope you enjoy this post I found in another group.';
segmentList.add((ConnectApi.MessageSegmentInput)textSegment);
body.messageSegments = segmentList;
input.body = body;

ConnectApi.ChatterFeeds.postFeedItem(null, ConnectApi.FeedType.UserProfile, 'me', input,
null);
```

Example for Posting a Mention to a User Profile Feed

To post to a user profile feed and include an @mention, call the `ConnectApi.ChatterFeeds.postFeedItem` method.

```
String communityId = null;
ConnectApi.FeedType feedType = ConnectApi.FeedType.UserProfile;

ConnectApi.FeedItemInput input = new ConnectApi.FeedItemInput();
ConnectApi.MessageBodyInput messageInput = new ConnectApi.MessageBodyInput();
ConnectApi.TextSegmentInput textSegment;
```

```
ConnectApi.MentionSegmentInput mentionSegment = new ConnectApi.MentionSegmentInput();

messageInput.messageSegments = new List<ConnectApi.MessageSegmentInput>();

textSegment = new ConnectApi.TextSegmentInput();
textSegment.text = 'Hey there ';
messageInput.messageSegments.add(textSegment);

mentionSegment.id = '005D0000001LLO1';
messageInput.messageSegments.add(mentionSegment);

textSegment = new ConnectApi.TextSegmentInput();
textSegment.text = '. How are you?';
messageInput.messageSegments.add(textSegment);

input.body = messageInput;

ConnectApi.FeedItem feedItemRep = ConnectApi.ChatterFeeds.postFeedItem(communityId, feedType,
 'me', input, null);
```

`searchFeedItems(communityId, q)`

Get the feed items that match the search criteria.

API Version

28.0–31.0

Important: In version 32.0 and later, use searchFeedElements(communityId, q).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage searchFeedItems(String communityId, String q)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedItems(communityId, q, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedItems(communityId, q, sortParam)`

Get the sorted feed items that match the search criteria.

API Version

28.0–31.0

Important: In version 32.0 and later, use searchFeedElements(communityId, q, sortParam).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage searchFeedItems(String communityId, String q,
ConnectApi.FeedSortOrder sortParam)
```

Parameters

communityId

Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q

Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

sortParam

Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedItems(communityId, q, sortParam, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedItems(communityId, q, pageParam, pageSize)`

Get a page of feed items that match the search criteria.

API Version

28.0–31.0

Important: In version 32.0 and later, use searchFeedElements(communityId, q, pageParam, pageSize).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage searchFeedItems(String communityId, String q,
String pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedItems(communityId, q, pageParam, pageSize, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedItems(communityId, q, pageParam, pageSize, sortParam)`

Get a page of sorted feed items that match the search criteria.

API Version

28.0–31.0

Important: In version 32.0 and later, use searchFeedElements(communityId, q, pageParam, pageSize, sortParam).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage searchFeedItems(String communityId, String q,
String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedItems(communityId, q, pageParam, pageSize, sortParam, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedItems(communityId, q, recentCommentCount, pageParam, pageSize, sortParam)`

Get a page of sorted feed items that match the search criteria.

API Version

29.0–31.0

Important: In version 32.0 and later, use searchFeedElements(communityId, q, recentCommentCount, pageParam, pageSize, sortParam).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage searchFeedItems(String communityId, String q,
Integer recentCommentCount, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder
sortParam)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`q`
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

`recentCommentCount`
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedItems(communityId, q, recentCommentCount, pageParam, pageSize, sortParam, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedItemsInFeed(communityId, feedType, q)`

Get the feed items from the `Company`, `Home`, and `Moderation` feeds that match the search criteria.

API Version

28.0–31.0

Important: In version 32.0 and later, use searchFeedElementsInFeed(communityId, feedType, q).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage searchFeedItemsInFeed(String communityId,
ConnectApi.FeedType feedType, String q)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedItemsInFeed(communityId, feedType, q, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedItemsInFeed(communityId, feedType, pageParam, pageSize, sortParam, q)`

Get a page of sorted feed items from the `Company`, `Home`, and `Moderation` feeds that match the search criteria.

API Version

28.0–31.0

Important: In version 32.0 and later, use searchFeedElementsInFeed(communityId, feedType, pageParam, pageSize, sortParam, q).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage searchFeedItemsInFeed(String communityId,
ConnectApi.FeedType feedType, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, String q)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

q

Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedItemsInFeed(communityId, feedType, pageParam, pageSize, sortParam, q, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedItemsInFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, q)`

Get a page of sorted feed items from the `Company`, `Home`, and `Moderation` feeds that match the search criteria. Each feed item includes no more than the specified number of comments.

API Version

29.0–31.0

Important: In version 32.0 and later, use searchFeedElementsInFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, q).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage searchFeedItemsInFeed(String communityId,
ConnectApi.FeedType feedType, Integer recentCommentCount, ConnectApi.FeedDensity density,
String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, String q)
```

Parameters

communityId

Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedItemsInFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, q, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedItemsInFeed(communityId, feedType, subjectId, q)`

Get the feed items from a feed that match the search criteria.

API Version

28.0–31.0

Important: In version 32.0 and later, use searchFeedElementsInFeed(communityId, feedType, subjectId, q).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage searchFeedItemsInFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, String q)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedType`
Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessages`, `Filter`, `Landing`, and `Streams`.

`subjectId`
Type: String

If *`feedType`* is `Record`, *`subjectId`* can be any record ID, including a group ID. If feed type is `UserProfile`, *`subjectId`* can be any user ID. If *`feedType`* is any other value, *`subjectId`* must be the ID of the context user or the alias `me`.

`q`
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedItemsInFeed(communityId, feedType, subjectId, q, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedItemsInFeed(communityId, feedType, subjectId, pageParam, pageSize, sortParam, q)`

Get a page of sorted feed items from a feed for a user or record that match the search criteria.

API Version

28.0–31.0

Important: In version 32.0 and later, use searchFeedElementsInFeed(communityId, feedType, subjectId, pageParam, pageSize, sortParam, q).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage searchFeedItemsInFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, String q)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedType`
Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessages`, `Filter`, `Landing`, and `Streams`.

`subjectId`
Type: String

If *`feedType`* is `Record`, *`subjectId`* can be any record ID, including a group ID. If *`feedType`* is `Streams`, *`subjectId`* must be a stream ID. If *`feedType`* is `Topics`, *`subjectId`* must be a topic ID. If *`feedType`* is `UserProfile`, *`subjectId`* can be any user ID. If the *`feedType`* is any other value, *`subjectId`* must be the ID of the context user or the alias `me`.

`pageParam`
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

`pageSize`
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

`sortParam`
Type: `ConnectApi.FeedSortOrder`

Order of feed items in the feed.

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

`q`
Type: String

Search term. Searches keywords in the user or group name. A minimum of one character is required. This parameter doesn't support wildcards. This parameter is required.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedItemsInFeed(communityId, feedType, subjectId, pageParam, pageSize, sortParam, q, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedItemsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q)`

Get a page of sorted feed items from a feed that match the search criteria. Each feed item includes no more than the specified number of comments.

API Version

29.0–31.0

Important: In version 32.0 and later, use searchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage searchFeedItemsInFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, String q)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessages`, `Filter`, `Landing`, and `Streams`.

subjectId
Type: String

If *`feedType`* is `Record`, *`subjectId`* can be any record ID, including a group ID. If *`feedType`* is `Streams`, *`subjectId`* must be a stream ID. If *`feedType`* is `Topics`, *`subjectId`* must be a topic ID. If *`feedType`* is `UserProfile`, *`subjectId`* can be any user ID. If the *`feedType`* is any other value, *`subjectId`* must be the ID of the context user or the alias `me`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedItemsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedItemsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, showInternalOnly)`

Get a page of sorted feed items from a feed for a user or record that match the search criteria. Each feed item includes no more than the specified number of comments. Specify whether to return feed items posted by internal (non-Experience Cloud site) users only.

API Version

30.0–31.0

Important: In version 32.0 and later, use searchFeedElementsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, showInternalOnly).

Available to Guest Users

31.0 only

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage searchFeedItemsInFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, String q, Boolean showInternalOnly)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Value must be `ConnectApi.FeedType.Record`.

subjectId
Type: String

Any record ID, including a group ID.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.

- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

`pageParam`
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

`pageSize`
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

`sortParam`
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

`q`
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

`showInternalOnly`
Type: Boolean

Specifies whether to show only feed items from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedItemsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, showInternalOnly, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedItemsInFilterFeed(communityId, subjectId, keyPrefix, q)`

Get the feed items that match the search criteria from a feed filtered by a key prefix for a user.

API Version

28.0–31.0

Important: In version 32.0 and later, use searchFeedElementsInFilterFeed(communityId, subjectId, keyPrefix, q).

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage searchFeedItemsInFilterFeed(String communityId,
String subjectId, String keyPrefix, String q)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedItemsInFilterFeed(communityId, subjectId, keyPrefix, q, result)

Apex Developer Guide: Testing ConnectApi Code

searchFeedItemsInFilterFeed(communityId, subjectId, keyPrefix, pageParam, pageSize, sortParam, q)

Get a page of sorted feed items that match the search criteria from a feed filtered by a key prefix for a user.

API Version

28.0–31.0

Important: In version 32.0 and later, use searchFeedElementsInFilterFeed(communityId, subjectId, keyPrefix, pageParam, pageSize, sortParam, q).

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage searchFeedItemsInFilterFeed(String communityId,
String subjectId, String keyPrefix, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, String q)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedItemsInFilterFeed(communityId, feedType, subjectId, keyPrefix, pageParam, pageSize, sortParam, q, result)

Apex Developer Guide: Testing ConnectApi Code

`searchFeedItemsInFilterFeed(communityId, subjectId, keyPrefix, recentCommentCount, density, pageParam, pageSize, sortParam, q)`

Get a page of sorted feed items that match the search criteria from a feed filtered by a key prefix for a user. Each feed item includes no more than the specified number of comments.

API Version

29.0–31.0

Important: In version 32.0 and later, use searchFeedElementsInFilterFeed(communityId, subjectId, keyPrefix, recentCommentCount, density, pageParam, pageSize, sortParam, q).

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItemPage searchFeedItemsInFilterFeed(String communityId,
String subjectId, String keyPrefix, Integer recentCommentCount, ConnectApi.FeedDensity
density, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, String
q)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

q

Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.FeedItemPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchFeedItemsInFilterFeed(communityId, feedType, subjectId, keyPrefix, recentCommentCount, density, pageParam, pageSize, sortParam, q, result)

Apex Developer Guide: Testing ConnectApi Code

`shareFeedElement(communityId, subjectId, feedElementType, originalFeedElementId)`

Share the *`originalFeedElementId`* as the context user.

API Version

31.0–38.0

Important: In version 39.0 and later, use `postFeedElement(communityId, feedElement)` or `updateFeedElement(communityId, feedElementId, feedElement)` with the `ConnectApi.FeedEntityShareCapabilityInput` to share a feed entity with a feed element.

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedElement shareFeedElement(String communityId, String
subjectId, ConnectApi.FeedElementType feedElementType, String originalFeedElementId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

The ID of the user or group with whom to share the feed element.

feedElementType
Type: `ConnectApi.FeedElementType`

Values are:

- `Bundle`—A container of feed elements. A bundle also has a body made up of message segments that can always be gracefully degraded to text-only values.
- `FeedItem`—A feed item has a single parent and is scoped to oneExperience Cloud site or across all Experience Cloud sites. A feed item can have capabilities such as bookmarks, canvas, content, comment, link, poll. Feed items have a body made up of message segments that can always be gracefully degraded to text-only values.
- `Recommendation`—A recommendation is a feed element with a recommendations capability. A recommendation suggests records to follow, groups to join, or applications that are helpful to the context user.

originalFeedElementId
Type: String

The ID of the feed element to share.

Return Value

Type: `ConnectApi.FeedElement`

Example

```
ConnectApi.ChatterFeeds.shareFeedElement(null, '0F9RR0000004CPw',
ConnectApi.FeedElementType.FeedItem, '0D5RR0000004Gxc');
```

shareFeedItem(communityId, feedType, subjectId, originalFeedItemId)

Share the *originalFeedItemId* to the feed specified by the *feedType*.

API Version

28.0–31.0

Important:

- In version 32.0–38.0, use `shareFeedElement(communityId, subjectId, feedElementType, originalFeedElementId)`.

- In version 39.0 and later, use `postFeedElement(communityId, feedElement)` or `updateFeedElement(communityId, feedElementId, feedElement)` with the `ConnectApi.FeedEntityShareCapabilityInput`.

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItem shareFeedItem(String communityId, ConnectApi.FeedType
feedType, String subjectId, String originalFeedItemId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

One of the following:

- `News`
- `Record`
- `UserProfile`

To share a feed item with a group, use `Record` and use a group ID as the *subjectId*.

subjectId
Type: String

The value depends on the value of *feedType*:

- `News`—*subjectId* must be the ID of the context user or the keyword `me`.
- `Record`—*subjectId* can be a group ID or the ID of the context user (or `me`).
- `UserProfile`—*subjectId* can be any user ID.

originalFeedItemId
Type: String

The ID of the feed item to share.

Return Value

Type: `ConnectApi.FeedItem`

Example

To share a feed item with a group, pass in the Experience Cloud site ID (or `null`), the feed type `Record`, the group ID, and the ID of the feed item to share.

```
ConnectApi.ChatterFeeds.shareFeedItem(null, ConnectApi.FeedType.Record, '0F9D00000000izf',
 '0D5D0000000JuAG');
```

updateBookmark(communityId, feedItemId, isBookmarkedByCurrentUser)

Bookmark a feed item or remove a bookmark from a feed item.

API Version

28.0–31.0

Important: In version 32.0 and later, use updateFeedElementBookmarks(communityId, feedElementId, bookmarks) or updateFeedElementBookmarks(communityId, feedElementId, isBookmarkedByCurrentUser).

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedItem updateBookmark(String communityId, String feedItemId,
Boolean isBookmarkedByCurrentUser)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedItemId
Type: String

ID for a feed item.

isBookmarkedByCurrentUser
Type: Boolean

—Specifying `true` adds the feed item to the list of bookmarks for the context user. Specify `false` to remove a bookmark.

Return Value

Type: `ConnectApi.FeedItem`

voteOnFeedPoll(communityId, feedItemId, myChoiceId)

Vote or change your vote on a feed poll.

API Version

28.0–31.0

Important: In version 32.0 and later, use voteOnFeedElementPoll(communityId, feedElementId, myChoiceId).

Requires Chatter

Yes

Signature

```
public static ConnectApi.FeedPoll voteOnFeedPoll(String communityId, String feedItemId,
String myChoiceId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedItemId
Type: String

ID of the feed item that is associated with the poll.

myChoiceId
Type: String

ID of the item in the poll you're voting for.

Return Value

Type: `ConnectApi.FeedPoll`

setTestGetFeedItemsFromFeed(communityId, feedType, result)

Register a `ConnectApi.FeedItemPage` object to be returned when `getFeedItemsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

API Version

28.0–31.0

Signature

```
public static Void setTestGetFeedItemsFromFeed(String communityId, ConnectApi.FeedType
feedType, ConnectApi.FeedItemPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

result
Type: `ConnectApi.FeedItemPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedItemsFromFeed(communityId, feedType)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedItemsFromFeed(communityId, feedType, pageParam, pageSize, sortParam, result)

Register a `ConnectApi.FeedItemPage` object to be returned when `getFeedItemsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

API Version

28.0–31.0

Signature

```
public static Void setTestGetFeedItemsFromFeed(String communityId, ConnectApi.FeedType feedType, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, ConnectApi.FeedItemPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

result
Type: `ConnectApi.FeedItemPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedItemsFromFeed(communityId, feedType, pageParam, pageSize, sortParam)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedItemsFromFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, result)

Register a `ConnectApi.FeedItemPage` object to be returned when `getFeedItemsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

API Version

29.0–31.0

Signature

```
public static Void setTestGetFeedItemsFromFeed(String communityId, ConnectApi.FeedType feedType, Integer recentCommentCount, ConnectApi.FeedDensity density, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, ConnectApi.FeedItemPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

result
Type: `ConnectApi.FeedItemPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedItemsFromFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedItemsFromFeed(communityId, feedType, subjectId, result)

Register a `ConnectApi.FeedItemPage` object to be returned when `getFeedItemsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

API Version

28.0–31.0

Signature

```
public static Void setTestGetFeedItemsFromFeed(String communityId, ConnectApi.FeedType feedType, String subjectId, ConnectApi.FeedItemPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessageModeration`, `DirectMessages`, `Filter`, `Home`, `Isolated`, `Landing`, `Moderation`, and `PendingReview`.

subjectId
Type: String

If *feedType* is `Record`, *subjectId* can be any record ID, including a group ID. If *feedType* is `Streams`, *subjectId* must be a stream ID. If *feedType* is `Topics`, *subjectId* must be a topic ID. If *feedType* is `UserProfile`, *subjectId* can be any user ID. If the *feedType* is any other value, *subjectId* must be the ID of the context user or the alias `me`.

result
Type: `ConnectApi.FeedItemPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedItemsFromFeed(communityId, feedType, subjectId)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedItemsFromFeed(communityId, feedType, subjectId, pageParam, pageSize, sortParam, result)

Register a `ConnectApi.FeedItemPage` object to be returned when `getFeedItemsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

API Version

28.0–31.0

Signature

```
public static Void setTestGetFeedItemsFromFeed(String communityId, ConnectApi.FeedType feedType, String subjectId, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, ConnectApi.FeedItemPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessageModeration`, `DirectMessages`, `Filter`, `Home`, `Isolated`, `Landing`, `Moderation`, and `PendingReview`.

subjectId
Type: String

If *feedType* is `Record`, *subjectId* can be any record ID, including a group ID. If *feedType* is `Streams`, *subjectId* must be a stream ID. If *feedType* is `Topics`, *subjectId* must be a topic ID. If *feedType* is `UserProfile`, *subjectId* can be any user ID. If the *feedType* is any other value, *subjectId* must be the ID of the context user or the alias `me`.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: ConnectApi.FeedSortOrder

Values are:

- CreatedDateAsc—Sorts by oldest creation date. This sort order is available only for DirectMessageModeration, Draft, Isolated, Moderation, and PendingReview feeds.
- CreatedDateDesc—Sorts by most recent creation date.
- LastModifiedDateDesc—Sorts by most recent activity.
- MostViewed—Sorts by most viewed content. This sort order is available only for Home feeds when the ConnectApi.FeedFilter is UnansweredQuestions.
- Relevance—Sorts by most relevant content. This sort order is available only for Company, Home, and Topics feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in null, the default value CreatedDateDesc is used.

result
Type: ConnectApi.FeedItemPage

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedItemsFromFeed(communityId, feedType, subjectId, pageParam, pageSize, sortParam)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedItemsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, result)

Register a ConnectApi.FeedItemPage object to be returned when getFeedItemsFromFeed is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

API Version

29.0–31.0

Signature

```
public static Void setTestGetFeedItemsFromFeed(String communityId, ConnectApi.FeedType
feedType, String subjectId, Integer recentCommentCount, ConnectApi.FeedDensity density,
String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam,
ConnectApi.FeedItemPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessageModeration`, `DirectMessages`, `Filter`, `Home`, `Isolated`, `Landing`, `Moderation`, and `PendingReview`.

subjectId
Type: String

If *feedType* is `Record`, *subjectId* can be any record ID, including a group ID. If *feedType* is `Streams`, *subjectId* must be a stream ID. If *feedType* is `Topics`, *subjectId* must be a topic ID. If *feedType* is `UserProfile`, *subjectId* can be any user ID. If the *feedType* is any other value, *subjectId* must be the ID of the context user or the alias `me`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

result
Type: `ConnectApi.FeedItemPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedItemsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedItemsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, showInternalOnly, result)

Register a `ConnectApi.FeedItemPage` object to be returned when `getFeedItemsFromFeed` is called with matching parameters in a test context. Use the get feed method with the same parameters or the code throws an exception.

API Version

30.0–31.0

Signature

```
public static Void setTestGetFeedItemsFromFeed(String communityId, ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount, ConnectApi.FeedDensity density, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, Boolean showInternalOnly, ConnectApi.FeedItemPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessageModeration`, `DirectMessages`, `Filter`, `Home`, `Isolated`, `Landing`, `Moderation`, and `PendingReview`.

subjectId
Type: String

If *feedType* is `Record`, *subjectId* can be any record ID, including a group ID. If *feedType* is `Streams`, *subjectId* must be a stream ID. If *feedType* is `Topics`, *subjectId* must be a topic ID. If *feedType* is `UserProfile`, *subjectId* can be any user ID. If the *feedType* is any other value, *subjectId* must be the ID of the context user or the alias `me`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

showInternalOnly
Type: Boolean

Specifies whether to show only feed items from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

result
Type: `ConnectApi.FeedItemPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedItemsFromFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, showInternalOnly)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedItemsFromFilterFeed(communityId, subjectId, keyPrefix, result)

Register a `ConnectApi.FeedItemPage` object to be returned when the matching `getFeedItemsFromFilterFeed` method is called in a test context. Use the method with the same parameters or the code throws an exception.

API Version

28.0–31.0

Signature

```
public static Void setTestGetFeedItemsFromFilterFeed(String communityId, String subjectId, String keyPrefix, ConnectApi.FeedItemPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

result
Type: `ConnectApi.FeedItemPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedItemsFromFilterFeed(communityId, subjectId, keyPrefix)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedItemsFromFilterFeed(communityId, subjectId, keyPrefix, pageParam, pageSize, sortParam, result)

Register a `ConnectApi.FeedItemPage` object to be returned when the matching `getFeedItemsFromFilterFeed` method is called in a test context. Use the method with the same parameters or the code throws an exception.

API Version

28.0–31.0

Signature

```
public static Void setTestGetFeedItemsFromFilterFeed(String communityId, String subjectId, String keyPrefix, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, ConnectApi.FeedItemPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

result
Type: `ConnectApi.FeedItemPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedItemsFromFilterFeed(communityId, subjectId, keyPrefix, pageParam, pageSize, sortParam)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedItemsFromFilterFeed(communityId, subjectId, keyPrefix, recentCommentCount, density, pageParam, pageSize, sortParam, result)

Register a `ConnectApi.FeedItemPage` object to be returned when the matching `getFeedItemsFromFilterFeed` method is called in a test context. Use the method with the same parameters or the code throws an exception.

API Version

29.0–31.0

Signature

```
public static Void setTestGetFeedItemsFromFilterFeed(String communityId, String subjectId, String keyPrefix, Integer recentCommentCount, ConnectApi.FeedDensity density, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, ConnectApi.FeedItemPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.

- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

`pageParam`
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

`pageSize`
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

`sortParam`
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

`result`
Type: `ConnectApi.FeedItemPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedItemsFromFilterFeed(communityId, subjectId, keyPrefix, recentCommentCount, density, pageParam, pageSize, sortParam)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetFeedItemsFromFilterFeedUpdatedSince(communityId, subjectId, keyPrefix, recentCommentCount, density, pageParam, pageSize, sortParam, updatedSince, result)`

Register a `ConnectApi.FeedItemPage` object to be returned when the `getFeedItemsFromFilterFeedUpdatedSince` method is called in a test context.

API Version

30.0–31.0

Signature

```
public static Void setTestGetFeedItemsFromFilterFeedUpdatedSince(String communityId,
String subjectId, String keyPrefix, Integer recentCommentCount, ConnectApi.FeedDensity
density, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, String
updatedSince, ConnectApi.FeedItemPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.

- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

`updatedSince`
Type: String

Opaque token containing information about the last modified date of the feed. Do not construct this token. To retrieve this token, call `getFeedItemsFromFilterFeed` and take the value from the `updatesToken` property of the `ConnectApi.FeedItemPage` response body.

`result`
Type: `ConnectApi.FeedItemPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedItemsFromFilterFeedUpdatedSince(communityId, subjectId, keyPrefix, recentCommentCount, density, pageParam, pageSize, updatedSince)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetFeedItemsUpdatedSince(communityId, feedType, recentCommentCount, density, pageParam, pageSize, updatedSince, ConnectApi.FeedItemPage, results)`

Register a `ConnectApi.FeedItemPage` object to be returned when `getFeedItemsUpdatedSince` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

30.0–31.0

Signature

```
public static Void setTestGetFeedItemsUpdatedSince(String communityId,
ConnectApi.FeedType feedType, Integer recentCommentCount, ConnectApi.FeedDensity density,
String pageParam, Integer pageSize, String updatedSince, ConnectApi.FeedItemPage results)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedType`
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

updatedSince
Type: String

An opaque token containing information about the last modified date of the feed. Do not construct this token. Retrieve this token from the `updatesToken` property of the `ConnectApi.FeedItemPage` response body.

result
Type: `ConnectApi.FeedItemPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedItemsUpdatedSince(communityId, feedType, recentCommentCount, density, pageParam, pageSize, updatedSince)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedItemsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, updatedSince, result)

Register a `ConnectApi.FeedItemPage` object to be returned when `getFeedItemsUpdatedSince` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

30.0–31.0

Signature

```
public static Void setTestGetFeedItemsUpdatedSince(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize, String updatedSince,
ConnectApi.FeedItemPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

One of these values:

- `Files`
- `Groups`
- `News`
- `People`
- `Record`

subjectId
Type: String

If *feedType* is `ConnectApi.Record`, *subjectId* can be any record ID, including a group ID. Otherwise, it must be the context user or the alias `me`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

updatedSince
Type: String

An opaque token containing information about the last modified date of the feed. Do not construct this token. Retrieve this token from the `updatesToken` property of the `ConnectApi.FeedItemPage` response body.

result
Type: `ConnectApi.FeedItemPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedItemsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, updatedSince)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFeedItemsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, updatedSince, showInternalOnly, result)

Register a `ConnectApi.FeedItemPage` object to be returned when `getFeedItemsUpdatedSince` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

30.0–31.0

Signature

```
public static Void setTestGetFeedItemsUpdatedSince(String communityId,
ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount,
ConnectApi.FeedDensity density, String pageParam, Integer pageSize, String updatedSince,
Boolean showInternalOnly, ConnectApi.FeedItemPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

One of these values:

- `Files`
- `Groups`
- `News`
- `People`
- `Record`

subjectId
Type: String

If *feedType* is `ConnectApi.Record`, *subjectId* can be any record ID, including a group ID. Otherwise, it must be the context user or the alias `me`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

updatedSince
Type: String

An opaque token containing information about the last modified date of the feed. Do not construct this token. Retrieve this token from the `updatesToken` property of the `ConnectApi.FeedItemPage` response body.

showInternalOnly
Type: Boolean

Specifies whether to show only feed items from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

result
Type: `ConnectApi.FeedItemPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFeedItemsUpdatedSince(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, updatedSince, showInternalOnly)

Apex Developer Guide: Testing ConnectApi Code

setTestSearchFeedItems(communityId, q, result)

Register a test feed item page to be returned when `searchFeedItems(communityId, q)` is called during a test.

API Version

28.0–31.0

Signature

```
public static Void searchFeedItems(String communityId, String q, ConnectApi.FeedItemPage result)
```

Parameters

communityId
: Type: String

  ID for an Experience Cloud site, `internal`, or `null`.

q
: Type: String

  Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

result
: Type: `ConnectApi.FeedItemPage`

  Object containing test data.

Return Value

Type: Void

SEE ALSO:

- searchFeedItems(communityId, q)
- *Apex Developer Guide*: Testing ConnectApi Code

setTestSearchFeedItems(communityId, q, sortParam, result)

Register a test feed item page to be returned when `searchFeedItems(String, String, ConnectApi.FeedSortOrder)` is called during a test.

API Version

28.0–31.0

Signature

```
public static Void setTestSearchFeedItems(String communityId, String q, ConnectApi.FeedSortOrder sortParam, ConnectApi.FeedItemPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

result
Type: `ConnectApi.FeedItemPage`

The feed item test page.

Return Value

Type: Void

SEE ALSO:

searchFeedItems(communityId, q, sortParam)

Apex Developer Guide: Testing ConnectApi Code

`setTestSearchFeedItems(communityId, q, pageParam, pageSize, result)`

Register a test feed item page to be returned when `searchFeedItems(String, String, String, Integer)` is called during a test.

API Version

28.0–31.0

Signature

```
public static Void setTestSearchFeedItems(String communityId, String q, String pageParam,
Integer pageSize, ConnectApi.FeedItemPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

result
Type: `ConnectApi.FeedItemPage`

The test feed item page.

Return Value

Type: Void

SEE ALSO:

searchFeedItems(communityId, q, pageParam, pageSize)

Apex Developer Guide: Testing ConnectApi Code

`setTestSearchFeedItems(communityId, q, pageParam, pageSize, sortParam, result)`

Register a test feed item page to be returned when `searchFeedItems(String, String, String, Integer, ConnectApi.FeedSortOrder)` is called during a test.

API Version

28.0–31.0

Signature

```
public static Void setTestSearchFeedItems(String communityId, String q, String pageParam,
Integer pageSize, ConnectApi.FeedSortOrder sortParam, ConnectApi.FeedItemPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

result
Type: `ConnectApi.FeedItemPage`

The test feed item page.

Return Value

Type: Void

SEE ALSO:

searchFeedItems(communityId, q, pageParam, pageSize, sortParam)

Apex Developer Guide: Testing ConnectApi Code

`setTestSearchFeedItems(communityId, q, recentCommentCount, pageParam, pageSize, sortParam, result)`

Register a test feed item page to be returned when `searchFeedItems(communityId, q, recentCommentCount, pageParam, pageSize, sortParam)` is called during a test.

API Version

29.0–31.0

Signature

```
public static Void setTestSearchFeedItems(String communityId, String q, Integer recentCommentCount, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, ConnectApi.FeedItemPage result)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`q`
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

`recentCommentCount`
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

`pageParam`
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

`pageSize`
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

`sortParam`
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

result
Type: `ConnectApi.FeedItemPage`

The test feed item page.

Return Value

Type: Void

SEE ALSO:

searchFeedItems(communityId, q, recentCommentCount, pageParam, pageSize, sortParam)

Apex Developer Guide: Testing ConnectApi Code

setTestSearchFeedItemsInFeed(communityId, feedType, q, result)

Register a `ConnectApi.FeedItemPage` object to be returned when the matching `ConnectApi.searchFeedItemsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

28.0–31.0

Signature

```
public static Void setTestSearchFeedItemsInFeed(String communityId, ConnectApi.FeedType feedType, String q, ConnectApi.FeedItemPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values are `Company`, `DirectMessageModeration`, `DirectMessages`, `Home`, `Isolated`, `Moderation`, and `PendingReview`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

result
Type: `ConnectApi.FeedItemPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedItemsInFeed(communityId, feedType, q)

Apex Developer Guide: Testing ConnectApi Code

setTestSearchFeedItemsInFeed(communityId, feedType, pageParam, pageSize, sortParam, q, result)

Register a `ConnectApi.FeedItemPage` object to be returned when the matching `ConnectApi.searchFeedItemsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

28.0–31.0

Signature

```
public static Void setTestSearchFeedItemsInFeed(String communityId, ConnectApi.FeedType feedType, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, String q, ConnectApi.FeedItemPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessageModeration`, `DirectMessages`, `Filter`, `Home`, `Isolated`, `Landing`, `Moderation`, and `PendingReview`.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.

- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

q

Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

result

Type: `ConnectApi.FeedItemPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedItemsInFeed(communityId, feedType, pageParam, pageSize, sortParam, q)

Apex Developer Guide: Testing ConnectApi Code

setTestSearchFeedItemsInFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, q, result)

Register a `ConnectApi.FeedItemPage` object to be returned when the matching `ConnectApi.searchFeedItemsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

29.0–31.0

Signature

```
public static Void setTestSearchFeedItemsInFeed(String communityId, ConnectApi.FeedType feedType, Integer recentCommentCount, ConnectApi.FeedDensity density, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, String q, ConnectApi.FeedItemPage result)
```

Parameters

communityId

Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: ConnectApi.FeedType

Type of feed. Valid values include every ConnectApi.FeedType except Company, DirectMessageModeration, DirectMessages, Filter, Home, Isolated, Landing, Moderation, and PendingReview.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: ConnectApi.FeedDensity

Specify the amount of content in a feed.

- AllUpdates—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- FewerUpdates—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, currentPageToken or nextPageToken. If you pass in null, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in null, the default size is 25.

sortParam
Type: ConnectApi.FeedSortOrder

Values are:

- CreatedDateAsc—Sorts by oldest creation date. This sort order is available only for DirectMessageModeration, Draft, Isolated, Moderation, and PendingReview feeds.
- CreatedDateDesc—Sorts by most recent creation date.
- LastModifiedDateDesc—Sorts by most recent activity.
- MostViewed—Sorts by most viewed content. This sort order is available only for Home feeds when the ConnectApi.FeedFilter is UnansweredQuestions.
- Relevance—Sorts by most relevant content. This sort order is available only for Company, Home, and Topics feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in null, the default value CreatedDateDesc is used.

q
Type: String

Required and can't be null. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

result
Type: ConnectApi.FeedItemPage

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedItemsInFeed(communityId, feedType, recentCommentCount, density, pageParam, pageSize, sortParam, q)

Apex Developer Guide: Testing ConnectApi Code

setTestSearchFeedItemsInFeed(communityId, feedType, subjectId, q, result)

Register a `ConnectApi.FeedItemPage` object to be returned when the matching `ConnectApi.searchFeedItemsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

28.0–31.0

Signature

```
public static Void setTestSearchFeedItemsInFeed(String communityId, ConnectApi.FeedType feedType, String subjectId, String q, ConnectApi.FeedItemPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessages`, `Filter`, `Landing`, and `Streams`.

subjectId
Type: String

If *feedType* is `Record`, *subjectId* can be any record ID, including a group ID. If *feedType* is `Streams`, *subjectId* must be a stream ID. If *feedType* is `Topics`, *subjectId* must be a topic ID. If *feedType* is `UserProfile`, *subjectId* can be any user ID. If the *feedType* is any other value, *subjectId* must be the ID of the context user or the alias `me`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

result
Type: `ConnectApi.FeedItemPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedItemsInFeed(communityId, feedType, subjectId, q)

Apex Developer Guide: Testing ConnectApi Code

setTestSearchFeedItemsInFeed(communityId, feedType, subjectId, pageParam, pageSize, sortParam, q, result)

Register a `ConnectApi.FeedItemPage` object to be returned when the matching `ConnectApi.searchFeedItemsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

28.0–31.0

Signature

```
public static Void setTestSearchFeedItemsInFeed(String communityId, ConnectApi.FeedType feedType, String subjectId, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, String q, ConnectApi.FeedItemPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessages`, `Filter`, `Landing`, and `Streams`.

subjectId
Type: String

If *`feedType`* is `Record`, *`subjectId`* can be any record ID, including a group ID. If *`feedType`* is `Streams`, *`subjectId`* must be a stream ID. If *`feedType`* is `Topics`, *`subjectId`* must be a topic ID. If *`feedType`* is `UserProfile`, *`subjectId`* can be any user ID. If the *`feedType`* is any other value, *`subjectId`* must be the ID of the context user or the alias `me`.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: ConnectApi.FeedSortOrder

Values are:

- CreatedDateAsc—Sorts by oldest creation date. This sort order is available only for DirectMessageModeration, Draft, Isolated, Moderation, and PendingReview feeds.
- CreatedDateDesc—Sorts by most recent creation date.
- LastModifiedDateDesc—Sorts by most recent activity.
- MostViewed—Sorts by most viewed content. This sort order is available only for Home feeds when the ConnectApi.FeedFilter is UnansweredQuestions.
- Relevance—Sorts by most relevant content. This sort order is available only for Company, Home, and Topics feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in null, the default value CreatedDateDesc is used.

q
Type: String

Required and can't be null. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

result
Type: ConnectApi.FeedItemPage

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedItemsInFeed(communityId, feedType, subjectId, pageParam, pageSize, sortParam, q)

Apex Developer Guide: Testing ConnectApi Code

setTestSearchFeedItemsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, result)

Register a ConnectApi.FeedItemPage object to be returned when the matching ConnectApi.searchFeedItemsInFeed method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

29.0–31.0

Signature

```
public static Void setTestSearchFeedItemsInFeed(String communityId, ConnectApi.FeedType
feedType, String subjectId, Integer recentCommentCount, ConnectApi.FeedDensity density,
String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, String q,
ConnectApi.FeedItemPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessages`, `Filter`, `Landing`, and `Streams`.

subjectId
Type: String

If *feedType* is `Record`, *subjectId* can be any record ID, including a group ID. If *feedType* is `Streams`, *subjectId* must be a stream ID. If *feedType* is `Topics`, *subjectId* must be a topic ID. If *feedType* is `UserProfile`, *subjectId* can be any user ID. If the *feedType* is any other value, *subjectId* must be the ID of the context user or the alias `me`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

q

Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

result

Type: `ConnectApi.FeedItemPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedItemsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q)

Apex Developer Guide: Testing ConnectApi Code

setTestSearchFeedItemsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, showInternalOnly, result)

Register a `ConnectApi.FeedItemPage` object to be returned when the matching `ConnectApi.searchFeedItemsInFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

29.0–31.0

Signature

```
public static Void setTestSearchFeedItemsInFeed(String communityId, ConnectApi.FeedType feedType, String subjectId, Integer recentCommentCount, ConnectApi.FeedDensity density, String pageParam, Integer pageSize, ConnectApi.FeedSortOrder sortParam, String q, Boolean showInternalOnly, ConnectApi.FeedItemPage result)
```

Parameters

communityId

Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType

Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessages`, `Filter`, `Landing`, and `Streams`.

subjectId
Type: String

If *feedType* is `Record`, *subjectId* can be any record ID, including a group ID. If *feedType* is `Streams`, *subjectId* must be a stream ID. If *feedType* is `Topics`, *subjectId* must be a topic ID. If *feedType* is `UserProfile`, *subjectId* can be any user ID. If the *feedType* is any other value, *subjectId* must be the ID of the context user or the alias `me`.

recentCommentCount
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

density
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

showInternalOnly
Type: Boolean

Specifies whether to show only feed items from internal (non-Experience Cloud site) users (`true`), or not (`false`). The default value is `false`.

result
Type: `ConnectApi.FeedItemPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchFeedItemsInFeed(communityId, feedType, subjectId, recentCommentCount, density, pageParam, pageSize, sortParam, q, showInternalOnly)

Apex Developer Guide: Testing ConnectApi Code

setTestSearchFeedItemsInFilterFeed(communityId, subjectId, keyPrefix, q, result)

Register a `ConnectApi.FeedItemPage` object to be returned when the matching `ConnectApi.searchFeedItemsInFilterFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

28.0–31.0

Signature

```
public static Void setTestSearchFeedItemsInFilterFeed(String communityId, String subjectId, String keyPrefix, String q, ConnectApi.FeedItemPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

result
Type: `ConnectApi.FeedItemPage`

Specify the test feed item page.

Return Value

Type: Void

SEE ALSO:

searchFeedItemsInFilterFeed(communityId, subjectId, keyPrefix, q)

Apex Developer Guide: Testing ConnectApi Code

setTestSearchFeedItemsInFilterFeed(communityId, feedType, subjectId, keyPrefix, pageParam, pageSize, sortParam, q, result)

Register a `ConnectApi.FeedItemPage` object to be returned when the matching `ConnectApi.searchFeedItemsInFilterFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

28.0–31.0

Signature

```
public static Void setTestSearchFeedItemsInFilterFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, String keyPrefix, String pageParam,
Integer pageSize, ConnectApi.FeedSortOrder sortParam, String q, ConnectApi.FeedItemPage
result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedType
Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessageModeration`, `DirectMessages`, `Filter`, `Home`, `Isolated`, `Landing`, `Moderation`, and `PendingReview`.

subjectId
Type: String

ID of the context user or the alias `me`.

keyPrefix
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

result
Type: `ConnectApi.FeedItemPage`

Specify the test feed item page.

Return Value

Type: Void

SEE ALSO:

searchFeedItemsInFilterFeed(communityId, subjectId, keyPrefix, pageParam, pageSize, sortParam, q)

Apex Developer Guide: Testing ConnectApi Code

`setTestSearchFeedItemsInFilterFeed(communityId, feedType, subjectId, keyPrefix, recentCommentCount, density, pageParam, pageSize, sortParam, q, result)`

Register a `ConnectApi.FeedItemPage` object to be returned when the matching `ConnectApi.searchFeedItemsInFilterFeed` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

29.0–31.0

Signature

```
public static Void setTestSearchFeedItemsInFilterFeed(String communityId,
ConnectApi.FeedType feedType, String subjectId, String keyPrefix, Integer
recentCommentCount, ConnectApi.FeedDensity density, String pageParam, Integer pageSize,
ConnectApi.FeedSortOrder sortParam, String q, ConnectApi.FeedItemPage result)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedType`
Type: `ConnectApi.FeedType`

Type of feed. Valid values include every `ConnectApi.FeedType` except `Company`, `DirectMessageModeration`, `DirectMessages`, `Filter`, `Home`, `Isolated`, `Landing`, `Moderation`, and `PendingReview`.

`subjectId`
Type: String

ID of the context user or the alias `me`.

`keyPrefix`
Type: String

A key prefix that specifies record type. A key prefix is the first three characters in the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

`recentCommentCount`
Type: Integer

Maximum number of comments to return with each feed item. The default value is 3.

`density`
Type: `ConnectApi.FeedDensity`

Specify the amount of content in a feed.

- `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations.
- `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Number of feed items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.FeedSortOrder`

Values are:

- `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `LastModifiedDateDesc`—Sorts by most recent activity.
- `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`.
- `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.

Sorts the returned feed by the most recently created feed item, or by the most recently modified feed item. If you pass in `null`, the default value `CreatedDateDesc` is used.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

result
Type: `ConnectApi.FeedItemPage`

Specify the test feed item page.

Return Value

Type: Void

SEE ALSO:

searchFeedItemsInFilterFeed(communityId, subjectId, keyPrefix, recentCommentCount, density, pageParam, pageSize, sortParam, q)

Apex Developer Guide: Testing ConnectApi Code

ChatterGroups Class

Information about groups, such as the group's members, photo, and the groups the specified user is a member of. Add members to a group, remove members, and change the group photo.

Namespace

ConnectApi

ChatterGroups Methods

These methods are for `ChatterGroups`. All methods are static.

All methods in this class require Chatter and are subject to the per user, per namespace, per hour rate limit.

IN THIS SECTION:

addMember(communityId, groupId, userId)
Add a user to a group as a standard member.

addMemberWithRole(communityId, groupId, userId, role)
Add a user with a role to a group.

addRecord(communityId, groupId, recordId)
Associate a record with a group.

createGroup(communityId, groupInput)
Create a group.

deleteBannerPhoto(communityId, groupId)
Delete the group banner photo.

deleteGroup(communityId, groupId)
Delete a group.

deleteMember(communityId, membershipId)
Remove a member from a group.

deletePhoto(communityId, groupId)
Delete the group photo.

getAnnouncements(communityId, groupId)
Get the first page of announcements in a group.

getAnnouncements(communityId, groupId, pageParam, pageSize)
Get a page of announcements in a group.

getBannerPhoto(communityId, groupId)
Get the group banner photo.

getGroup(communityId, groupId)
Get information about a group.

getGroupBatch(communityId, groupIds)
Get information about a list of groups.

getGroupMembershipRequest(communityId, requestId)
Get information about a request to join a private group.

getGroupMembershipRequests(communityId, groupId)
Get information about every request to join a private group.

getGroupMembershipRequests(communityId, groupId, status)
Get information about every request to join a private group that has a specified status.

getGroups(communityId)
Get the first page of groups.

getGroups(communityId, pageParam, pageSize)
Get a page of groups.

getGroups(communityId, pageParam, pageSize, archiveStatus)
Get a page of groups with an archive status.

getMember(communityId, membershipId)
Get information about a group member.

getMembers(communityId, groupId)
Get the first page of information about the members of a group.

getMembers(communityId, groupId, pageParam, pageSize)
Get a page of information about the members of a group.

getMembershipBatch(communityId, membershipIds)
Get information about a list of group memberships.

getMyChatterSettings(communityId, groupId)
Get the context user's Chatter settings for a group.

getPhoto(communityId, groupId)
Get the photo for a group.

getRecord(communityId, groupRecordId)
Get a record associated with a group.

getRecords(communityId, groupId)
Get the first page of records associated with a group.

getRecords(communityId, groupId, pageParam, pageSize)
Get a page of records associated with a group.

inviteUsers(groupId, invite)
Invite internal and external users to join a group.

postAnnouncement(communityId, groupId, announcement)
Post an announcement to a group.

removeRecord(communityId, groupRecordId)
Remove the association of a record with a group.

requestGroupMembership(communityId, groupId)
Request membership in a private group.

searchGroups(communityId, q)
Get the first page of groups that match the search criteria.

searchGroups(communityId, q, pageParam, pageSize)
Get a page of groups that match the search criteria.

searchGroups(communityId, q, archiveStatus, pageParam, pageSize)
Get a page of groups with the archive status that match the search criteria.

setBannerPhoto(communityId, groupId, fileId, versionNumber)
Set an uploaded file as the group banner photo.

setBannerPhoto(communityId, groupId, fileUpload)
Set a file that hasn't been uploaded as the group banner photo.

setBannerPhotoWithAttributes(communityId, groupId, bannerPhoto)
Set and crop an uploaded file as the group banner photo.

setBannerPhotoWithAttributes(communityId, groupId, bannerPhoto, fileUpload)
Set and crop a file that hasn't been uploaded as the group banner photo.

setPhoto(communityId, groupId, fileId, versionNumber)
Set an uploaded file as the group photo.

setPhoto(communityId, groupId, fileUpload)
Set a file that hasn't been uploaded as the group photo.

setPhotoWithAttributes(communityId, groupId, photo)
Set and crop an uploaded file as the group photo.

setPhotoWithAttributes(communityId, groupId, photo, fileUpload)
Set and crop a file that hasn't been uploaded as the group photo.

updateGroup(communityId, groupId, groupInput)
Update the settings of a group.

updateGroupMember(communityId, membershipId, role)
Update the role of a group member.

updateMyChatterSettings(communityId, groupId, emailFrequency)
Update the context user's email frequency for a group.

updateRequestStatus(communityId, requestId, status)
Update a request to join a private group.

updateRequestStatus(communityId, requestId, status, responseMessage)
Update a request to join a private group and optionally provide a message when the request is denied.

`addMember(communityId, groupId, userId)`

Add a user to a group as a standard member.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.GroupMember addMember(String communityId, String groupId,
String userId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

ID for a group.

userId
Type: String

ID for a user.

Return Value

Type: `ConnectApi.GroupMember`

Usage

To execute this method, the context user must be the group owner or moderator.

addMemberWithRole(communityId, groupId, userId, role)

Add a user with a role to a group.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.GroupMember addMemberWithRole(String communityId, String
groupId, String userId, ConnectApi.GroupMembershipType role)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

ID for a group.

userId
Type: String

ID for a user.

role
Type: ConnectApi.GroupMembershipType

The group membership type. One of these values:

- GroupManager
- StandardMember

Return Value

Type: ConnectApi.GroupMember

Usage

To execute this method, the context user must be the group owner or moderator.

addRecord(communityId, groupId, recordId)

Associate a record with a group.

API Version

34.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.GroupRecord addRecord(String communityId, String groupId,
String recordId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, internal, or null.

groupId
Type: String

ID of the group with which to associate the record.

recordId
Type: String

ID of the record to associate with the group.

Return Value

Type: ConnectApi.GroupRecord

createGroup(communityId, groupInput)

Create a group.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterGroupDetail createGroup(String, communityId,
ConnectApi.ChatterGroupInput groupInput)
```

Parameters

communityId
Type: String,

ID for an Experience Cloud site, `internal`, or `null`.

groupInput
Type: `ConnectApi.ChatterGroupInput`

The properties of the group.

Return Value

Type: `ConnectApi.ChatterGroupDetail`

deleteBannerPhoto(communityId, groupId)

Delete the group banner photo.

API Version

36.0

Requires Chatter

Yes

Signature

```
public static Void deleteBannerPhoto(String communityId, String groupId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

ID of the group.

Return Value

Type: Void

Usage

This method is successful only when the context user is the group manager or owner, or has Modify All Data permission.

deleteGroup(communityId, groupId)

Delete a group.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static Void deleteGroup(String communityId, String groupId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

ID for a group.

Return Value

Type: Void

deleteMember(communityId, membershipId)

Remove a member from a group.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static Void deleteMember(String communityId, String membershipId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

membershipId
Type: String

ID for a membership.

Return Value

Type: Void

Usage

This method is successful only when the context user is the group manager or owner, or has Modify All Data permission.

deletePhoto(communityId, groupId)

Delete the group photo.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static Void deletePhoto(String communityId, String groupId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

ID for a group.

Return Value

Type: Void

Usage

This method is only successful when the context user is the group manager or owner, or has Modify All Data permission.

getAnnouncements(communityId, groupId)

Get the first page of announcements in a group.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.AnnouncementPage getAnnouncements(String communityId, String groupId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

ID for a group.

Return Value

Type: `ConnectApi.AnnouncementPage`

Usage

To post an announcement, get information about an announcement, update the expiration date of an announcement, or delete an announcement, use the methods of the `ConnectApi.Announcements` class.

getAnnouncements(communityId, groupId, pageParam, pageSize)

Get a page of announcements in a group.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.AnnouncementPage getAnnouncements(String communityId, String groupId, Integer pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

ID for a group.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.AnnouncementPage`

Usage

To post an announcement, get information about an announcement, update the expiration date of an announcement, or delete an announcement, use the methods of the `ConnectApi.Announcements` class.

`getBannerPhoto(communityId, groupId)`

Get the group banner photo.

API Version

36.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.BannerPhoto getBannerPhoto(String communityId, String groupId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

The ID of the group.

Return Value

Type: `ConnectApi.BannerPhoto`

getGroup(communityId, groupId)

Get information about a group.

API Version

28.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterGroupDetail getGroup(String communityId, String groupId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

ID for a group.

Return Value

Type: `ConnectApi.ChatterGroupDetail`

getGroupBatch(communityId, groupIds)

Get information about a list of groups.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.BatchResult[] getGroupBatch(String communityId, List<String>
groupIds)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupIds
Type: List<String>

A list of up to 500 group IDs.

Return Value

Type: `ConnectApi.BatchResult`[]

The `ConnectApi.BatchResult.getResult()` method returns a `ConnectApi.ChatterGroup` object and errors embedded in the results for groups that didn't load.

Example

```
// Create a list of groups.
ConnectApi.ChatterGroupPage groupPage = ConnectApi.ChatterGroups.getGroups(null);

// Create a list of group IDs.
List<String> groupIds = new List<String>();
for (ConnectApi.ChatterGroup aGroup : groupPage.groups){
    groupIds.add(aGroup.id);
}

// Get info about all the groups in the list.
ConnectApi.BatchResult[] batchResults = ConnectApi.ChatterGroups.getGroupBatch(null,
groupIds);

for (ConnectApi.BatchResult batchResult : batchResults) {
    if (batchResult.isSuccess()) {
        // Operation was successful.
        // Print the number of members in each group.
        ConnectApi.ChatterGroup aGroup;
        if(batchResult.getResult() instanceof ConnectApi.ChatterGroup) {
           aGroup = (ConnectApi.ChatterGroup) batchResult.getResult();
        }
```

```
        System.debug('SUCCESS');
        System.debug(aGroup.memberCount);
    }
    else {
        // Operation failed. Print errors.
        System.debug('FAILURE');
        System.debug(batchResult.getErrorMessage());
    }
}
```

SEE ALSO:

getMembershipBatch(communityId, membershipIds)

getGroupMembershipRequest(communityId, requestId)

Get information about a request to join a private group.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.GroupMembershipRequest getGroupMembershipRequest(String
communityId, String requestId)
```

Parameters

communityId

Type: String

ID for an Experience Cloud site, `internal`, or `null`.

requestId

Type: String

The ID of a request to join a private group.

Return Value

Type: `ConnectApi.GroupMembershipRequest`

Usage

This method is successful only when the context user is the group manager or owner, or has Modify All Data permission.

getGroupMembershipRequests(communityId, groupId)

Get information about every request to join a private group.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.GroupMembershipRequests getGroupMembershipRequests(String communityId, String groupId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

ID for a group.

Return Value

Type: `ConnectApi.GroupMembershipRequests`

Usage

This method is successful only when the context user is the group manager or owner, or has Modify All Data permission.

getGroupMembershipRequests(communityId, groupId, status)

Get information about every request to join a private group that has a specified status.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.GroupMembershipRequests getGroupMembershipRequests(String communityId, String groupId, ConnectApi.GroupMembershipRequestStatus status)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

ID for a group.

status
Type: `ConnectApi.GroupMembershipRequestStatus`

status—Status of a request to join a private group.

- `Accepted`
- `Declined`
- `Pending`

Return Value

Type: `ConnectApi.GroupMembershipRequests`

Usage

This method is successful only when the context user is the group manager or owner, or has Modify All Data permission.

`getGroups(communityId)`

Get the first page of groups.

API Version

28.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterGroupPage getGroups(String communityId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

Return Value

Type: ConnectApi.ChatterGroupPage

getGroups(communityId, pageParam, pageSize)

Get a page of groups.

API Version

28.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterGroupPage getGroups(String communityId, Integer pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: ConnectApi.ChatterGroupPage

getGroups(communityId, pageParam, pageSize, archiveStatus)

Get a page of groups with an archive status.

API Version

29.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterGroupPage getGroups(String communityId, Integer
pageParam, Integer pageSize, ConnectApi.GroupArchiveStatus archiveStatus)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

archiveStatus
Type: `ConnectApi.GroupArchiveStatus`

Archive status of groups.

- `All`—All groups, including groups that are archived and groups that aren't archived.
- `Archived`—Groups that are archived.
- `NotArchived`—Groups that aren't archived.

If you pass in `null`, the default value is `All`.

Return Value

Type: `ConnectApi.ChatterGroupPage`

`getMember(communityId, membershipId)`

Get information about a group member.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.GroupMember getMember(String communityId, String membershipId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

membershipId
Type: String

ID for a membership.

Return Value

Type: `ConnectApi.GroupMember`

`getMembers(communityId, groupId)`

Get the first page of information about the members of a group.

API Version

28.0

Available to Guest Users

36.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.GroupMemberPage getMembers(String communityId, String groupId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

ID for a group.

Return Value

Type: `ConnectApi.GroupMemberPage`

getMembers(communityId, groupId, pageParam, pageSize)

Get a page of information about the members of a group.

API Version

28.0

Available to Guest Users

36.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.GroupMemberPage getMembers(String communityId, String groupId,
Integer pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

ID for a group.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.GroupMemberPage`

getMembershipBatch(communityId, membershipIds)

Get information about a list of group memberships.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.BatchResult[] getMembershipBatch(String communityId,
List<String> membershipIds)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

membershipIds
Type: List<String>

A list of up to 500 group membership IDs.

Return Value

Type: `ConnectApi.BatchResult`[]

The `ConnectApi.BatchResult.getResult()` method returns a `ConnectApi.GroupMember` object and errors embedded in the results for group memberships that didn't load.

Example

```
// Get members of a group.
ConnectApi.GroupMemberPage membersPage = ConnectApi.ChatterGroups.getMembers(null,
'0F9D00000000oOT');

// Create a list of membership IDs.
List<String> membersList = new List<String>();
for (ConnectApi.GroupMember groupMember : membersPage.members){
    membersList.add(groupMember.id);
}

// Get info about all group memberships in the list.
ConnectApi.BatchResult[] batchResults = ConnectApi.ChatterGroups.getMembershipBatch(null,
 membersList);

for (ConnectApi.BatchResult batchResult : batchResults) {
    if (batchResult.isSuccess()) {
        // Operation was successful.
        // Print the first name of each member.
        ConnectApi.GroupMember groupMember;
        if(batchResult.getResult() instanceof ConnectApi.GroupMember) {
            groupMember = (ConnectApi.GroupMember) batchResult.getResult();
        }
        System.debug('SUCCESS');
        System.debug(groupMember.user.firstName);
    }
```

```
    else {
        // Operation failed. Print errors.
        System.debug('FAILURE');
        System.debug(batchResult.getErrorMessage());
    }
}
```

SEE ALSO:

getGroupBatch(communityId, groupIds)

getMyChatterSettings(communityId, groupId)

Get the context user's Chatter settings for a group.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.GroupChatterSettings getMyChatterSettings(String communityId,
String groupId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

ID for a group.

Return Value

Type: `ConnectApi.GroupChatterSettings`

getPhoto(communityId, groupId)

Get the photo for a group.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Photo getPhoto(String communityId, String groupId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

ID for a group.

Return Value

Type: `ConnectApi.Photo`

getRecord(communityId, groupRecordId)

Get a record associated with a group.

API Version

34.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.GroupRecord getRecord(String communityId, String groupRecordId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupRecordId
Type: String

ID of the group record.

Return Value

Type: `ConnectApi.GroupRecord`

getRecords(communityId, groupId)

Get the first page of records associated with a group.

API Version

33.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.GroupRecordPage getRecords(String communityId, String groupId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

ID for a group.

Return Value

Type: `ConnectApi.GroupRecordPage`

getRecords(communityId, groupId, pageParam, pageSize)

Get a page of records associated with a group.

API Version

33.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.GroupRecordPage getRecords(String communityId, String groupId,
Integer pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

ID for a group.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.GroupRecordPage`

inviteUsers(groupId, invite)

Invite internal and external users to join a group.

API Version

39.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Invitations inviteUsers(String groupId, ConnectApi.InviteInput
invite)
```

Parameters

groupId
Type: String

ID of the group.

invite
Type: `ConnectApi.InviteInput`

A `ConnectApi.InviteInput` body.

Return Value

Type: `ConnectApi.Invitations`

postAnnouncement(communityId, groupId, announcement)

Post an announcement to a group.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Announcement postAnnouncement(String communityId, String groupId, ConnectApi.AnnouncementInput announcement)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

ID for a group.

announcement
Type: `ConnectApi.AnnouncementInput`

A `ConnectApi.AnnouncementInput` object.

Return Value

Type: `ConnectApi.Announcement`

Usage

Use an announcement to highlight information. Users can discuss, like, and post comments on announcements. Deleting the feed post deletes the announcement.

To post an announcement, get information about an announcement, update the expiration date of an announcement, or delete an announcement, use the methods of the `ConnectApi.Announcements` class.

removeRecord(communityId, groupRecordId)

Remove the association of a record with a group.

API Version

34.0

Requires Chatter

Yes

Signature

```
public static Void removeRecord(String communityId, String groupRecordId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupRecordId
Type: String

ID of the group record.

Return Value

Type: Void

requestGroupMembership(communityId, groupId)

Request membership in a private group.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.GroupMembershipRequest requestGroupMembership(String
communityId, String groupId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

ID for a group.

Return Value

Type: ConnectApi.GroupMembershipRequest

Sample: Requesting to Join a Private Group

This sample code calls `ConnectApi.ChatterGroups.requestGroupMembership` to request to join a private group.

```
String communityId = null;
ID groupId = '0F9x00000000hAZ';

ConnectApi.GroupMembershipRequest membershipRequest =
ConnectApi.ChatterGroups.requestGroupMembership(communityId, groupId);
```

searchGroups(communityId, q)

Get the first page of groups that match the search criteria.

API Version

28.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterGroupPage searchGroups(String communityId, String q)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards. Can be specified as `null`.

Return Value

Type: ConnectApi.ChatterGroupPage

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchGroups(communityId, q, result)

Apex Developer Guide: Testing ConnectApi Code

`searchGroups(communityId, q, pageParam, pageSize)`

Get a page of groups that match the search criteria.

API Version

28.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterGroupPage searchGroups(String communityId, String q,
Integer pageParam, Integer pageSize)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`q`
Type: String

Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards. Can be specified as `null`.

`pageParam`
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

`pageSize`
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.ChatterGroupPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchGroups(communityId, q, pageParam, pageSize, result)

Apex Developer Guide: Testing ConnectApi Code

`searchGroups(communityId, q, archiveStatus, pageParam, pageSize)`

Get a page of groups with the archive status that match the search criteria.

API Version

29.0

Available to Guest Users

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterGroupPage searchGroups(String communityId, String q,
ConnectApi.GroupArchiveStatus archiveStatus, Integer pageParam, Integer pageSize)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`q`
Type: String

Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards. Can be specified as `null`.

`archiveStatus`
Type: `ConnectApi.GroupArchiveStatus`

Archive status of groups.

- `All`—All groups, including groups that are archived and groups that aren't archived.
- `Archived`—Groups that are archived.

- `NotArchived`—Groups that aren't archived.

`pageParam`
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

`pageSize`
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.ChatterGroupPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchGroups(communityId, q, archiveStatus, pageParam, pageSize, result)

Apex Developer Guide: Testing ConnectApi Code

`setBannerPhoto(communityId, groupId, fileId, versionNumber)`

Set an uploaded file as the group banner photo.

API Version

36.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.BannerPhoto setBannerPhoto(String communityId, String groupId,
String fileId, Integer versionNumber)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`groupId`
Type: String

The ID of the group.

fileId
Type: String

The ID of the already uploaded file. The key prefix must be 069, and the image must be smaller than 8 MB.

versionNumber
Type: Integer

Version number of the existing file. Specify either an existing version number or, to get the latest version, specify `null`.

Return Value

Type: `ConnectApi.BannerPhoto`

Usage

This method is successful only when the context user is the group manager or owner, or has Modify All Data permission.

Photos are processed asynchronously and might not be visible right away.

`setBannerPhoto(communityId, groupId, fileUpload)`

Set a file that hasn't been uploaded as the group banner photo.

API Version

36.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.BannerPhoto setBannerPhoto(String communityId, String groupId,
ConnectApi.BinaryInput fileUpload)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

The ID of the group.

fileUpload
Type: `ConnectApi.BinaryInput`

File to use as the photo. The content type must be usable as an image.

Return Value

Type: ConnectApi.BannerPhoto

Usage

This method is successful only when the context user is the group manager or owner, or has Modify All Data permission.

Photos are processed asynchronously and might not be visible right away.

setBannerPhotoWithAttributes(communityId, groupId, bannerPhoto)

Set and crop an uploaded file as the group banner photo.

API Version

36.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.BannerPhoto setBannerPhotoWithAttributes(String communityId,
String groupId, ConnectApi.BannerPhotoInput bannerPhoto)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

The ID of the group.

bannerPhoto
Type: ConnectApi.BannerPhotoInput

A `ConnectApi.BannerPhotoInput` object that specifies the ID and version of the file, and how to crop the file.

Return Value

Type: ConnectApi.BannerPhoto

Usage

This method is successful only when the context user is the group manager or owner, or has Modify All Data permission.

Photos are processed asynchronously and might not be visible right away.

setBannerPhotoWithAttributes(communityId, groupId, bannerPhoto, fileUpload)

Set and crop a file that hasn't been uploaded as the group banner photo.

API Version

36.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.BannerPhoto setBannerPhotoWithAttributes(String communityId,
String groupId, ConnectApi.BannerPhotoInput bannerPhoto, ConnectApi.BinaryInput
fileUpload)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

The ID of the group.

bannerPhoto
Type: `ConnectApi.BannerPhotoInput`

A `ConnectApi.BannerPhotoInput` object specifying the cropping parameters.

fileUpload
Type: `ConnectApi.BinaryInput`

File to use as the photo. The content type must be usable as an image.

Return Value

Type: `ConnectApi.BannerPhoto`

Usage

This method is successful only when the context user is the group manager or owner, or has Modify All Data permission.

Photos are processed asynchronously and might not be visible right away.

setPhoto(communityId, groupId, fileId, versionNumber)

Set an uploaded file as the group photo.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Photo setPhoto(String communityId, String groupId, String
fileId, Integer versionNumber)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

ID for a group.

fileId
Type: String

ID of a file already uploaded. The key prefix must be 069, and the file must be an image that is smaller than 2 GB.

versionNumber
Type: Integer

Version number of the existing file. Specify either an existing version number or, to get the latest version, specify `null`.

Return Value

Type: `ConnectApi.Photo`

Usage

This method is successful only when the context user is the group manager or owner, or has Modify All Data permission.

Photos are processed asynchronously and might not be visible right away.

Sample: Updating a Group Photo with an Existing File

When a group is created, it doesn't have a group photo. You can set an existing photo that has already been uploaded to Salesforce as the group photo. The key prefix must be 069 and the file size must be less than 2 GB.

```
String communityId = null;
ID groupId = '0F9x00000000hAK';
ID fileId = '069x00000001Ion';

// Set photo
ConnectApi.Photo photo = ConnectApi.ChatterGroups.setPhoto(communityId, groupId, fileId,
null);
```

setPhoto(communityId, groupId, fileUpload)

Set a file that hasn't been uploaded as the group photo.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Photo setPhoto(String communityId, String groupId,
ConnectApi.BinaryInput fileUpload)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

ID for a group.

fileUpload
Type: `ConnectApi.BinaryInput`

File to use as the photo. The content type must be usable as an image.

Return Value

Type: `ConnectApi.Photo`

Usage

This method is successful only when the context user is the group manager or owner, or has Modify All Data permission.

Photos are processed asynchronously and might not be visible right away.

Sample: Uploading a New File and Using It as a Group Photo

When a group is created, it doesn't have a group photo. You can upload a photo and set it as the group photo.

```
String communityId = null;
ID groupId = '0F9x00000000hAP';
ID photoId = '069x00000001Ioo';

// Set photo
List<ContentVersion> groupPhoto = [Select c.VersionData From ContentVersion c where
ContentDocumentId=:photoId];
ConnectApi.BinaryInput binary = new ConnectApi.BinaryInput(groupPhoto.get(0).VersionData,
```

```
 'image/png', 'image.png');
ConnectApi.Photo photo = ConnectApi.ChatterGroups.setPhoto(communityId, groupId, binary);
```

setPhotoWithAttributes(communityId, groupId, photo)

Set and crop an uploaded file as the group photo.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Photo setPhotoWithAttributes(String communityId, String groupId,
ConnectApi.PhotoInput photo)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

ID for a group.

photo
Type: `ConnectApi.PhotoInput`

A `ConnectApi.PhotoInput` object that specifies the ID and version of the file, and how to crop the file.

Return Value

Type: `ConnectApi.Photo`

Usage

This method is successful only when the context user is the group manager or owner, or has Modify All Data permission.

Photos are processed asynchronously and might not be visible right away.

setPhotoWithAttributes(communityId, groupId, photo, fileUpload)

Set and crop a file that hasn't been uploaded as the group photo.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Photo setPhotoWithAttributes(String communityId, String groupId,
ConnectApi.PhotoInput photo, ConnectApi.BinaryInput fileUpload)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

ID for a group.

photo
Type: `ConnectApi.PhotoInput`

A `ConnectApi.PhotoInput` object that specifies how to crop the file specified in *`fileUpload`*.

fileUpload
Type: `ConnectApi.BinaryInput`

File to use as the photo. The content type must be usable as an image.

Return Value

Type: `ConnectApi.Photo`

Usage

This method is successful only when the context user is the group manager or owner, or has Modify All Data permission.

Photos are processed asynchronously and might not be visible right away.

`updateGroup(communityId, groupId, groupInput)`

Update the settings of a group.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterGroup updateGroup(String communityId, String groupId,
ConnectApi.ChatterGroupInput groupInput)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

ID for a group.

groupInput
Type: `ConnectApi.ChatterGroupInput`

A `ConnectApi.ChatterGroupInput` object.

Return Value

Type: `ConnectApi.ChatterGroup`

Usage

This method is successful only when the context user is the group manager or owner, or has Modify All Data permission. Use this method to update any settings in the `ConnectApi.ChatterGroupInput` class. These settings include the group title and text in the "Information" section, whether the group is public or private, and whether the group is archived.

Example

This example archives a group.

```
String groupId = '0F9D00000000qSz';
String communityId = null;

ConnectApi.ChatterGroupInput groupInput = new ConnectApi.ChatterGroupInput();
groupInput.isArchived = true;

ConnectApi.ChatterGroups.updateGroup(communityId, groupId, groupInput);
```

updateGroupMember(communityId, membershipId, role)

Update the role of a group member.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterGroup updateGroupMember(String communityId, String
membershipId, ConnectApi.GroupMembershipType role)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

membershipId
Type: String

ID for a membership.

role
Type: `ConnectApi.GroupMembershipType`

The group membership type. One of these values:

- `GroupManager`
- `StandardMember`

Return Value

Type: `ConnectApi.ChatterGroup`

Usage

This method is successful only when the context user is the group manager or owner, or has Modify All Data permission.

updateMyChatterSettings(communityId, groupId, emailFrequency)

Update the context user's email frequency for a group.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.GroupChatterSettings updateMyChatterSettings(String communityId,
String groupId, ConnectApi.GroupEmailFrequency emailFrequency)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

ID for a group.

emailFrequency
Type: `ConnectApi.GroupEmailFrequency`

Frequency with which a user receives email.

- `EachPost`
- `DailyDigest`
- `WeeklyDigest`
- `Never`
- `UseDefault`

The value `UseDefault` uses the value set in a call to `updateChatterSettings(communityId, userId, defaultGroupEmailFrequency)`.

Return Value

Type: `ConnectApi.GroupChatterSettings`

updateRequestStatus(communityId, requestId, status)

Update a request to join a private group.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.GroupMembershipRequest updateRequestStatus(String communityId,
String requestId, ConnectApi.GroupMembershipRequestStatus status)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

requestId
Type: String

ID for a request to join a private group.

status
Type: `ConnectApi.GroupMembershipRequestStatus`

Status of the request:

- `Accepted`
- `Declined`

The `Pending` value of the enum is not valid in this method.

Return Value

Type: `ConnectApi.GroupMembershipRequest`

Usage

This method is successful only when the context user is the group manager or owner, or has Modify All Data permission.

Sample: Accepting or Declining a Request to Join a Private Group

This sample code calls `ConnectApi.ChatterGroups.updateRequestStatus` and passes it the membership request ID and an `ConnectApi.GroupMembershipRequestStatus.Accepted` status. You can also pass `ConnectApi.GroupMembershipRequestStatus.Declined`.

```
String communityId = null;
ID groupId = '0F9x00000000hAZ';
String requestId = '0I5x000000001snCAA';

ConnectApi.GroupMembershipRequest membershipRequestRep =
ConnectApi.ChatterGroups.updateRequestStatus(communityId, requestId,
ConnectApi.GroupMembershipRequestStatus.Accepted);
```

updateRequestStatus(communityId, requestId, status, responseMessage)

Update a request to join a private group and optionally provide a message when the request is denied.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.GroupMembershipRequest updateRequestStatus(String communityId,
String requestId, ConnectApi.GroupMembershipRequestStatus status, String responseMessage)
```

Parameters

communityId

Type: String

ID for an Experience Cloud site, `internal`, or `null`.

requestId
Type: String

ID for a request to join a private group.

status
Type: `ConnectApi.GroupMembershipRequestStatus`

Status of the request:

- `Accepted`
- `Declined`

The `Pending` value of the enum is not valid in this method.

responseMessage
Type: String

Provide a message to the user if their membership request is declined. The value of this property is used only when the value of the `status` property is `Declined`.

The maximum length is 756 characters.

Return Value

Type: `ConnectApi.GroupMembershipRequest`

Usage

This method is successful only when the context user is the group manager or owner, or has Modify All Data permission.

ChatterGroups Test Methods

These test methods are for `ChatterGroups`. All methods are static.

For information about using these methods to test your `ConnectApi` code, see Testing ConnectApi Code.

`setTestSearchGroups(communityId, q, result)`

Register a `ConnectApi.ChatterGroupPage` object to be returned when the matching `ConnectApi.searchGroups` method is called in a test context. Use the test method with the same parameters or you receive an exception.

API Version

29.0

Signature

```
public static Void setTestSearchGroups(String communityId, String q,
ConnectApi.ChatterGroupPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards. Can be specified as `null`.

result
Type: `ConnectApi.ChatterGroupPage`

Test `ConnectApi.ChatterGroupPage` object.

Return Value

Type: Void

SEE ALSO:

searchGroups(communityId, q)

Apex Developer Guide: Testing ConnectApi Code

`setTestSearchGroups(communityId, q, pageParam, pageSize, result)`

Register a `ConnectApi.ChatterGroupPage` object to be returned when the matching `ConnectApi.searchGroups` method is called in a test context. Use the test method with the same parameters or you receive an exception.

API Version

28.0

Signature

```
public static Void setTestSearchGroups(String communityId, String q, Integer pageParam,
Integer pageSize, ConnectApi.ChatterGroupPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards. Can be specified as `null`.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

result
Type: `ConnectApi.ChatterGroupPage`

Test `ConnectApi.ChatterGroupPage` object.

Return Value

Type: Void

SEE ALSO:

searchGroups(communityId, q, pageParam, pageSize)

Apex Developer Guide: Testing ConnectApi Code

`setTestSearchGroups(communityId, q, archiveStatus, pageParam, pageSize, result)`

Register a `ConnectApi.ChatterGroupPage` object to be returned when the matching `ConnectApi.searchGroups` method is called in a test context. Use the test method with the same parameters or you receive an exception.

API Version

29.0

Signature

```
public static Void setTestSearchGroups(String communityId, String q,
ConnectApi.GroupArchiveStatus, archiveStatus, Integer pageParam, Integer pageSize,
ConnectApi.ChatterGroupPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards. Can be specified as `null`.

archiveStatus
Type: `ConnectApi.GroupArchiveStatus`

Archive status of groups.

- `All`—All groups, including groups that are archived and groups that aren't archived.
- `Archived`—Groups that are archived.
- `NotArchived`—Groups that aren't archived.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

result
Type: `ConnectApi.ChatterGroupPage`

Test `ConnectApi.ChatterGroupPage` object.

Return Value

Type: Void

SEE ALSO:

searchGroups(communityId, q, archiveStatus, pageParam, pageSize)

Apex Developer Guide: Testing ConnectApi Code

ChatterMessages Class

Get, send, search, and reply to private messages. You can also get and search private conversations, mark conversations as read, and get a count of unread private messages.

Namespace

ConnectApi

Usage

Private messages and direct messages are different features. Direct messages are newer and offer a richer feature set for private communication in Experience Cloud sites. Direct messages are a capability within Chatter feeds. To work with direct messages, use the ChatterFeeds Class.

ChatterMessages Methods

These methods are for `ChatterMessages`. All methods are static.

All methods in this class require Chatter and are subject to the per user, per namespace, per hour rate limit.

IN THIS SECTION:

getConversation(conversationId)
Get a conversation.

getConversation(conversationId, pageParam, pageSize)
Get a page of a conversation.

getConversation(communityId, conversationId)
Get a conversation from an Experience Cloud site.

getConversation(communityId, conversationId, pageParam, pageSize)
Get a page of a conversation from an Experience Cloud site.

getConversations()
Get the most recent conversations.

getConversations(pageParam, pageSize)
Get a page of conversations.

getConversations(communityId)
Get the most recent conversations from an Experience Cloud site.

getConversations(communityId, pageParam, pageSize)
Get a page of conversations from an Experience Cloud site.

getMessage(messageId)
Get a message.

getMessage(communityId, messageId)
Get a message from an Experience Cloud site.

getMessages()
Get the most recent messages.

getMessages(pageParam, pageSize)
Get a page of messages.

getMessages(communityId)
Get the most recent messages from an Experience Cloud site.

getMessages(communityId, pageParam, pageSize)
Get a page of messages from an Experience Cloud site.

getUnreadCount()
Get the number of conversations that are marked unread.

getUnreadCount(communityId)
Get the number of conversations that are marked unread in an Experience Cloud site.

markConversationRead(conversationId, read)
Mark a conversation as read or unread.

markConversationRead(communityId, conversationID, read)
Mark a conversation as read or unread in an Experience Cloud site.

replyToMessage(text, inReplyTo)
Reply to a message.

replyToMessage(communityId, text, inReplyTo)
Reply to a message in an Experience Cloud site.

searchConversation(conversationId, q)
Get a conversation that matches the search criteria.

searchConversation(conversationId, pageParam, pageSize, q)
Get a conversation with a page of messages that match the search criteria.

searchConversation(communityId, conversationId, q)
Get a conversation with messages that match the search criteria in an Experience Cloud site.

searchConversation(communityId, conversationId, pageParam, pageSize, q)
Get a conversation with a page of messages that match the search criteria in an Experience Cloud site.

searchConversations(q)
Get conversations in which member names and messages match the search criteria.

searchConversations(pageParam, pageSize, q)
Get a page of conversations in which member names and messages match the search criteria.

searchConversations(communityId, q)
Get conversations in which member names and messages match the search criteria in an Experience Cloud site.

searchConversations(communityId, pageParam, pageSize, q)
Get a page of conversations in which member names and messages match the search criteria in an Experience Cloud site.

searchMessages(q)
Get messages that match the search criteria.

searchMessages(pageParam, pageSize, q)
Get a page of messages that match the search criteria.

searchMessages(communityId, q)
Get messages that match the search criteria in an Experience Cloud site.

searchMessages(communityId, pageParam, pageSize, q)
Get a page of messages that match the search criteria in an Experience Cloud site.

sendMessage(text, recipients)
Send a message to a list of recipients.

sendMessage(communityId, text, recipients)
Send a message to a list of recipients in an Experience Cloud site.

`getConversation(conversationId)`

Get a conversation.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterConversation getConversation(String conversationId)
```

Parameters

`conversationId`
Type: String

ID for the conversation.

Return Value

Type: ConnectApi.ChatterConversation

getConversation(conversationId, pageParam, pageSize)

Get a page of a conversation.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterConversation getConversation(String conversationId,
String pageParam, Integer pageSize)
```

Parameters

conversationId
Type: String

ID for the conversation.

pageParam
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: ConnectApi.ChatterConversation

getConversation(communityId, conversationId)

Get a conversation from an Experience Cloud site.

API Version

30.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterConversation getConversation(String communityId, String conversationId)
```

Parameters

communityId
Type:String

ID for an Experience Cloud site, `internal`, or `null`.

conversationId
Type: String

ID for the conversation.

Return Value

Type: `ConnectApi.ChatterConversation`

getConversation(communityId, conversationId, pageParam, pageSize)

Get a page of a conversation from an Experience Cloud site.

API Version

30.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterConversation getConversation(String communityId, String conversationId, String pageParam, String pageSize)
```

Parameters

communityId
Type:String

ID for an Experience Cloud site, `internal`, or `null`.

conversationId
Type: String

ID for the conversation.

pageParam
Type:String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.ChatterConversation`

getConversations()

Get the most recent conversations.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterConversationPage getConversations()
```

Return Value

Type: `ConnectApi.ChatterConversationPage`

getConversations(pageParam, pageSize)

Get a page of conversations.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterConversationPage getConversations(String pageParam,
Integer pageSize)
```

Parameters

pageParam
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.ChatterConversationPage`

getConversations(communityId)

Get the most recent conversations from an Experience Cloud site.

API Version

30.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterConversationPage getConversations(String communityId)
```

Parameters

communityId
Type:String

ID for an Experience Cloud site, `internal`, or `null`.

Return Value

Type: `ConnectApi.ChatterConversationPage`

getConversations(communityId, pageParam, pageSize)

Get a page of conversations from an Experience Cloud site.

API Version

30.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterConversationPage getConversations(String communityId,
String pageParam, Integer pageSize)
```

Parameters

communityId
Type:String

ID for an Experience Cloud site, `internal`, or `null`.

pageParam
Type:String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.ChatterConversationPage`

getMessage(messageId)

Get a message.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterMessage getMessage(String messageId)
```

Parameters

messageId
Type: String

ID for the message.

Return Value

Type: ConnectApi.ChatterMessage

getMessage(communityId, messageId)

Get a message from an Experience Cloud site.

API Version

30.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterMessage getMessage(String communityId, String messageId)
```

Parameters

communityId
Type:String

ID for an Experience Cloud site, `internal`, or `null`.

messageId
Type: String

ID for the message.

Return Value

Type:ConnectApi.ChatterMessage

getMessages()

Get the most recent messages.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterMessagePage getMessages()
```

Return Value

Type: ConnectApi.ChatterMessagePage

getMessages(pageParam, pageSize)

Get a page of messages.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterMessagePage getMessages(String pageParam, Integer pageSize)
```

Parameters

pageParam
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: ConnectApi.ChatterMessagePage

getMessages(communityId)

Get the most recent messages from an Experience Cloud site.

API Version

30.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterMessagePage getMessages(String communityId)
```

Parameters

communityId
Type:String

ID for an Experience Cloud site, `internal`, or `null`.

Return Value

Type: `ConnectApi.ChatterMessagePage`

getMessages(communityId, pageParam, pageSize)

Get a page of messages from an Experience Cloud site.

API Version

30.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterMessagePage getMessages(String communityId, String pageParam, Integer pageSize)
```

Parameters

communityId
Type:String

ID for an Experience Cloud site, `internal`, or `null`.

pageParam
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.ChatterMessagePage`

getUnreadCount()

Get the number of conversations that are marked unread.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UnreadConversationCount getUnreadCount()
```

Return Value

Type: `ConnectApi.UnreadConversationCount`

If there are fewer than 50 unread conversations, `ConnectApi.UreadConversationCount` returns the exact number of unread conversations and the `hasMore` property is `false`. If there are more than 50 unread conversations, `ConnectApi.UreadConversationCount` returns 50 unread conversations and the `hasMore` property is `true`.

Example

```
ConnectApi.UnreadConversationCount unread = ConnectApi.ChatterMessages.getUnreadCount();
```

`getUnreadCount(communityId)`

Get the number of conversations that are marked unread in an Experience Cloud site.

API Version

30.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UnreadConversationCount getUnreadCount(String communityId)
```

Parameters

`communityId`
Type:String

ID for an Experience Cloud site, `internal`, or `null`.

Return Value

Type: `ConnectApi.UnreadConversationCount`

If there are fewer than 50 unread conversations, `ConnectApi.UreadConversationCount` returns the exact number of unread conversations and the `hasMore` property is `false`. If there are more than 50 unread conversations, `ConnectApi.UreadConversationCount` returns 50 unread conversations and the `hasMore` property is `true`.

markConversationRead(conversationId, read)

Mark a conversation as read or unread.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterConversationSummary markConversationRead(String conversationId, Boolean read)
```

Parameters

conversationId
Type: String

ID for the conversation.

read
Type: Boolean

Specify whether the conversation is read (`true`) or not (`false`).

Return Value

Type: `ConnectApi.ChatterConversationSummary`

markConversationRead(communityId, conversationID, read)

Mark a conversation as read or unread in an Experience Cloud site.

API Version

30.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterConversationSummary markConversationRead(String communityId, String conversationID, Boolean read)
```

Parameters

communityId
Type:String

ID for an Experience Cloud site, `internal`, or `null`.

conversationId
Type: String

ID for the conversation.

read
Type: Boolean

Specify whether the conversation is read (`true`) or not (`false`).

Return Value

Type: `ConnectApi.ChatterConversationSummary`

replyToMessage(text, inReplyTo)

Reply to a message.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterMessage replyToMessage(String text, String inReplyTo)
```

Parameters

text
Type: String

Text of the message. Can't be empty or over 10,000 characters.

inReplyTo
Type: String

ID of the message that is being responded to.

Return Value

Type: `ConnectApi.ChatterMessage`

replyToMessage(communityId, text, inReplyTo)

Reply to a message in an Experience Cloud site.

API Version

30.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterMessage replyToMessage(String communityId, String text,
String inReplyTo)
```

Parameters

communityId
Type:String

ID for an Experience Cloud site, `internal`, or `null`.

text
Type: String

Text of the message. Can't be empty or over 10,000 characters.

inReplyTo
Type: String

ID of the message that is being responded to.

Return Value

Type: `ConnectApi.ChatterMessage`

searchConversation(conversationId, q)

Get a conversation that matches the search criteria.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterConversation searchConversation(String conversationId,
String q)
```

Parameters

conversationId
Type: String

ID for the conversation.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.ChatterConversation`

searchConversation(conversationId, pageParam, pageSize, q)

Get a conversation with a page of messages that match the search criteria.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterConversation searchConversation(String conversationId,
String pageParam, Integer pageSize, String q)
```

Parameters

conversationId
Type: String

ID for the conversation.

pageParam
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: ConnectApi.ChatterConversation

searchConversation(communityId, conversationId, q)

Get a conversation with messages that match the search criteria in an Experience Cloud site.

API Version

30.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterConversation searchConversation(String communityId,
String conversationId, String q)
```

Parameters

communityId
Type:String

ID for an Experience Cloud site, `internal`, or `null`.

conversationId
Type: String

ID for the conversation.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: ConnectApi.ChatterConversation

searchConversation(communityId, conversationId, pageParam, pageSize, q)

Get a conversation with a page of messages that match the search criteria in an Experience Cloud site.

API Version

30.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterConversation searchConversation(String communityId,
String conversationId, String pageParam, Integer pageSize, String q)
```

Parameters

communityId
Type:String

ID for an Experience Cloud site, `internal`, or `null`.

conversationId
Type: String

ID for the conversation.

pageParam
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.ChatterConversation`

searchConversations(q)

Get conversations in which member names and messages match the search criteria.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterConversationPage searchConversations(String q)
```

Parameters

q

Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.ChatterConversationPage`

searchConversations(pageParam, pageSize, q)

Get a page of conversations in which member names and messages match the search criteria.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterConversationPage searchConversations(String pageParam,
Integer pageSize, String q)
```

Parameters

pageParam

Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize

Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

q

Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.ChatterConversationPage`

searchConversations(communityId, q)

Get conversations in which member names and messages match the search criteria in an Experience Cloud site.

API Version

30.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterConversationPage searchConversations(String communityId,
String q)
```

Parameters

communityId
Type:String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.ChatterConversationPage`

searchConversations(communityId, pageParam, pageSize, q)

Get a page of conversations in which member names and messages match the search criteria in an Experience Cloud site.

API Version

30.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterConversationPage searchConversations(String communityId,
String pageParam, Integer pageSize, String q)
```

Parameters

communityId
Type:String

ID for an Experience Cloud site, `internal`, or `null`.

pageParam
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.ChatterConversationPage`

searchMessages(q)

Get messages that match the search criteria.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterMessagePage searchMessages(String q)
```

Parameters

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.ChatterMessagePage`

searchMessages(pageParam, pageSize, q)

Get a page of messages that match the search criteria.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterMessagePage searchMessages(String pageParam, Integer pageSize, String q)
```

Parameters

pageParam
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.ChatterMessagePage`

searchMessages(communityId, q)

Get messages that match the search criteria in an Experience Cloud site.

API Version

30.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterMessagePage searchMessages(String communityId, String q)
```

Parameters

communityId
Type:String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.ChatterMessagePage`

searchMessages(communityId, pageParam, pageSize, q)

Get a page of messages that match the search criteria in an Experience Cloud site.

API Version

30.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterMessagePage searchMessages(String communityId, String pageParam, Integer pageSize, String q)
```

Parameters

communityId
Type:String

ID for an Experience Cloud site, `internal`, or `null`.

pageParam
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: ConnectApi.ChatterMessagePage

sendMessage(text, recipients)

Send a message to a list of recipients.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterMessage sendMessage(String text, String recipients)
```

Parameters

text
Type: String

Text of the message. Can't be empty or over 10,000 characters.

recipients
Type: String

Up to nine comma-separated IDs of the users receiving the message.

Return Value

Type: ConnectApi.ChatterMessage

sendMessage(communityId, text, recipients)

Send a message to a list of recipients in an Experience Cloud site.

API Version

30.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterMessage sendMessage(String communityId, String text,
String recipients)
```

Parameters

communityId
Type:String

ID for an Experience Cloud site, `internal`, or `null`.

text
Type: String

Text of the message. Can't be empty or over 10,000 characters.

recipients
Type: String

Up to nine comma-separated IDs of users to receive the message.

Return Value

Type: `ConnectApi.ChatterMessage`

ChatterUsers Class

Access information about users, such as activity, followers, subscriptions, files, and groups.

Namespace

ConnectApi

ChatterUsers Methods

These methods are for `ChatterUsers`. All methods are static.

All methods in this class require Chatter and are subject to the per user, per namespace, per hour rate limit.

IN THIS SECTION:

exportUserActivities(communityId, userId)
Export Chatter-related user activity, such as bookmarks, topic endorsements, and votes.

follow(communityId, userId, subjectId)
Follow a user or record.

getChatterSettings(communityId, userId)
Get the default Chatter settings for a user.

getFollowers(communityId, userId)
Get the first page of followers for a user.

getFollowers(communityId, userId, pageParam, pageSize)
Get a page of followers for a user.

getFollowings(communityId, userId)
Get the first page of users and records that a user follows.

getFollowings(communityId, userId, pageParam)
Get a page of users and records that a user follows.

getFollowings(communityId, userId, pageParam, pageSize)
Get a page with the specified number of users and records that a user follows.

getFollowings(communityId, userId, filterType)
Get the first page of records, filtered by key prefix, that a user follows.

getFollowings(communityId, userId, filterType, pageParam)
Get a page of records, filtered by key prefix, that a user follows.

getFollowings(communityId, userId, filterType, pageParam, pageSize)
Get a page with the specified number of records, filtered by key prefix, that a user follows.

getReputation(communityId, userId)
Get a user's reputation.

getUser(communityId, userId)
Get information about a user.

getUserBatch(communityId, userIds)
Get information about a list of users.

getUserGroups(communityId, userId)
Get a user's groups.

getUserGroups(communityId, userId, pageParam, pageSize)
Get a page of a user's groups.

getUsers(communityId)
Get the first page of users.

getUsers(communityId, pageParam, pageSize)
Get a page of users.

purgeUserActivities(communityId, userId)
Start a job to purge Chatter-related user activity, such as bookmarks, topic endorsements, and votes.

searchUserGroupDetails(communityId, userId, q)
Get the user's groups that match the search criteria.

searchUserGroupDetails(communityId, userId, q, pageParam, pageSize)
Get a page of a user's groups that match the search criteria.

searchUsers(communityId, q)
Get the first page of users that match the search criteria.

searchUsers(communityId, q, pageParam, pageSize)
Get a page of users that match the search criteria.

searchUsers(communityId, q, searchContextId, pageParam, pageSize)
Get a page of users that match the search criteria.

updateChatterSettings(communityId, userId, defaultGroupEmailFrequency)
Update the default Chatter settings for a user.

updateUser(communityId, userId, userInput)
Update the About Me section for a user.

`exportUserActivities(communityId, userId)`

Export Chatter-related user activity, such as bookmarks, topic endorsements, and votes.

API Version

42.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserActivitiesJob exportUserActivities(String communityId,
String userId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID of the user.

Return Value

Type: `ConnectApi.UserActivitiesJob`

Usage

The following activities can be exported.

- `Bookmark`—User bookmarked a post.
- `ChatterActivity`—Total counts of posts and comments made and likes and comments received for a user.
- `ChatterLike`—User liked a post or comment.
- `Comment`—User commented on a post.
- `CompanyVerify`—User verified comment.
- `DownVote`—User downvoted a post or comment.
- `FeedEntityRead`—User read a post.
- `FeedRead`—User read a feed.
- `Mute`—User muted a post.
- `Post`—User made a post.
- `TopicEndorsement`—User endorsed another user on a topic or received endorsement on a topic.
- `UpVote`—User upvoted a post or comment.

`follow(communityId, userId, subjectId)`

Follow a user or record.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Subscription follow(String communityId, String userId, String subjectId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for the context user or the keyword `me`.

subjectId
Type: String

ID of the user or record to follow.

Return Value

Type: `ConnectApi.Subscription`

Example

```
ChatterUsers.ConnectApi.Subscription subscriptionToRecord =
ConnectApi.ChatterUsers.follow(null, 'me', '001RR000002G4Y0');
```

Usage

This method creates an EntitySubscription record, which requires certain permissions. See the Usage section of the EntitySubscription object for more information.

SEE ALSO:

Unfollow a Record

getChatterSettings(communityId, userId)

Get the default Chatter settings for a user.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserChatterSettings getChatterSettings(String communityId,
String userId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for the context user or the keyword `me`.

Return Value

Type: `ConnectApi.UserChatterSettings`

getFollowers(communityId, userId)

Get the first page of followers for a user.

API Version

28.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FollowerPage getFollowers(String communityId, String userId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for a user.

Return Value

Type: `ConnectApi.FollowerPage`

getFollowers(communityId, userId, pageParam, pageSize)

Get a page of followers for a user.

API Version

28.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FollowerPage getFollowers(String communityId, String userId,
Integer pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for a user.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.FollowerPage`

getFollowings(communityId, userId)

Get the first page of users and records that a user follows.

API Version

28.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FollowingPage getFollowings(String communityId, String userId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for a user.

Return Value

Type: `ConnectApi.FollowingPage`

getFollowings(communityId, userId, pageParam)

Get a page of users and records that a user follows.

API Version

28.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FollowingPage getFollowings(String communityId, String userId,
Integer pageParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for a user.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

Return Value

Type: `ConnectApi.FollowingPage`

getFollowings(communityId, userId, pageParam, pageSize)

Get a page with the specified number of users and records that a user follows.

API Version

28.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FollowingPage getFollowings(String communityId, String userId,
Integer pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for a user.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.FollowingPage`

getFollowings(communityId, userId, filterType)

Get the first page of records, filtered by key prefix, that a user follows.

API Version

28.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FollowingPage getFollowings(String communityId, String userId,
String filterType)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for a user.

filterType
Type: String

Specifies the key prefix to filter the type of objects returned. A key prefix is the first three characters of the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

Return Value

Type: `ConnectApi.FollowingPage`

getFollowings(communityId, userId, filterType, pageParam)

Get a page of records, filtered by key prefix, that a user follows.

API Version

28.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FollowingPage getFollowings(String communityId, String userId,
String filterType, Integer pageParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for a user.

filterType
Type: String

Specifies the key prefix to filter the type of objects returned. A key prefix is the first three characters of the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

Return Value

Type: ConnectApi.FollowingPage

getFollowings(communityId, userId, filterType, pageParam, pageSize)

Get a page with the specified number of records, filtered by key prefix, that a user follows.

API Version

28.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.FollowingPage getFollowings(String communityId, String userId,
String filterType, Integer pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for a user.

filterType
Type: String

Specifies the key prefix to filter the type of objects returned. A key prefix is the first three characters of the object ID, which specifies the object type. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: ConnectApi.FollowingPage

getReputation(communityId, userId)

Get a user's reputation.

API Version

32.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Reputation getReputation(String communityId, String userId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID of the user.

Return Value

Type: `ConnectApi.Reputation`

getUser(communityId, userId)

Get information about a user.

API Version

28.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserSummary getUser(String communityId, String userId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for a user.

Return Value

Type: `ConnectApi.UserDetail`

Usage

If the user is external, the properties that the `ConnectApi.UserDetail` output class shares with the `ConnectApi.UserSummary` output class can have non-null values. Other properties are always `null`.

getUserBatch(communityId, userIds)

Get information about a list of users.

API Version

31.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.BatchResult[] getUserBatch(String communityId, List<String>
userIds)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userIds
Type: List<String>

A list of up to 500 user IDs.

Return Value

Type: `ConnectApi.BatchResult[]`

The `ConnectApi.BatchResult.getResult()` method returns a `ConnectApi.User` object and errors for users that didn't load.

Example

```
// Get users in an organization.
ConnectApi.UserPage userPage = ConnectApi.ChatterUsers.getUsers(null);

// Create a list of user IDs.
List<String> userList = new List<String>();
for (ConnectApi.User user : userPage.users){
    userList.add(user.id);
}

// Get info about all users in the list.
ConnectApi.BatchResult[] batchResults = ConnectApi.ChatterUsers.getUserBatch(null, userList);

for (ConnectApi.BatchResult batchResult : batchResults) {
    if (batchResult.isSuccess()) {
        // Operation was successful.
        // Print each user's username.
        ConnectApi.UserDetail user;
        if(batchResult.getResult() instanceof ConnectApi.UserDetail) {
            user = (ConnectApi.UserDetail) batchResult.getResult();
        }
        System.debug('SUCCESS');
        System.debug(user.username);
    }
    else {
        // Operation failed. Print errors.
        System.debug('FAILURE');
        System.debug(batchResult.getErrorMessage());
    }
}
```

getUserGroups(communityId, userId)

Get a user's groups.

API Version

45.0

Available to Guest Users

45.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserGroupDetailPage getUserGroups(String communityId, String
userId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for a user.

Return Value

Type: `ConnectApi.UserGroupDetailPage`

getUserGroups(communityId, userId, pageParam, pageSize)

Get a page of a user's groups.

API Version

45.0

Available to Guest Users

45.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserGroupDetailPage getUserGroups(String communityId, String
userId, Integer pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for a user.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.UserGroupDetailPage`

`getUsers(communityId)`

Get the first page of users.

API Version

28.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserPage getUsers(String communityId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

Return Value

Type: `ConnectApi.UserPage`

`getUsers(communityId, pageParam, pageSize)`

Get a page of users.

API Version

28.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserPage getUsers(String communityId, Integer pageParam,
Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.UserPage`

`purgeUserActivities(communityId, userId)`

Start a job to purge Chatter-related user activity, such as bookmarks, topic endorsements, and votes.

API Version

42.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserActivitiesJob purgeUserActivities(String communityId,
String userId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID of the user.

Return Value

Type: `ConnectApi.UserActivitiesJob`

Usage

The following activities can be purged with this method.

- `Bookmark`—User bookmarked a post.
- `ChatterActivity`—Total counts of posts and comments made and likes and comments received for a user.
- `ChatterLike`—User liked a post or comment.
- `CompanyVerify`—User verified comment.
- `DownVote`—User downvoted a post or comment.
- `FeedEntityRead`—User read a post.
- `FeedRead`—User read a feed.
- `Mute`—User muted a post.
- `TopicEndorsement`—User endorsed another user on a topic or received endorsement on a topic.
- `UpVote`—User upvoted a post or comment.

To delete a user's posts and comments, use these methods, respectively.

- `deleteFeedElement(communityId, feedElementId)`
- `deleteComment(communityId, commentId)`

`searchUserGroupDetails(communityId, userId, q)`

Get the user's groups that match the search criteria.

API Version

45.0

Available to Guest Users

45.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserGroupDetailPage searchUserGroupDetails(String communityId,
String userId, String q)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for a user.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.UserGroupDetailPage`

searchUserGroupDetails(communityId, userId, q, pageParam, pageSize)

Get a page of a user's groups that match the search criteria.

API Version

45.0

Available to Guest Users

45.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserGroupDetailPage searchUserGroupDetails(String communityId,
String userId, String q, Integer pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for a user.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.UserGroupDetailPage`

searchUsers(communityId, q)

Get the first page of users that match the search criteria.

API Version

28.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserPage searchUsers(String communityId, String q)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q

Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.UserPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchUsers(communityId, q, result)

Apex Developer Guide: Testing ConnectApi Code

`searchUsers(communityId, q, pageParam, pageSize)`

Get a page of users that match the search criteria.

API Version

28.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserPage searchUsers(String communityId, String q, Integer
pageParam, Integer pageSize)
```

Parameters

communityId

Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q

Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.UserPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchUsers(communityId, q, pageParam, pageSize, result)

Apex Developer Guide: Testing ConnectApi Code

`searchUsers(communityId, q, searchContextId, pageParam, pageSize)`

Get a page of users that match the search criteria.

API Version

28.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserPage searchUsers(String communityId, String q, String
searchContextId, Integer pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

searchContextId
Type: String

A feed item ID that filters search results for feed @mentions. More useful results are listed first. When you specify this argument, you cannot query more than 500 results and you cannot use wildcards in the search term.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.UserPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchUsers(communityId, q, searchContextId, pageParam, pageSize, result)

Apex Developer Guide: Testing ConnectApi Code

`updateChatterSettings(communityId, userId, defaultGroupEmailFrequency)`

Update the default Chatter settings for a user.

API Version

28.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserChatterSettings updateChatterSettings(String communityId,
String userId, ConnectApi.GroupEmailFrequency defaultGroupEmailFrequency)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for the context user or the keyword `me`.

defaultGroupEmailFrequency
Type: `ConnectApi.GroupEmailFrequency`

Frequency with which a user receives email. Values are:

- `EachPost`
- `DailyDigest`
- `WeeklyDigest`
- `Never`
- `UseDefault`

Don't pass the value `UseDefault` for the *defaultGroupEmailFrequency* parameter because calling `updateChatterSettings` sets the default value.

Return Value

Type: `ConnectApi.UserChatterSettings`

updateUser(communityId, userId, userInput)

Update the About Me section for a user.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserDetail updateUser(String communityId, String userId,
ConnectApi.UserInput userInput)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for the context user or the keyword `me`.

userInput
Type: `ConnectApi.UserInput`

Specifies the updated information.

Return Value

Type: `ConnectApi.UserDetail`

ChatterUsers Test Methods

These test methods are for `ChatterUsers`. All methods are static.

For information about using these methods to test your `ConnectApi` code, see Testing ConnectApi Code.

IN THIS SECTION:

setTestSearchUsers(communityId, q, result)
Register a `ConnectApi.UserPage` object to be returned when the matching `ConnectApi.searchUsers` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchUsers(communityId, q, pageParam, pageSize, result)
Register a `ConnectApi.UserPage` object to be returned when the matching `ConnectApi.searchUsers` method is called in a test context. Use the method with the same parameters or you receive an exception.

setTestSearchUsers(communityId, q, searchContextId, pageParam, pageSize, result)
Register a `ConnectApi.UserPage` object to be returned when the matching `ConnectApi.searchUsers` method is called in a test context. Use the method with the same parameters or you receive an exception.

`setTestSearchUsers(communityId, q, result)`

Register a `ConnectApi.UserPage` object to be returned when the matching `ConnectApi.searchUsers` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

28.0

Signature

```
public static Void setTestSearchUsers(String communityId, String q, ConnectApi.UserPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

result
Type: `ConnectApi.UserPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchUsers(communityId, q)

Apex Developer Guide: Testing ConnectApi Code

setTestSearchUsers(communityId, q, pageParam, pageSize, result)

Register a `ConnectApi.UserPage` object to be returned when the matching `ConnectApi.searchUsers` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

28.0

Signature

```
public static Void setTestSearchUsers(String communityId, String q, Integer pageParam,
Integer pageSize, ConnectApi.UserPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

result
Type: `ConnectApi.UserPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchUsers(communityId, q, pageParam, pageSize)

Apex Developer Guide: Testing ConnectApi Code

setTestSearchUsers(communityId, q, searchContextId, pageParam, pageSize, result)

Register a `ConnectApi.UserPage` object to be returned when the matching `ConnectApi.searchUsers` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

28.0

Signature

```
public static Void setTestSearchUsers(String communityId, String q, String
searchContextId, Integer pageParam, Integer pageSize, ConnectApi.UserPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

searchContextId
Type: String

A feed item ID that filters search results for feed @mentions. More useful results are listed first. When you specify this argument, you cannot query more than 500 results and you cannot use wildcards in the search term.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

result
Type: `ConnectApi.UserPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

searchUsers(communityId, q, searchContextId, pageParam, pageSize)

Apex Developer Guide: Testing ConnectApi Code

Retired ChatterUsers Methods

These methods for `ChatterUsers` are retired.

All methods in this class require Chatter and are subject to the per user, per namespace, per hour rate limit.

IN THIS SECTION:

deletePhoto(communityId, userId)
Delete a user's photo.

getGroups(communityId, userId)
Get the groups that a user is a member of.

getGroups(communityId, userId, pageParam, pageSize)
Get a page of groups that a user is a member of.

getPhoto(communityId, userId)
Get a user's photo.

searchUserGroups(communityId, userId, q)
Get the user's groups that match the search criteria.

searchUserGroups(communityId, userId, q, pageParam, pageSize)
Get a page of a user's groups that match the search criteria.

setPhoto(communityId, userId, fileId, versionNumber)
Set an uploaded file as a user's photo.

setPhoto(communityId, userId, fileUpload)
Set a file that hasn't been uploaded as the user's photo.

setPhotoWithAttributes(communityId, userId, photo)
Set and crop an uploaded file as a user's photo.

setPhotoWithAttributes(communityId, userId, photo, fileUpload)
Set and crop a file that hasn't been uploaded as a user's photo.

`deletePhoto(communityId, userId)`

Delete a user's photo.

API Version

28.0–34.0

Important: In version 35.0 and later, use `ConnectApi.UserProfiles.deletePhoto(communityId, userId)`

Signature

```
public static Void deletePhoto(String communityId, String userId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for the context user or the keyword `me`.

Return Value

Type: Void

getGroups(communityId, userId)

Get the groups that a user is a member of.

API Version

28.0–44.0

Important: In version 45.0 and later, use `getUserGroups(communityId, userId)`.

Available to Guest Users

32.0–44.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserGroupPage getGroups(String communityId, String userId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for a user.

Return Value

Type: `ConnectApi.UserGroupPage`

getGroups(communityId, userId, pageParam, pageSize)

Get a page of groups that a user is a member of.

API Version

28.0–44.0

Important: In version 45.0 and later, use `getUserGroups(communityId, userId, pageParam, pageSize)`.

Available to Guest Users

32.0–44.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserGroupPage getGroups(String communityId, String userId,
Integer pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for a user.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.UserGroupPage`

getPhoto(communityId, userId)

Get a user's photo.

API Version

28.0–34.0

Important: In version 35.0 and later, use `ConnectApi.UserProfiles.getPhoto(communityId, userId)`.

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Photo getPhoto(String communityId, String userId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for a user.

Return Value

Type: `ConnectApi.Photo`

searchUserGroups(communityId, userId, q)

Get the user's groups that match the search criteria.

API Version

30.0–44.0

Important: In version 45.0 and later, use `searchUserGroupDetails(communityId, userId, q)`.

Available to Guest Users

32.0–44.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserGroupPage searchUserGroups(String communityId, String userId, String q)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for a user.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

Return Value

Type: `ConnectApi.UserGroupPage`

searchUserGroups(communityId, userId, q, pageParam, pageSize)

Get a page of a user's groups that match the search criteria.

API Version

30.0–44.0

Important: In version 45.0 and later, use `searchUserGroupDetails(communityId, userId, q, pageParam, pageSize)`.

Available to Guest Users

32.0–44.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserGroupPage searchUserGroups(String communityId, String userId, String q, Integer pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for a user.

q
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.UserGroupPage`

setPhoto(communityId, userId, fileId, versionNumber)

Set an uploaded file as a user's photo.

API Version

28.0–34.0

Important: In version 35.0 and later, use `ConnectApi.UserProfiles.setPhoto(communityId, userId, fileId, versionNumber)`

Requires Chatter

Yes

Signature

```
public static ConnectApi.Photo setPhoto(String communityId, String userId, String fileId, Integer versionNumber)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for the context user or the keyword `me`.

fileId
Type: String

ID of a file already uploaded. The file must be an image, and be smaller than 2 GB.

versionNumber
Type: Integer

Version number of the existing file. Specify either an existing version number, or `null` to get the latest version.

Return Value

Type: `ConnectApi.Photo`

Usage

Photos are processed asynchronously and might not be visible right away.

`setPhoto(communityId, userId, fileUpload)`

Set a file that hasn't been uploaded as the user's photo.

API Version

28.0–34.0

Important: In version 35.0 and later, use `ConnectApi.UserProfiles.setPhoto(communityId, userId, fileUpload)`

Requires Chatter

Yes

Signature

```
public static ConnectApi.Photo setPhoto(String communityId, String userId,
ConnectApi.BinaryInput fileUpload)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for the context user or the keyword `me`.

fileUpload
Type: `ConnectApi.BinaryInput`

File to use as the photo. The content type must be usable as an image.

Return Value

Type: `ConnectApi.Photo`

Usage

Photos are processed asynchronously and might not be visible right away.

`setPhotoWithAttributes(communityId, userId, photo)`

Set and crop an uploaded file as a user's photo.

API Version

29.0–34.0

Important: In version 35.0 and later, use `ConnectApi.UserProfiles.setPhotoWithAttributes(communityId, userId, photo)`

Requires Chatter

Yes

Signature

```
public static ConnectApi.Photo setPhotoWithAttributes(String communityId, String userId,
ConnectApi.PhotoInput photo)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for the context user or the keyword `me`.

photo
Type: `ConnectApi.PhotoInput`

A `ConnectApi.PhotoInput` object specifying the file ID, version number, and cropping parameters.

Return Value

Type: `ConnectApi.Photo`

Usage

Photos are processed asynchronously and might not be visible right away.

`setPhotoWithAttributes(communityId, userId, photo, fileUpload)`

Set and crop a file that hasn't been uploaded as a user's photo.

API Version

29.0–34.0

Important: In version 35.0 and later, use `ConnectApi.UserProfiles.setPhotoWithAttributes(communityId, userId, photo, fileUpload)`

Requires Chatter

Yes

Signature

```
public static ConnectApi.Photo setPhotoWithAttributes(String communityId, String userId,
ConnectApi.PhotoInput photo, ConnectApi.BinaryInput fileUpload)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for the context user or the keyword `me`.

photo
Type: `ConnectApi.PhotoInput`

A `ConnectApi.PhotoInput` object specifying the cropping parameters.

fileUpload
Type: `ConnectApi.BinaryInput`

File to use as the photo. The content type must be usable as an image.

Return Value

Type: `ConnectApi.Photo`

Usage

Photos are processed asynchronously and might not be visible right away.

Clm Class

Create and update Contract Lifecycle Management (CLM) contracts using object ID.

Namespace

ConnectApi

Clm Methods

These methods are for `Clm`. All methods are static.

IN THIS SECTION:

createContract(contractInputPayload)
Create contracts using the object ID.

updateContract(contractInputPayload)
Update contracts using the object ID.

`createContract(contractInputPayload)`

Create contracts using the object ID.

API Version

56.0

Requires Chatter

No

Signature

```
public static ConnectApi.ContractOutputRepresentation
createContract(ConnectApi.ContractInputRepresentation contractInputPayload)
```

Parameters

`contractInputPayload`
Type: `ConnectApi.ContractInputRepresentation` on page 1843

Input payload to create contract.

Return Value

Type: `ConnectApi.ContractOutputRepresentation` on page 2060

`updateContract(contractInputPayload)`

Update contracts using the object ID.

API Version

56.0

Requires Chatter

No

Signature

```
public static ConnectApi.ContractOutputRepresentation
updateContract(ConnectApi.ContractInputRepresentation contractInputPayload)
```

Parameters

`contractInputPayload`
Type: `ConnectApi.ContractInputRepresentation` on page 1843

Input payload to update contract.

Return Value

Type: `ConnectApi.ContractOutputRepresentation` on page 2060

CommerceBuyerExperience Class

Create, delete, or get commerce addresses. Get order delivery group, order item, order shipments, shipment items, and order summaries. Get adjustments for order items and order summaries.

Namespace

ConnectApi

CommerceBuyerExperience Methods

These methods are for `CommerceBuyerExperience`. All methods are static.

IN THIS SECTION:

addOrderToCart(webstoreId, orderSummaryId, orderToCartInput)
Add an order to a cart using a webstore order summary.

addOrderToCart(webstoreId, orderSummaryId, orderToCartInput, effectiveAccountId)
Add an order to a cart for a specific account using a webstore order summary.

calculateAdjustmentAggregates(webstoreId, orderSummaryIds)
Submit a job to calculate adjustment aggregates for a list of order summary IDs.

createCommerceAccountAddress(webstoreId, accountId, addressInput)
Create a Commerce account address for a webstore account.

deleteCommerceAccountAddress(webstoreId, accountId, addressId)
Delete a Commerce account address for a webstore.

getCommerceAccountAddress(webstoreId, accountId)
Get a Commerce account address for a webstore.

getCommerceAccountAddress(webstoreId, accountId, defaultOnly)
Get Commerce account addresses for a webstore and account.

getCommerceAccountAddress(webstoreId, accountId, defaultOnly, addressType, fields, pageToken, pageSize, sortOrder)
Get Commerce account addresses for a webstore and account.

getCommerceAccountAddress(webstoreId, accountId, addressType, excludeUnsupportedCountries)
Get Commerce account addresses for a webstore and account.

getCommerceAccountAddress(webstoreId, accountId, defaultOnly, addressType, excludeUnsupportedCountries)
Get Commerce account addresses for a webstore and account.

getCommerceAccountAddress(webstoreId, accountId, defaultOnly, addressType, excludeUnsupportedCountries, fields, pageToken, pageSize, sortOrder)
Get Commerce account addresses for a webstore and account.

getOrderDeliveryGroupSummaries(webstoreId, effectiveAccountId, orderSummaryId)
Get order delivery group summaries.

getOrderDeliveryGroupSummaries(webstoreId, effectiveAccountId, orderSummaryId, pageSize)
Get order delivery group summaries.

getOrderDeliveryGroupSummaries(webstoreId, effectiveAccountId, orderSummaryId, pageParam)
Get a page of order delivery group summaries.

getOrderDeliveryGroupSummaries(webstoreId, effectiveAccountId, orderSummaryId, fields)
Get order delivery group summaries with specific fields.

getOrderDeliveryGroupSummaries(webstoreId, effectiveAccountId, orderSummaryId, pageParam, fields)
Get a page of order delivery group summaries with specific fields.

getOrderDeliveryGroupSummaries(webstoreId, effectiveAccountId, orderSummaryId, fields, pageSize)
Get order delivery group summaries with specific fields.

getOrderDeliveryGroupSummaries(webstoreId, effectiveAccountId, orderSummaryId, fields, sortParam)
Get a sorted list of order delivery group summaries with specific fields.

getOrderDeliveryGroupSummaries(webstoreId, effectiveAccountId, orderSummaryId, fields, pageSize, sortParam)
Get a sorted list of order delivery group summaries with specific fields.

getOrderItemSummaries(webstoreId, effectiveAccountId, orderSummaryId)
Get order item summaries.

getOrderItemSummaries(webstoreId, effectiveAccountId, orderSummaryId, pageSize)
Get order item summaries.

getOrderItemSummaries(webstoreId, effectiveAccountId, orderSummaryId, orderDeliveryGroupSummaryId)
Get order item summaries for a delivery group summary.

getOrderItemSummaries(webstoreId, effectiveAccountId, orderSummaryId, orderDeliveryGroupSummaryId, pageSize)
Get order item summaries for a delivery group summary.

getOrderItemSummaries(webstoreId, effectiveAccountId, orderSummaryId, orderDeliveryGroupSummaryId, pageParam)
Get a page of order item summaries for a delivery group summary.

getOrderItemSummaries(webstoreId, effectiveAccountId, orderSummaryId, orderDeliveryGroupSummaryId, fields)
Get order item summaries for a delivery group summary with specific fields.

getOrderItemSummaries(webstoreId, effectiveAccountId, orderSummaryId, orderDeliveryGroupSummaryId, fields, pageSize)
Get order item summaries for a delivery group summary with specific fields.

getOrderItemSummaries(webstoreId, effectiveAccountId, orderSummaryId, orderDeliveryGroupSummaryId, fields, pageParam)
Get a page of order item summaries for a delivery group summary with specific fields.

getOrderItemSummaries(webstoreId, effectiveAccountId, orderSummaryId, orderDeliveryGroupSummaryId, fields, sortParam)
Get a sorted list of order item summaries for a delivery group summary with specific fields.

getOrderItemSummaries(webstoreId, effectiveAccountId, orderSummaryId, orderDeliveryGroupSummaryId, fields, pageSize, sortParam)
Get a sorted page of order item summaries for a delivery group summary with specific fields.

getOrderItemSummaries(webstoreId, effectiveAccountId, orderSummaryId, orderDeliveryGroupSummaryId, fields, pageParam, pageSize, sortParam, includeAdjustmentDetails)
Get a sorted page of order item summaries for a delivery group summary with specific fields and include adjustment details.

getOrderItemSummaryAdjustments(webstoreId, orderSummaryId, orderItemSummaryAdjustmentCollectionInput)
Get adjustments for order items.

getOrderItemSummaryAdjustments(webstoreId, orderSummaryId, orderItemSummaryAdjustmentCollectionInput, effectiveAccountId)
Get adjustments for order items.

getOrderShipmentItems(webstoreId, shipmentId)
Get order shipment items.

getOrderShipmentItems(webstoreId, shipmentId, effectiveAccountId)
Get order shipment items.

getOrderShipmentItems(webstoreId, shipmentId, effectiveAccountId, fields)
Get order shipment items with specific fields.

getOrderShipmentItems(webstoreId, shipmentId, effectiveAccountId, fields, pageToken, pageSize)
Get a page of order shipment items with specific fields.

getOrderShipmentItems(webstoreId, shipmentId, effectiveAccountId, fields, pageToken, pageSize, sortOrder)
Get a sorted page of order shipment items.

getOrderShipments(webstoreId, orderSummaryId)
Get order shipments.

getOrderShipments(webstoreId, orderSummaryId, effectiveAccountId)
Get order shipments.

getOrderShipments(webstoreId, orderSummaryId, effectiveAccountId, fields)
Get order shipments with specific fields.

getOrderShipments(webstoreId, orderSummaryId, effectiveAccountId, fields, pageSize, pageToken)
Get a page of order shipments with specific fields.

getOrderShipments(webstoreId, orderSummaryId, effectiveAccountId, fields, pageSize, pageToken, sortOrder)
Get a sorted page of order shipments with specific fields.

getOrderSummaries(webstoreId)
Get order summaries.

getOrderSummaries(webstoreId, effectiveAccountId)
Get order summaries.

getOrderSummaries(webstoreId, effectiveAccountId, fields)
Get order summaries with specific fields.

getOrderSummaries(webstoreId, effectiveAccountId, fields, pageSize, pageToken)
Get a page of order summaries with specific fields.

getOrderSummaries(webstoreId, effectiveAccountId, fields, pageSize, pageToken, sortOrder)
Get a sorted page of order summaries with specific fields.

getOrderSummaries(webstoreId, effectiveAccountId, fields, pageSize, pageToken, sortOrder, earliestDate, latestDate)
Get a sorted page of order summaries with specific fields within a specific date range.

getOrderSummaries(webstoreId, effectiveAccountId, fields, pageSize, pageToken, sortOrder, earliestDate, latestDate, ownerScoped)
Get a sorted page of order summaries with specific fields within a specific date range and scoped to orders owned by the context user.

getOrderSummaries(webstoreId, effectiveAccountId, fields, pageSize, pageToken, sortOrder, earliestDate, latestDate, ownerScoped, includeAdjustmentDetails)
Get a sorted page of order summaries with specific fields within a specific date range and scoped to orders owned by the context user.

getOrderSummary(webstoreId, orderSummaryId, effectiveAccountId)
Get an order summary.

getOrderSummary(webstoreId, orderSummaryId, effectiveAccountId, fields)
Get an order summary with fields.

getOrderSummary(webstoreId, orderSummaryId, effectiveAccountId, fields, includeAdjustmentDetails)
Get an order summary with fields and include adjustment details.

getOrderSummaryAdjustments(webstoreId, orderSummaryId)
Get adjustments for an order summary.

getOrderSummaryAdjustments(webstoreId, orderSummaryId, effectiveAccountId)
Get adjustments for an order summary.

lookupOrderSummary(webstoreId, effectiveAccountId, fields, excludeLineItems, excludeDeliveryGroups, excludeAdjustmentAggregates, excludeAdjustments, deliveryGroupId, orderSummaryLookupInput) (Developer Preview)
Look up details about an order summary for a guest shopper or a registered buyer using the effective account ID, requested fields, line items, delivery groups, adjustments aggregates, and adjustments.

lookupOrderSummary(webstoreId, effectiveAccountId, fields, orderSummaryLookupInput) (Developer Preview)
Look up details about an order summary for a guest shopper or a registered buyer using the effective account ID and requested fields.

lookupOrderSummary(webstoreId, effectiveAccountId, orderSummaryLookupInput) (Developer Preview)
Look up details about an order summary for a guest shopper or a registered buyer using the effective account ID.

updateCommerceAccountAddress(webstoreId, accountId, addressId, addressInput)
Update a Commerce account address for a webstore.

`addOrderToCart(webstoreId, orderSummaryId, orderToCartInput)`

Add an order to a cart using a webstore order summary.

API Version

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderToCartResult addOrderToCart(String webstoreId, String orderSummaryId, ConnectApi.OrderToCartInput orderToCartInput)
```

Parameters

webstoreId
Type: String

ID of the webstore.

orderSummaryId
Type: String

ID of the order summary.

orderToCartInput
Type: ConnectApi.OrderToCartInput

Input value indicating which cart the order should be added to.

Return Value

Type: ConnectApi.OrderToCartResult

addOrderToCart(webstoreId, orderSummaryId, orderToCartInput, effectiveAccountId)

Add an order to a cart for a specific account using a webstore order summary.

API Version

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderToCartResult addOrderToCart(String webstoreId, String orderSummaryId, ConnectApi.OrderToCartInput orderToCartInput, String effectiveAccountId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

orderSummaryId
Type: String

ID of the order summary.

orderToCartInput
Type: ConnectApi.OrderToCartInput

Input value indicating which cart the order should be added to.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

Return Value

Type: ConnectApi.OrderToCartResult

calculateAdjustmentAggregates(webstoreId, orderSummaryIds)

Submit a job to calculate adjustment aggregates for a list of order summary IDs.

API Version

55.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderSummaryAdjustmentAggregatesAsyncOutput
calculateAdjustmentAggregates(String webstoreId,
ConnectApi.OrderSummaryAdjustmentAggregatesAsyncInput orderSummaryIds)
```

Parameters

webstoreId
Type: String

ID of the webstore.

orderSummaryIds
Type: ConnectApi.OrderSummaryAdjustmentAggregatesAsyncInput

A `ConnectApi.OrderSummaryAdjustmentAggregatesAsyncInput` class with a list of order summary IDs.

Return Value

Type: ConnectApi.OrderSummaryAdjustmentAggregatesAsyncOutput

createCommerceAccountAddress(webstoreId, accountId, addressInput)

Create a Commerce account address for a webstore account.

API Version

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.CommerceAddressOutput createCommerceAccountAddress(String
webstoreId, String accountId, ConnectApi.commerceAddressInput addressInput)
```

Parameters

webstoreId
Type: String

ID of the webstore.

accountId
Type: String

ID of the account.

addressInput
Type: ConnectApi.commerceAddressInput

Information about the address you want to create.

Return Value

Type: ConnectApi.CommerceAddressOutput

deleteCommerceAccountAddress(webstoreId, accountId, addressId)

Delete a Commerce account address for a webstore.

API Version

54.0

Requires Chatter

No

Signature

```
public static Void deleteCommerceAccountAddress(String webstoreId, String accountId,
String addressId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

accountId
Type: String

ID of the account.

addressId
Type: String

ID of the address.

Return Value

Type: Void

getCommerceAccountAddress(webstoreId, accountId)

Get a Commerce account address for a webstore.

API Version

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.CommerceAddressCollection getCommerceAccountAddress(String
webstoreId, String accountId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

accountId
Type: String

ID of the account.

Return Value

Type: ConnectApi.CommerceAddressCollection

getCommerceAccountAddress(webstoreId, accountId, defaultOnly)

Get Commerce account addresses for a webstore and account.

You can get the default address by itself, or you can get all of the addresses for the account.

API Version

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.CommerceAddressCollection getCommerceAccountAddress(String webstoreId, String accountId, Boolean defaultOnly)
```

Parameters

webstoreId
Type: String

ID of the webstore.

accountId
Type: String

ID of the account.

defaultOnly
Type: Boolean

Indicate if you only want the default address (`true`) or all addresses for the account (`false`). The default value is `false`.

Return Value

Type: ConnectApi.CommerceAddressCollection

getCommerceAccountAddress(webstoreId, accountId, defaultOnly, addressType, fields, pageToken, pageSize, sortOrder)

Get Commerce account addresses for a webstore and account.

API Version

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.CommerceAddressCollection getCommerceAccountAddress(String
webstoreId, String accountId, Boolean defaultOnly, List<String> addressType, List<String>
fields, String pageToken, Integer pageSize, ConnectApi.CommerceAddressSort sortOrder)
```

Parameters

webstoreId
Type: String

ID of the webstore.

accountId
Type: String

ID of the account.

defaultOnly
Type: Boolean

Indicate if you want only the default address (`true`) or all addresses for the account (`false`). The default value is `false`.

addressType
Type: List<String>

Type of address, for example, `Billing` or `Shipping`.

fields
Type: List<String>

A list of custom fields for the address.

pageToken
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortOrder
Type: `ConnectApi.CommerceAddressSort`

Sort order for Commerce addresses.

- `CreatedDateAsc`—Sort in ascending order of created date.
- `CreatedDateDesc`—Sort in descending order of created date.
- `NameAsc`—Sort in ascending order of name.
- `NameDesc`—Sort in descending order of name.

Return Value

Type: ConnectApi.CommerceAddressCollection

getCommerceAccountAddress(webstoreId, accountId, addressType, excludeUnsupportedCountries)

Get Commerce account addresses for a webstore and account.

API Version

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CommerceAddressCollection getCommerceAccountAddress(String webstoreId, String accountId, List<String> addressType, Boolean excludeUnsupportedCountries)
```

Parameters

webstoreId
Type: String

ID of the webstore.

accountId
Type: String

ID of the account.

addressType
Type: List<String>

Type of address, for example, `Billing` or `Shipping`.

excludeUnsupportedCountries
Type: Boolean

Indicate if you want to retrieve all addresses (`false`) or only addresses of type Shipping that are in countries included in the store's shipToCountries list (`true`). The default value is `false`.

Return Value

Type: ConnectApi.CommerceAddressCollection

getCommerceAccountAddress(webstoreId, accountId, defaultOnly, addressType, excludeUnsupportedCountries)

Get Commerce account addresses for a webstore and account.

API Version

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CommerceAddressCollection getCommerceAccountAddress(String webstoreId, String accountId, Boolean defaultOnly, List<String> addressType, Boolean excludeUnsupportedCountries)
```

Parameters

webstoreId
Type: String

ID of the webstore.

accountId
Type: String

ID of the account.

defaultOnly
Type: Boolean

Indicate if you want only the default address (`true`) or all addresses for the account (`false`). The default value is `false`.

addressType
Type: List<String>

Type of address, for example, `Billing` or `Shipping`.

excludeUnsupportedCountries
Type: Boolean

Indicate if you want to retrieve all addresses (`false`) or only addresses of type Shipping that are in countries included in the store's shipToCountries list (`true`). The default value is `false`.

Return Value

Type: `ConnectApi.CommerceAddressCollection`

`getCommerceAccountAddress(webstoreId, accountId, defaultOnly, addressType, excludeUnsupportedCountries, fields, pageToken, pageSize, sortOrder)`

Get Commerce account addresses for a webstore and account.

API Version

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CommerceAddressCollection getCommerceAccountAddress(String webstoreId, String accountId, Boolean defaultOnly, List<String> addressType, Boolean excludeUnsupportedCountries, List<String> fields, String pageToken, Integer pageSize, ConnectApi.CommerceAddressSort sortOrder)
```

Parameters

webstoreId
Type: String

ID of the webstore.

accountId
Type: String

ID of the account.

defaultOnly
Type: Boolean

Indicate if you want only the default address (`true`) or all addresses for the account (`false`). The default value is `false`.

addressType
Type: List<String>

Type of address, for example, `Billing` or `Shipping`.

excludeUnsupportedCountries
Type: Boolean

Indicate if you want to retrieve all addresses (`false`) or only addresses of type Shipping that are in countries included in the store's shipToCountries list (`true`). The default value is `false`.

fields
Type: List<String>

A list of custom fields for the address.

pageToken
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortOrder
Type: `ConnectApi.CommerceAddressSort`

Sort order for Commerce addresses. Values are:

- `CreatedDateAsc`—Sort in ascending order of created date.
- `CreatedDateDesc`—Sort in descending order of created date.

- `NameAsc`—Sort in ascending order of name.
- `NameDesc`—Sort in descending order of name.

Return Value

Type: `ConnectApi.CommerceAddressCollection`

getOrderDeliveryGroupSummaries(webstoreId, effectiveAccountId, orderSummaryId)

Get order delivery group summaries.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderDeliveryGroupSummaryCollection
getOrderDeliveryGroupSummaries(String webstoreId, String effectiveAccountId, String
orderSummaryId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

orderSummaryId
Type: String

ID of the order summary.

Return Value

Type: `ConnectApi.OrderDeliveryGroupSummaryCollection`

getOrderDeliveryGroupSummaries(webstoreId, effectiveAccountId, orderSummaryId, pageSize)

Get order delivery group summaries.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderDeliveryGroupSummaryCollection
getOrderDeliveryGroupSummaries(String webstoreId, String effectiveAccountId, String
orderSummaryId, Integer pageSize)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

orderSummaryId
Type: String

ID of the order summary.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.OrderDeliveryGroupSummaryCollection`

`getOrderDeliveryGroupSummaries(webstoreId, effectiveAccountId, orderSummaryId, pageParam)`

Get a page of order delivery group summaries.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderDeliveryGroupSummaryCollection
getOrderDeliveryGroupSummaries(String webstoreId, String effectiveAccountId, String
orderSummaryId, String pageParam)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

orderSummaryId
Type: String

ID of the order summary.

pageParam
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

Return Value

Type: `ConnectApi.OrderDeliveryGroupSummaryCollection`

`getOrderDeliveryGroupSummaries(webstoreId, effectiveAccountId, orderSummaryId, fields)`

Get order delivery group summaries with specific fields.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderDeliveryGroupSummaryCollection
getOrderDeliveryGroupSummaries(String webstoreId, String effectiveAccountId, String
orderSummaryId, List<String> fields)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

orderSummaryId
Type: String

ID of the order summary.

fields
Type: List<String>

List of up to 15 order delivery group summary or order delivery method fields to display in the UI in each item row. For example, `fields=OrderDeliveryGroupSummary.DeliveryAddress, OrderDeliveryMethod.Name`.

Return Value

Type: `ConnectApi.OrderDeliveryGroupSummaryCollection`

getOrderDeliveryGroupSummaries(webstoreId, effectiveAccountId, orderSummaryId, pageParam, fields)

Get a page of order delivery group summaries with specific fields.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderDeliveryGroupSummaryCollection
getOrderDeliveryGroupSummaries(String webstoreId, String effectiveAccountId, String
orderSummaryId, String pageParam, List<String> fields)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

orderSummaryId
Type: String

ID of the order summary.

pageParam
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

fields
Type: List<String>

List of up to 15 order delivery group summary or order delivery method fields to display in the UI in each item row. For example, `fields=OrderDeliveryGroupSummary.DeliveryAddress, OrderDeliveryMethod.Name`.

Return Value

Type: `ConnectApi.OrderDeliveryGroupSummaryCollection`

getOrderDeliveryGroupSummaries(webstoreId, effectiveAccountId, orderSummaryId, fields, pageSize)

Get order delivery group summaries with specific fields.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderDeliveryGroupSummaryCollection
getOrderDeliveryGroupSummaries(String webstoreId, String effectiveAccountId, String
orderSummaryId, List<String> fields, Integer pageSize)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

orderSummaryId
Type: String

ID of the order summary.

fields
Type: List<String>

List of up to 15 order delivery group summary or order delivery method fields to display in the UI in each item row. For example, `fields=OrderDeliveryGroupSummary.DeliveryAddress, OrderDeliveryMethod.Name`.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.OrderDeliveryGroupSummaryCollection`

getOrderDeliveryGroupSummaries(webstoreId, effectiveAccountId, orderSummaryId, fields, sortParam)

Get a sorted list of order delivery group summaries with specific fields.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderDeliveryGroupSummaryCollection
getOrderDeliveryGroupSummaries(String webstoreId, String effectiveAccountId, String
orderSummaryId, List<String> fields, ConnectApi.OrderDeliveryGroupSummarySort sortParam)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

orderSummaryId
Type: String

ID of the order summary.

fields
Type: List<String>

List of up to 15 order delivery group summary or order delivery method fields to display in the UI in each item row. For example, `fields=OrderDeliveryGroupSummary.DeliveryAddress, OrderDeliveryMethod.Name`.

sortParam
Type: ConnectApi.OrderDeliveryGroupSummarySort

Sort order for order delivery group summaries. Values are:

- IdAsc—Sorts by ID in ascending alphanumeric order (A–Z, 0–9).
- IdDesc—Sorts by ID in descending alphanumeric order (Z–A, 9–0).

If null, IdAsc is the default sort order.

Return Value

Type: ConnectApi.OrderDeliveryGroupSummaryCollection

getOrderDeliveryGroupSummaries(webstoreId, effectiveAccountId, orderSummaryId, fields, pageSize, sortParam)

Get a sorted list of order delivery group summaries with specific fields.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderDeliveryGroupSummaryCollection
getOrderDeliveryGroupSummaries(String webstoreId, String effectiveAccountId, String
orderSummaryId, List<String> fields, Integer pageSize,
ConnectApi.OrderDeliveryGroupSummarySort sortParam)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If null, defaults to the account ID for the context user.

orderSummaryId
Type: String

ID of the order summary.

fields
Type: List<String>

List of up to 15 order delivery group summary or order delivery method fields to display in the UI in each item row. For example, fields=OrderDeliveryGroupSummary.DeliveryAddress, OrderDeliveryMethod.Name.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.OrderDeliveryGroupSummarySort`

Sort order for order delivery group summaries. Values are:

- `IdAsc`—Sorts by ID in ascending alphanumeric order (A–Z, 0–9).
- `IdDesc`—Sorts by ID in descending alphanumeric order (Z–A, 9–0).

If `null`, `IdAsc` is the default sort order.

Return Value

Type: `ConnectApi.OrderDeliveryGroupSummaryCollection`

`getOrderItemSummaries(webstoreId, effectiveAccountId, orderSummaryId)`

Get order item summaries.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderItemSummaryCollection getOrderItemSummaries(String
webstoreId, String effectiveAccountId, String orderSummaryId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

orderSummaryId
Type: String

ID of the order summary.

Return Value

Type: `ConnectApi.OrderItemSummaryCollection`

getOrderItemSummaries(webstoreId, effectiveAccountId, orderSummaryId, pageSize)

Get order item summaries.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderItemSummaryCollection getOrderItemSummaries(String webstoreId, String effectiveAccountId, String orderSummaryId, Integer pageSize)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

orderSummaryId
Type: String

ID of the order summary.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.OrderItemSummaryCollection`

getOrderItemSummaries(webstoreId, effectiveAccountId, orderSummaryId, orderDeliveryGroupSummaryId)

Get order item summaries for a delivery group summary.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderItemSummaryCollection getOrderItemSummaries(String
webstoreId, String effectiveAccountId, String orderSummaryId, String
orderDeliveryGroupSummaryId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

orderSummaryId
Type: String

ID of the order summary.

orderDeliveryGroupSummaryId
Type: String

ID of the order delivery group summary.

Return Value

Type: `ConnectApi.OrderItemSummaryCollection`

`getOrderItemSummaries(webstoreId, effectiveAccountId, orderSummaryId, orderDeliveryGroupSummaryId, pageSize)`

Get order item summaries for a delivery group summary.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderItemSummaryCollection getOrderItemSummaries(String
webstoreId, String effectiveAccountId, String orderSummaryId, String
orderDeliveryGroupSummaryId, Integer pageSize)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

orderSummaryId
Type: String

ID of the order summary.

orderDeliveryGroupSummaryId
Type: String

ID of the order delivery group summary.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.OrderItemSummaryCollection`

getOrderItemSummaries(webstoreId, effectiveAccountId, orderSummaryId, orderDeliveryGroupSummaryId, pageParam)

Get a page of order item summaries for a delivery group summary.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderItemSummaryCollection getOrderItemSummaries(String
webstoreId, String effectiveAccountId, String orderSummaryId, String
orderDeliveryGroupSummaryId, String pageParam)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

orderSummaryId
Type: String

ID of the order summary.

orderDeliveryGroupSummaryId
Type: String

ID of the order delivery group summary.

pageParam
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

Return Value

Type: `ConnectApi.OrderItemSummaryCollection`

getOrderItemSummaries(webstoreId, effectiveAccountId, orderSummaryId, orderDeliveryGroupSummaryId, fields)

Get order item summaries for a delivery group summary with specific fields.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderItemSummaryCollection getOrderItemSummaries(String webstoreId, String effectiveAccountId, String orderSummaryId, String orderDeliveryGroupSummaryId, List<String> fields)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

orderSummaryId
Type: String

ID of the order summary.

orderDeliveryGroupSummaryId
Type: String

ID of the order delivery group summary.

fields
Type: List<String>

List of up to 15 order item summary or product fields to display in the UI in each item row.

Return Value

Type: `ConnectApi.OrderItemSummaryCollection`

getOrderItemSummaries(webstoreId, effectiveAccountId, orderSummaryId, orderDeliveryGroupSummaryId, fields, pageSize)

Get order item summaries for a delivery group summary with specific fields.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderItemSummaryCollection getOrderItemSummaries(String
webstoreId, String effectiveAccountId, String orderSummaryId, String
orderDeliveryGroupSummaryId, List<String> fields, Integer pageSize)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

orderSummaryId
Type: String

ID of the order summary.

orderDeliveryGroupSummaryId
Type: String

ID of the order delivery group summary.

fields
Type: List<String>

List of up to 15 order item summary or product fields to display in the UI in each item row.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.OrderItemSummaryCollection`

getOrderItemSummaries(webstoreId, effectiveAccountId, orderSummaryId, orderDeliveryGroupSummaryId, fields, pageParam)

Get a page of order item summaries for a delivery group summary with specific fields.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderItemSummaryCollection getOrderItemSummaries(String webstoreId, String effectiveAccountId, String orderSummaryId, String orderDeliveryGroupSummaryId, List<String> fields, String pageParam)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

orderSummaryId
Type: String

ID of the order summary.

orderDeliveryGroupSummaryId
Type: String

ID of the order delivery group summary.

fields
Type: List<String>

List of up to 15 order item summary or product fields to display in the UI in each item row.

pageParam
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

Return Value

Type: `ConnectApi.OrderItemSummaryCollection`

getOrderItemSummaries(webstoreId, effectiveAccountId, orderSummaryId, orderDeliveryGroupSummaryId, fields, sortParam)

Get a sorted list of order item summaries for a delivery group summary with specific fields.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderItemSummaryCollection getOrderItemSummaries(String webstoreId, String effectiveAccountId, String orderSummaryId, String orderDeliveryGroupSummaryId, List<String> fields, ConnectApi.OrderItemSummarySort sortParam)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

orderSummaryId
Type: String

ID of the order summary.

orderDeliveryGroupSummaryId
Type: String

ID of the order delivery group summary.

fields
Type: List<String>

List of up to 15 order item summary or product fields to display in the UI in each item row.

sortParam
Type: `ConnectApi.OrderItemSummarySort`

Sort order for order item summaries. Values are:

- `IdAsc`—Sorts by ID in ascending alphanumeric order (A–Z, 0–9).
- `IdDesc`—Sorts by ID in descending alphanumeric order (Z–A, 9–0).

If `null`, `IdAsc` is the default sort order.

Return Value

Type: `ConnectApi.OrderItemSummaryCollection`

`getOrderItemSummaries(webstoreId, effectiveAccountId, orderSummaryId, orderDeliveryGroupSummaryId, fields, pageSize, sortParam)`

Get a sorted page of order item summaries for a delivery group summary with specific fields.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderItemSummaryCollection getOrderItemSummaries(String
webstoreId, String effectiveAccountId, String orderSummaryId, String
orderDeliveryGroupSummaryId, List<String> fields, Integer pageSize,
ConnectApi.OrderItemSummarySort sortParam)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

orderSummaryId
Type: String

ID of the order summary.

orderDeliveryGroupSummaryId
Type: String

ID of the order delivery group summary.

fields
Type: List<String>

List of up to 15 order item summary or product fields to display in the UI in each item row.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.OrderItemSummarySort`

Sort order for order item summaries. Values are:

- `IdAsc`—Sorts by ID in ascending alphanumeric order (A–Z, 0–9).
- `IdDesc`—Sorts by ID in descending alphanumeric order (Z–A, 9–0).

If `null`, `IdAsc` is the default sort order.

Return Value

Type: `ConnectApi.OrderItemSummaryCollection`

getOrderItemSummaries(webstoreId, effectiveAccountId, orderSummaryId, orderDeliveryGroupSummaryId, fields, pageParam, pageSize, sortParam, includeAdjustmentDetails)

Get a sorted page of order item summaries for a delivery group summary with specific fields and include adjustment details.

API Version

56.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderItemSummaryCollection getOrderItemSummaries(String
webstoreId, String effectiveAccountId, String orderSummaryId, String
orderDeliveryGroupSummaryId, List<String> fields, String pageParam, Integer pageSize,
ConnectApi.OrderItemSummarySort sortParam, Boolean includeAdjustmentDetails)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

orderSummaryId
Type: String

ID of the order summary.

orderDeliveryGroupSummaryId
Type: String

ID of the order delivery group summary.

fields
Type: List<String>

List of up to 15 order item summary or product fields to display in the UI in each item row.

pageParam
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.OrderItemSummarySort`

Sort order for order item summaries. Values are:

- `IdAsc`—Sorts by ID in ascending alphanumeric order (A–Z, 0–9).
- `IdDesc`—Sorts by ID in descending alphanumeric order (Z–A, 9–0).

If `null`, `IdAsc` is the default sort order.

includeAdjustmentDetails
Type: Boolean

Specifies whether to return adjustment details (`true`) or not (`false`).

Return Value

Type: `ConnectApi.OrderItemSummaryCollection`

getOrderItemSummaryAdjustments(webstoreId, orderSummaryId, orderItemSummaryAdjustmentCollectionInput)

Get adjustments for order items.

API Version

53.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderItemSummaryAdjustmentCollection
getOrderItemSummaryAdjustments(String webstoreId, String orderSummaryId,
ConnectApi.OrderItemSummaryAdjustmentCollectionInput
orderItemSummaryAdjustmentCollectionInput)
```

Parameters

webstoreId
Type: String

ID of the webstore.

orderSummaryId
Type: String

ID of the order summary.

orderItemSummaryAdjustmentCollectionInput
Type: ConnectApi.OrderItemSummaryAdjustmentCollectionInput

Collection of order item summaries to get adjustments for.

Return Value

Type: ConnectApi.OrderItemSummaryAdjustmentCollection

getOrderItemSummaryAdjustments(webstoreId, orderSummaryId, orderItemSummaryAdjustmentCollectionInput, effectiveAccountId)

Get adjustments for order items.

API Version

53.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderItemSummaryAdjustmentCollection
getOrderItemSummaryAdjustments(String webstoreId, String orderSummaryId,
ConnectApi.OrderItemSummaryAdjustmentCollectionInput
orderItemSummaryAdjustmentCollectionInput, String effectiveAccountId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

orderSummaryId
Type: String

ID of the order summary.

orderItemSummaryAdjustmentCollectionInput
Type: ConnectApi.OrderItemSummaryAdjustmentCollectionInput

Collection of order item summaries to get adjustments for.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

Return Value

Type: ConnectApi.OrderItemSummaryAdjustmentCollection

getOrderShipmentItems(webstoreId, shipmentId)

Get order shipment items.

API Version

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderShipmentItemCollection getOrderShipmentItems(String
webstoreId, String shipmentId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

shipmentId
Type: String

ID of the shipment.

Return Value

Type: ConnectApi.OrderShipmentItemCollection

getOrderShipmentItems(webstoreId, shipmentId, effectiveAccountId)

Get order shipment items.

API Version

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderShipmentItemCollection getOrderShipmentItems(String webstoreId, String shipmentId, String effectiveAccountId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

shipmentId
Type: String

ID of the shipment.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

Return Value

Type: ConnectApi.OrderShipmentItemCollection

getOrderShipmentItems(webstoreId, shipmentId, effectiveAccountId, fields)

Get order shipment items with specific fields.

API Version

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderShipmentItemCollection getOrderShipmentItems(String
webstoreId, String shipmentId, String effectiveAccountId, List<String> fields)
```

Parameters

webstoreId
Type: String

ID of the webstore.

shipmentId
Type: String

ID of the shipment.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

fields
Type: List<String>

List of up to 15 additional shipment items, order item summary, and product fields to display in the UI in each item row.

Return Value

Type: `ConnectApi.OrderShipmentItemCollection`

getOrderShipmentItems(webstoreId, shipmentId, effectiveAccountId, fields, pageToken, pageSize)

Get a page of order shipment items with specific fields.

API Version

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderShipmentItemCollection getOrderShipmentItems(String
webstoreId, String shipmentId, String effectiveAccountId, List<String> fields, String
pageToken, Integer pageSize)
```

Parameters

webstoreId
Type: String

ID of the webstore.

shipmentId
Type: String

ID of the shipment.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

fields
Type: List<String>

List of up to 15 additional shipment items, order item summary, and product fields to display in the UI in each item row.

pageToken
Type: String

Specifies the base64 encoded page token. Page tokens are returned as part of the response. If unspecified, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.OrderShipmentItemCollection`

getOrderShipmentItems(webstoreId, shipmentId, effectiveAccountId, fields, pageToken, pageSize, sortOrder)

Get a sorted page of order shipment items.

API Version

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderShipmentItemCollection getOrderShipmentItems(String
webstoreId, String shipmentId, String effectiveAccountId, List<String> fields, String
pageToken, Integer pageSize, ConnectApi.OrderShipmentItemSort sortOrder)
```

Parameters

webstoreId
Type: String

ID of the webstore.

shipmentId
Type: String

ID of the shipment.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

fields
Type: List<String>

List of up to 15 additional shipment items, order item summary, and product fields to display in the UI in each item row.

pageToken
Type: String

Specifies the base64 encoded page token. Page tokens are returned as part of the response. If unspecified, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortOrder
Type: `ConnectApi.OrderShipmentItemSort`

Sort order for order shipment items. Values are:

- `IdAsc`—Sorts by ID in ascending alphanumeric order (A–Z, 0–9).
- `IdDesc`—Sorts by ID in descending alphanumeric order (Z–A, 9–0).

If unspecified, defaults to `IdAsc`.

Return Value

Type: `ConnectApi.OrderShipmentItemCollection`

getOrderShipments(webstoreId, orderSummaryId)

Get order shipments.

API Version

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderShipmentCollection getOrderShipments(String webstoreId,
String orderSummaryId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

orderSummaryId
Type: String

ID of the order summary.

Return Value

Type: ConnectApi.OrderShipmentCollection

getOrderShipments(webstoreId, orderSummaryId, effectiveAccountId)

Get order shipments.

API Version

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderShipmentCollection getOrderShipments(String webstoreId,
String orderSummaryId, String effectiveAccountId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

orderSummaryId
Type: String

ID of the order summary.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

Return Value

Type: ConnectApi.OrderShipmentCollection

getOrderShipments(webstoreId, orderSummaryId, effectiveAccountId, fields)

Get order shipments with specific fields.

API Version

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderShipmentCollection getOrderShipments(String webstoreId,
String orderSummaryId, String effectiveAccountId, List<String> fields)
```

Parameters

webstoreId
Type: String

ID of the webstore.

orderSummaryId
Type: String

ID of the order summary.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

fields
Type: List<String>

List of up to 15 additional shipment and order delivery method fields to display in the UI in each item row.

Return Value

Type: `ConnectApi.OrderShipmentCollection`

`getOrderShipments(webstoreId, orderSummaryId, effectiveAccountId, fields, pageSize, pageToken)`

Get a page of order shipments with specific fields.

API Version

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderShipmentCollection getOrderShipments(String webstoreId,
String orderSummaryId, String effectiveAccountId, List<String> fields, Integer pageSize,
String pageToken)
```

Parameters

webstoreId
Type: String

ID of the webstore.

orderSummaryId
Type: String

ID of the order summary.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

fields
Type: List<String>

List of up to 15 additional shipment and order delivery method fields to display in the UI in each item row.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

pageToken
Type: String

Specifies the base64 encoded page token. Page tokens are returned as part of the response. If unspecified, the first page is returned.

Return Value

Type: `ConnectApi.OrderShipmentCollection`

`getOrderShipments(webstoreId, orderSummaryId, effectiveAccountId, fields, pageSize, pageToken, sortOrder)`

Get a sorted page of order shipments with specific fields.

API Version

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderShipmentCollection getOrderShipments(String webstoreId,
String orderSummaryId, String effectiveAccountId, List<String> fields, Integer pageSize,
String pageToken, ConnectApi.OrderShipmentSort sortOrder)
```

Parameters

webstoreId
Type: String

ID of the webstore.

orderSummaryId
Type: String

ID of the order summary.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

fields
Type: List<String>

List of up to 15 additional shipment and order delivery method fields to display in the UI in each item row.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

pageToken
Type: String

Specifies the base64 encoded page token. Page tokens are returned as part of the response. If unspecified, the first page is returned.

sortOrder
Type: `ConnectApi.OrderShipmentSort`

Sort order for order shipments. Values are:

- `ExpectedDeliveryDateAsc`—Sorts by the oldest expected delivery date.
- `ExpectedDeliveryDateDesc`—Sorts by the most recent expected delivery date.
- `ShipmentNumberAsc`—Sorts by shipment number in ascending order (0–9).
- `ShipmentNumberDesc`—Sorts by shipment number in descending order (9–0).

If unspecified, defaults to `ShipmentNumberAsc`.

If you're sorting by expected delivery date, make sure the expected delivery date is populated on your shipment records. A `null` value isn't supported and results in an error.

Return Value

Type: `ConnectApi.OrderShipmentCollection`

`getOrderSummaries(webstoreId)`

Get order summaries.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderSummaryCollectionRepresentation getOrderSummaries(String webstoreId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

Return Value

Type: ConnectApi.OrderSummaryCollectionRepresentation

getOrderSummaries(webstoreId, effectiveAccountId)

Get order summaries.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderSummaryCollectionRepresentation getOrderSummaries(String webstoreId, String effectiveAccountId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

Return Value

Type: ConnectApi.OrderSummaryCollectionRepresentation

getOrderSummaries(webstoreId, effectiveAccountId, fields)

Get order summaries with specific fields.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderSummaryCollectionRepresentation getOrderSummaries(String webstoreId, String effectiveAccountId, List<String> fields)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

fields
Type: List<String>

List of up to 35 additional order summary fields to display in the UI in each item row.

These order summary fields are returned regardless of fields specified.

- `createdDate`
- `orderSummaryId`
- `orderNumber`
- `orderedDate`
- `ownerId`
- `status`
- `totalAmount`

Return Value

Type: ConnectApi.OrderSummaryCollectionRepresentation

`getOrderSummaries(webstoreId, effectiveAccountId, fields, pageSize, pageToken)`

Get a page of order summaries with specific fields.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderSummaryCollectionRepresentation getOrderSummaries(String webstoreId, String effectiveAccountId, List<String> fields, Integer pageSize, String pageToken)
```

Parameters

`webstoreId`
Type: String

ID of the webstore.

`effectiveAccountId`
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

`fields`
Type: List<String>

List of up to 35 additional order summary fields to display in the UI in each item row.

These order summary fields are returned regardless of fields specified.

- `createdDate`
- `orderSummaryId`
- `orderNumber`
- `orderedDate`
- `ownerId`
- `status`
- `totalAmount`

`pageSize`
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

`pageToken`
Type: String

Specifies the base64 encoded page token. Page tokens are returned as part of the response. If unspecified, the first page is returned.

Return Value

Type: ConnectApi.OrderSummaryCollectionRepresentation

getOrderSummaries(webstoreId, effectiveAccountId, fields, pageSize, pageToken, sortOrder)

Get a sorted page of order summaries with specific fields.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderSummaryCollectionRepresentation getOrderSummaries(String webstoreId, String effectiveAccountId, List<String> fields, Integer pageSize, String pageToken, ConnectApi.OrderSummarySortOrder sortOrder)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

fields
Type: List<String>

List of up to 35 additional order summary fields to display in the UI in each item row.

These order summary fields are returned regardless of fields specified.

- `createdDate`
- `orderSummaryId`
- `orderNumber`
- `orderedDate`
- `ownerId`
- `status`
- `totalAmount`

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

pageToken
Type: String

Specifies the base64 encoded page token. Page tokens are returned as part of the response. If unspecified, the first page is returned.

sortOrder
Type: `ConnectApi.OrderSummarySortOrder`

Sort order for order summaries. Values are:

- `CreatedDateAsc`—Sorts by the oldest created date.
- `CreatedDateDesc`—Sorts by the most recent created date.
- `OrderedDateAsc`—Sorts by the oldest ordered date.
- `OrderedDateDesc`—Sorts by the most recent ordered date.

If unspecified, defaults to `OrderedDateDesc`.

If you're sorting by ordered date, make sure the ordered date is populated on your order summary records. A `null` value isn't supported and results in an error.

Return Value

Type: `ConnectApi.OrderSummaryCollectionRepresentation`

`getOrderSummaries(webstoreId, effectiveAccountId, fields, pageSize, pageToken, sortOrder, earliestDate, latestDate)`

Get a sorted page of order summaries with specific fields within a specific date range.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderSummaryCollectionRepresentation getOrderSummaries(String webstoreId, String effectiveAccountId, List<String> fields, Integer pageSize, String pageToken, ConnectApi.OrderSummarySortOrder sortOrder, String earliestDate, String latestDate)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

fields
Type: List<String>

List of up to 35 additional order summary fields to display in the UI in each item row.

These order summary fields are returned regardless of fields specified.

- `createdDate`
- `orderSummaryId`
- `orderNumber`
- `orderedDate`
- `ownerId`
- `status`
- `totalAmount`

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

pageToken
Type: String

Specifies the base64 encoded page token. Page tokens are returned as part of the response. If unspecified, the first page is returned.

sortOrder
Type: `ConnectApi.OrderSummarySortOrder`

Sort order for order summaries. Values are:

- `CreatedDateAsc`—Sorts by the oldest created date.
- `CreatedDateDesc`—Sorts by the most recent created date.
- `OrderedDateAsc`—Sorts by the oldest ordered date.
- `OrderedDateDesc`—Sorts by the most recent ordered date.

If unspecified, defaults to `OrderedDateDesc`.

If you're sorting by ordered date, make sure the ordered date is populated on your order summary records. A `null` value isn't supported and results in an error.

earliestDate
Type: String

Oldest created or ordered date, depending on the `sortOrder` value, for order summaries to return. Results include any orders on and after this date..

latestDate
Type: String

Most recent created or ordered date, depending on the `sortOrder` value, for order summaries to return. Results include any orders before this date.

Return Value

Type: `ConnectApi.OrderSummaryCollectionRepresentation`

`getOrderSummaries(webstoreId, effectiveAccountId, fields, pageSize, pageToken, sortOrder, earliestDate, latestDate, ownerScoped)`

Get a sorted page of order summaries with specific fields within a specific date range and scoped to orders owned by the context user.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderSummaryCollectionRepresentation getOrderSummaries(String webstoreId, String effectiveAccountId, List<String> fields, Integer pageSize, String pageToken, ConnectApi.OrderSummarySortOrder sortOrder, String earliestDate, String latestDate, Boolean ownerScoped)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

fields
Type: List<String>

List of up to 35 additional order summary fields to display in the UI in each item row.

These order summary fields are returned regardless of fields specified.

- `createdDate`
- `orderSummaryId`
- `orderNumber`
- `orderedDate`
- `ownerId`
- `status`
- `totalAmount`

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

pageToken
Type: String

Specifies the base64 encoded page token. Page tokens are returned as part of the response. If unspecified, the first page is returned.

sortOrder
Type: ConnectApi.OrderSummarySortOrder

Sort order for order summaries. Values are:

- `CreatedDateAsc`—Sorts by the oldest created date.
- `CreatedDateDesc`—Sorts by the most recent created date.
- `OrderedDateAsc`—Sorts by the oldest ordered date.
- `OrderedDateDesc`—Sorts by the most recent ordered date.

If unspecified, defaults to `OrderedDateDesc`.

If you're sorting by ordered date, make sure the ordered date is populated on your order summary records. A `null` value isn't supported and results in an error.

earliestDate
Type: String

Oldest created or ordered date, depending on the `sortOrder` value, for order summaries to return. Results include any orders on and after this date..

latestDate
Type: String

Most recent created or ordered date, depending on the `sortOrder` value, for order summaries to return. Results include any orders before this date.

ownerScoped
Type: Boolean

Specifies whether the results are scoped to orders owned by the context user (`true`) or to orders owned by and shared with the context user (`false`). If unspecified, defaults to `true`.

Return Value

Type: ConnectApi.OrderSummaryCollectionRepresentation

getOrderSummaries(webstoreId, effectiveAccountId, fields, pageSize, pageToken, sortOrder, earliestDate, latestDate, ownerScoped, includeAdjustmentDetails)

Get a sorted page of order summaries with specific fields within a specific date range and scoped to orders owned by the context user.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderSummaryCollectionRepresentation getOrderSummaries(String webstoreId, String effectiveAccountId, List<String> fields, Integer pageSize, String
```

```
pageToken, ConnectApi.OrderSummarySortOrder sortOrder, String earliestDate, String
latestDate, Boolean ownerScoped, Boolean includeAdjustmentDetails)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

fields
Type: List<String>

List of up to 35 additional order summary fields to display in the UI in each item row.

These order summary fields are returned regardless of fields specified.

- `createdDate`
- `orderSummaryId`
- `orderNumber`
- `orderedDate`
- `ownerId`
- `status`
- `totalAmount`

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

pageToken
Type: String

Specifies the base64 encoded page token. Page tokens are returned as part of the response. If unspecified, the first page is returned.

sortOrder
Type: `ConnectApi.OrderSummarySortOrder`

Sort order for order summaries. Values are:

- `CreatedDateAsc`—Sorts by the oldest created date.
- `CreatedDateDesc`—Sorts by the most recent created date.
- `OrderedDateAsc`—Sorts by the oldest ordered date.
- `OrderedDateDesc`—Sorts by the most recent ordered date.

If unspecified, defaults to `OrderedDateDesc`.

If you're sorting by ordered date, make sure the ordered date is populated on your order summary records. A `null` value isn't supported and results in an error.

earliestDate
Type: String

Oldest created or ordered date, depending on the `sortOrder` value, for order summaries to return. Results include any orders on and after this date..

latestDate
Type: String

Most recent created or ordered date, depending on the `sortOrder` value, for order summaries to return. Results include any orders before this date.

ownerScoped
Type: Boolean

Specifies whether the results are scoped to orders owned by the context user (`true`) or to orders owned by and shared with the context user (`false`). If unspecified, defaults to `true`.

includeAdjustmentDetails
Type: Boolean

Specifies whether to fetch price adjustment details based on their type (`true`). If unspecified, defaults to `false`.

Return Value

Type: `ConnectApi.OrderSummaryCollectionRepresentation`

getOrderSummary(webstoreId, orderSummaryId, effectiveAccountId)

Get an order summary.

API Version

55.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderSummaryRepresentation getOrderSummary(String webstoreId,
String orderSummaryId, String effectiveAccountId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

orderSummaryId
Type: String

ID of the order summary.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

Return Value

Type: ConnectApi.OrderSummaryRepresentation

getOrderSummary(webstoreId, orderSummaryId, effectiveAccountId, fields)

Get an order summary with fields.

API Version

55.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderSummaryRepresentation getOrderSummary(String webstoreId,
String orderSummaryId, String effectiveAccountId, List<String> fields)
```

Parameters

webstoreId
Type: String

ID of the webstore.

orderSummaryId
Type: String

ID of the order summary.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

fields
Type: List<String>

List of up to 35 additional order summary fields to display in the UI in each item row.

These order summary fields are returned regardless of fields specified.

- `createdDate`
- `orderSummaryId`
- `orderNumber`
- `orderedDate`
- `ownerId`
- `status`
- `totalAmount`

Return Value

Type: ConnectApi.OrderSummaryRepresentation

getOrderSummary(webstoreId, orderSummaryId, effectiveAccountId, fields, includeAdjustmentDetails)

Get an order summary with fields and include adjustment details.

API Version

55.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderSummaryRepresentation getOrderSummary(String webstoreId, String orderSummaryId, String effectiveAccountId, List<String> fields, Boolean includeAdjustmentDetails)
```

Parameters

webstoreId
Type: String

ID of the webstore.

orderSummaryId
Type: String

ID of the order summary.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

fields
Type: List<String>

List of up to 35 additional order summary fields to display in the UI in each item row.

These order summary fields are returned regardless of fields specified.

- `createdDate`
- `orderSummaryId`
- `orderNumber`
- `orderedDate`
- `ownerId`
- `status`
- `totalAmount`

includeAdjustmentDetails
Type: Boolean

Specifies whether to return adjustment details (`true`) or not (`false`). If unspecified, the default value is `false`.

Return Value

Type: `ConnectApi.OrderSummaryRepresentation`

getOrderSummaryAdjustments(webstoreId, orderSummaryId)

Get adjustments for an order summary.

API Version

53.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderSummaryAdjustmentCollection
getOrderSummaryAdjustments(String webstoreId, String orderSummaryId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

orderSummaryId
Type: String

ID of the order summary.

Return Value

Type: `ConnectApi.OrderSummaryAdjustmentCollection`

getOrderSummaryAdjustments(webstoreId, orderSummaryId, effectiveAccountId)

Get adjustments for an order summary.

API Version

53.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderSummaryAdjustmentCollection
getOrderSummaryAdjustments(String webstoreId, String orderSummaryId, String
effectiveAccountId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

orderSummaryId
Type: String

ID of the order summary.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

Return Value

Type: `ConnectApi.OrderSummaryAdjustmentCollection`

`lookupOrderSummary(webstoreId, effectiveAccountId, fields, excludeLineItems, excludeDeliveryGroups, excludeAdjustmentAggregates, excludeAdjustments, deliveryGroupId, orderSummaryLookupInput)` (Developer Preview)

Look up details about an order summary for a guest shopper or a registered buyer using the effective account ID, requested fields, line items, delivery groups, adjustments aggregates, and adjustments.

Note: This API is available as a developer preview. It isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools.

API Version

58.0

Available to Guest Users

58.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderSummaryLookupOutput lookupOrderSummary(String webstoreId,
String effectiveAccountId, List<String> fields, Boolean excludeLineItems, Boolean
excludeDeliveryGroups, Boolean excludeAdjustmentAggregates, Boolean excludeAdjustments,
String deliveryGroupId, ConnectApi.OrderSummaryLookupInput orderSummaryLookupInput)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If unspecified, defaults to the account ID for the context user or, for guest users, the guest buyer profile ID of the current store.

fields
Type: List<String>

List of specific fields, including custom fields, to return in the response along with default fields. For example, `OrderSummary.TotalAmount`, `OrderItemSummary.Quantity`, `Product2.Description`, `OrderDeliveryGroupSummary.GrandTotalAmount`, `OrderDeliveryMethod.Carrier`.

excludeLineItems
Type: Boolean

Specifies whether to exclude line items from the response. If unspecified, the default value is `false`.

excludeDeliveryGroups
Type: Boolean

Specifies whether to exclude delivery groups from the response. If unspecified, the default value is `false`.

excludeAdjustmentAggregates
Type: Boolean

Specifies whether to exclude adjustment aggregates associated with an order summary. Adjustment aggregates include fields detailing promotional amounts by price, tax, and total. Aggregates are calculated asynchronously and results returned to the order summary. If unspecified, the default value is `false`.

excludeAdjustments
Type: Boolean

Specifies whether to exclude adjustments associated with an order summary. Adjustments include promotional discounts. If unspecified, the default value is `false`.

deliveryGroupId
Type: String

ID of the delivery group associated with the order summary.

orderSummaryLookupInput
Type: `ConnectApi.OrderSummaryLookupInput`

Order summary lookup input representation.

Return Value

Type: ConnectApi.OrderSummaryLookupOutput

lookupOrderSummary(webstoreId, effectiveAccountId, fields, orderSummaryLookupInput) (Developer Preview)

Look up details about an order summary for a guest shopper or a registered buyer using the effective account ID and requested fields.

Note: This API is available as a developer preview. It isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools.

API Version

58.0

Available to Guest Users

58.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderSummaryLookupOutput lookupOrderSummary(String webstoreId,
String effectiveAccountId, List<String> fields, ConnectApi.OrderSummaryLookupInput
orderSummaryLookupInput)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If unspecified, defaults to the account ID for the context user or, for guest users, the guest buyer profile ID of the current store.

fields
Type: List<String>

List of specific fields, including custom fields, to return in the response along with default fields. For example, `OrderSummary.TotalAmount`, `OrderItemSummary.Quantity`, `Product2.Description`, `OrderDeliveryGroupSummary.GrandTotalAmount`, `OrderDeliveryMethod.Carrier`.

orderSummaryLookupInput
Type: ConnectApi.OrderSummaryLookupInput

Order summary lookup input representation.

Return Value

Type: `ConnectApi.OrderSummaryLookupOutput`

`lookupOrderSummary(webstoreId, effectiveAccountId, orderSummaryLookupInput)` (Developer Preview)

Look up details about an order summary for a guest shopper or a registered buyer using the effective account ID.

Note: This API is available as a developer preview. It isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools.

API Version

58.0

Available to Guest Users

58.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderSummaryLookupOutput lookupOrderSummary(String webstoreId,
String effectiveAccountId, ConnectApi.OrderSummaryLookupInput orderSummaryLookupInput)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If unspecified, defaults to the account ID for the context user or, for guest users, the guest buyer profile ID of the current store.

orderSummaryLookupInput
Type: `ConnectApi.OrderSummaryLookupInput`

Order summary lookup input representation.

Return Value

Type: `ConnectApi.OrderSummaryLookupOutput`

`updateCommerceAccountAddress(webstoreId, accountId, addressId, addressInput)`

Update a Commerce account address for a webstore.

API Version

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.CommerceAddressOutput updateCommerceAccountAddress(String webstoreId, String accountId, String addressId, ConnectApi.commerceAddressInput addressInput)
```

Parameters

webstoreId
Type: String

ID of the webstore.

accountId
Type: String

ID of the account.

addressId
Type: String

ID of the address.

addressInput
Type: `ConnectApi.commerceAddressInput`

Information about the address fields you want to update.

Return Value

Type: `ConnectApi.CommerceAddressOutput`

CommerceCart Class

Get, create, update, calculate, and delete carts. Get cart items, add items to carts, update and delete cart items.

Namespace

ConnectApi

CommerceCart Methods

These methods are for `CommerceCart`. All methods are static.

IN THIS SECTION:

addItemToCart(webstoreId, effectiveAccountId, activeCartOrId, cartItemInput, currencyIsoCode)
Add an item to a cart of a specific currency.

addItemsToCart(webstoreId, effectiveAccountId, activeCartOrId, cartItems)
Add a batch of up to 100 items to a cart.

addItemsToCart(webstoreId, effectiveAccountId, activeCartOrId, cartItems, currencyIsoCode)
Add a batch of up to 100 items to a cart of a specific currency.

addItemToCart(webstoreId, effectiveAccountId, activeCartOrId, cartItemInput)
Add an item to a cart.

applyCartCoupon(webstoreId, effectiveAccountId, activeCartOrId, cartCouponInput)
Apply a coupon to a cart.

applyCartCoupon(webstoreId, effectiveAccountId, activeCartOrId, cartCouponInput, currencyIsoCode)
Apply a coupon to a cart.

copyCartToWishlist(webstoreId, effectiveAccountId, activeCartOrId, cartToWishlistInput)
Copy the products from a cart to a wishlist.

calculateCart(webstoreId, activeCartOrId, effectiveAccountId)
Calculate a cart.

calculateCart(webstoreId, activeCartOrId, effectiveAccountId, calculateCartInput)
Calculate a cart.

createCart(webstoreId, cart)
Create a cart.

cloneCart(webstoreId, activeCartOrId, targetEffectiveAccountId, targetType)
Clones an existing cart to create a secondary, read-only cart to support Pay Now functionality. Sets the guest cart status to `PendingDelete` in a B2B store or `Closed` in a D2C store.

deleteCart(webstoreId, effectiveAccountId, activeCartOrId)
Delete a cart. Sets the guest cart status to `PendingDelete` in a B2B store or `Closed` in a D2C store.

deleteCartCoupon(webstoreId, effectiveAccountId, activeCartOrId, cartCouponId)
Delete a coupon from a cart.

deleteCartCoupon(webstoreId, effectiveAccountId, activeCartOrId, cartCouponId, currencyIsoCode)
Delete a coupon from a cart.

deleteCartItem(webstoreId, effectiveAccountId, activeCartOrId, cartItemId)
Delete an item from a cart.

deleteInventoryReservation(webstoreId, activeCartOrId, effectiveAccountId) (Pilot)
Delete an inventory reservation.

evaluateShipping(webstoreId, activeCartOrId, effectiveAccountId, cartEvaluateShippingInput)
Evaluate shipping costs for a cart.

evaluateTaxes(webstoreId, activeCartOrId, effectiveAccountId, cartEvaluateTaxInput)

Evaluate taxes for a cart.

getCartCoupons(webstoreId, effectiveAccountId, activeCartOrId)

Get coupons for a cart.

getCartCoupons(webstoreId, effectiveAccountId, activeCartOrId, currencyIsoCode)

Get coupons for a cart.

getCartItemPromotion(webstoreId, effectiveAccountId, activeCartOrId, cartItemPromotionCollectionInput)

Get promotions for a cart item.

getCartItemPromotion(webstoreId, effectiveAccountId, activeCartOrId, cartItemPromotionCollectionInput, currencyIsoCode)

Get a promotion for a cart item.

getCartItems(webstoreId, effectiveAccountId, activeCartOrId)

Get items in a cart.

getCartItems(webstoreId, effectiveAccountId, activeCartOrId, pageParam)

Get a page of items in a cart.

getCartItems(webstoreId, effectiveAccountId, activeCartOrId, pageParam, sortParam)

Get a sorted page of items in a cart.

getCartItems(webstoreId, effectiveAccountId, activeCartOrId, pageParam, pageSize)

Get a specified size page of items in a cart.

getCartItems(webstoreId, effectiveAccountId, activeCartOrId, pageParam, pageSize, sortParam)

Get a specified size, sorted page of items in a cart.

getCartItems(webstoreId, effectiveAccountId, activeCartOrId, productFields, pageParam, pageSize, sortParam)

Get a specified size, sorted page of items filtered by product fields in a cart.

getCartItems(webstoreId, effectiveAccountId, activeCartOrId, productFields, pageParam, pageSize, sortParam, currencyIsoCode)

Get a specified size, sorted page of items filtered by product fields in a cart.

getCartItems(webstoreId, effectiveAccountId, activeCartOrId, productFields, pageParam, pageSize, sortParam, currencyIsoCode, includePromotions, includeCoupons)

Get a sorted page of items in a cart, including coupons and promotions.

getCartItems(webstoreId, effectiveAccountId, activeCartOrId, productFields, pageParam, pageSize, sortParam, currencyIsoCode, includePromotions, includeCoupons, pageNumber)

Get a specific, sorted page of items in a cart, including coupons and promotions.

getCartPromotions(webstoreId, effectiveAccountId, activeCartOrId)

Get promotions for a cart.

getCartPromotions(webstoreId, effectiveAccountId, activeCartOrId, currencyIsoCode)

Get promotions for a cart in a specific currency.

getCartSummary(webstoreId, effectiveAccountId, activeCartOrId)

Get a cart.

getCartSummary(webstoreId, effectiveAccountId, activeCartOrId, currencyIsoCode)

Get a cart in a specific currency.

getOrCreateActiveCartSummary(webstoreId, effectiveAccountId, activeCartOrId)

Get a cart or create an active cart if one doesn't exist.

getOrCreateActiveCartSummary(webstoreId, effectiveAccountId, activeCartOrId, currencyIsoCode)
Get a cart in a specific currency, or create an active cart if one doesn't exist.

getProductCartItem(webstoreId, effectiveAccountId, activeCartOrId, productId, currencyIsoCode)
Get cart items of a specific product.

getProductCartItems(webstoreId, effectiveAccountId, activeCartOrId, pageSize, pageNumber, currencyIsoCode)
Get the items in a cart, sorted by product ID.

makeCartPrimary(webstoreId, activeCartOrId, effectiveAccountId)
Make a secondary cart a primary cart.

preserveGuestCart(webstoreId, effectiveAccountId, activeCartOrId, currencyIsoCode)
Preserve cart contents when a guest logs in as an authenticated customer. Sets the guest cart status to `PendingDelete` in a B2B store or `Closed` in a D2C store.

setCartMessagesVisibility(webstoreId, activeCartOrId, effectiveAccountId, messageVisibility)
Set the visibility for cart messages.

updateCartItem(webstoreId, effectiveAccountId, activeCartOrId, cartItemId, cartItem)
Update an item in a cart.

updateCartItem(webstoreId, effectiveAccountId, activeCartOrId, cartItemId, cartItem, currencyIsoCode)
Update an item in a cart of a specific currency.

upsertInventoryReservation(webstoreId, activeCartOrId, effectiveAccountId, cartInventoryReservationInput) (Pilot)
Create or update an inventory reservation.

`addItemToCart(webstoreId, effectiveAccountId, activeCartOrId, cartItemInput, currencyIsoCode)`

Add an item to a cart of a specific currency.

API Version

57.0

Available to Guest Users

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartItem addItemToCart(String webstoreId, String
effectiveAccountId, String activeCartOrId, ConnectApi.CartItemInput cartItemInput,
String currencyIsoCode)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value indicates a cart with a status that isn't `Closed` or `PendingDelete`.

cartItemInput
Type: `ConnectApi.CartItemInput`

A `ConnectApi.CartItemInput` object representing an item to add to the cart.

currencyIsoCode
Type: String

The currency ISO code of the cart.

Return Value

Type: `ConnectApi.CartItem`

Usage

Buyers with read access to carts can add, update, and delete items in carts.

This method respects buyer View Product entitlements and only users entitled to view product data can access it.

`addItemsToCart(webstoreId, effectiveAccountId, activeCartOrId, cartItems)`

Add a batch of up to 100 items to a cart.

API Version

49.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.BatchResult[] addItemsToCart(String webstoreId, String
effectiveAccountId, String activeCartOrId, List<ConnectApi.BatchInput> cartItems)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`.

cartItems
Type: List<`ConnectApi.BatchInput`>

The list can contain up to 100 `ConnectApi.BatchInput` objects. In the `ConnectApi.BatchInput` constructor, the input object must be `ConnectApi.CartItemInput`.

Return Value

Type: `ConnectApi.BatchResult`[]

The `ConnectApi.BatchResult.getResult()` method returns a `ConnectApi.CartItem` object.

The returned objects correspond to each of the input objects and are returned in the same order as the input objects.

The method call fails only if an error occurs that affects the entire operation (such as a parsing failure). If an individual object causes an error, the error is embedded within the `ConnectApi.BatchResult` list.

Usage

Buyers with read access to carts can add, update, and delete items in carts.

This method respects buyer View Product entitlements and only users entitled to view product data can access it.

`addItemsToCart(webstoreId, effectiveAccountId, activeCartOrId, cartItems, currencyIsoCode)`

Add a batch of up to 100 items to a cart of a specific currency.

API Version

57.0

Available to Guest Users

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.BatchResult[] addItemsToCart(String webstoreId, String effectiveAccountId, String activeCartOrId, List<ConnectApi.BatchInput> cartItems, String currencyIsoCode)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`.

cartItems
Type: List<`ConnectApi.BatchInput`>

The list can contain up to 100 `ConnectApi.BatchInput` objects. In the `ConnectApi.BatchInput` constructor, the input object must be `ConnectApi.CartItemInput`.

currencyIsoCode
Type: String

The currency ISO code of the cart.

Return Value

Type: `ConnectApi.BatchResult`[]

The `ConnectApi.BatchResult.getResult()` method returns a `ConnectApi.CartItem` object.

The returned objects correspond to each of the input objects and are returned in the same order as the input objects.

The method call fails only if an error occurs that affects the entire operation (such as a parsing failure). If an individual object causes an error, the error is embedded within the `ConnectApi.BatchResult` list.

Usage

Buyers with read access to carts can add, update, and delete items in carts.

This method respects buyer View Product entitlements and only users entitled to view product data can access it.

addItemToCart(webstoreId, effectiveAccountId, activeCartOrId, cartItemInput)

Add an item to a cart.

API Version

49.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartItem addItemToCart(String webstoreId, String
effectiveAccountId, String activeCartOrId, ConnectApi.CartItemInput cartItemInput)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`.

cartItemInput
Type: `ConnectApi.CartItemInput`

A `ConnectApi.CartItemInput` object representing an item to add to the cart.

Return Value

Type: `ConnectApi.CartItem`

Usage

Buyers with read access to carts can add, update, and delete items in carts.

This method respects buyer View Product entitlements and only users entitled to view product data can access it.

applyCartCoupon(webstoreId, effectiveAccountId, activeCartOrId, cartCouponInput)

Apply a coupon to a cart.

API Version

54.0

Available to Guest Users

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartCouponCollection applyCartCoupon(String webstoreId, String effectiveAccountId, String activeCartOrId, ConnectApi.cartCouponInput cartCouponInput)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`.

cartCouponInput
Type: `ConnectApi.cartCouponInput`

Coupon code for the coupon.

Return Value

Type: `ConnectApi.CartCouponCollection`

applyCartCoupon(webstoreId, effectiveAccountId, activeCartOrId, cartCouponInput, currencyIsoCode)

Apply a coupon to a cart.

API Version

57.0

Available to Guest Users

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartCouponCollection applyCartCoupon(String webstoreId, String effectiveAccountId, String activeCartOrId, ConnectApi.cartCouponInput cartCouponInput, String currencyIsoCode)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value indicates a cart with a status that isn't `Closed` or `PendingDelete`.

cartCouponInput
Type: `ConnectApi.cartCouponInput`

Coupon code for the coupon.

currencyIsoCode
Type: String

Currency ISO code of the cart.

Return Value

Type: `ConnectApi.CartCouponCollection`

copyCartToWishlist(webstoreId, effectiveAccountId, activeCartOrId, cartToWishlistInput)

Copy the products from a cart to a wishlist.

API Version

50.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartToWishlistResult copyCartToWishlist(String webstoreId,
String effectiveAccountId, String activeCartOrId, ConnectApi.CartToWishlistInput
cartToWishlistInput)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`.

cartToWishlistInput
Type: `ConnectApi.CartToWishlistInput`

A `ConnectApi.CartToWishlistInput` object indicating the wishlist to copy products to.

Return Value

Type: `ConnectApi.CartToWishlistResult`

calculateCart(webstoreId, activeCartOrId, effectiveAccountId)

Calculate a cart.

API Version

62.0

Available to Guest Users

62.0

Requires Chatter

No

Signature

```
public static ConnectApi.CalculateCartResult calculateCart(String webstoreId, String
activeCartOrId, String effectiveAccountId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value indicates a cart with a status that isn't `Closed` or `PendingDelete`.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

Return Value

Type: `ConnectApi.CalculateCartResult`

Usage

Buyers with read access to carts can add, update, and delete items in carts.

This method respects buyer View Product entitlements and only users entitled to view product data can access it.

calculateCart(webstoreId, activeCartOrId, effectiveAccountId, calculateCartInput)

Calculate a cart.

API Version

63.0

Available to Guest Users

62.0

Requires Chatter

No

Signature

```
public static ConnectApi.CalculateCartResult calculateCart(String webstoreId, String
activeCartOrId, String effectiveAccountId, ConnectApi.CalculateCartInput
calculateCartInput)
```

Parameters

webstoreId
Type: String

ID of the webstore.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value indicates a cart with a status that isn't `Closed` or `PendingDelete`.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

calculateCartInput
Type: `ConnectApi.CalculateCartInput`

A `ConnectApi.CalculateCartInput` object representing any custom fields for the cart.

Return Value

Type: `ConnectApi.CalculateCartResult`

Usage

Buyers with read access to carts can add, update, and delete items in carts.

This method respects buyer View Product entitlements and only users entitled to view product data can access it.

`createCart(webstoreId, cart)`

Create a cart.

API Version

49.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartSummary createCart(String webstoreId, ConnectApi.CartInput
cart)
```

Parameters

webstoreId
Type: String

ID of the webstore.

cart
Type: `ConnectApi.CartInput`

A `ConnectApi.CartInput` object representing a cart.

Return Value

Type: `ConnectApi.CartSummary`

Usage

Buyers with read access to carts can create and delete carts.

`cloneCart(webstoreId, activeCartOrId, targetEffectiveAccountId, targetType)`

Clones an existing cart to create a secondary, read-only cart to support Pay Now functionality. Sets the guest cart status to `PendingDelete` in a B2B store or `Closed` in a D2C store.

API Version

60.0

Available to Guest Users

60.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartSummary cloneCart(String webstoreId, String activeCartOrId,
String targetEffectiveAccountId, String targetType)
```

Parameters

webstoreId
Type: String

ID of the webstore.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`.

targetEffectiveAccountId
Type: String

Effective Account ID associated with the cloned cart.

targetType
Type: String

Type of the cloned cart. Value is `PayNowReadOnly`.

Return Value

Type: `ConnectApi.CartSummary` on page 2009

Usage

The cloneCart method is valid only for the Pay Now feature. See Salesforce Pay Now for Embedded Payment Solutions.

`deleteCart(webstoreId, effectiveAccountId, activeCartOrId)`

Delete a cart. Sets the guest cart status to `PendingDelete` in a B2B store or `Closed` in a D2C store.

API Version

49.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static Void deleteCart(String webstoreId, String effectiveAccountId, String activeCartOrId)
```

Parameters

`webstoreId`
Type: String

ID of the webstore.

`effectiveAccountId`
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

`activeCartOrId`
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`.

Return Value

Type: Void

Usage

Buyers with read access to carts can create and delete carts.

deleteCartCoupon(webstoreId, effectiveAccountId, activeCartOrId, cartCouponId)

Delete a coupon from a cart.

API Version

54.0

Available to Guest Users

57.0

Requires Chatter

No

Signature

```
public static Void deleteCartCoupon(String webstoreId, String effectiveAccountId, String activeCartOrId, String cartCouponId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`.

cartCouponId
Type: String

ID of the cart coupon.

Return Value

Type: Void

deleteCartCoupon(webstoreId, effectiveAccountId, activeCartOrId, cartCouponId, currencyIsoCode)

Delete a coupon from a cart.

API Version

57.0

Available to Guest Users

57.0

Requires Chatter

No

Signature

```
public static Void deleteCartCoupon(String webstoreId, String effectiveAccountId, String activeCartOrId, String cartCouponId, String currencyIsoCode)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value indicates a cart with a status that isn't `Closed` or `PendingDelete`.

cartCouponId
Type: String

ID of the cart coupon.

currencyIsoCode
Type: String

Currency ISO code of the cart.

Return Value

Type: Void

deleteCartItem(webstoreId, effectiveAccountId, activeCartOrId, cartItemId)

Delete an item from a cart.

API Version

49.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static Void deleteCartItem(String webstoreId, String effectiveAccountId, String activeCartOrId, String cartItemId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`.

cartItemId
Type: String

ID of the cart item.

Return Value

Type: Void

Usage

Buyers with read access to carts can add, update, and delete items in carts.

`deleteInventoryReservation(webstoreId, activeCartOrId, effectiveAccountId)` (Pilot)

Delete an inventory reservation.

Note: This feature is not generally available and is being piloted with certain Customers subject to additional terms and conditions. It is not part of your purchased Services. This feature is subject to change, may be discontinued with no notice at any time in Salesforce's sole discretion, and Salesforce may never make this feature generally available. Make your purchase decisions only on

the basis of generally available products and features. This feature is made available on an AS IS basis and use of this feature is at your sole risk.

API Version

58.0

Available to Guest Users

58.0

Requires Chatter

No

Signature

```
public static Void deleteInventoryReservation(String webstoreId, String activeCartOrId,
String effectiveAccountId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value indicates a cart with a status that isn't `Closed` or `PendingDelete`.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

Return Value

Type: Void

evaluateShipping(webstoreId, activeCartOrId, effectiveAccountId, cartEvaluateShippingInput)

Evaluate shipping costs for a cart.

API Version

63.0

Available to Guest Users

63.0

Requires Chatter

No

Signature

```
public static ConnectApi.CalculateCartResult evaluateShipping(String webstoreId, String activeCartOrId, String effectiveAccountId, ConnectApi.CartEvaluateShippingInput cartEvaluateShippingInput)
```

Parameters

webstoreId
Type: String

ID of the webstore.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value indicates a cart with a status that isn't `Closed` or `PendingDelete`.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

cartEvaluateShippingInput
Type: `ConnectApi.CartEvaluateShippingInput`

A `ConnectApi.CartEvaluateShippingInput` object representing a shipping address and any custom fields.

Return Value

Type: `ConnectApi.CalculateCartResult`

Usage

Buyers with read access to carts can add, update, and delete items in carts.

This method respects buyer View Product entitlements and only users entitled to view product data can access it.

`evaluateTaxes(webstoreId, activeCartOrId, effectiveAccountId, cartEvaluateTaxInput)`

Evaluate taxes for a cart.

API Version

63.0

Available to Guest Users

63.0

Requires Chatter

No

Signature

```
public static ConnectApi.CalculateCartResult evaluateTaxes(String webstoreId, String activeCartOrId, String effectiveAccountId, ConnectApi.CartEvaluateTaxInput cartEvaluateTaxInput)
```

Parameters

webstoreId
Type: String

ID of the webstore.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value indicates a cart with a status that isn't `Closed` or `PendingDelete`.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

cartEvaluateTaxInput
Type: `ConnectApi.CartEvaluateTaxInput`

A `ConnectApi.CartEvaluateTaxInput` object representing a shipping address and any custom fields.

Return Value

Type: `ConnectApi.CalculateCartResult`

Usage

Buyers with read access to carts can add, update, and delete items in carts.

This method respects buyer View Product entitlements and only users entitled to view product data can access it.

`getCartCoupons(webstoreId, effectiveAccountId, activeCartOrId)`

Get coupons for a cart.

API Version

54.0

Available to Guest Users

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartCouponCollection getCartCoupons(String webstoreId, String effectiveAccountId, String activeCartOrId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`.

Return Value

Type: `ConnectApi.CartCouponCollection`

`getCartCoupons(webstoreId, effectiveAccountId, activeCartOrId, currencyIsoCode)`

Get coupons for a cart.

API Version

57.0

Available to Guest Users

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartCouponCollection getCartCoupons(String webstoreId, String effectiveAccountId, String activeCartOrId, String currencyIsoCode)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value indicates a cart with a status that isn't `Closed` or `PendingDelete`.

Return Value

Type: `ConnectApi.CartCouponCollection`

getCartItemPromotion(webstoreId, effectiveAccountId, activeCartOrId, cartItemPromotionCollectionInput)

Get promotions for a cart item.

API Version

52.0

Available to Guest Users

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartItemPromotionCollectionOutputRepresentation
getCartItemPromotion(String webstoreId, String effectiveAccountId, String activeCartOrId,
ConnectApi.CartItemPromotionCollectionInputRepresentation
cartItemPromotionCollectionInput)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value indicates a cart with a status that isn't `Closed` or `PendingDelete`.

cartItemPromotionCollectionInput
Type: `ConnectApi.CartItemPromotionCollectionInputRepresentation`

Promotions for a cart item.

Return Value

Type: `ConnectApi.CartItemPromotionCollectionOutputRepresentation`

`getCartItemPromotion(webstoreId, effectiveAccountId, activeCartOrId, cartItemPromotionCollectionInput, currencyIsoCode)`

Get a promotion for a cart item.

API Version

57.0

Available to Guest Users

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartItemPromotionCollectionOutputRepresentation
getCartItemPromotion(String webstoreId, String effectiveAccountId, String activeCartOrId,
ConnectApi.CartItemPromotionCollectionInputRepresentation
cartItemPromotionCollectionInput, String currencyIsoCode)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value indicates a cart with a status that isn't `Closed` or `PendingDelete`.

cartItemPromotionCollectionInput
Type: ConnectApi.CartItemPromotionCollectionInputRepresentation

Promotions for a cart item.

currencyIsoCode
Type: String

Currency ISO code of the cart.

Return Value

Type: ConnectApi.CartItemPromotionCollectionOutputRepresentation

getCartItems(webstoreId, effectiveAccountId, activeCartOrId)

Get items in a cart.

API Version

49.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartItemCollection getCartItems(String webstoreId, String effectiveAccountId, String activeCartOrId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`.

Return Value

Type: ConnectApi.CartItemCollection

getCartItems(webstoreId, effectiveAccountId, activeCartOrId, pageParam)

Get a page of items in a cart.

API Version

49.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartItemCollection getCartItems(String webstoreId, String effectiveAccountId, String activeCartOrId, String pageParam)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`.

pageParam
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

Return Value

Type: ConnectApi.CartItemCollection

`getCartItems(webstoreId, effectiveAccountId, activeCartOrId, pageParam, sortParam)`

Get a sorted page of items in a cart.

API Version

49.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartItemCollection getCartItems(String webstoreId, String effectiveAccountId, String activeCartOrId, String pageParam, ConnectApi.CartItemSortOrder sortParam)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`.

pageParam
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

sortParam
Type: `ConnectApi.CartItemSortOrder`

Sort order for items in a cart. Values are:

- `CreatedDateAsc`—Sorts by oldest creation date.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `NameAsc`—Sorts by name in ascending alphabetical order (A–Z).
- `NameDesc`—Sorts by name in descending alphabetical order (Z–A).

- `SalesPriceAsc`—Sorts from lowest to highest negotiated price.
- `SalesPriceDesc`—Sorts from highest to lowest negotiated price.

If `null`, the default is `CreatedDateDesc`.

Return Value

Type: `ConnectApi.CartItemCollection`

`getCartItems(webstoreId, effectiveAccountId, activeCartOrId, pageParam, pageSize)`

Get a specified size page of items in a cart.

API Version

49.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartItemCollection getCartItems(String webstoreId, String effectiveAccountId, String activeCartOrId, String pageParam, Integer pageSize)
```

Parameters

`webstoreId`
Type: String

ID of the webstore.

`effectiveAccountId`
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

`activeCartOrId`
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`.

`pageParam`
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.CartItemCollection`

getCartItems(webstoreId, effectiveAccountId, activeCartOrId, pageParam, pageSize, sortParam)

Get a specified size, sorted page of items in a cart.

API Version

49.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartItemCollection getCartItems(String webstoreId, String effectiveAccountId, String activeCartOrId, String pageParam, Integer pageSize, ConnectApi.CartItemSortOrder sortParam)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`.

pageParam
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.CartItemSortOrder`

Sort order for items in a cart. Values are:

- `CreatedDateAsc`—Sorts by oldest creation date.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `NameAsc`—Sorts by name in ascending alphabetical order (A–Z).
- `NameDesc`—Sorts by name in descending alphabetical order (Z–A).
- `SalesPriceAsc`—Sorts from lowest to highest negotiated price.
- `SalesPriceDesc`—Sorts from highest to lowest negotiated price.

If `null`, the default is `CreatedDateDesc`.

Return Value

Type: `ConnectApi.CartItemCollection`

getCartItems(webstoreId, effectiveAccountId, activeCartOrId, productFields, pageParam, pageSize, sortParam)

Get a specified size, sorted page of items filtered by product fields in a cart.

API Version

49.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartItemCollection getCartItems(String webstoreId, String effectiveAccountId, String activeCartOrId, String productFields, String pageParam, Integer pageSize, ConnectApi.CartItemSortOrder sortParam)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`.

productFields
Type: String

Comma-separated list of up to 15 product fields. Results include fields that you have access to. Some product fields (such as `productName` and `sku`) are returned even when not included in the *productFields* parameter.

pageParam
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.CartItemSortOrder`

Sort order for items in a cart. Values are:

- `CreatedDateAsc`—Sorts by oldest creation date.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `NameAsc`—Sorts by name in ascending alphabetical order (A–Z).
- `NameDesc`—Sorts by name in descending alphabetical order (Z–A).
- `SalesPriceAsc`—Sorts from lowest to highest negotiated price.
- `SalesPriceDesc`—Sorts from highest to lowest negotiated price.

If `null`, the default is `CreatedDateDesc`.

Return Value

Type: `ConnectApi.CartItemCollection`

getCartItems(webstoreId, effectiveAccountId, activeCartOrId, productFields, pageParam, pageSize, sortParam, currencyIsoCode)

Get a specified size, sorted page of items filtered by product fields in a cart.

API Version

57.0

Available to Guest Users

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartItemCollection getCartItems(String webstoreId, String effectiveAccountId, String activeCartOrId, String productFields, String pageParam, Integer pageSize, ConnectApi.CartItemSortOrder sortParam, String currencyIsoCode)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`.

productFields
Type: String

Comma-separated list of up to 15 product fields. Results include fields that you have access to. Some product fields (such as `productName` and `sku`) are returned even when not included in the *productFields* parameter.

pageParam
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

currencyIsoCode
Type: String

Currency ISO code of the cart.

Return Value

Type: `ConnectApi.CartItemCollection`

`getCartItems(webstoreId, effectiveAccountId, activeCartOrId, productFields, pageParam, pageSize, sortParam, currencyIsoCode, includePromotions, includeCoupons)`

Get a sorted page of items in a cart, including coupons and promotions.

API Version

59.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartItemCollection getCartItems(String webstoreId, String effectiveAccountId, String activeCartOrId, String productFields, String pageParam, Integer pageSize, ConnectApi.CartItemSortOrder sortParam, String currencyIsoCode, Boolean includePromotions, Boolean includeCoupons)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`.

productFields
Type: String

Comma-separated list of up to 15 product fields. Results include fields that you have access to. Some product fields (such as `productName` and `sku`) are returned even when not included in the *`productFields`* parameter.

pageParam
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

includePromotions
Type: Boolean

Indicates whether to include coupons (`True`) or not (`False`)

includeCoupons
Type: Boolean

Indicates whether to include promotions (`True`) or not (`False`).

Return Value

Type: `ConnectApi.CartItemCollection`

getCartItems(webstoreId, effectiveAccountId, activeCartOrId, productFields, pageParam, pageSize, sortParam, currencyIsoCode, includePromotions, includeCoupons, pageNumber)

Get a specific, sorted page of items in a cart, including coupons and promotions.

API Version

60.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartItemCollection getCartItems(String webstoreId, String effectiveAccountId, String activeCartOrId, String productFields, String pageParam, Integer pageSize, ConnectApi.CartItemSortOrder sortParam, String currencyIsoCode, Boolean includePromotions, Boolean includeCoupons, Integer pageNumber)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`.

`productFields`
Type: String

Comma-separated list of up to 15 product fields. Results include fields that you have access to. Some product fields (such as `productName` and `sku`) are returned even when not included in the *`productFields`* parameter.

`pageParam`
Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

Description

`pageSize`
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

`sortParam`
Type: `ConnectApi.CartItemSortOrder`

Sort order for items in a cart. Values are:

- `CreatedDateAsc`—Sorts by oldest creation date.
- `CreatedDateDesc`—Sorts by most recent creation date.
- `NameAsc`—Sorts by name in ascending alphabetical order (A–Z).
- `NameDesc`—Sorts by name in descending alphabetical order (Z–A).
- `SalesPriceAsc`—Sorts from lowest to highest negotiated price.
- `SalesPriceDesc`—Sorts from highest to lowest negotiated price.

If `null`, the default is `CreatedDateDesc`.

`includePromotions`
Type: Boolean

Indicates whether to include coupons (`True`) or not (`False`)

`includeCoupons`
Type: Boolean

Indicates whether to include promotions (`True`) or not (`False`).

`pageNumber`
Type: Integer

Specifies the requested page number.

Return Value

Type: `ConnectApi.CartItemCollection`

`getCartPromotions(webstoreId, effectiveAccountId, activeCartOrId)`

Get promotions for a cart.

API Version

53.0

Available to Guest Users

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartPromotionCollection getCartPromotions(String webstoreId,
String effectiveAccountId, String activeCartOrId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value indicates a cart with a status that isn't `Closed` or `PendingDelete`.

Return Value

Type: `ConnectApi.CartPromotionCollection`

`getCartPromotions(webstoreId, effectiveAccountId, activeCartOrId, currencyIsoCode)`

Get promotions for a cart in a specific currency.

API Version

57.0

Available to Guest Users

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartPromotionCollection getCartPromotions(String webstoreId,
String effectiveAccountId, String activeCartOrId, String currencyIsoCode)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value indicates a cart with a status that isn't `Closed` or `PendingDelete`.

currencyIsoCode
Type: String

Currency ISO code of the cart.

Return Value

Type: `ConnectApi.CartPromotionCollection`

`getCartSummary(webstoreId, effectiveAccountId, activeCartOrId)`

Get a cart.

API Version

49.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartSummary getCartSummary(String webstoreId, String
effectiveAccountId, String activeCartOrId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`. If you specify `active` and there isn't an active cart, you get an error.

Return Value

Type: `ConnectApi.CartSummary`

getCartSummary(webstoreId, effectiveAccountId, activeCartOrId, currencyIsoCode)

Get a cart in a specific currency.

API Version

57.0

Available to Guest Users

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartSummary getCartSummary(String webstoreId, String
effectiveAccountId, String activeCartOrId, String currencyIsoCode)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`. If you specify `active` and there isn't an active cart, you get an error.

currencyIsoCode
Type: String

Currency ISO code of the cart.

Return Value

Type: `ConnectApi.CartSummary`

getOrCreateActiveCartSummary(webstoreId, effectiveAccountId, activeCartOrId)

Get a cart or create an active cart if one doesn't exist.

API Version

49.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartSummary getOrCreateActiveCartSummary(String webstoreId,
String effectiveAccountId, String activeCartOrId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`. If you specify `active` and there isn't an active cart, one is created.

Return Value

Type: `ConnectApi.CartSummary`

Usage

Buyers with read access to carts can create and delete carts.

`getOrCreateActiveCartSummary(webstoreId, effectiveAccountId, activeCartOrId, currencyIsoCode)`

Get a cart in a specific currency, or create an active cart if one doesn't exist.

API Version

57.0

Available to Guest Users

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartSummary getOrCreateActiveCartSummary(String webstoreId,
String effectiveAccountId, String activeCartOrId, String currencyIsoCode)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`. If you specify `active` and there isn't an active cart, one is created.

currencyIsoCode
Type: String

Currency ISO code of the cart.

Return Value

Type: `ConnectApi.CartSummary`

Usage

Buyers with read access to carts can create and delete carts.

getProductCartItem(webstoreId, effectiveAccountId, activeCartOrId, productId, currencyIsoCode)

Get cart items of a specific product.

API Version

60.0

Available to Guest Users

60.0

Requires Chatter

No

Signature

```
public static ConnectApi.ProductCartItem getProductCartItem(String webstoreId, String effectiveAccountId, String activeCartOrId, String productId, String currencyIsoCode)
```

Parameters

effectiveAccountId
: Type: String

  ID of the buyer account or guest buyer profile for which the request is made. If unspecified, the default value is determined from context.

currencyIsoCode
: Type: String

  Currency ISO code of the cart.

Return Value

Type: `ConnectApi.ProductCartItem` on page 2240

getProductCartItems(webstoreId, effectiveAccountId, activeCartOrId, pageSize, pageNumber, currencyIsoCode)

Get the items in a cart, sorted by product ID.

API Version

60.0

Available to Guest Users

60.0

Requires Chatter

No

Signature

```
public static ConnectApi.ProductCartItemCollection getProductCartItems(String webstoreId,
String effectiveAccountId, String activeCartOrId, Integer pageSize, Integer pageNumber,
String currencyIsoCode)
```

Parameters

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If unspecified, the default value is determined from context.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you don't specify a value, the default size is 25.

pageNumber
Type: Integer

Specifies the requested page number.

currencyIsoCode
Type: String

Currency ISO code of the cart.

Return Value

Type: `ConnectApi.ProductCartItemCollection` on page 2240

makeCartPrimary(webstoreId, activeCartOrId, effectiveAccountId)

Make a secondary cart a primary cart.

API Version

53.0

Available to Guest Users

56.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.CommerceActionResult makeCartPrimary(String webstoreId, String activeCartOrId, String effectiveAccountId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

Return Value

Type: `ConnectApi.CommerceActionResult`

preserveGuestCart(webstoreId, effectiveAccountId, activeCartOrId, currencyIsoCode)

Preserve cart contents when a guest logs in as an authenticated customer. Sets the guest cart status to `PendingDelete` in a B2B store or `Closed` in a D2C store.

API Version

60.0

Requires Chatter

No

Signature

```
public static ConnectApi.PreserveCart preserveGuestCart(String webstoreId, String effectiveAccountId, String activeCartOrId, String currencyIsoCode)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If null, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`.

currencyIsoCode
Type: String

The currency ISO code of the cart.

Return Value

Type: `ConnectApi.PreserveCart` on page 2233

setCartMessagesVisibility(webstoreId, activeCartOrId, effectiveAccountId, messageVisibility)

Set the visibility for cart messages.

API Version

50.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartMessagesVisibilityResult setCartMessagesVisibility(String webstoreId, String activeCartOrId, String effectiveAccountId, ConnectApi.CartMessagesVisibilityInput messageVisibility)
```

Parameters

webstoreId
Type: String

ID of the webstore.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

`messageVisibility`
Type: `ConnectApi.CartMessagesVisibilityInput`

A `ConnectApi.CartMessagesVisibilityInput` object specifying the visibility setting.

Return Value

Type: `ConnectApi.CartMessagesVisibilityResult`

`updateCartItem(webstoreId, effectiveAccountId, activeCartOrId, cartItemId, cartItem)`

Update an item in a cart.

API Version

49.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartItem updateCartItem(String webstoreId, String effectiveAccountId, String activeCartOrId, String cartItemId, ConnectApi.CartItemInput cartItem)
```

Parameters

`webstoreId`
Type: String

ID of the webstore.

`effectiveAccountId`
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

`activeCartOrId`
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`.

`cartItemId`
Type: String

ID of the cart item.

cartItem
Type: ConnectApi.CartItemInput

A ConnectApi.CartItemInput object representing a cart item to update.

Return Value

Type: ConnectApi.CartItem

Usage

Buyers with read access to carts can add, update, and delete items in carts.

This method respects buyer View Product entitlements and only users entitled to view product data can access it.

updateCartItem(webstoreId, effectiveAccountId, activeCartOrId, cartItemId, cartItem, currencyIsoCode)

Update an item in a cart of a specific currency.

API Version

57.0

Available to Guest Users

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartItem updateCartItem(String webstoreId, String
effectiveAccountId, String activeCartOrId, String cartItemId, ConnectApi.CartItemInput
cartItem, String currencyIsoCode)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If null, the default value is determined from context.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value is available in version 50.0 and later and indicates a cart with a status that isn't `Closed` or `PendingDelete`.

`cartItemId`
Type: String

ID of the cart item.

`cartItem`
Type: `ConnectApi.CartItemInput`

A `ConnectApi.CartItemInput` object representing a cart item to update.

`currencyIsoCode`
Type: String

The currency ISO code of the cart.

Return Value

Type: `ConnectApi.CartItem`

Usage

Buyers with read access to carts can add, update, and delete items in carts.

This method respects buyer View Product entitlements and only users entitled to view product data can access it.

`upsertInventoryReservation(webstoreId, activeCartOrId, effectiveAccountId, cartInventoryReservationInput)` (Pilot)

Create or update an inventory reservation.

Note: This feature is not generally available and is being piloted with certain Customers subject to additional terms and conditions. It is not part of your purchased Services. This feature is subject to change, may be discontinued with no notice at any time in Salesforce's sole discretion, and Salesforce may never make this feature generally available. Make your purchase decisions only on the basis of generally available products and features. This feature is made available on an AS IS basis and use of this feature is at your sole risk.

API Version

58.0

Available to Guest Users

58.0

Requires Chatter

No

Signature

```
public static ConnectApi.CartInventoryReservationOutputRepresentation
upsertInventoryReservation(String webstoreId, String activeCartOrId, String
```

```
effectiveAccountId, ConnectApi.CartInventoryReservationInputRepresentation
cartInventoryReservationInput)
```

Parameters

webstoreId
Type: String

ID of the webstore.

activeCartOrId
Type: String

ID of the cart, `active`, or `current`. The `current` value indicates a cart with a status that isn't `Closed` or `PendingDelete`.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

cartInventoryReservationInput
Type: `ConnectApi.CartInventoryReservationInputRepresentation`

A `ConnectApi.CartInventoryReservationInputRepresentation` input class indicating the reservation duration.

Return Value

Type: `ConnectApi.CartInventoryReservationOutputRepresentation`

CommerceCatalog Class

Get products, product categories, and product category paths.

Namespace

ConnectApi

CommerceCatalog Methods

These methods are for `CommerceCatalog`. All methods are static.

IN THIS SECTION:

getProduct(webstoreId, productId, effectiveAccountId, fields, excludeFields, mediaGroups, excludeMedia, excludeEntitlementDetails, excludePrimaryProductCategory)

Get a product.

getProduct(webstoreId, productId, effectiveAccountId, fields, excludeFields, mediaGroups, excludeMedia, excludeEntitlementDetails, excludePrimaryProductCategory, excludeVariationInfo, excludeAttributeSetInfo)

Get a product with variation and attribute information.

getProduct(webstoreId, productId, effectiveAccountId, fields, excludeFields, mediaGroups, excludeMedia, excludeEntitlementDetails, excludePrimaryProductCategory, excludeVariationInfo, excludeAttributeSetInfo, excludeQuantityRule)

Get a product with quantity rule information.

getProduct(webstoreId, productId, effectiveAccountId, fields, excludeFields, mediaGroups, excludeMedia, excludeEntitlementDetails, excludePrimaryProductCategory, excludeVariationInfo, excludeAttributeSetInfo, excludeQuantityRule, excludeProductSellingModels)
Get detailed information for a product, optionally including information about its product selling models.

getProduct(webstoreId, productId, effectiveAccountId, fields, mediaGroups, excludeFields, excludeMedia, excludePrimaryProductCategory, excludeVariationInfo, excludeAttributeSetInfo, excludeQuantityRule, excludeProductSellingModels)
Get detailed information for a product without its entitlement information.

getProductCategory(webstoreId, productCategoryId, effectiveAccountId, fields, excludeFields, mediaGroups, excludeMedia)
Get a product category.

getProductCategoryChildren(webstoreId, effectiveAccountId, parentProductCategoryId, fields, excludeFields, mediaGroups, excludeMedia)
Get product categories.

getProductCategoryPath(webstoreId, productCategoryId)
Get the product category path from the root category to the current category.

getProductChildCollection(webstoreId, productId, effectiveAccountId, fields, mediaGroups, excludeFields, excludeMedia, excludeAttributeSetInfo, excludeQuantityRule, pageToken, pageSize)
Get a collection of child products related to a parent product.

getProducts(webstoreId, effectiveAccountId, ids, skus, fields, excludeMedia, excludePrices)
Get fields, prices, and default images for a list of products.

getProducts(webstoreId, effectiveAccountId, ids, skus, fields, excludeMedia)
Get fields and default images for a list of products.

`getProduct(webstoreId, productId, effectiveAccountId, fields, excludeFields, mediaGroups, excludeMedia, excludeEntitlementDetails, excludePrimaryProductCategory)`

Get a product.

API Version

49.0

Available to Guest Users

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.ProductDetail getProduct(String webstoreId, String productId,
String effectiveAccountId, List<String> fields, Boolean excludeFields, List<String>
mediaGroups, Boolean excludeMedia, Boolean excludeEntitlementDetails, Boolean
excludePrimaryProductCategory)
```

Parameters

webstoreId
Type: String

ID of the webstore.

productId
Type: String

ID of the product.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

fields
Type: List<String>

Comma-separated list of field names.

If this list is empty or unspecified, all fields are returned. There is no limit to the number of fields you can specify. The number of fields and number of characters in the field name may affect the URL size limit. If `excludeFields` and `fields` are specified, the `excludeFields` parameter takes precedence.

excludeFields
Type: Boolean

Specifies whether the fields are returned (`false`) or not (`true`). If unspecified, defaults to `false`.

mediaGroups
Type: List<String>

Comma-separated list of developer names of media group records.

If this list is empty or unspecified, all media groups are returned. If `excludeMedia` and `mediaGroups` are specified, the `excludeMedia` parameter takes precedence.

excludeMedia
Type: Boolean

Specifies whether the media groups and default images of the product are returned (`false`) or not (`true`). If unspecified, defaults to `false`.

excludeEntitlementDetails
Type: Boolean

Specifies whether the entitlement details of the product are returned (`false`) or not (`true`). If unspecified, defaults to `false`.

excludePrimaryProductCategory
Type: Boolean

Specifies whether the primary category path of the product is returned (`false`) or not (`true`). If unspecified, defaults to `false`.

Return Value

Type: `ConnectApi.ProductDetail`

Usage

This method respects buyer View Product entitlements and only users entitled to view product data can access it.

`getProduct(webstoreId, productId, effectiveAccountId, fields, excludeFields, mediaGroups, excludeMedia, excludeEntitlementDetails, excludePrimaryProductCategory, excludeVariationInfo, excludeAttributeSetInfo)`

Get a product with variation and attribute information.

API Version

50.0

Available to Guest Users

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.ProductDetail getProduct(String webstoreId, String productId, String effectiveAccountId, List<String> fields, Boolean excludeFields, List<String> mediaGroups, Boolean excludeMedia, Boolean excludeEntitlementDetails, Boolean excludePrimaryProductCategory, Boolean excludeVariationInfo, Boolean excludeAttributeSetInfo)
```

Parameters

webstoreId
Type: String

ID of the webstore.

productId
Type: String

ID of the product.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

fields
Type: List<String>

Comma-separated list of field names.

If this list is empty or unspecified, all fields are returned. There is no limit to the number of fields you can specify. The number of fields and number of characters in the field name may affect the URL size limit. If `excludeFields` and `fields` are specified, the `excludeFields` parameter takes precedence.

excludeFields
Type: Boolean

Specifies whether the fields are returned (`false`) or not (`true`). If unspecified, defaults to `false`.

mediaGroups
Type: List<String>

Comma-separated list of developer names of media group records.

If this list is empty or unspecified, all media groups are returned. If `excludeMedia` and `mediaGroups` are specified, the `excludeMedia` parameter takes precedence.

excludeMedia
Type: Boolean

Specifies whether the media groups and default images of the product are returned (`false`) or not (`true`). If unspecified, defaults to `false`.

excludeEntitlementDetails
Type: Boolean

Specifies whether the entitlement details of the product are returned (`false`) or not (`true`). If unspecified, defaults to `false`.

excludePrimaryProductCategory
Type: Boolean

Specifies whether the primary category path of the product is returned (`false`) or not (`true`). If unspecified, defaults to `false`.

excludeVariationInfo
Type: Boolean

Specifies whether the variation information for the product is returned (`false`) or not (`true`). If unspecified, defaults to `false`.

excludeAttributeSetInfo
Type: Boolean

Specifies whether the attribute set information for the product is returned (`false`) or not (`true`). If unspecified, defaults to `false`.

Return Value

Type: `ConnectApi.ProductDetail`

Usage

This method respects buyer View Product entitlements and only users entitled to view product data can access it.

getProduct(webstoreId, productId, effectiveAccountId, fields, excludeFields, mediaGroups, excludeMedia, excludeEntitlementDetails, excludePrimaryProductCategory, excludeVariationInfo, excludeAttributeSetInfo, excludeQuantityRule)

Get a product with quantity rule information.

API Version

52.0

Available to Guest Users

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.ProductDetail getProduct(String webstoreId, String productId,
String effectiveAccountId, List<String> fields, Boolean excludeFields, List<String>
mediaGroups, Boolean excludeMedia, Boolean excludeEntitlementDetails, Boolean
excludePrimaryProductCategory, Boolean excludeVariationInfo, Boolean
excludeAttributeSetInfo, Boolean excludeQuantityRule)
```

Parameters

webstoreId
Type: String

ID of the webstore.

productId
Type: String

ID of the product.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

fields
Type: List<String>

Comma-separated list of field names.

If this list is empty or unspecified, all fields are returned. There is no limit to the number of fields you can specify. The number of fields and number of characters in the field name may affect the URL size limit. If `excludeFields` and `fields` are specified, the `excludeFields` parameter takes precedence.

excludeFields
Type: Boolean

Specifies whether the fields are returned (`false`) or not (`true`). If unspecified, defaults to `false`.

mediaGroups
Type: List<String>

Comma-separated list of developer names of media group records.

If this list is empty or unspecified, all media groups are returned. If `excludeMedia` and `mediaGroups` are specified, the `excludeMedia` parameter takes precedence.

excludeMedia
Type: Boolean

Specifies whether the media groups and default images of the product are returned (`false`) or not (`true`). If unspecified, defaults to `false`.

excludeEntitlementDetails
Type: Boolean

Specifies whether the entitlement details of the product are returned (`false`) or not (`true`). If unspecified, defaults to `false`.

excludePrimaryProductCategory
Type: Boolean

Specifies whether the primary category path of the product is returned (`false`) or not (`true`). If unspecified, defaults to `false`.

excludeVariationInfo
Type: Boolean

Specifies whether the variation information for the product is returned (`false`) or not (`true`). If unspecified, defaults to `false`.

excludeAttributeSetInfo
Type: Boolean

Specifies whether the attribute set information for the product is returned (`false`) or not (`true`). If unspecified, defaults to `false`.

excludeQuantityRule
Type: Boolean

Specifies whether the quantity rule information for the product is returned (`false`) or not (`true`). If unspecified, defaults to `false`.

Return Value

Type: `ConnectApi.ProductDetail`

Usage

This method respects buyer View Product entitlements and only users entitled to view product data can access it.

getProduct(webstoreId, productId, effectiveAccountId, fields, excludeFields, mediaGroups, excludeMedia, excludeEntitlementDetails, excludePrimaryProductCategory, excludeVariationInfo, excludeAttributeSetInfo, excludeQuantityRule, excludeProductSellingModels)

Get detailed information for a product, optionally including information about its product selling models.

API Version

56.0

Available to Guest Users

56.0

Requires Chatter

No

Signature

```
public static ConnectApi.ProductDetail getProduct(String webstoreId, String productId,
String effectiveAccountId, List<String> fields, Boolean excludeFields, List<String>
mediaGroups, Boolean excludeMedia, Boolean excludeEntitlementDetails, Boolean
excludePrimaryProductCategory, Boolean excludeVariationInfo, Boolean
```

```
excludeAttributeSetInfo, Boolean excludeQuantityRule, Boolean
excludeProductSellingModels)
```

Parameters

webstoreId
Type: String

ID of the webstore.

productId
Type: String

ID of the product.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

fields
Type: List<String>

Comma-separated list of field names.

If this list is empty or unspecified, all fields are returned. There is no limit to the number of fields you can specify. The number of fields and number of characters in the field name may affect the URL size limit. If `excludeFields` and `fields` are specified, the `excludeFields` parameter takes precedence.

excludeFields
Type: Boolean

Specifies whether the fields are returned (`false`) or not (`true`). If unspecified, defaults to `false`.

mediaGroups
Type: List<String>

Comma-separated list of developer names of media group records.

If this list is empty or unspecified, all media groups are returned. If `excludeMedia` and `mediaGroups` are specified, the `excludeMedia` parameter takes precedence.

excludeMedia
Type: Boolean

Specifies whether the media groups and default images of the product are returned (`false`) or not (`true`). If unspecified, defaults to `false`.

excludeEntitlementDetails
Type: Boolean

Specifies whether the entitlement details of the product are returned (`false`) or not (`true`). If unspecified, defaults to `false`.

excludePrimaryProductCategory
Type: Boolean

Specifies whether the primary category path of the product is returned (`false`) or not (`true`). If unspecified, defaults to `false`.

excludeVariationInfo
Type: Boolean

Specifies whether the variation information for the product is returned (`false`) or not (`true`). If unspecified, defaults to `false`.

excludeAttributeSetInfo
Type: Boolean

Specifies whether the attribute set information for the product is returned (`false`) or not (`true`). If unspecified, defaults to `false`.

excludeQuantityRule
Type: Boolean

Specifies whether the quantity rule information for the product is returned (`false`) or not (`true`). If unspecified, defaults to `false`.

excludeProductSellingModels
Type: Boolean

Specifies whether product selling models are returned or not. The behavior of this parameter depends on whether you turn on the CommerceSubscription permission. If the permission is on, and if you set the parameter to `false` (or if you omit the parameter), product selling models are returned. If the permission is on, and if you set the parameter to `true`, product selling models are not returned. If the permission is off, product selling models are not returned, regardless of whether you omit the parameter or provide a value.

Return Value

Type: `ConnectApi.ProductDetail`

Usage

This method respects buyer View Product entitlements and only users entitled to view product data can access it.

getProduct(webstoreId, productId, effectiveAccountId, fields, mediaGroups, excludeFields, excludeMedia, excludePrimaryProductCategory, excludeVariationInfo, excludeAttributeSetInfo, excludeQuantityRule, excludeProductSellingModels)

Get detailed information for a product without its entitlement information.

API Version

57.0

Available to Guest Users

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.ProductDetail getProduct(String webstoreId, String productId,
String effectiveAccountId, List<String> fields, List<String> mediaGroups, Boolean
excludeFields, Boolean excludeMedia, Boolean excludePrimaryProductCategory, Boolean
excludeVariationInfo, Boolean excludeAttributeSetInfo, Boolean excludeQuantityRule,
Boolean excludeProductSellingModels)
```

Parameters

webstoreId
Type: String

ID of the webstore.

productId
Type: String

ID of the product.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

fields
Type: List<String>

Comma-separated list of field names.

If this list is empty or unspecified, all fields are returned. There is no limit to the number of fields you can specify. The number of fields and number of characters in the field name may affect the URL size limit. If `excludeFields` and `fields` are specified, the `excludeFields` parameter takes precedence.

mediaGroups
Type: List<String>

Comma-separated list of developer names of media group records.

If this list is empty or unspecified, all media groups are returned. If `excludeMedia` and `mediaGroups` are specified, the `excludeMedia` parameter takes precedence.

excludeFields
Type: Boolean

Specifies whether the fields are returned (`false`) or not (`true`). If unspecified, defaults to `false`.

excludeMedia
Type: Boolean

Specifies whether the media groups and default images of the product are returned (`false`) or not (`true`). If unspecified, defaults to `false`.

excludePrimaryProductCategory
Type: Boolean

Specifies whether the primary category path of the product is returned (`false`) or not (`true`). If unspecified, defaults to `false`.

excludeVariationInfo
Type: Boolean

Specifies whether the variation information for the product is returned (`false`) or not (`true`). If unspecified, defaults to `false`.

excludeAttributeSetInfo
Type: Boolean

Specifies whether the attribute set information for the product is returned (`false`) or not (`true`). If unspecified, defaults to `false`.

excludeQuantityRule
Type: Boolean

Specifies whether the quantity rule information for the product is returned (`false`) or not (`true`). If unspecified, defaults to `false`.

excludeProductSellingModels
Type: Boolean

Specifies whether product selling models are returned or not. The behavior of this parameter depends on whether you turn on the CommerceSubscription permission. If the permission is on, and if you set the parameter to `false` (or if you omit the parameter), product selling models are returned. If the permission is on, and if you set the parameter to `true`, product selling models are not returned. If the permission is off, product selling models are not returned, regardless of whether you omit the parameter or provide a value.

Return Value

Type: `ConnectApi.ProductDetail`

Usage

This method respects buyer View Product entitlements and only users entitled to view product data can access it.

getProductCategory(webstoreId, productCategoryId, effectiveAccountId, fields, excludeFields, mediaGroups, excludeMedia)

Get a product category.

API Version

49.0

Available to Guest Users

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.ProductCategoryDetail getProductCategory(String webstoreId,
String productCategoryId, String effectiveAccountId, List<String> fields, Boolean
excludeFields, List<String> mediaGroups, Boolean excludeMedia)
```

Parameters

webstoreId
Type: String

ID of the webstore.

productCategoryId
Type: String

ID of the product category.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

fields
Type: List<String>

Comma-separated list of field names.

If this list is empty or unspecified, all fields are returned. There is no limit to the number of fields you can specify. The number of fields and number of characters in the field name may affect the URL size limit. If `excludeFields` and `fields` are specified, the `excludeFields` parameter takes precedence.

excludeFields
Type: Boolean

Specifies whether the fields are returned (`false`) or not (`true`). If unspecified, defaults to `false`.

mediaGroups
Type: List<String>

Comma-separated list of developer names of media group records.

If this list is empty or unspecified, all media groups are returned. If `excludeMedia` and `mediaGroups` are specified, the `excludeMedia` parameter takes precedence.

excludeMedia
Type: Boolean

Specifies whether the media groups and default images of the product are returned (`false`) or not (`true`). If unspecified, defaults to `false`.

Return Value

Type: `ConnectApi.ProductCategoryDetail`

getProductCategoryChildren(webstoreId, effectiveAccountId, parentProductCategoryId, fields, excludeFields, mediaGroups, excludeMedia)

Get product categories.

API Version

52.0

Available to Guest Users

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.ProductCategoryDetailCollection
getProductCategoryChildren(String webstoreId, String effectiveAccountId, String
parentProductCategoryId, List<String> fields, Boolean excludeFields, List<String>
mediaGroups, Boolean excludeMedia)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

parentProductCategoryId
Type: String

ID of the product category for which you want to get all the children product categories. If `null`, returns all the top-level product categories for the store.

fields
Type: List<String>

Comma-separated list of field names.

If this list is empty or unspecified, all fields are returned. There is no limit to the number of fields you can specify. The number of fields and number of characters in the field name may affect the URL size limit. If `excludeFields` and `fields` are specified, the `excludeFields` parameter takes precedence.

excludeFields
Type: Boolean

Specifies whether the fields are returned (`false`) or not (`true`). If unspecified, defaults to `false`.

mediaGroups
Type: List<String>

Comma-separated list of developer names of media group records.

If this list is empty or unspecified, all media groups are returned. If `excludeMedia` and `mediaGroups` are specified, the `excludeMedia` parameter takes precedence.

excludeMedia
Type: Boolean

Specifies whether the media groups and default images of the product are returned (`false`) or not (`true`). If unspecified, defaults to `false`.

Return Value

Type: `ConnectApi.ProductCategoryDetailCollection`

`getProductCategoryPath(webstoreId, productCategoryId)`

Get the product category path from the root category to the current category.

API Version

49.0

Available to Guest Users

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.ProductCategoryPath getProductCategoryPath(String webstoreId,
String productCategoryId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

productCategoryId
Type: String

ID of the product category.

Return Value

Type: `ConnectApi.ProductCategoryPath`

Usage

This method respects buyer View Product entitlements and only users entitled to view product data can access it.

getProductChildCollection(webstoreId, productId, effectiveAccountId, fields, mediaGroups, excludeFields, excludeMedia, excludeAttributeSetInfo, excludeQuantityRule, pageToken, pageSize)

Get a collection of child products related to a parent product.

API Version

57.0

Available to Guest Users

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.ProductChildCollection getProductChildCollection(String
webstoreId, String productId, String effectiveAccountId, List<String> fields,
List<String> mediaGroups, Boolean excludeFields, Boolean excludeMedia, Boolean
excludeAttributeSetInfo, Boolean excludeQuantityRule, String pageToken, Integer pageSize)
```

Parameters

webstoreId
Type: String

ID of the webstore.

productId
Type: String

ID of the product.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If unspecified, the default value is determined from context.

fields
Type: List<String>

Comma-separated list of field names.

If this list is empty or unspecified, all fields are returned. There's no limit to the number of fields you can specify. The number of fields and number of characters in the field name may affect the URL size limit. If `excludeFields` and `fields` are specified, the `excludeFields` parameter takes precedence.

mediaGroups
Type: List<String>

Comma-separated list of developer names of media group records. Possible values:

- `Attachment`
- `productDetailImage`
- `productListImage`

If this list is empty or unspecified, all media groups are returned. If `excludeMedia` and `mediaGroups` are specified, the `excludeMedia` parameter takes precedence.

For product bundles, only the `producListImage` is returned.

excludeFields
Type: Boolean

Specifies whether the fields are returned (`false`) or not (`true`). If unspecified, defaults to `false`.

excludeMedia
Type: Boolean

Specifies whether the media groups and default images of the product are returned (`false`) or not (`true`). If unspecified, defaults to `false`.

excludeAttributeSetInfo
Type: Boolean

Specifies whether the attribute set information for the product is returned (`false`) or not (`true`). If unspecified, defaults to `false`.

excludeQuantityRule
Type: Boolean

Specifies whether the quantity rule information for the product is returned (`false`) or not (`true`). If unspecified, defaults to `false`.

pageToken
Type: String

Specifies the base64 encoded page token. Page tokens are returned as part of the response. If unspecified, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you don't specify a value, the default size is 20.

Return Value

Type: `ConnectApi.ProductChildCollection`

getProducts(webstoreId, effectiveAccountId, ids, skus, fields, excludeMedia, excludePrices)

Get fields, prices, and default images for a list of products.

API Version

54.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.ProductOverviewCollection getProducts(String webstoreId, String effectiveAccountId, List<String> ids, List<String> skus, List<String> fields, Boolean excludeMedia, Boolean excludePrices)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If unspecified, the default value is determined from context.

ids
Type: List<String>

List of product IDs. Specify either a list of up to 20 product IDs or SKUs, but not both.

skus
Type: List<String>

List of SKUs. Specify either a list of up to 20 SKUs or product IDs, but not both.

fields
Type: List<String>

Comma-separated list of field names to return for each product. If the list is empty or not specified, all fields are returned. You can specify any number of fields.

excludeMedia
Type: Boolean

Specifies whether default images are returned for the products (`false`) or not (`true`). The default is `false`.

excludePrices
Type: Boolean

Specifies whether prices are returned for the products (`false`) or not (`true`). The default is `false`.

Note: In version 58.0 and later, prices aren't returned for products regardless of this parameter. To get pricing information for products in version 58.0 and later, use the CommerceStorePricing Class.

Return Value

Type: `ConnectApi.ProductOverviewCollection`

getProducts(webstoreId, effectiveAccountId, ids, skus, fields, excludeMedia)

Get fields and default images for a list of products.

API Version

58.0

Available to Guest Users

58.0

Requires Chatter

No

Signature

```
public static ConnectApi.ProductOverviewCollection getProducts(String webstoreId, String effectiveAccountId, List<String> ids, List<String> skus, List<String> fields, Boolean excludeMedia)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

ids
Type: List<String>

List of product IDs. Specify either a list of up to 20 product IDs or SKUs, but not both.

skus
Type: List<String>

List of SKUs. Specify either a list of up to 20 SKUs or product IDs, but not both.

fields
Type: List<String>

Comma-separated list of field names to return for each product. If the list is empty or not specified, all fields are returned. You can specify any number of fields.

excludeMedia
Type: Boolean

Specifies whether default images are returnedSpecifies whether the media groups and default images of the product are returned (`false`) or not (`true`). If unspecified, defaults to `false`.

Return Value

Type: `ConnectApi.ProductOverviewCollection`

CommerceCatalogManagement Class

Create or update a composite product. Create a variation product.

Namespace

ConnectApi

CommerceCatalogManagement Methods

These methods are for `CommerceCatalogManagement`. All methods are static.

compositeCommerceProductCreate(webstoreId, compositeCommerceProductInputRepresentation)

Create a composite product.

API Version

61.0

Requires Chatter

No

Signature

```
public static ConnectApi.CompositeCommerceProductOutputRepresentation
compositeCommerceProductCreate(String webstoreId,
ConnectApi.CompositeCommerceProductInputRepresentation
compositeCommerceProductInputRepresentation)
```

Parameters

webstoreId
Type: String

ID of the webstore.

compositeCommerceProductInputRepresentation
Type: ConnectApi.CompositeCommerceProductInputRepresentation

Details used to create the composite product.

Return Value

Type: ConnectApi.CompositeCommerceProductOutputRepresentation

compositeCommerceProductUpdate(webstoreId, productId, compositeCommerceProductInputRepresentation)

Update a composite product.

API Version

61.0

Requires Chatter

No

Signature

```
public static ConnectApi.CompositeCommerceProductOutputRepresentation
compositeCommerceProductUpdate(String webstoreId, String productId,
```

```
ConnectApi.CompositeCommerceProductInputRepresentation
compositeCommerceProductInputRepresentation)
```

Parameters

webstoreId
Type: String

ID of the webstore.

productId
Type: String

ID of the composite product.

compositeCommerceProductInputRepresentation
Type: ConnectApi.CompositeCommerceProductInputRepresentation

Details used to update the composite product.

Return Value

Type: ConnectApi.CompositeCommerceProductOutputRepresentation

compositeCommerceVariationCreate(webstoreId, compositeCommerceVariationInputRepresentation)

Create a variation product.

API Version

62.0

Requires Chatter

No

Signature

```
public static ConnectApi.CompositeCommerceVariationOutputRepresentation
compositeCommerceVariationCreate(String webstoreId,
ConnectApi.CompositeCommerceVariationInputRepresentation
compositeCommerceVariationInputRepresentation)
```

Parameters

webstoreId
Type: String

ID of the webstore.

compositeCommerceVariationInputRepresentation
Type: ConnectApi.CompositeCommerceVariationInputRepresentation

Details used to create the variation product.

Return Value

Type: `ConnectApi.CompositeCommerceVariationOutputRepresentation`

CommercePromotions Class

Evaluate promotions for Commerce orders. Get coupon code redemption usage.

Namespace

ConnectApi

CommercePromotions Methods

These methods are for `CommercePromotions`. All methods are static.

Note: Don't write an Apex test that calls these `CommercePromotions` APIs within the context of a guest.

IN THIS SECTION:

decreaseRedemption(couponCodeRedemption)
Get coupon code redemption usage to revert a previously redeemed coupon code.

evaluate(salesTransaction)
Determine which promotions the customer is eligible for based on the store and buyer group, and compute the applicable price adjustments based on the coupons and the items in the cart. This API evaluates only the first 50 active manual promotions and first 50 active automatic promotions, based on priority. This API computes and returns applicable price adjustments, but it does not apply those adjustments to the webcart record. If you want to enable promotions based on shipping, contact Salesforce Customer Support.

increaseRedemption(couponCodeRedemption)
Get coupon code redemption addition (increase) usage.

`decreaseRedemption(couponCodeRedemption)`

Get coupon code redemption usage to revert a previously redeemed coupon code.

API Version

58.0

Available to Guest Users

58.0

Requires Chatter

No

Signature

```
public static ConnectApi.CouponCodeRedemptionCollection
decreaseRedemption(ConnectApi.CouponCodeRedemptionInput couponCodeRedemption)
```

Parameters

couponCodeRedemption
Type: ConnectApi.CouponCodeRedemptionInput on page 1844

Tracks each coupon code redemption.

Return Value

Type: ConnectApi.CouponCodeRedemptionCollection on page 2061

evaluate(salesTransaction)

Determine which promotions the customer is eligible for based on the store and buyer group, and compute the applicable price adjustments based on the coupons and the items in the cart. This API evaluates only the first 50 active manual promotions and first 50 active automatic promotions, based on priority. This API computes and returns applicable price adjustments, but it does not apply those adjustments to the webcart record. If you want to enable promotions based on shipping, contact Salesforce Customer Support.

API Version

57.0

Available to Guest Users

57.0

Requires Chatter

No

Signature

```
global static ConnectApi.PromotionEvaluation evaluate(ConnectApi.PromotionEvaluateInput
salesTransaction)
```

Parameters

salesTransaction
Type: ConnectApi.PromotionEvaluateInput

Find promotions that the customer is eligible for and compute their discounts.

Return Value

Type: ConnectApi.PromotionEvaluation

`increaseRedemption(couponCodeRedemption)`

Get coupon code redemption addition (increase) usage.

API Version

58.0

Available to Guest Users

58.0

Requires Chatter

No

Signature

```
public static ConnectApi.CouponCodeRedemptionCollection
increaseRedemption(ConnectApi.CouponCodeRedemptionInput couponCodeRedemption)
```

Parameters

couponCodeRedemption
Type: `ConnectApi.CouponCodeRedemptionInput` on page 1844

Tracks each coupon code redemption.

Return Value

Type: `ConnectApi.CouponCodeRedemptionCollection` on page 2061

CommerceSearch Class

Get sort rules for the live search index. Get product search suggestions. Search products.

Namespace

ConnectApi

CommerceSearch Methods

These methods are for `CommerceSearch`. All methods are static.

IN THIS SECTION:

getSortRules(webstoreId)
Get sort rules for the live index.

getSuggestions(webstoreId, effectiveAccountId, searchTerm, maxResults)
Get suggestions for product searches.

getSuggestions(webstoreId, effectiveAccountId, searchTerm, maxResults, includeSuggestedProducts, maxSuggestedProducts)
Get suggestions for product searches.

searchProducts(webstoreId, effectiveAccountId, productSearchInput)
Search products.

`getSortRules(webstoreId)`

Get sort rules for the live index.

API Version

52.0

Available to Guest Users

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.SortRulesCollection getSortRules(String webstoreId)
```

Parameters

`webstoreId`
Type: String

ID of the webstore.

Return Value

Type: `ConnectApi.SortRulesCollection`

`getSuggestions(webstoreId, effectiveAccountId, searchTerm, maxResults)`

Get suggestions for product searches.

API Version

52.0

Available to Guest Users

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.ProductSearchSuggestionsResults getSuggestions(String
webstoreId, String effectiveAccountId, String searchTerm, Integer maxResults)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

searchTerm
Type: String

Search term with up to 100 characters. If specified, the service returns autocomplete suggestions from the user's recent searches. If unspecified, the service returns suggestions from the user's recent searches.

maxResults
Type: Integer

Maximum number of suggestions. Values are from 1 through 10. If unspecified, defaults to 10.

Return Value

Type: `ConnectApi.ProductSearchSuggestionsResults`

`getSuggestions(webstoreId, effectiveAccountId, searchTerm, maxResults, includeSuggestedProducts, maxSuggestedProducts)`

Get suggestions for product searches.

API Version

58.0

Available to Guest Users

58.0

Requires Chatter

No

Signature

```
public static ConnectApi.ProductSearchSuggestionsResults getSuggestions(String
webstoreId, String effectiveAccountId, String searchTerm, Integer maxResults, Boolean
includeSuggestedProducts, Integer maxSuggestedProducts)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

searchTerm
Type: String

Search term with up to 100 characters. If specified, the service returns autocomplete suggestions from the user's recent searches. If unspecified, the service returns suggestions from the user's recent searches.

maxResults
Type: Integer

Maximum number of suggestions. Values are from 1 through 10. If unspecified, defaults to 10.

includeSuggestedProducts
Type: Boolean

Specifies whether a search term returns product suggestions (`true`) or not (`false`). If unspecified, defaults to `false`. If `true`, the service returns the suggested product IDs.

maxSuggestedProducts
Type: String

Maximum number of product suggestions. Values are from 1 through 10. If unspecified, defaults to 4.

Return Value

Type: `ConnectApi.ProductSearchSuggestionsResults`

searchProducts(webstoreId, effectiveAccountId, productSearchInput)

Search products.

API Version

52.0

Available to Guest Users

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.ProductSearchResults searchProducts(String webstoreId, String
effectiveAccountId, ConnectApi.ProductSearchInput productSearchInput)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

productSearchInput
Type: `ConnectApi.ProductSearchInput`

A `ConnectApi.ProductSearchInput` body with either a category ID or search terms.

Return Value

Type: `ConnectApi.ProductSearchResults`

Usage

Searching products respects buyer View Product entitlements and only users entitled to view product data can access this resource.

CommerceSearchConnectFamily Class

Search products by search term or category in a webstore.

Namespace

ConnectApi

CommerceSearchConnectFamily Methods

These methods are for `CommerceSearchConnectFamily`. All methods are static.

IN THIS SECTION:

searchProducts(webstoreId, searchTerm, categoryId, sortRuleId, grouping, fields, refinements, pageParam, pageSize, effectiveAccountId, includeQuantityRule)
Search products by search term or category in a webstore.

`searchProducts(webstoreId, searchTerm, categoryId, sortRuleId, grouping, fields, refinements, pageParam, pageSize, effectiveAccountId, includeQuantityRule)`

Search products by search term or category in a webstore.

API Version

58.0

Available to Guest Users

58.0

Requires Chatter

No

Signature

```
public static ConnectApi.CommerceProductSearchResults searchProducts(String webstoreId,
String searchTerm, String categoryId, String sortRuleId, String grouping, List<String>
fields, String refinements, Integer pageParam, Integer pageSize, String
effectiveAccountId, Boolean includeQuantityRule)
```

Parameters

webstoreId
Type: String

ID of the webstore.

searchTerm
Type: String

List of up to 32 space-separated search terms.

categoryId
Type: String

Category ID returns results for products in this category or its subcategories.

sortRuleId
Type: String

ID of the sort rule that specifies the order of products in the search results. If unspecified, the default sort type is relevancy.

grouping
Type: String

Grouping option for search results. If unspecified, the default is the value specified in **Search** > **Results Display Settings** > **Results Grouping**.

fields
Type: List<String>

Product fields to return in search results. Search results include fields you have access to.

refinements
Type: String

List up to nine refinements (facets) for search results. Buyers or shoppers can select up to 20 values for each refinement. The `refinements` parameter is encoded as a Base64 string from the JSON representation of ConnectApi.DistinctValueRefinementInput .

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 200. If unspecified, defaults to 20.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If unspecified, the default value is determined from context.

includeQuantityRule
Type: Boolean

Specifies whether to include purchase quantity rule information for products in search results (`true`) or not (`false`). If unspecified, defaults to `false`.

Return Value

Type: `ConnectApi.CommerceProductSearchResults`

Usage

Searching products respects buyer View Product entitlements and only users entitled to view product data can access this resource.

CommerceSearchSettings Class

Get indexes. Get index logs. Create an index of a product catalog.

Namespace

ConnectApi

CommerceSearchSettings Methods

These methods are for `CommerceSearchSettings`. All methods are static.

IN THIS SECTION:

createCommerceSearchIndex(webstoreId, indexBuildType)
Create an index of a product catalog.

getCommerceSearchIndex(webstoreId, indexId)
Get a Commerce search index.

getCommerceSearchIndexes(webstoreId)
Get Commerce search indexes.

getCommerceSearchIndexLogs(webstoreId)
Get Commerce search index logs.

`createCommerceSearchIndex(webstoreId, indexBuildType)`

Create an index of a product catalog.

API Version

57.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.CommerceSearchIndex createCommerceSearchIndex(String webstoreId,
String indexBuildType)
```

Parameters

webstoreId
Type: String

ID of the webstore.

indexBuildType
Type: ConnectApi.CommerceSearchIndexBuildType

Build type of the index. Values are:

- `Full`
- `Incremental`

Return Value

Type: ConnectApi.CommerceSearchIndex

Usage

This method creates a live index that replaces the current live index. Any indexes that are in progress are removed when you manually create an index with this method.

`getCommerceSearchIndex(webstoreId, indexId)`

Get a Commerce search index.

API Version

52.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.CommerceSearchIndex getSingleCommerceSearchIndex(String
webstoreId, String indexId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

indexId
Type: String

ID of the index.

Return Value

Type: ConnectApi.CommerceSearchIndex

getCommerceSearchIndexes(webstoreId)

Get Commerce search indexes.

API Version

52.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.CommerceSearchIndexCollection getCommerceSearchIndexes(String webstoreId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

Return Value

Type: ConnectApi.CommerceSearchIndexCollection

getCommerceSearchIndexLogs(webstoreId)

Get Commerce search index logs.

API Version

57.0

Requires Chatter

No

Signature

```
public static ConnectApi.CommerceSearchIndexLogCollection
getCommerceSearchIndexLogs(String webstoreId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

Return Value

Type: `ConnectApi.CommerceSearchIndexLogCollection`

CommerceStorePricing Class

Get product prices.

Namespace

ConnectApi

CommerceStorePricing Methods

These methods are for `CommerceStorePricing`. All methods are static.

IN THIS SECTION:

getProductPrice(webstoreId, productId, effectiveAccountId)
Get the list and buyer price for a product.

getProductPrice(webstoreId, productId, effectiveAccountId, productSellingModelIds)
Get a product's list and buyer price for specified product selling models.

getProductPrices(webstoreId, effectiveAccountId, pricingInput)
Get the prices for multiple products.

`getProductPrice(webstoreId, productId, effectiveAccountId)`

Get the list and buyer price for a product.

API Version

49.0

Available to Guest Users

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.ProductPrice getProductPrice(String webstoreId, String
productId, String effectiveAccountId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

productId
Type: String

ID of the product.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made.

Return Value

Type: `ConnectApi.ProductPrice`

Usage

This method respects buyer entitlements and only users entitled to view product and price data can access it.

If a store is segmented into markets, this API looks at the language parameter appended to the URL to determine the shopper's locale and returns the appropriate values.

`getProductPrice(webstoreId, productId, effectiveAccountId, productSellingModelIds)`

Get a product's list and buyer price for specified product selling models.

API Version

56.0

Available to Guest Users

56.0

Requires Chatter

No

Signature

```
public static ConnectApi.ProductPrice getProductPrice(String webstoreId, String
productId, String effectiveAccountId, List<String> productSellingModelIds)
```

Parameters

webstoreId
Type: String

ID of the webstore.

productId
Type: String

ID of the product.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made.

productSellingModelIds
Type: List<String>

List of product selling model IDs for the product.

Return Value

Type: ConnectApi.ProductPrice

getProductPrices(webstoreId, effectiveAccountId, pricingInput)

Get the prices for multiple products.

API Version

49.0

Available to Guest Users

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.PricingResult getProductPrices(String webstoreId, String
effectiveAccountId, ConnectApi.PricingInput pricingInput)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the buyer account or guest buyer profile for which the request is made. If `null`, the default value is determined from context.

pricingInput
Type: `ConnectApi.PricingInput`

A `ConnectApi.PricingInput` body with the list of line items to price.

Return Value

Type: `ConnectApi.PricingResult`

Usage

This method respects buyer entitlements and only users entitled to view product and price data can access it.

If a store is segmented into markets, this API looks at the language parameter appended to the URL to determine the shopper's locale and returns the appropriate values.

CommerceWishlist Class

Get, create, update, and delete wishlists. Add wishlists to carts. Get wishlist items, add items to wishlists, and delete wishlist items.

Namespace

ConnectApi

CommerceWishlist Methods

These methods are for `CommerceWishlist`. All methods are static.

IN THIS SECTION:

addItemToWishlist(webstoreId, wishlistId, wishlistItemInput)
Add an item to a wishlist for the context user.

addItemToWishlist(webstoreId, effectiveAccountId, wishlistId, wishlistItemInput)
Add an item to a wishlist.

addWishlistToCart(webstoreId, wishlistId)
Add a wishlist to the active cart for the context user.

addWishlistToCart(webstoreId, wishlistId, effectiveAccountId)
Add a wishlist to the active cart.

addWishlistToCartWithCartId(webstoreId, wishlistId, cartId)
Add a wishlist to a cart.

addWishlistToCartWithCartId(webstoreId, wishlistId, cartId, effectiveAccountId)
Add a wishlist to a cart.

createWishlist(webstoreId, wishlistInput)
Create a wishlist for the context user.

createWishlist(webstoreId, effectiveAccountId, wishlistInput)
Create a wishlist.

deleteWishlist(webstoreId, wishlistId)
Delete a wishlist for the context user.

deleteWishlist(webstoreId, effectiveAccountId, wishlistId)
Delete a wishlist.

getWishlist(webstoreId, effectiveAccountId, wishlistId, productFields, sortItemsBy)
Get a wishlist with product fields sorted by items.

getWishlist(webstoreId, effectiveAccountId, wishlistId, productFields, pageSize, sortItemsBy)
Get a wishlist with product fields sorted by items with a specified number of items per page.

getWishlistItems(webstoreId, effectiveAccountId, wishlistId, productFields, pageParam, sortItemsBy)
Get a page of sorted wishlist items with product fields.

getWishlistItems(webstoreId, effectiveAccountId, wishlistId, productFields, pageParam, pageSize, sortItemsBy)
Get a page of specified size of sorted wishlist items with product fields.

getWishlistSummaries(webstoreId, effectiveAccountId, includeDisplayedList)
Get wishlist summaries.

getWishlistSummaries(webstoreId, effectiveAccountId, includeDisplayedList, productFields, sortItemsBy)
Get wishlist summaries with product fields sorted by items.

getWishlistSummaries(webstoreId, effectiveAccountId, includeDisplayedList, productFields, pageSize, sortItemsBy)
Get wishlist summaries with product fields sorted by items with a specified number of items per page.

removeWishlistItem(webstoreId, effectiveAccountId, wishlistId, wishlistItemId)
Remove an item from a wishlist.

updateWishlist(webstoreId, wishlistId, wishlistUpdateInput)
Update the name of a wishlist for the context user.

updateWishlist(webstoreId, effectiveAccountId, wishlistId, wishlistUpdateInput)
Update the name of a wishlist.

`addItemToWishlist(webstoreId, wishlistId, wishlistItemInput)`

Add an item to a wishlist for the context user.

API Version

49.0

Requires Chatter

No

Signature

```
public static ConnectApi.WishlistItem addItemToWishlist(String webstoreId, String wishlistId, ConnectApi.WishlistItemInput wishlistItemInput)
```

Parameters

webstoreId
Type: String

ID of the webstore.

wishlistId
Type: String

ID of the wishlist.

wishlistItemInput
Type: `ConnectApi.WishlistItemInput`

A `ConnectApi.WishlistItemInput` body with the item to add to the wishlist.

Return Value

Type: `ConnectApi.WishlistItem`

Usage

This method respects buyer View Product entitlements and only users entitled to view product data can access it.

addItemToWishlist(webstoreId, effectiveAccountId, wishlistId, wishlistItemInput)

Add an item to a wishlist.

API Version

49.0

Requires Chatter

No

Signature

```
public static ConnectApi.WishlistItem addItemToWishlist(String webstoreId, String effectiveAccountId, String wishlistId, ConnectApi.WishlistItemInput wishlistItemInput)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

wishlistId
Type: String

ID of the wishlist.

wishlistItemInput
Type: `ConnectApi.WishlistItemInput`

A `ConnectApi.WishlistItemInput` body with the item to add to the wishlist.

Return Value

Type: `ConnectApi.WishlistItem`

Usage

This method respects buyer View Product entitlements and only users entitled to view product data can access it.

addWishlistToCart(webstoreId, wishlistId)

Add a wishlist to the active cart for the context user.

API Version

49.0

Requires Chatter

No

Signature

```
public static ConnectApi.WishlistToCartResult addWishlistToCart(String webstoreId,
String wishlistId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

wishlistId
Type: String

ID of the wishlist.

Return Value

Type: `ConnectApi.WishlistToCartResult`

Usage

This method respects buyer View Product entitlements and only users entitled to view product data can access it.

addWishlistToCart(webstoreId, wishlistId, effectiveAccountId)

Add a wishlist to the active cart.

API Version

49.0

Requires Chatter

No

Signature

```
public static ConnectApi.WishlistToCartResult addWishlistToCart(String webstoreId,
String wishlistId, String effectiveAccountId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

wishlistId
Type: String

ID of the wishlist.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

Return Value

Type: `ConnectApi.WishlistToCartResult`

Usage

This method respects buyer View Product entitlements and only users entitled to view product data can access it.

addWishlistToCartWithCartId(webstoreId, wishlistId, cartId)

Add a wishlist to a cart.

API Version

49.0

Requires Chatter

No

Signature

```
public static ConnectApi.WishlistToCartResult addWishlistToCartWithCartId(String
webstoreId, String wishlistId, String cartId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

wishlistId
Type: String

ID of the wishlist.

cartId
Type: String

ID of the cart. If `null`, wishlist items are added to the active cart.

Return Value

Type: `ConnectApi.WishlistToCartResult`

Usage

This method respects buyer View Product entitlements and only users entitled to view product data can access it.

addWishlistToCartWithCartId(webstoreId, wishlistId, cartId, effectiveAccountId)

Add a wishlist to a cart.

API Version

49.0

Requires Chatter

No

Signature

```
public static ConnectApi.WishlistToCartResult addWishlistToCartWithCartId(String
webstoreId, String wishlistId, String cartId, String effectiveAccountId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

wishlistId
Type: String

ID of the wishlist.

cartId
Type: String

ID of the cart. If `null`, wishlist items are added to the active cart.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

Return Value

Type: `ConnectApi.WishlistToCartResult`

Usage

This method respects buyer View Product entitlements and only users entitled to view product data can access it.

createWishlist(webstoreId, wishlistInput)

Create a wishlist for the context user.

API Version

49.0

Requires Chatter

No

Signature

```
public static ConnectApi.Wishlist createWishlist(String webstoreId,
ConnectApi.WishlistInput wishlistInput)
```

Parameters

webstoreId
Type: String

ID of the webstore.

wishlistInput
Type: `ConnectApi.WishlistInput`

A `ConnectApi.WishlistInput` body that includes the wishlist name and items.

Return Value

Type: `ConnectApi.Wishlist`

createWishlist(webstoreId, effectiveAccountId, wishlistInput)

Create a wishlist.

API Version

49.0

Requires Chatter

No

Signature

```
public static ConnectApi.Wishlist createWishlist(String webstoreId, String
effectiveAccountId, ConnectApi.WishlistInput wishlistInput)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

wishlistInput
Type: `ConnectApi.WishlistInput`

A `ConnectApi.WishlistInput` body that includes the wishlist name and items.

Return Value

Type: `ConnectApi.Wishlist`

deleteWishlist(webstoreId, wishlistId)

Delete a wishlist for the context user.

API Version

49.0

Requires Chatter

No

Signature

```
public static Void deleteWishlist(String webstoreId, String wishlistId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

wishlistId
Type: String

ID of the wishlist.

Return Value

Type: Void

deleteWishlist(webstoreId, effectiveAccountId, wishlistId)

Delete a wishlist.

API Version

51.0

Requires Chatter

No

Signature

```
public static Void deleteWishlist(String webstoreId, String effectiveAccountId, String
wishlistId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

wishlistId
Type: String

ID of the wishlist.

Return Value

Type: Void

getWishlist(webstoreId, effectiveAccountId, wishlistId, productFields, sortItemsBy)

Get a wishlist with product fields sorted by items.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.Wishlist getWishlist(String webstoreId, String
effectiveAccountId, String wishlistId, String productFields,
ConnectApi.WishlistItemSortOrder sortItemsBy)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

wishlistId
Type: String

ID of the wishlist.

productFields
Type: String

Comma-separated list of custom product fields. Fields aren't case-sensitive. For example, `ProductCode` and `productcode` return the same results.

sortItemsBy
Type: `ConnectApi.WishlistItemSortOrder`

Sort order for wishlist items. Values are:

- `CreatedDateAsc`—Sorts by oldest creation date.
- `CreatedDateDesc`—Sorts by most recent creation date.

The default sort order is `CreatedDateDesc`.

Return Value

Type: ConnectApi.Wishlist

getWishlist(webstoreId, effectiveAccountId, wishlistId, productFields, pageSize, sortItemsBy)

Get a wishlist with product fields sorted by items with a specified number of items per page.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.Wishlist getWishlist(String webstoreId, String
effectiveAccountId, String wishlistId, String productFields, Integer pageSize,
ConnectApi.WishlistItemSortOrder sortItemsBy)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

wishlistId
Type: String

ID of the wishlist.

productFields
Type: String

Comma-separated list of custom product fields. Fields aren't case-sensitive. For example, `ProductCode` and `productcode` return the same results.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortItemsBy
Type: ConnectApi.WishlistItemSortOrder

Sort order for wishlist items. Values are:

- `CreatedDateAsc`—Sorts by oldest creation date.
- `CreatedDateDesc`—Sorts by most recent creation date.

The default sort order is `CreatedDateDesc`.

Return Value

Type: `ConnectApi.Wishlist`

`getWishlistItems(webstoreId, effectiveAccountId, wishlistId, productFields, pageParam, sortItemsBy)`

Get a page of sorted wishlist items with product fields.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.WishlistItemCollection getWishlistItems(String webstoreId, String effectiveAccountId, String wishlistId, String productFields, String pageParam, ConnectApi.WishlistItemSortOrder sortItemsBy)
```

Parameters

`webstoreId`
Type: String

ID of the webstore.

`effectiveAccountId`
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

`wishlistId`
Type: String

ID of the wishlist.

`productFields`
Type: String

Comma-separated list of custom product fields. Fields aren't case-sensitive. For example, `ProductCode` and `productcode` return the same results.

`pageParam`
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

`sortItemsBy`
Type: `ConnectApi.WishlistItemSortOrder`

Sort order for wishlist items. Values are:

- `CreatedDateAsc`—Sorts by oldest creation date.
- `CreatedDateDesc`—Sorts by most recent creation date.

The default sort order is `CreatedDateDesc`.

Return Value

Type: `ConnectApi.WishlistItemCollection`

getWishlistItems(webstoreId, effectiveAccountId, wishlistId, productFields, pageParam, pageSize, sortItemsBy)

Get a page of specified size of sorted wishlist items with product fields.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.WishlistItemCollection getWishlistItems(String webstoreId,
String effectiveAccountId, String wishlistId, String productFields, String pageParam,
Integer pageSize, ConnectApi.WishlistItemSortOrder sortItemsBy)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

wishlistId
Type: String

ID of the wishlist.

productFields
Type: String

Comma-separated list of custom product fields. Fields aren't case-sensitive. For example, `ProductCode` and `productcode` return the same results.

pageParam
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 200. If you pass in `null`, the default size is 25.

sortItemsBy
Type: `ConnectApi.WishlistItemSortOrder`

Sort order for wishlist items. Values are:

- `CreatedDateAsc`—Sorts by oldest creation date.
- `CreatedDateDesc`—Sorts by most recent creation date.

The default sort order is `CreatedDateDesc`.

Return Value

Type: `ConnectApi.WishlistItemCollection`

getWishlistSummaries(webstoreId, effectiveAccountId, includeDisplayedList)

Get wishlist summaries.

API Version

49.0

Requires Chatter

No

Signature

```
public static ConnectApi.WishlistsSummary getWishlistSummaries(String webstoreId, String effectiveAccountId, Boolean includeDisplayedList)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

includeDisplayedList
Type: Boolean

Specifies whether to include the displayed list (`true`) or not (`false`). If `null`, defaults to `false`.

Return Value

Type: `ConnectApi.WishlistsSummary`

`getWishlistSummaries(webstoreId, effectiveAccountId, includeDisplayedList, productFields, sortItemsBy)`

Get wishlist summaries with product fields sorted by items.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.WishlistsSummary getWishlistSummaries(String webstoreId, String effectiveAccountId, Boolean includeDisplayedList, String productFields, ConnectApi.WishlistItemSortOrder sortItemsBy)
```

Parameters

`webstoreId`
Type: String

ID of the webstore.

`effectiveAccountId`
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

`includeDisplayedList`
Type: Boolean

Specifies whether to include the displayed list (`true`) or not (`false`).

`productFields`
Type: String

Comma-separated list of custom product fields. Fields aren't case-sensitive. For example, `ProductCode` and `productcode` return the same results.

If *`includeDisplayedList`* is `false`, *`productFields`* is ignored.

`sortItemsBy`
Type: `ConnectApi.WishlistItemSortOrder`

Sort order for wishlist items. Values are:

- `CreatedDateAsc`—Sorts by oldest creation date.
- `CreatedDateDesc`—Sorts by most recent creation date.

The default sort order is `CreatedDateDesc`.

If *includeDisplayedList* is `false`, *sortItemsBy* is ignored.

Return Value

Type: `ConnectApi.WishlistsSummary`

getWishlistSummaries(webstoreId, effectiveAccountId, includeDisplayedList, productFields, pageSize, sortItemsBy)

Get wishlist summaries with product fields sorted by items with a specified number of items per page.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.WishlistsSummary getWishlistSummaries(String webstoreId, String effectiveAccountId, Boolean includeDisplayedList, Integer pageSize, String productFields, pageSize, ConnectApi.WishlistItemSortOrder sortItemsBy)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

includeDisplayedList
Type: Boolean

Specifies whether to include the displayed list (`true`) or not (`false`).

productFields
Type: String

Comma-separated list of custom product fields. Fields aren't case-sensitive. For example, `ProductCode` and `productcode` return the same results.

If *includeDisplayedList* is `false`, *productFields* is ignored.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortItemsBy
Type: `ConnectApi.WishlistItemSortOrder`

Sort order for wishlist items. Values are:

- `CreatedDateAsc`—Sorts by oldest creation date.
- `CreatedDateDesc`—Sorts by most recent creation date.

The default sort order is `CreatedDateDesc`.

If *`includeDisplayedList`* is `false`, *`sortItemsBy`* is ignored.

Return Value

Type: `ConnectApi.WishlistsSummary`

removeWishlistItem(webstoreId, effectiveAccountId, wishlistId, wishlistItemId)

Remove an item from a wishlist.

API Version

49.0

Requires Chatter

No

Signature

```
public static Void removeWishlistItem(String webstoreId, String effectiveAccountId,
String wishlistId, String wishlistItemId)
```

Parameters

`webstoreId`
Type: String

ID of the webstore.

`effectiveAccountId`
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

`wishlistId`
Type: String

ID of the wishlist.

`wishlistItemId`
Type: String

ID of the wishlist item to remove.

Return Value

Type: Void

updateWishlist(webstoreId, wishlistId, wishlistUpdateInput)

Update the name of a wishlist for the context user.

API Version

50.0

Requires Chatter

No

Signature

```
public static ConnectApi.WishlistSummary updateWishlist(String webstoreId, String wishlistId, ConnectApi.WishlistUpdateInput wishlistUpdateInput)
```

Parameters

webstoreId
Type: String

ID of the webstore.

wishlistId
Type: String

ID of the wishlist.

wishlistUpdateInput
Type: ConnectApi.WishlistUpdateInput

A `ConnectApi.WishlistUpdateInput` body with the wishlist name to update.

Return Value

Type: ConnectApi.WishlistSummary

updateWishlist(webstoreId, effectiveAccountId, wishlistId, wishlistUpdateInput)

Update the name of a wishlist.

API Version

50.0

Requires Chatter

No

Signature

```
public static ConnectApi.WishlistSummary updateWishlist(String webstoreId, String effectiveAccountId, String wishlistId, ConnectApi.WishlistUpdateInput wishlistUpdateInput)
```

Parameters

webstoreId
Type: String

ID of the webstore.

effectiveAccountId
Type: String

ID of the account for which the request is made. If `null`, defaults to the account ID for the context user.

wishlistId
Type: String

ID of the wishlist.

wishlistUpdateInput
Type: `ConnectApi.WishlistUpdateInput`

A `ConnectApi.WishlistUpdateInput` body with the wishlist name to update.

Return Value

Type: `ConnectApi.WishlistSummary`

Communities Class

Get information about Experience Cloud sites in your org.

Namespace

ConnectApi

Communities Methods

These methods are for `Communities`. All methods are static.

IN THIS SECTION:

getCommunities()
Get a list of Experience Cloud sites that the context user has access to.

getCommunities(communityStatus)
Get a list of Experience Cloud sites with the specified status that the context user has access to.

getCommunity(communityId)
Get information about an Experience Cloud site.

getCommunities()

Get a list of Experience Cloud sites that the context user has access to.

API Version

28.0

Requires Chatter

No

Signature

```
public static ConnectApi.CommunityPage getCommunities()
```

Return Value

Type: ConnectApi.CommunityPage

getCommunities(communityStatus)

Get a list of Experience Cloud sites with the specified status that the context user has access to.

API Version

28.0

Requires Chatter

No

Signature

```
public static ConnectApi.CommunityPage getCommunities(ConnectApi.CommunityStatus
communityStatus)
```

Parameters

communityStatus
Type: ConnectApi.CommunityStatus

communityStatus—Status of the Experience Cloud site. Values are:

- Live
- Inactive
- UnderConstruction

Return Value

Type: ConnectApi.CommunityPage

`getCommunity(communityId)`

Get information about an Experience Cloud site.

API Version

28.0

Available to Guest Users

35.0

Requires Chatter

No

Signature

```
public static ConnectApi.Community getCommunity(String communityId)
```

Parameters

`communityId`
Type: String

ID of an Experience Cloud site. You can't specify `null` or `internal`.

Return Value

Type: `ConnectApi.Community`

CommunityModeration Class

Get information about flagged feed items and comments in an Experience Cloud site. Add and remove flags from comments and feed items.

Namespace

ConnectApi

CommunityModeration Methods

These methods are for `CommunityModeration`. All methods are static.

All methods in this class require Chatter and are subject to the per user, per namespace, per hour rate limit.

IN THIS SECTION:

addFlagToComment(communityId, commentId)
Add a moderation flag to a comment.

addFlagToComment(communityId, commentId, visibility)
Add a moderation flag of the specified visibility to a comment.

addFlagToComment(communityId, commentId, type)
Add a moderation flag of the specified type to a comment.

addFlagToComment(communityId, commentId, note)
Add a moderation flag with a note to a comment.

addFlagToComment(communityId, commentId, type, note)
Add a moderation flag of the specified type with a note to a comment.

addFlagToComment(communityId, commentId, type, visibility)
Add a moderation flag of the specified type and visibility to a comment.

addFlagToComment(communityId, commentId, visibility, note)
Add a moderation flag of the specified visibility with a note to a comment.

addFlagToComment(communityId, commentId, type, visibility, note)
Add a moderation flag of the specified type and visibility with a note to a comment.

addFlagToFeedElement(communityId, feedElementId)
Add a moderation flag to a feed element.

addFlagToFeedElement(communityId, feedElementId, visibility)
Add a moderation flag of the specified visibility to a feed element.

addFlagToFeedElement(communityId, feedElementId, type)
Add a moderation flag of the specified type to a feed element.

addFlagToFeedElement(communityId, feedElementId, note)
Add a moderation flag with a note to a feed element.

addFlagToFeedElement(communityId, feedElementId, type, note)
Add a moderation flag of the specified type with a note to a feed element.

addFlagToFeedElement(communityId, feedElementId, type, visibility)
Add a moderation flag of the specified type and visibility to a feed element.

addFlagToFeedElement(communityId, feedElementId, visibility, note)
Add a moderation flag of the specified visibility with a note to a feed element.

addFlagToFeedElement(communityId, feedElementId, type, visibility, note)
Add a moderation flag of the specified type and visibility with a note to a feed element.

getFlagsOnComment(communityId, commentId)
Get the moderation flags on a comment.

getFlagsOnComment(communityId, commentId, visibility)
Get the moderation flags with specified visibility on a comment.

getFlagsOnComment(communityId, commentId, pageSize, pageParam)
Get a page of moderation flags on a comment.

getFlagsOnComment(communityId, commentId, visibility, pageSize, pageParam)
Get a page of moderation flags with specified visibility on a comment.

getFlagsOnFeedElement(communityId, feedElementId)
Get the moderation flags on a feed element.

getFlagsOnFeedElement(communityId, feedElementId, visibility)
Get the moderation flags with specified visibility on a feed element.

getFlagsOnFeedElement(communityId, feedElementId, pageParam, pageSize)
Get a page of moderation flags on a feed element.

getFlagsOnFeedElement(communityId, feedElementId, visibility, pageSize, pageParam)
Get a page of moderation flags with specified visibility on a feed element.

removeFlagFromComment(communityId, commentId, userId)
Remove a moderation flag from a comment.

removeFlagFromFeedElement(communityId, feedElementId, userId)
Remove a moderation flag from a feed element.

`addFlagToComment(communityId, commentId)`

Add a moderation flag to a comment.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationFlags addFlagToComment(String communityId, String commentId)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`commentId`
Type: String

ID for a comment.

Return Value

Type: `ConnectApi.ModerationFlags`

Usage

To add a flag to a comment, Allow members to flag content must be selected for an Experience Cloud site.

addFlagToComment(communityId, commentId, visibility)

Add a moderation flag of the specified visibility to a comment.

API Version

30.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationFlags addFlagToComment(String communityId, String commentId, ConnectApi.CommunityFlagVisibility visibility)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

commentId
Type: String

ID for a comment.

visibility
Type: `ConnectApi.CommunityFlagVisibility`

Visibility behavior of a flag for various user types.

- `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item.
- `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.

Return Value

Type: `ConnectApi.ModerationFlags`

Usage

To add a flag to a comment, Allow members to flag content must be selected for an Experience Cloud site.

addFlagToComment(communityId, commentId, type)

Add a moderation flag of the specified type to a comment.

API Version

38.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationFlags addFlagToComment(String communityId, String commentId, ConnectApi.CommunityFlagType type)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

commentId
Type: String

ID for a comment.

type
Type: `ConnectApi.CommunityFlagType`

Type of moderation flag.

- `FlagAsInappropriate`—Flag for inappropriate content.
- `FlagAsSpam`—Flag for spam.

If a type isn't specified, it defaults to `FlagAsInappropriate`.

Return Value

Type: `ConnectApi.ModerationFlags`

Usage

To add a flag to a comment, Allow members to flag content must be selected for an Experience Cloud site.

`addFlagToComment(communityId, commentId, note)`

Add a moderation flag with a note to a comment.

API Version

38.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationFlags addFlagToComment(String communityId, String commentId, String note)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

commentId
Type: String

ID for a comment.

note
Type: String

A note of up to 4,000 characters about the flag.

Return Value

Type: `ConnectApi.ModerationFlags`

Usage

To add a flag to a comment, Allow members to flag content must be selected for an Experience Cloud site.

addFlagToComment(communityId, commentId, type, note)

Add a moderation flag of the specified type with a note to a comment.

API Version

38.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationFlags addFlagToComment(String communityId, String
commentId, ConnectApi.CommunityFlagType type, String note)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

commentId
Type: String

ID for a comment.

type
Type: `ConnectApi.CommunityFlagType`

Type of moderation flag.

- `FlagAsInappropriate`—Flag for inappropriate content.
- `FlagAsSpam`—Flag for spam.

If a type isn't specified, it defaults to `FlagAsInappropriate`.

note
Type: String

A note of up to 4,000 characters about the flag.

Return Value

Type: `ConnectApi.ModerationFlags`

Usage

To add a flag to a comment, Allow members to flag content must be selected for an Experience Cloud site.

addFlagToComment(communityId, commentId, type, visibility)

Add a moderation flag of the specified type and visibility to a comment.

API Version

38.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationFlags addFlagToComment(String communityId, String commentId, ConnectApi.CommunityFlagType type, ConnectApi.CommunityFlagVisibility visibility)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

commentId
Type: String

ID for a comment.

type
Type: `ConnectApi.CommunityFlagType`

Type of moderation flag.

- `FlagAsInappropriate`—Flag for inappropriate content.

- `FlagAsSpam`—Flag for spam.

If a type isn't specified, it defaults to `FlagAsInappropriate`.

`visibility`
Type: `ConnectApi.CommunityFlagVisibility`

Visibility behavior of a flag for various user types.

- `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item.
- `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.

Return Value

Type: `ConnectApi.ModerationFlags`

Usage

To add a flag to a comment, Allow members to flag content must be selected for an Experience Cloud site.

`addFlagToComment(communityId, commentId, visibility, note)`

Add a moderation flag of the specified visibility with a note to a comment.

API Version

38.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationFlags addFlagToComment(String communityId, String commentId, ConnectApi.CommunityFlagVisibility visibility, String note)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`commentId`
Type: String

ID for a comment.

`visibility`
Type: `ConnectApi.CommunityFlagVisibility`

Visibility behavior of a flag for various user types.

- `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item.

- `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.

note
Type: String

A note of up to 4,000 characters about the flag.

Return Value

Type: `ConnectApi.ModerationFlags`

Usage

To add a flag to a comment, Allow members to flag content must be selected for an Experience Cloud site.

addFlagToComment(communityId, commentId, type, visibility, note)

Add a moderation flag of the specified type and visibility with a note to a comment.

API Version

38.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationFlags addFlagToComment(String communityId, String commentId, ConnectApi.CommunityFlagType type, ConnectApi.CommunityFlagVisibility visibility, String note)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

commentId
Type: String

ID for a comment.

type
Type: `ConnectApi.CommunityFlagType`

Type of moderation flag.

- `FlagAsInappropriate`—Flag for inappropriate content.
- `FlagAsSpam`—Flag for spam.

If a type isn't specified, it defaults to `FlagAsInappropriate`.

visibility
Type: ConnectApi.CommunityFlagVisibility

Visibility behavior of a flag for various user types.

- `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item.
- `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.

note
Type: String

A note of up to 4,000 characters about the flag.

Return Value

Type: ConnectApi.ModerationFlags

Usage

To add a flag to a comment, Allow members to flag content must be selected for an Experience Cloud site.

addFlagToFeedElement(communityId, feedElementId)

Add a moderation flag to a feed element.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationCapability addFlagToFeedElement(String communityId,
String feedElementId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

Return Value

Type: ConnectApi.ModerationCapability

If the feed element doesn't support this capability, the return value is `ConnectApi.NotFoundException`.

Usage

To add a flag to a feed element, Allow members to flag content must be selected for an Experience Cloud site.

addFlagToFeedElement(communityId, feedElementId, visibility)

Add a moderation flag of the specified visibility to a feed element.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationCapability addFlagToFeedElement(String communityId,
String feedElementId, ConnectApi.CommunityFlagVisibility visibility)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

visibility
Type: `ConnectApi.CommunityFlagVisibility`

Visibility behavior of a flag for various user types. One of these values:

- `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item.
- `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.

Return Value

Type: `ConnectApi.ModerationCapability`

If the feed element doesn't support this capability, the return value is `ConnectApi.NotFoundException`.

Usage

To add a flag to a feed element, Allow members to flag content must be selected for an Experience Cloud site.

addFlagToFeedElement(communityId, feedElementId, type)

Add a moderation flag of the specified type to a feed element.

API Version

38.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationCapability addFlagToFeedElement(String communityId,
String feedElementId, ConnectApi.CommunityFlagType type)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

type
Type: `ConnectApi.CommunityFlagType`

Type of moderation flag.

- `FlagAsInappropriate`—Flag for inappropriate content.
- `FlagAsSpam`—Flag for spam.

If a type isn't specified, it defaults to `FlagAsInappropriate`.

Return Value

Type: `ConnectApi.ModerationCapability`

Usage

To add a flag to a feed element, Allow members to flag content must be selected for an Experience Cloud site.

addFlagToFeedElement(communityId, feedElementId, note)

Add a moderation flag with a note to a feed element.

API Version

38.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationCapability addFlagToFeedElement(String communityId,
String feedElementId, String note)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

note
Type: String

A note of up to 4,000 characters about the flag.

Return Value

Type: `ConnectApi.ModerationCapability`

Usage

To add a flag to a feed element, Allow members to flag content must be selected for an Experience Cloud site.

addFlagToFeedElement(communityId, feedElementId, type, note)

Add a moderation flag of the specified type with a note to a feed element.

API Version

38.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationCapability addFlagToFeedElement(String communityId,
String feedElementId, ConnectApi.CommunityFlagType type, String note)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

type
Type: `ConnectApi.CommunityFlagType`

Type of moderation flag.

- `FlagAsInappropriate`—Flag for inappropriate content.
- `FlagAsSpam`—Flag for spam.

If a type isn't specified, it defaults to `FlagAsInappropriate`.

note
Type: String

A note of up to 4,000 characters about the flag.

Return Value

Type: `ConnectApi.ModerationCapability`

Usage

To add a flag to a feed element, Allow members to flag content must be selected for an Experience Cloud site.

`addFlagToFeedElement(communityId, feedElementId, type, visibility)`

Add a moderation flag of the specified type and visibility to a feed element.

API Version

38.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationCapability addFlagToFeedElement(String communityId,
String feedElementId, ConnectApi.CommunityFlagType type,
ConnectApi.CommunityFlagVisibility visibility)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

type
Type: `ConnectApi.CommunityFlagType`

Type of moderation flag.

- `FlagAsInappropriate`—Flag for inappropriate content.
- `FlagAsSpam`—Flag for spam.

If a type isn't specified, it defaults to `FlagAsInappropriate`.

visibility
Type: `ConnectApi.CommunityFlagVisibility`

Visibility behavior of a flag for various user types. One of these values:

- `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item.
- `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.

Return Value

Type: `ConnectApi.ModerationCapability`

Usage

To add a flag to a feed element, Allow members to flag content must be selected for an Experience Cloud site.

`addFlagToFeedElement(communityId, feedElementId, visibility, note)`

Add a moderation flag of the specified visibility with a note to a feed element.

API Version

38.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationCapability addFlagToFeedElement(String communityId,
String feedElementId, ConnectApi.CommunityFlagVisibility visibility, String note)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

visibility
Type: `ConnectApi.CommunityFlagVisibility`

Visibility behavior of a flag for various user types. One of these values:

- `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item.
- `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.

note
Type: String

A note of up to 4,000 characters about the flag.

Return Value

Type: `ConnectApi.ModerationCapability`

Usage

To add a flag to a feed element, Allow members to flag content must be selected for an Experience Cloud site.

`addFlagToFeedElement(communityId, feedElementId, type, visibility, note)`

Add a moderation flag of the specified type and visibility with a note to a feed element.

API Version

38.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationCapability addFlagToFeedElement(String communityId,
String feedElementId, ConnectApi.CommunityFlagType type,
ConnectApi.CommunityFlagVisibility visibility, String note)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedElementId`
Type: String

ID of the feed element.

`type`
Type: `ConnectApi.CommunityFlagType`

Type of moderation flag.

- `FlagAsInappropriate`—Flag for inappropriate content.
- `FlagAsSpam`—Flag for spam.

If a type isn't specified, it defaults to `FlagAsInappropriate`.

`visibility`
Type: `ConnectApi.CommunityFlagVisibility`

Visibility behavior of a flag for various user types. One of these values:

- `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item.
- `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.

`note`
Type: String

A note of up to 4,000 characters about the flag.

Return Value

Type: `ConnectApi.ModerationCapability`

Usage

To add a flag to a feed element, Allow members to flag content must be selected for an Experience Cloud site.

`getFlagsOnComment(communityId, commentId)`

Get the moderation flags on a comment.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationFlags getFlagsOnComment(String communityId, String commentId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

commentId
Type: String

ID for a comment.

Return Value

Type: `ConnectApi.ModerationFlags`

Usage

To get moderation flags, the context user must have the Moderate Experiences Feeds permission.

`getFlagsOnComment(communityId, commentId, visibility)`

Get the moderation flags with specified visibility on a comment.

API Version

30.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationFlags getFlagsOnComment(String communityId, String
commentId, ConnectApi.CommunityFlagVisibility visibility)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

commentId
Type: String

ID for a comment.

visibility
Type: `ConnectApi.CommunityFlagVisibility`

Visibility behavior of a flag for various user types.

- `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item.

- `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.

Return Value

Type: `ConnectApi.ModerationFlags`

Usage

To get moderation flags, the context user must have the Moderate Experiences Feeds permission.

`getFlagsOnComment(communityId, commentId, pageSize, pageParam)`

Get a page of moderation flags on a comment.

API Version

40.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationFlags getFlagsOnComment(String communityId, String commentId, Integer pageSize, String pageParam)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`commentId`
Type: String

ID for a comment.

`pageSize`
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. The default size is 0.

`pageParam`
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

Return Value

Type: `ConnectApi.ModerationFlags`

Usage

To get moderation flags, the context user must have the Moderate Experiences Feeds permission.

getFlagsOnComment(communityId, commentId, visibility, pageSize, pageParam)

Get a page of moderation flags with specified visibility on a comment.

API Version

40.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationFlags getFlagsOnComment(String communityId, String commentId, ConnectApi.CommunityFlagVisibility visibility, Integer pageSize, String pageParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

commentId
Type: String

ID for a comment.

visibility
Type: `ConnectApi.CommunityFlagVisibility`

Visibility behavior of a flag for various user types.

- `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item.
- `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. The default size is 0.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

Return Value

Type: `ConnectApi.ModerationFlags`

Usage

To get moderation flags, the context user must have the Moderate Experiences Feeds permission.

`getFlagsOnFeedElement(communityId, feedElementId)`

Get the moderation flags on a feed element.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationCapability getFlagsOnFeedElement(String communityId,
String feedElementId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

Return Value

Type: `ConnectApi.ModerationCapability`

If the feed element doesn't support this capability, the return value is `ConnectApi.NotFoundException`.

Usage

To get moderation flags, the context user must have the Moderate Experiences Feeds permission.

`getFlagsOnFeedElement(communityId, feedElementId, visibility)`

Get the moderation flags with specified visibility on a feed element.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationCapability getFlagsOnFeedElement(String communityId,
String feedElementId, ConnectApi.CommunityFlagVisibility visibility)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

visibility
Type: `ConnectApi.CommunityFlagVisibility`

Visibility behavior of a flag for various user types. One of these values:

- `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item.
- `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.

Return Value

Type: `ConnectApi.ModerationCapability`

If the feed element doesn't support this capability, the return value is `ConnectApi.NotFoundException`.

Usage

To get moderation flags, the context user must have the Moderate Experiences Feeds permission.

`getFlagsOnFeedElement(communityId, feedElementId, pageParam, pageSize)`

Get a page of moderation flags on a feed element.

API Version

40.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationCapability getFlagsOnFeedElement(String communityId,
String feedElementId, String pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

pageParam
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. The default size is 0.

Return Value

Type: `ConnectApi.ModerationCapability`

If the feed element doesn't support this capability, the return value is `ConnectApi.NotFoundException`.

Usage

To get moderation flags, the context user must have the Moderate Experiences Feeds permission.

`getFlagsOnFeedElement(communityId, feedElementId, visibility, pageSize, pageParam)`

Get a page of moderation flags with specified visibility on a feed element.

API Version

40.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationCapability getFlagsOnFeedElement(String communityId,
String feedElementId, ConnectApi.CommunityFlagVisibility visibility, Integer pageSize,
String pageParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedElementId`
Type: String

ID of the feed element.

`visibility`
Type: `ConnectApi.CommunityFlagVisibility`

Visibility behavior of a flag for various user types.

- `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item.
- `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.

`pageSize`
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. The default size is 0.

`pageParam`
Type: String

Page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

Return Value

Type: `ConnectApi.ModerationCapability`

If the feed element doesn't support this capability, the return value is `ConnectApi.NotFoundException`.

Usage

To get moderation flags, the context user must have the Moderate Experiences Feeds permission.

`removeFlagFromComment(communityId, commentId, userId)`

Remove a moderation flag from a comment.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationFlags removeFlagFromComment(String communityId,
String commentId, String userId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

commentId
Type: String

ID for a comment.

userId
Type: String

ID of the context user for whom the flag is removed. Specify `null` to remove the flag for all users.

Return Value

Type: Void

Usage

To remove a moderation flag, the context user must have added the flag or must have the Moderate Experiences Feeds permission.

removeFlagFromFeedElement(communityId, feedElementId, userId)

Remove a moderation flag from a feed element.

API Version

31.0

Requires Chatter

Yes

Signature

```
public static void removeFlagFromFeedElement(String communityId, String feedElementId,
String userId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

userId
Type: String

ID of the context user for whom the flag is removed. Specify `null` to remove the flag for all users.

Return Value

Type: `ConnectApi.ModerationCapability`

If the feed element doesn't support this capability, the return value is `ConnectApi.NotFoundException`.

Usage

To remove a moderation flag, the context user must have added the flag or must have the Moderate Experiences Feeds permission.

Retired CommunityModeration Methods

These methods for `CommunityModeration` are retired.

IN THIS SECTION:

addFlagToFeedItem(communityId, feedItemId)
Add a moderation flag to a feed item.

addFlagToFeedItem(communityId, feedItemId, visibility)
Add a moderation flag with specified visibility to a feed item.

getFlagsOnFeedItem(communityId, feedItemId)
Get the moderation flags on a feed item.

getFlagsOnFeedItem(communityId, feedItemId, visibility)
Get the moderation flags with specified visibility on a feed item.

removeFlagsOnFeedItem(communityId, feedItemId, userId)
Remove a moderation flag from a feed item.

`addFlagToFeedItem(communityId, feedItemId)`

Add a moderation flag to a feed item.

API Version

29.0–31.0

Important: In version 32.0 and later, use addFlagToFeedElement(communityId, feedElementId).

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationFlags addFlagToFeedItem(String communityId, String feedItemId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedItemId
Type: String

ID for a feed item.

Return Value

Type: `ConnectApi.ModerationFlags`

Usage

To add a flag to a feed item, Allow members to flag content must be selected for an Experience Cloud site.

`addFlagToFeedItem(communityId, feedItemId, visibility)`

Add a moderation flag with specified visibility to a feed item.

API Version

30.0–31.0

Important: In version 32.0 and later, use addFlagToFeedElement(communityId, feedElementId, visibility).

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationFlags addFlagToFeedItem(String communityId, String feedItemId, ConnectApi.CommunityFlagVisibility visibility)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedItemId
Type: String

ID for a feed item.

visibility
Type: `ConnectApi.CommunityFlagVisibility`

Visibility behavior of a flag for various user types.

- `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item.

- `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.

Return Value

Type: `ConnectApi.ModerationFlags`

Usage

To add a flag to a feed item, Allow members to flag content must be selected for an Experience Cloud site.

`getFlagsOnFeedItem(communityId, feedItemId)`

Get the moderation flags on a feed item.

API Version

29.0–31.0

Important: In version 32.0 and later, use getFlagsOnFeedElement(communityId, feedElementId).

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationFlags getFlagsOnFeedItem(String communityId, String feedItemId)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`feedItemId`
Type: String

ID for a feed item.

Return Value

Type: `ConnectApi.ModerationFlags`

Usage

To get moderation flags, the context user must have the Moderate Experiences Feeds permission.

`getFlagsOnFeedItem(communityId, feedItemId, visibility)`

Get the moderation flags with specified visibility on a feed item.

API Version

30.0–31.0

Important: In version 32.0 and later, use getFlagsOnFeedElement(communityId, feedElementId, visibility).

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationFlags getFlagsOnFeedItem(String communityId, String feedItemId, ConnectApi.CommunityFlagVisibility visibility)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedItemId
Type: String

ID for a feed item.

visibility
Type: `ConnectApi.CommunityFlagVisibility`

Visibility behavior of a flag for various user types. Values are:

- `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item.
- `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.

Return Value

Type: `ConnectApi.ModerationFlags`

Usage

To get moderation flags, the context user must have the Moderate Experiences Feeds permission.

`removeFlagsOnFeedItem(communityId, feedItemId, userId)`

Remove a moderation flag from a feed item.

API Version

29.0–31.0

Important: In version 32.0 and later, use removeFlagFromFeedElement(communityId, feedElementId, userId).

Requires Chatter

Yes

Signature

```
public static ConnectApi.ModerationFlags removeFlagsOnFeedItem(String communityId,
String feedItemId, String userId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedItemId
Type: String

ID for a feed item.

userId
Type: String

ID of the context user for whom the flag is removed. Specify `null` to remove the flag for all users.

Return Value

Type: Void

Usage

To remove a moderation flag, the context user must have added the flag or must have the Moderate Experiences Feeds permission.

ContentHub Class

Access Files Connect repositories and their files and folders.

Namespace

ConnectApi

ContentHub Methods

These methods are for `ContentHub`. All methods are static.

Use `ContentHub` methods to work with Files Connect repositories.

IN THIS SECTION:

addRepositoryItem(repositoryId, repositoryFolderId, file)
Add a repository item.

addRepositoryItem(communityId, repositoryId, repositoryFolderId, file)
Add a repository item in an Experience Cloud site.

addRepositoryItem(repositoryId, repositoryFolderId, file, fileData)
Add a repository item, including the binary file.

addRepositoryItem(communityId, repositoryId, repositoryFolderId, file, fileData)
Add a repository item, including the binary file, in an Experience Cloud site.

getAllowedItemTypes(repositoryId, repositoryFolderId)
Get the item types that the context user is allowed to create in the repository folder.

getAllowedItemTypes(repositoryId, repositoryFolderId, filter)
Get the item types, filtered by type, that the context user is allowed to create in the repository folder.

getAllowedItemTypes(communityId, repositoryId, repositoryFolderId)
Get the item types that the context user is allowed to create in the repository folder in an Experience Cloud site.

getAllowedItemTypes(communityId, repositoryId, repositoryFolderId, filter)
Get the item types, filtered by type, that the context user is allowed to create in the repository folder in an Experience Cloud site.

getFilePreview(repositoryId, repositoryFileId, formatType)
Get a repository file preview.

getFilePreview(repositoryId, repositoryFileId, formatType, startPageNumber, endPageNumber)
Get a page or page range of a repository file preview.

getFilePreview(communityId, repositoryId, repositoryFileId, formatType)
Get a repository file preview in an Experience Cloud site.

getFilePreview(communityId, repositoryId, repositoryFileId, formatType, startPageNumber, endPageNumber)
Get a page or page range of a repository file preview in an Experience Cloud site.

getItemType(repositoryId, repositoryItemTypeId)
Get information about an item type associated with a repository.

getItemType(communityId, repositoryId, repositoryItemTypeId)
Get information about an item type associated with a repository in an Experience Cloud site.

getPreviews(repositoryId, repositoryFileId)
Get information about a repository file's supported previews.

getPreviews(communityId, repositoryId, repositoryFileId)
Get information about a repository file's supported previews in an Experience Cloud site.

getRepositories()
Get a list of repositories.

getRepositories(communityId)
Get a list of repositories in an Experience Cloud site.

getRepositories(pageParam, pageSize)
Get a page of repositories.

getRepositories(communityId, pageParam, pageSize)
Get a page of repositories in an Experience Cloud site.

getRepository(repositoryId)
Get a repository.

getRepository(communityId, repositoryId)
Get a repository in an Experience Cloud site.

getRepositoryFile(repositoryId, repositoryFileId)
Get a repository file.

getRepositoryFile(repositoryId, repositoryFileId, includeExternalFilePermissionsInfo)
Get a repository file with or without permissions information.

getRepositoryFile(communityId, repositoryId, repositoryFileId)
Get a repository file in an Experience Cloud site.

getRepositoryFile(communityId, repositoryId, repositoryFileId, includeExternalFilePermissionsInfo)
Get a repository file with or without permissions information in an Experience Cloud site.

getRepositoryFolder(repositoryId, repositoryFolderId)
Get a repository folder.

getRepositoryFolder(communityId, repositoryId, repositoryFolderId)
Get a repository folder in an Experience Cloud site.

getRepositoryFolderItems(repositoryId, repositoryFolderId)
Get repository folder items.

getRepositoryFolderItems(communityId, repositoryId, repositoryFolderId)
Get repository folder items in an Experience Cloud site.

getRepositoryFolderItems(repositoryId, repositoryFolderId, pageParam, pageSize)
Get a page of repository folder items.

getRepositoryFolderItems(communityId, repositoryId, repositoryFolderId, pageParam, pageSize)
Get a page of repository folder items in an Experience Cloud site.

updateRepositoryFile(repositoryId, repositoryFileId, file)
Update the name of a repository file.

updateRepositoryFile(repositoryId, repositoryFileId, file, fileData)
Update the content of a repository file.

updateRepositoryFile(communityId, repositoryId, repositoryFileId, file)
Update the name of a repository file in an Experience Cloud site.

updateRepositoryFile(communityId, repositoryId, repositoryFileId, file, fileData)
Update the content of a repository file in an Experience Cloud site.

`addRepositoryItem(repositoryId, repositoryFolderId, file)`

Add a repository item.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.RepositoryFolderItem addRepositoryItem(String repositoryId,
String repositoryFolderId, ConnectApi.ContentHubItemInput file)
```

Parameters

repositoryId
Type: String

The ID of the repository.

repositoryFolderId
Type: String

The ID of the repository folder.

file
Type: ConnectApi.ContentHubItemInput

The item type ID and fields of the item type.

Return Value

Type: ConnectApi.RepositoryFolderItem

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

Example

This example creates a file without binary content (metadata only) in a Google Drive repository folder. After the file is created, we show the file's ID, name, description, external URL, and download URL.

```
final String gDriveRepositoryId = '0XCxx00000000ODGAY', gDriveFolderId =
'folder:0B0lTys1KmM3sSVJ2bjIzTGFqSWs';

final ConnectApi.ContentHubItemInput newItem = new ConnectApi.ContentHubItemInput();
newItem.itemTypeId = 'document'; //see getAllowedTypes for any file item types available
for creation/update
newItem.fields = new List<ConnectApi.ContentHubFieldValueInput>();

//Metadata: name field
final ConnectApi.ContentHubFieldValueInput fieldValueInput = new
ConnectApi.ContentHubFieldValueInput();
fieldValueInput.name = 'name';
fieldValueInput.value = 'new folder item name.txt';
newItem.fields.add(fieldValueInput);

//Metadata: description field
final ConnectApi.ContentHubFieldValueInput fieldValueInputDesc = new
ConnectApi.ContentHubFieldValueInput();
fieldValueInputDesc.name = 'description';
fieldValueInputDesc.value = 'It does describe it';
```

```
newItem.fields.add(fieldValueInputDesc);

final ConnectApi.RepositoryFolderItem newFolderItem =
ConnectApi.ContentHub.addRepositoryItem(gDriveRepositoryId, gDriveFolderId, newItem);
final ConnectApi.RepositoryFileSummary newFile = newFolderItem.file;
System.debug(String.format('New file - id: \'\'{0}\'\', name: \'\'{1}\'\', description:
\'\'{2}\'\' \n external URL: \'\'{3}\'\', download URL: \'\'{4}\'\'', new String[]{
newFile.id, newFile.name, newFile.description, newFile.externalDocumentUrl,
newFile.downloadUrl}));
```

SEE ALSO:

- setTestAddRepositoryItem(repositoryId, repositoryFolderId, file, result)
- *Apex Developer Guide*: Testing ConnectApi Code

addRepositoryItem(communityId, repositoryId, repositoryFolderId, file)

Add a repository item in an Experience Cloud site.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.RepositoryFolderItem addRepositoryItem(String communityId,
String repositoryId, String repositoryFolderId, ConnectApi.ContentHubItemInput file)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

repositoryId
Type: String

The ID of the repository.

repositoryFolderId
Type: String

The ID of the repository folder.

file
Type: `ConnectApi.ContentHubItemInput`

The item type ID and fields of the item type.

Return Value

Type: `ConnectApi.RepositoryFolderItem`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestAddRepositoryItem(communityId, repositoryId, repositoryFolderId, file, result)

Apex Developer Guide: Testing ConnectApi Code

addRepositoryItem(repositoryId, repositoryFolderId, file, fileData)

Add a repository item, including the binary file.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.RepositoryFolderItem addRepositoryItem(String repositoryId,
String repositoryFolderId, ConnectApi.ContentHubItemInput file, ConnectApi.BinaryInput
fileData)
```

Parameters

repositoryId
Type: String

The ID of the repository.

repositoryFolderId
Type: String

The ID of the repository folder.

file
Type: `ConnectApi.ContentHubItemInput`

The item type ID and fields of the item type.

fileData
Type: `ConnectApi.BinaryInput`

The binary file.

Return Value

Type: ConnectApi.RepositoryFolderItem

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

Example

This example creates a file with binary content and metadata in a Google Drive repository folder. After the file is created, we show the file's ID, name, description, external URL, and download URL.

```
final String gDriveRepositoryId = '0XCxx00000000ODGAY', gDriveFolderId =
'folder:0B0lTys1KmM3sSVJ2bjIzTGFqSWs';

final ConnectApi.ContentHubItemInput newItem = new ConnectApi.ContentHubItemInput();
newItem.itemTypeId = 'document'; //see getAllowedTypes for any file item types available
for creation/update
newItem.fields = new List<ConnectApi.ContentHubFieldValueInput>();

//Metadata: name field
Final String newFileName = 'new folder item name.txt';
final ConnectApi.ContentHubFieldValueInput fieldValueInput = new
ConnectApi.ContentHubFieldValueInput();
fieldValueInput.name = 'name';
fieldValueInput.value = newFileName;
newItem.fields.add(fieldValueInput);

//Metadata: description field
final ConnectApi.ContentHubFieldValueInput fieldValueInputDesc = new
ConnectApi.ContentHubFieldValueInput();
fieldValueInputDesc.name = 'description';
fieldValueInputDesc.value = 'It does describe it';
newItem.fields.add(fieldValueInputDesc);

//Binary content
final Blob newFileBlob = Blob.valueOf('awesome content for brand new file');
final String newFileMimeType = 'text/plain';
final ConnectApi.BinaryInput fileBinaryInput = new ConnectApi.BinaryInput(newFileBlob,
newFileMimeType, newFileName);

final ConnectApi.RepositoryFolderItem newFolderItem =
ConnectApi.ContentHub.addRepositoryItem(gDriveRepositoryId, gDriveFolderId, newItem,
fileBinaryInput);
final ConnectApi.RepositoryFileSummary newFile = newFolderItem.file;
System.debug(String.format('New file - id: \'\'{0}\'\', name: \'\'{1}\'\', description:
\'\'{2}\'\' \n external URL: \'\'{3}\'\', download URL: \'\'{4}\'\'', new String[]{
```

```
newFile.id, newFile.name, newFile.description, newFile.externalDocumentUrl,
newFile.downloadUrl}));
```

SEE ALSO:

setTestAddRepositoryItem(repositoryId, repositoryFolderId, file, fileData, result)

Apex Developer Guide: Testing ConnectApi Code

addRepositoryItem(communityId, repositoryId, repositoryFolderId, file, fileData)

Add a repository item, including the binary file, in an Experience Cloud site.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.RepositoryFolderItem addRepositoryItem(String communityId,
String repositoryId, String repositoryFolderId, ConnectApi.ContentHubItemInput file,
ConnectApi.BinaryInput fileData)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

repositoryId
Type: String

The ID of the repository.

repositoryFolderId
Type: String

The ID of the repository folder.

file
Type: `ConnectApi.ContentHubItemInput`

The item type ID and fields of the item type.

fileData
Type: `ConnectApi.BinaryInput`

The binary file.

Return Value

Type: `ConnectApi.RepositoryFolderItem`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestAddRepositoryItem(communityId, repositoryId, repositoryFolderId, file, fileData, result)

Apex Developer Guide: Testing ConnectApi Code

`getAllowedItemTypes(repositoryId, repositoryFolderId)`

Get the item types that the context user is allowed to create in the repository folder.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.ContentHubAllowedItemTypeCollection getAllowedItemTypes(String repositoryId, String repositoryFolderId)
```

Parameters

`repositoryId`
Type: String

The ID of the repository.

`repositoryFolderId`
Type: String

The ID of the repository folder.

Return Value

Type: `ConnectApi.ContentHubAllowedItemTypeCollection`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetAllowedItemTypes(repositoryId, repositoryFolderId, result)

Apex Developer Guide: Testing ConnectApi Code

`getAllowedItemTypes(repositoryId, repositoryFolderId, filter)`

Get the item types, filtered by type, that the context user is allowed to create in the repository folder.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.ContentHubAllowedItemTypeCollection getAllowedItemTypes(String
repositoryId, String repositoryFolderId, ConnectApi.ConnectContentHubItemType filter)
```

Parameters

`repositoryId`
Type: String

The ID of the repository.

`repositoryFolderId`
Type: String

The ID of the repository folder.

`filter`
Type: `ConnectApi.ContentHubItemType`

Item types. Values are:

- `Any`—Includes files and folders.
- `FilesOnly`—Includes files only.
- `FoldersOnly`—Includes folders only.

Return Value

Type: `ConnectApi.ContentHubAllowedItemTypeCollection`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

Example

This example calls `getAllowedItemTypes(repositoryId, repositoryFolderId, ConnectApi.ContentHubItemType.FilesOnly)` to get the first `ConnectApi.ContentHubItemTypeSummary.id` of a file. The context user can create allowed files in a repository folder in the external system.

```
final ConnectApi.ContentHubAllowedItemTypeCollection allowedItemTypesColl =
ConnectApi.ContentHub.getAllowedItemTypes(repositoryId, repositoryFolderId,
ConnectApi.ContentHubItemType.FilesOnly);
final List<ConnectApi.ContentHubItemTypeSummary> allowedItemTypes =
allowedItemTypesColl.allowedItemTypes;
string allowedFileItemTypeId = null;
if(allowedItemTypes.size() > 0){
   ConnectApi.ContentHubItemTypeSummary allowedItemTypeSummary = allowedItemTypes.get(0);

   allowedFileItemTypeId = allowedItemTypeSummary.id;
}
```

SEE ALSO:

- setTestGetAllowedItemTypes(repositoryId, repositoryFolderId, filter, result)
- *Apex Developer Guide*: Testing ConnectApi Code

`getAllowedItemTypes(communityId, repositoryId, repositoryFolderId)`

Get the item types that the context user is allowed to create in the repository folder in an Experience Cloud site.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.ContentHubAllowedItemTypeCollection getAllowedItemTypes(String
communityId, String repositoryId, String repositoryFolderId)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

repositoryId
Type: String

The ID of the repository.

repositoryFolderId
Type: String

The ID of the repository folder.

Return Value

Type: `ConnectApi.ContentHubAllowedItemTypeCollection`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetAllowedItemTypes(communityId, repositoryId, repositoryFolderId, result)

Apex Developer Guide: Testing ConnectApi Code

`getAllowedItemTypes(communityId, repositoryId, repositoryFolderId, filter)`

Get the item types, filtered by type, that the context user is allowed to create in the repository folder in an Experience Cloud site.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.ContentHubAllowedItemTypeCollection getAllowedItemTypes(String
communityId, String repositoryId, String repositoryFolderId,
ConnectApi.ConnectContentHubItemType filter)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

repositoryId
Type: String

The ID of the repository.

repositoryFolderId
Type: String

The ID of the repository folder.

filter
Type: `ConnectApi.ContentHubItemType`

Item types. Values are:

- `Any`—Includes files and folders.
- `FilesOnly`—Includes files only.
- `FoldersOnly`—Includes folders only.

Return Value

Type: `ConnectApi.ContentHubAllowedItemTypeCollection`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetAllowedItemTypes(communityId, repositoryId, repositoryFolderId, filter, result)

Apex Developer Guide: Testing ConnectApi Code

`getFilePreview(repositoryId, repositoryFileId, formatType)`

Get a repository file preview.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.FilePreview getFilePreview(String repositoryId, String
repositoryFileId, ConnectApi.FilePreviewFormat formatType)
```

Parameters

repositoryId
Type: String

The ID of the repository.

repositoryFileId
Type: String

The ID of the repository file.

formatType

Type: ConnectApi.FilePreviewFormat

Specifies the format of the file preview. Values are:

- `Jpg`—Preview format is JPG.
- `Pdf`—Preview format is PDF.
- `Svg`—Preview format is compressed SVG.
- `Thumbnail`—Preview format is 240 x 180 PNG.
- `ThumbnailBig`—Preview format is 720 x 480 PNG.
- `ThumbnailTiny`—Preview format is 120 x 90 PNG.

PDF previews are available for files of type DOC, DOCX, PPT, PPTX, TEXT, XLS, and XLSX. SVG files are generated on demand.

If you're concerned that feature-rich SVG previews don't work in your org, choose alternative file previews. To use JPG file previews, enter *general* in the Quick Find box in Setup. Select General Settings, and then select **Display alternative file previews**.

Return Value

Type: ConnectApi.FilePreview

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

Example

This example calls `getFilePreview(repositoryId, repositoryFileId, ConnectApi.FilePreviewFormat.Thumbnail)` to get the thumbnail format preview along with its respective URL and number of thumbnail renditions. For each thumbnail format, we show every rendition URL available.

```
final String gDriveRepositoryId = '0XCxx00000000ODGAY', gDriveFileId =
'document:1-zcA1BaeoQbo2_yNFiHCcK6QJTPmOke-kHFC4TYg3rk';final ConnectApi.FilePreview
filePreview =
ConnectApi.ContentHub.getFilePreview(gDriveRepositoryId, gDriveFileId,
ConnectApi.FilePreviewFormat.Thumbnail);System.debug(String.format('Preview - URL:
\'\'{0}\'\', format: \'\'{1}\'\', nbr of
renditions for this format: {2}', new String[]{ filePreview.url,
filePreview.format.name(),String.valueOf(filePreview.previewUrls.size())}));for(ConnectApi.FilePreviewUrl
 filePreviewUrl : filePreview.previewUrls){
   System.debug('-----> Rendition URL: ' + filePreviewUrl.previewUrl);
}
```

SEE ALSO:

setTestGetFilePreview(repositoryId, repositoryFileId, formatType, result)

Apex Developer Guide: Testing ConnectApi Code

getFilePreview(repositoryId, repositoryFileId, formatType, startPageNumber, endPageNumber)

Get a page or page range of a repository file preview.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.FilePreview getFilePreview(String repositoryId, String repositoryFileId, ConnectApi.FilePreviewFormat formatType, Integer startPageNumber, Integer endPageNumber)
```

Parameters

repositoryId
Type: String

The ID of the repository.

repositoryFileId
Type: String

The ID of the repository file.

formatType
Type: ConnectApi.FilePreviewFormat

Specifies the format of the file preview. Values are:

- `Jpg`—Preview format is JPG.
- `Pdf`—Preview format is PDF.
- `Svg`—Preview format is compressed SVG.
- `Thumbnail`—Preview format is 240 x 180 PNG.
- `ThumbnailBig`—Preview format is 720 x 480 PNG.
- `ThumbnailTiny`—Preview format is 120 x 90 PNG.

PDF previews are available for files of type DOC, DOCX, PPT, PPTX, TEXT, XLS, and XLSX. SVG files are generated on demand.

If you're concerned that feature-rich SVG previews don't work in your org, choose alternative file previews. To use JPG file previews, enter *general* in the Quick Find box in Setup. Select General Settings, and then select **Display alternative file previews**.

startPageNumber
Type: Integer

The starting page number in the range of file preview URLs.

endPageNumber
Type: Integer

The ending page number in the range of file preview URLs.

Return Value

Type: `ConnectApi.FilePreview`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFilePreview(repositoryId, repositoryFileId, formatType, startPageNumber, endPageNumber, result)

Apex Developer Guide: Testing ConnectApi Code

`getFilePreview(communityId, repositoryId, repositoryFileId, formatType)`

Get a repository file preview in an Experience Cloud site.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.FilePreview getFilePreview(String communityId, String
repositoryId, String repositoryFileId, ConnectApi.FilePreviewFormat formatType)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`repositoryId`
Type: String

The ID of the repository.

`repositoryFileId`
Type: String

The ID of the repository file.

`formatType`
Type: `ConnectApi.FilePreviewFormat`

Specifies the format of the file preview. Values are:

- `Jpg`—Preview format is JPG.
- `Pdf`—Preview format is PDF.
- `Svg`—Preview format is compressed SVG.

- `Thumbnail`—Preview format is 240 x 180 PNG.
- `ThumbnailBig`—Preview format is 720 x 480 PNG.
- `ThumbnailTiny`—Preview format is 120 x 90 PNG.

PDF previews are available for files of type DOC, DOCX, PPT, PPTX, TEXT, XLS, and XLSX. SVG files are generated on demand.

If you're concerned that feature-rich SVG previews don't work in your org, choose alternative file previews. To use JPG file previews, enter *general* in the Quick Find box in Setup. Select General Settings, and then select **Display alternative file previews**.

Return Value

Type: `ConnectApi.FilePreview`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFilePreview(communityId, repositoryId, repositoryFileId, formatType, result)

Apex Developer Guide: Testing ConnectApi Code

`getFilePreview(communityId, repositoryId, repositoryFileId, formatType, startPageNumber, endPageNumber)`

Get a page or page range of a repository file preview in an Experience Cloud site.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.FilePreview getFilePreview(String communityId, String repositoryId, String repositoryFileId, ConnectApi.FilePreviewFormat formatType, Integer startPageNumber, Integer endPageNumber)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

repositoryId
Type: String

The ID of the repository.

repositoryFileId
Type: String

The ID of the repository file.

formatType
Type: `ConnectApi.FilePreviewFormat`

Specifies the format of the file preview. Values are:

- `Jpg`—Preview format is JPG.
- `Pdf`—Preview format is PDF.
- `Svg`—Preview format is compressed SVG.
- `Thumbnail`—Preview format is 240 x 180 PNG.
- `ThumbnailBig`—Preview format is 720 x 480 PNG.
- `ThumbnailTiny`—Preview format is 120 x 90 PNG.

PDF previews are available for files of type DOC, DOCX, PPT, PPTX, TEXT, XLS, and XLSX. SVG files are generated on demand.

If you're concerned that feature-rich SVG previews don't work in your org, choose alternative file previews. To use JPG file previews, enter *general* in the Quick Find box in Setup. Select General Settings, and then select **Display alternative file previews**.

startPageNumber
Type: Integer

The starting page number in the range of file preview URLs.

endPageNumber
Type: Integer

The ending page number in the range of file preview URLs.

Return Value

Type: `ConnectApi.FilePreview`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetFilePreview(communityId, repositoryId, repositoryFileId, formatType, startPageNumber, endPageNumber, result)

Apex Developer Guide: Testing ConnectApi Code

`getItemType(repositoryId, repositoryItemTypeId)`

Get information about an item type associated with a repository.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.ContentHubItemTypeDetail getItemType(String repositoryId,
String repositoryItemTypeId)
```

Parameters

repositoryId
Type: String

The ID of the repository.

repositoryItemTypeId
Type: String

The ID of the repository item type.

Return Value

Type: `ConnectApi.ContentHubItemTypeDetail`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetItemType(repositoryId, repositoryItemTypeId, result)

Apex Developer Guide: Testing ConnectApi Code

`getItemType(communityId, repositoryId, repositoryItemTypeId)`

Get information about an item type associated with a repository in an Experience Cloud site.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.ContentHubItemTypeDetail getItemType(String communityId, String
repositoryId, String repositoryItemTypeId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

repositoryId
Type: String

The ID of the repository.

repositoryItemTypeId
Type: String

The ID of the repository item type.

Return Value

Type: `ConnectApi.ContentHubItemTypeDetail`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetItemType(communityId, repositoryId, repositoryItemTypeId, result)

Apex Developer Guide: Testing ConnectApi Code

`getPreviews(repositoryId, repositoryFileId)`

Get information about a repository file's supported previews.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.FilePreviewCollection getPreviews(String repositoryId, String repositoryFileId)
```

Parameters

repositoryId
Type: String

The ID of the repository.

repositoryFileId
Type: String

The ID of the repository file.

Return Value

Type: `ConnectApi.FilePreviewCollection`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

Example

This example gets all supported preview formats and their respective URLs and number of renditions. For each supported preview format, we show every rendition URL available.

```
final String gDriveRepositoryId = '0XCxx00000000ODGAY', gDriveFileId =
'document:1-zcA1BaeoQbo2_yNFiHCcK6QJTPmOke-kHFC4TYg3rk';
final ConnectApi.FilePreviewCollection previewsCollection =
ConnectApi.ContentHub.getPreviews(gDriveRepositoryId, gDriveFileId);
for(ConnectApi.FilePreview filePreview : previewsCollection.previews){
   System.debug(String.format('Preview - URL: \'\'{0}\'\', format: \'\'{1}\'\', nbr of
renditions for this format: {2}', new String[]{ filePreview.url,
filePreview.format.name(),String.valueOf(filePreview.previewUrls.size())}));
   for(ConnectApi.FilePreviewUrl filePreviewUrl : filePreview.previewUrls){
      System.debug('-----> Rendition URL: ' + filePreviewUrl.previewUrl);
      }
}
```

SEE ALSO:

setTestGetPreviews(repositoryId, repositoryFileId, result)

Apex Developer Guide: Testing ConnectApi Code

getPreviews(communityId, repositoryId, repositoryFileId)

Get information about a repository file's supported previews in an Experience Cloud site.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.FilePreviewCollection getPreviews(String communityId, String repositoryId, String repositoryFileId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

repositoryId
Type: String

The ID of the repository.

repositoryFileId
Type: String

The ID of the repository file.

Return Value

Type: `ConnectApi.FilePreviewCollection`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetPreviews(communityId, repositoryId, repositoryFileId, result)

Apex Developer Guide: Testing ConnectApi Code

`getRepositories()`

Get a list of repositories.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.ContentHubRepositoryCollection getRepositories()
```

Return Value

Type: ConnectApi.ContentHubRepositoryCollection

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

Example

This example gets all repositories and gets the first SharePoint online repository found.

```
final string sharePointOnlineProviderType ='ContentHubSharepointOffice365';
final ConnectApi.ContentHubRepositoryCollection repositoryCollection =
ConnectApi.ContentHub.getRepositories();
ConnectApi.ContentHubRepository sharePointOnlineRepository = null;
for(ConnectApi.ContentHubRepository repository : repositoryCollection.repositories){
   if(sharePointOnlineProviderType.equalsIgnoreCase(repository.providerType.type)){
      sharePointOnlineRepository = repository;
      break;
   }
}
```

SEE ALSO:

setTestGetRepositories(result)

Apex Developer Guide: Testing ConnectApi Code

`getRepositories(communityId)`

Get a list of repositories in an Experience Cloud site.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.ContentHubRepositoryCollection getRepositories(String
communityId)
```

Parameters

communityId

Type: String

ID for an Experience Cloud site, `internal`, or `null`.

Return Value

Type: `ConnectApi.ContentHubRepositoryCollection`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

- setTestGetRepositories(communityId, result)
- *Apex Developer Guide*: Testing ConnectApi Code

`getRepositories(pageParam, pageSize)`

Get a page of repositories.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.ContentHubRepositoryCollection getRepositories(Integer
pageParam, Integer pageSize)
```

Parameters

`pageParam`
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

`pageSize`
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default page size is 25.

Return Value

Type: `ConnectApi.ContentHubRepositoryCollection`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetRepositories(pageParam, pageSize, result)

Apex Developer Guide: Testing ConnectApi Code

`getRepositories(communityId, pageParam, pageSize)`

Get a page of repositories in an Experience Cloud site.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.ContentHubRepositoryCollection getRepositories(String
communityId, Integer pageParam, Integer pageSize)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`pageParam`
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

`pageSize`
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default page size is 25.

Return Value

Type: `ConnectApi.ContentHubRepositoryCollection`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetRepositories(communityId, pageParam, pageSize, result)

Apex Developer Guide: Testing ConnectApi Code

`getRepository(repositoryId)`

Get a repository.

API Version

369.0

Requires Chatter

No

Signature

```
public static ConnectApi.ContentHubRepository getRepository(String repositoryId)
```

Parameters

`repositoryId`
Type: String

The ID of the repository.

Return Value

Type: `ConnectApi.ContentHubRepository`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

Example

```
final string repositoryId = '0XCxx0000000123GAA';
final ConnectApi.ContentHubRepository repository =
ConnectApi.ContentHub.getRepository(repositoryId);
```

SEE ALSO:

setTestGetRepository(repositoryId, result)

Apex Developer Guide: Testing ConnectApi Code

`getRepository(communityId, repositoryId)`

Get a repository in an Experience Cloud site.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.ContentHubRepository getRepository(String communityId, String repositoryId)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`repositoryId`
Type: String

The ID of the repository.

Return Value

Type: `ConnectApi.ContentHubRepository`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetRepository(communityId, repositoryId, result)

Apex Developer Guide: Testing ConnectApi Code

`getRepositoryFile(repositoryId, repositoryFileId)`

Get a repository file.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.RepositoryFileDetail getRepositoryFile(String repositoryId,
String repositoryFileId)
```

Parameters

repositoryId
Type: String

The ID of the repository.

repositoryFileId
Type: String

The ID of the repository file.

Return Value

Type: `ConnectApi.RepositoryFileDetail`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

Example

```
final String gDriveRepositoryId = '0XCxx00000000ODGAY', gDriveFileId =
'file:0B0lTys1KmM3sTmxKNjVJbWZja00';
final ConnectApi.RepositoryFileDetail file =
ConnectApi.ContentHub.getRepositoryFile(gDriveRepositoryId, gDriveFileId);
System.debug(String.format('File - name: \'\'{0}\'\', size: {1}, external URL: \'\'{2}\'\',
 download URL: \'\'{3}\'\'',
   new String[]{ file.name, String.valueOf(file.contentSize), file.externalDocumentUrl,
file.downloadUrl}));
```

SEE ALSO:

setTestGetRepositoryFile(repositoryId, repositoryFileId, result)

Apex Developer Guide: Testing ConnectApi Code

getRepositoryFile(repositoryId, repositoryFileId, includeExternalFilePermissionsInfo)

Get a repository file with or without permissions information.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.RepositoryFileDetail getRepositoryFile(String repositoryId,
String repositoryFileId, Boolean includeExternalFilePermissionsInfo)
```

Parameters

repositoryId
Type: String

The ID of the repository.

repositoryFileId
Type: String

The ID of the repository file.

includeExternalFilePermissionsInfo
Type: Boolean

Specifies whether to include permission information, such as whether the file is shared and what are the available permission types.

Managing external file permissions is supported for Google Drive, SharePoint Online, and OneDrive for Business.

Return Value

Type: ConnectApi.RepositoryFileDetail

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

Example

```
final String gDriveRepositoryId = '0XCxx00000000ODGAY', gDriveFileId =
'file:0B0lTys1KmM3sTmxKNjVJbWZja00';

final ConnectApi.RepositoryFileDetail file =
ConnectApi.ContentHub.getRepositoryFile(gDriveRepositoryId, gDriveFileId, true);
System.debug(String.format('File - name: \'\'{0}\'\', size: {1}, external URL: \'\'{2}\'\',
 download URL: \'\'{3}\'\'', new String[]{ file.name, String.valueOf(file.contentSize),
file.externalDocumentUrl, file.downloadUrl}));
final ConnectApi.ExternalFilePermissionInformation externalFilePermInfo =
file.externalFilePermissionInformation;

//permission types
final List<ConnectApi.ContentHubPermissionType> permissionTypes =
```

```
externalFilePermInfo.externalFilePermissionTypes;
for(ConnectApi.ContentHubPermissionType permissionType : permissionTypes){
   System.debug(String.format('Permission type - id: \'\'{0}\'\', label: \'\'{1}\'\'', new
 String[]{ permissionType.id, permissionType.label}));
}

//permission groups
final List<ConnectApi.RepositoryGroupSummary> groups =
externalFilePermInfo.repositoryPublicGroups;
for(ConnectApi.RepositoryGroupSummary ggroup : groups){
   System.debug(String.format('Group - id: \'\'{0}\'\', name: \'\'{1}\'\', type:
\'\'{2}\'\'', new String[]{ ggroup.id, ggroup.name, ggroup.type.name()}));
}
```

SEE ALSO:

setTestGetRepositoryFile(repositoryId, repositoryFileId, includeExternalFilePermissionsInfo, result)

Apex Developer Guide: Testing ConnectApi Code

`getRepositoryFile(communityId, repositoryId, repositoryFileId)`

Get a repository file in an Experience Cloud site.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.RepositoryFileDetail getRepositoryFile(String communityId,
String repositoryId, String repositoryFileId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

repositoryId
Type: String

The ID of the repository.

repositoryFileId
Type: String

The ID of the repository file.

Return Value

Type: `ConnectApi.RepositoryFileDetail`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetRepositoryFile(communityId, repositoryId, repositoryFileId, result)

Apex Developer Guide: Testing ConnectApi Code

`getRepositoryFile(communityId, repositoryId, repositoryFileId, includeExternalFilePermissionsInfo)`

Get a repository file with or without permissions information in an Experience Cloud site.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.RepositoryFileDetail getRepositoryFile(String communityId,
String repositoryId, String repositoryFileId, Boolean includeExternalFilePermissionsInfo)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`repositoryId`
Type: String

The ID of the repository.

`repositoryFileId`
Type: String

The ID of the repository file.

`includeExternalFilePermissionsInfo`
Type: Boolean

Specifies whether to include permission information, such as whether the file is shared and what are the available permission types.

Managing external file permissions is supported for Google Drive, SharePoint Online, and OneDrive for Business.

Return Value

Type: ConnectApi.RepositoryFileDetail

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetRepositoryFile(communityId, repositoryId, repositoryFileId, includeExternalFilePermissionsInfo, result)

Apex Developer Guide: Testing ConnectApi Code

`getRepositoryFolder(repositoryId, repositoryFolderId)`

Get a repository folder.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.RepositoryFolderDetail getRepositoryFolder(String repositoryId,
String repositoryFolderId)
```

Parameters

`repositoryId`
Type: String

The ID of the repository.

`repositoryFolderId`
Type: String

The ID of the repository folder.

Return Value

Type: ConnectApi.RepositoryFolderDetail

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

Example

```
final String gDriveRepositoryId = '0XCxx00000000ODGAY', gDriveFolderId =
'folder:0B0lTys1KmM3sSVJ2bjIzTGFqSWs';
final ConnectApi.RepositoryFolderDetail folder =
ConnectApi.ContentHub.getRepositoryFolder(gDriveRepositoryId, gDriveFolderId);
System.debug(String.format('Folder - name: \'\'{0}\'\', description: \'\'{1}\'\', external
 URL: \'\'{2}\'\', folder items URL: \'\'{3}\'\'',
   new String[]{ folder.name, folder.description, folder.externalFolderUrl,
folder.folderItemsUrl}));
```

SEE ALSO:

setTestGetRepositoryFolder(repositoryId, repositoryFolderId, result)

Apex Developer Guide: Testing ConnectApi Code

`getRepositoryFolder(communityId, repositoryId, repositoryFolderId)`

Get a repository folder in an Experience Cloud site.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.RepositoryFolderDetail getRepositoryFolder(String communityId,
String repositoryId, String repositoryFolderId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

repositoryId
Type: String

The ID of the repository.

repositoryFolderId
Type: String

The ID of the repository folder.

Return Value

Type: `ConnectApi.RepositoryFolderDetail`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetRepositoryFolder(communityId, repositoryId, repositoryFolderId, result)

Apex Developer Guide: Testing ConnectApi Code

`getRepositoryFolderItems(repositoryId, repositoryFolderId)`

Get repository folder items.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.RepositoryFolderItemsCollection getRepositoryFolderItems(String repositoryId, String repositoryFolderId)
```

Parameters

`repositoryId`
Type: String

The ID of the repository.

`repositoryFolderId`
Type: String

The ID of the repository folder.

Return Value

Type: `ConnectApi.RepositoryFolderItemsCollection`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

Example

This example gets the collection of items in a repository folder. For files, we show the file's name, size, external URL, and download URL. For folders, we show the folder's name, description, and external URL.

```
final String gDriveRepositoryId = '0XCxx00000000ODGAY', gDriveFolderId =
'folder:0B0lTys1KmM3sSVJ2bjIzTGFqSWs';
final ConnectApi.RepositoryFolderItemsCollection folderItemsColl =
ConnectApi.ContentHub.getRepositoryFolderItems(gDriveRepositoryId,gDriveFolderId);
final List<ConnectApi.RepositoryFolderItem> folderItems = folderItemsColl.items;
System.debug('Number of items in repository folder: ' + folderItems.size());
for(ConnectApi.RepositoryFolderItem item : folderItems){
   ConnectApi.RepositoryFileSummary fileSummary = item.file;
   if(fileSummary != null){
      System.debug(String.format('File item - name: \'\'{0}\'\', size: {1}, external URL:
 \'\'{2}\'\', download URL: \'\'{3}\'\'', new String[]{ fileSummary.name,
String.valueOf(fileSummary.contentSize), fileSummary.externalDocumentUrl,
fileSummary.downloadUrl}));
      }else{
         ConnectApi.RepositoryFolderSummary folderSummary = item.folder;
         System.debug(String.format('Folder item - name: \'\'{0}\'\', description:
\'\'{1}\'\'',  new String[]{ folderSummary.name, folderSummary.description}));
      }
}
```

SEE ALSO:

setTestGetRepositoryFolderItems(repositoryId, repositoryFolderId, result)

Apex Developer Guide: Testing ConnectApi Code

`getRepositoryFolderItems(communityId, repositoryId, repositoryFolderId)`

Get repository folder items in an Experience Cloud site.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.RepositoryFolderItemsCollection getRepositoryFolderItems(String
communityId, String repositoryId, String repositoryFolderId)
```

Parameters

`communityId`

Type: String

ID for an Experience Cloud site, `internal`, or `null`.

repositoryId
Type: String

The ID of the repository.

repositoryFolderId
Type: String

The ID of the repository folder.

Return Value

Type: `ConnectApi.RepositoryFolderItemsCollection`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetRepositoryFolderItems(communityId, repositoryId, repositoryFolderId, result)

Apex Developer Guide: Testing ConnectApi Code

`getRepositoryFolderItems(repositoryId, repositoryFolderId, pageParam, pageSize)`

Get a page of repository folder items.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.RepositoryFolderItemsCollection getRepositoryFolderItems(String repositoryId, String repositoryFolderId, Integer pageParam, Integer pageSize)
```

Parameters

repositoryId
Type: String

The ID of the repository.

repositoryFolderId
Type: String

The ID of the repository folder.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default page size is 25.

Return Value

Type: `ConnectApi.RepositoryFolderItemsCollection`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetRepositoryFolderItems(repositoryId, repositoryFolderId, pageParam, pageSize, result)

Apex Developer Guide: Testing ConnectApi Code

`getRepositoryFolderItems(communityId, repositoryId, repositoryFolderId, pageParam, pageSize)`

Get a page of repository folder items in an Experience Cloud site.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.RepositoryFolderItemsCollection getRepositoryFolderItems(String communityId, String repositoryId, String repositoryFolderId, Integer pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

repositoryId
Type: String

The ID of the repository.

repositoryFolderId
Type: String

The ID of the repository folder.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default page size is 25.

Return Value

Type: `ConnectApi.RepositoryFolderItemsCollection`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetRepositoryFolderItems(communityId, repositoryId, repositoryFolderId, pageParam, pageSize, result)

Apex Developer Guide: Testing ConnectApi Code

updateRepositoryFile(repositoryId, repositoryFileId, file)

Update the name of a repository file.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.RepositoryFileDetail updateRepositoryFile(String repositoryId,
String repositoryFileId, ConnectApi.ContentHubItemInput file)
```

Parameters

repositoryId
Type: String

The ID of the repository.

repositoryFileId
Type: String

The ID of the repository file.

file

Type: ConnectApi.ContentHubItemInput

The item type ID and fields of the item type.

When updating the metadata of a repository file, only the name field can be updated.

Return Value

Type: ConnectApi.RepositoryFileDetail

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

Example

This example updates the name of a file in a Google Drive repository. After the file is updated, we show the file's ID, name, description, external URL, download URL.

```
final String gDriveRepositoryId = '0XCxx00000000ODGAY', gDriveFolderId =
'folder:0B0lTys1KmM3sSVJ2bjIzTGFqSWs', gDriveFileId =
'document:1q9OatVpcyYBK-JWzp_PhR75ulQghwFP15zhkamKrRcQ';

final ConnectApi.ContentHubItemInput updatedItem = new ConnectApi.ContentHubItemInput();
updatedItem.itemTypeId = 'document'; //see getAllowedTypes for any file item types available
 for creation/update
updatedItem.fields = new List<ConnectApi.ContentHubFieldValueInput>();

//Metadata: name field
final ConnectApi.ContentHubFieldValueInput fieldValueInputName = new
ConnectApi.ContentHubFieldValueInput();
fieldValueInputName.name = 'name';
fieldValueInputName.value =  'updated file name.txt';
updatedItem.fields.add(fieldValueInputName);

final ConnectApi.RepositoryFileDetail updatedFile =
ConnectApi.ContentHub.updateRepositoryFile(gDriveRepositoryId, gDriveFileId, updatedItem);
System.debug(String.format('Updated file - id: \'\'{0}\'\', name: \'\'{1}\'\', description:
 \'\'{2}\'\',\n external URL: \'\'{3}\'\', download URL: \'\'{4}\'\'',  new String[]{
updatedFile.id, updatedFile.name, updatedFile.description, updatedFile.externalDocumentUrl,
 updatedFile.downloadUrl}));
```

SEE ALSO:

setTestUpdateRepositoryFile(communityId, repositoryId, repositoryFileId, file, fileData, result)

Apex Developer Guide: Testing ConnectApi Code

updateRepositoryFile(repositoryId, repositoryFileId, file, fileData)

Update the content of a repository file.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.RepositoryFileDetail updateRepositoryFile(String repositoryId,
String repositoryFileId, ConnectApi.ContentHubItemInput file, ConnectApi.BinaryInput
fileData)
```

Parameters

repositoryId
Type: String

The ID of the repository.

repositoryFileId
Type: String

The ID of the repository file.

file
Type: ConnectApi.ContentHubItemInput

The item type ID and fields of the item type.

When updating the metadata of a repository file, only the name field can be updated.

fileData
Type: ConnectApi.BinaryInput

The binary file.

Return Value

Type: ConnectApi.RepositoryFileDetail

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

Example

This example updates the content and name of a file in a Google Drive repository. After the file is updated, we show the file's ID, name, description, external URL, and download URL.

```
final String gDriveRepositoryId = '0XCxx00000000ODGAY', gDriveFolderId =
'folder:0B0lTys1KmM3sSVJ2bjIzTGFqSWs', gDriveFileId =
'document:1q9OatVpcyYBK-JWzp_PhR75ulQghwFP15zhkamKrRcQ';

final ConnectApi.ContentHubItemInput updatedItem = new ConnectApi.ContentHubItemInput();
```

```
updatedItem.itemTypeId = 'document'; //see getAllowedTypes for any file item types available
 for creation/update
updatedItem.fields = new List<ConnectApi.ContentHubFieldValueInput>();

//Metadata: name field
final ConnectApi.ContentHubFieldValueInput fieldValueInputName = new
ConnectApi.ContentHubFieldValueInput();
fieldValueInputName.name = 'name';
fieldValueInputName.value =  'updated file name.txt';
updatedItem.fields.add(fieldValueInputName);

//Binary content
final Blob updatedFileBlob = Blob.valueOf('even more awesome content for updated file');
final String updatedFileMimeType = 'text/plain';
final ConnectApi.BinaryInput fileBinaryInput = new ConnectApi.BinaryInput(updatedFileBlob,
 updatedFileMimeType, updatedFileName);

final ConnectApi.RepositoryFileDetail updatedFile =
ConnectApi.ContentHub.updateRepositoryFile(gDriveRepositoryId, gDriveFileId, updatedItem);
System.debug(String.format('Updated file - id: \'\'{0}\'\', name: \'\'{1}\'\', description:
 \'\'{2}\'\',\n external URL: \'\'{3}\'\', download URL: \'\'{4}\'\'',  new String[]{
updatedFile.id, updatedFile.name, updatedFile.description, updatedFile.externalDocumentUrl,
 updatedFile.downloadUrl}));
```

SEE ALSO:

setTestUpdateRepositoryFile(repositoryId, repositoryFileId, file, result)

Apex Developer Guide: Testing ConnectApi Code

updateRepositoryFile(communityId, repositoryId, repositoryFileId, file)

Update the name of a repository file in an Experience Cloud site.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.RepositoryFileDetail updateRepositoryFile(String communityId,
String repositoryId, String repositoryFileId, ConnectApi.ContentHubItemInput file)
```

Parameters

communityId

Type: String

ID for an Experience Cloud site, `internal`, or `null`.

repositoryId
Type: String

The ID of the repository.

repositoryFileId
Type: String

The ID of the repository file.

file
Type: `ConnectApi.ContentHubItemInput`

The item type ID and fields of the item type.

When updating the metadata of a repository file, only the name field can be updated.

Return Value

Type: `ConnectApi.RepositoryFileDetail`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestUpdateRepositoryFile(repositoryId, repositoryFileId, file, fileData, result)

Apex Developer Guide: Testing ConnectApi Code

`updateRepositoryFile(communityId, repositoryId, repositoryFileId, file, fileData)`

Update the content of a repository file in an Experience Cloud site.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.RepositoryFileDetail updateRepositoryFile(String communityId,
String repositoryId, String repositoryFileId, ConnectApi.ContentHubItemInput file,
ConnectApi.BinaryInput fileData)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

repositoryId
Type: String

The ID of the repository.

repositoryFileId
Type: String

The ID of the repository file.

file
Type: `ConnectApi.ContentHubItemInput`

The item type ID and fields of the item type.

When updating the metadata of a repository file, only the name field can be updated.

fileData
Type: `ConnectApi.BinaryInput`

The binary file.

Return Value

Type: `ConnectApi.RepositoryFileDetail`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestUpdateRepositoryFile(communityId, repositoryId, repositoryFileId, file, result)

Apex Developer Guide: Testing ConnectApi Code

ContentHub Test Methods

These test methods are for `ContentHub`. All methods are static.

For information about using these methods to test your `ConnectApi` code, see Testing ConnectApi Code.

`setTestAddRepositoryItem(repositoryId, repositoryFolderId, file, result)`

Register a `ConnectApi.RepositoryFolderItem` object to be returned when the matching `addRepositoryItem(repositoryId, repositoryFolderId, file)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestAddRepositoryItem(String repositoryId, String
repositoryFolderId, ConnectApi.ContentHubItemInput file, ConnectApi.RepositoryFolderItem
result)
```

Parameters

repositoryId
Type: String

The ID of the repository.

repositoryFolderId
Type: String

The ID of the repository folder.

file
Type: `ConnectApi.ContentHubItemInput`

The item type ID and fields of the item type.

result
Type: `ConnectApi.RepositoryFolderItem`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

addRepositoryItem(repositoryId, repositoryFolderId, file)

Apex Developer Guide: Testing ConnectApi Code

setTestAddRepositoryItem(communityId, repositoryId, repositoryFolderId, file, result)

Register a `ConnectApi.RepositoryFolderItem` object to be returned when the matching `addRepositoryItem(communityId, repositoryId, repositoryFolderId, file)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestAddRepositoryItem(String communityId, String repositoryId,
String repositoryFolderId, ConnectApi.ContentHubItemInput file,
ConnectApi.RepositoryFolderItem result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

repositoryId
Type: String

The ID of the repository.

repositoryFolderId
Type: String

The ID of the repository folder.

file
Type: `ConnectApi.ContentHubItemInput`

The item type ID and fields of the item type.

result
Type: `ConnectApi.RepositoryFolderItem`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

addRepositoryItem(communityId, repositoryId, repositoryFolderId, file)

Apex Developer Guide: Testing ConnectApi Code

`setTestAddRepositoryItem(repositoryId, repositoryFolderId, file, fileData, result)`

Register a `ConnectApi.RepositoryFolderItem` object to be returned when the matching `addRepositoryItem(repositoryId, repositoryFolderId, file, fileData)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestAddRepositoryItem(String repositoryId, String
repositoryFolderId, ConnectApi.ContentHubItemInput file, ConnectApi.BinaryInput fileData,
ConnectApi.RepositoryFolderItem result)
```

Parameters

repositoryId
Type: String

The ID of the repository.

repositoryFolderId
Type: String

The ID of the repository folder.

file
Type: ConnectApi.ContentHubItemInput

The item type ID and fields of the item type.

fileData
Type: ConnectApi.BinaryInput

The binary file.

result
Type: ConnectApi.RepositoryFolderItem

Object containing test data.

Return Value

Type: Void

SEE ALSO:

addRepositoryItem(repositoryId, repositoryFolderId, file, fileData)

Apex Developer Guide: Testing ConnectApi Code

setTestAddRepositoryItem(communityId, repositoryId, repositoryFolderId, file, fileData, result)

Register a `ConnectApi.RepositoryFolderItem` object to be returned when the matching `addRepositoryItem(communityId, repositoryId, repositoryFolderId, file, fileData)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestAddRepositoryItem(String communityId, String repositoryId,
String repositoryFolderId, ConnectApi.ContentHubItemInput file, ConnectApi.BinaryInput
fileData, ConnectApi.RepositoryFolderItem result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

repositoryId
Type: String

The ID of the repository.

repositoryFolderId
Type: String

The ID of the repository folder.

file
Type: `ConnectApi.ContentHubItemInput`

The item type ID and fields of the item type.

fileData
Type: `ConnectApi.BinaryInput`

The binary file.

result
Type: `ConnectApi.RepositoryFolderItem`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

addRepositoryItem(communityId, repositoryId, repositoryFolderId, file, fileData)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetAllowedItemTypes(repositoryId, repositoryFolderId, result)`

Register a `ConnectApi.ContentHubAllowedItemTypeCollection` object to be returned when the matching `getAllowedItemTypes(repositoryId, repositoryFolderId)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetAllowedItemTypes(String repositoryId, String
repositoryFolderId, ConnectApi.ContentHubAllowedItemTypeCollection result)
```

Parameters

repositoryId
Type: String

The ID of the repository.

repositoryFolderId
Type: String

The ID of the repository folder.

result
Type: ConnectApi.ContentHubAllowedItemTypeCollection

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getAllowedItemTypes(repositoryId, repositoryFolderId)

Apex Developer Guide: Testing ConnectApi Code

setTestGetAllowedItemTypes(repositoryId, repositoryFolderId, filter, result)

Register a `ConnectApi.ContentHubAllowedItemTypeCollection` object to be returned when the matching `getAllowedItemTypes(repositoryId, repositoryFolderId, filter)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetAllowedItemTypes(String repositoryId, String
repositoryFolderId, ConnectApi.ContentHubItemType filter,
ConnectApi.ContentHubAllowedItemTypeCollection result)
```

Parameters

repositoryId
Type: String

The ID of the repository.

repositoryFolderId
Type: String

The ID of the repository folder.

filter
Type: ConnectApi.ContentHubItemType

Item types. Values are:

- `Any`—Includes files and folders.
- `FilesOnly`—Includes files only.
- `FoldersOnly`—Includes folders only.

result
Type: `ConnectApi.ContentHubAllowedItemTypeCollection`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getAllowedItemTypes(repositoryId, repositoryFolderId, filter)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetAllowedItemTypes(communityId, repositoryId, repositoryFolderId, result)`

Register a `ConnectApi.ContentHubAllowedItemTypeCollection` object to be returned when the matching `getAllowedItemTypes(communityId, repositoryId, repositoryFolderId)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetAllowedItemTypes(String communityId, String repositoryId,
String repositoryFolderId, ConnectApi.ContentHubAllowedItemTypeCollection result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

repositoryId
Type: String

The ID of the repository.

repositoryFolderId
Type: String

The ID of the repository folder.

result
Type: `ConnectApi.ContentHubAllowedItemTypeCollection`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getAllowedItemTypes(communityId, repositoryId, repositoryFolderId)

Apex Developer Guide: Testing ConnectApi Code

setTestGetAllowedItemTypes(communityId, repositoryId, repositoryFolderId, filter, result)

Register a `ConnectApi.ContentHubAllowedItemTypeCollection` object to be returned when the matching `getAllowedItemTypes(communityId, repositoryId, repositoryFolderId, filter)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetAllowedItemTypes(String communityId, String repositoryId,
String repositoryFolderId, ConnectApi.ContentHubItemType filter,
ConnectApi.ContentHubAllowedItemTypeCollection result)
```

Parameters

communityId

Type: String

ID for an Experience Cloud site, `internal`, or `null`.

repositoryId

Type: String

The ID of the repository.

repositoryFolderId

Type: String

The ID of the repository folder.

filter

Type: `ConnectApi.ContentHubItemType`

Item types. Values are:

- `Any`—Includes files and folders.
- `FilesOnly`—Includes files only.
- `FoldersOnly`—Includes folders only.

result
Type: ConnectApi.ContentHubAllowedItemTypeCollection

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getAllowedItemTypes(communityId, repositoryId, repositoryFolderId, filter)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFilePreview(repositoryId, repositoryFileId, formatType, result)

Register a `ConnectApi.FilePreview` object to be returned when the matching `getFilePreview(repositoryId, repositoryFileId, formatType)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetFilePreview(String repositoryId, String repositoryFileId,
ConnectApi.FilePreviewFormat formatType, ConnectApi.FilePreview result)
```

Parameters

repositoryId
Type: String

The ID of the repository.

repositoryFileId
Type: String

The ID of the repository file.

formatType
Type: ConnectApi.FilePreviewFormat

Specifies the format of the file preview. Values are:

- `Jpg`—Preview format is JPG.
- `Pdf`—Preview format is PDF.
- `Svg`—Preview format is compressed SVG.
- `Thumbnail`—Preview format is 240 x 180 PNG.
- `ThumbnailBig`—Preview format is 720 x 480 PNG.
- `ThumbnailTiny`—Preview format is 120 x 90 PNG.

PDF previews are available for files of type DOC, DOCX, PPT, PPTX, TEXT, XLS, and XLSX. SVG files are generated on demand.

If you're concerned that feature-rich SVG previews don't work in your org, choose alternative file previews. To use JPG file previews, enter *general* in the Quick Find box in Setup. Select General Settings, and then select **Display alternative file previews**.

result
Type: ConnectApi.FilePreview

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFilePreview(repositoryId, repositoryFileId, formatType)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFilePreview(repositoryId, repositoryFileId, formatType, startPageNumber, endPageNumber, result)

Register a `ConnectApi.FilePreview` object to be returned when the matching `getFilePreview(repositoryId, repositoryFileId, formatType, startPageNumber, endPageNumber)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetFilePreview(String repositoryId, String repositoryFileId,
ConnectApi.FilePreviewFormat formatType, Integer startPageNumber, Integer endPageNumber,
ConnectApi.FilePreview result)
```

Parameters

repositoryId
Type: String

The ID of the repository.

repositoryFileId
Type: String

The ID of the repository file.

formatType
Type: ConnectApi.FilePreviewFormat

Specifies the format of the file preview. Values are:

- `Jpg`—Preview format is JPG.
- `Pdf`—Preview format is PDF.
- `Svg`—Preview format is compressed SVG.

- `Thumbnail`—Preview format is 240 x 180 PNG.
- `ThumbnailBig`—Preview format is 720 x 480 PNG.
- `ThumbnailTiny`—Preview format is 120 x 90 PNG.

PDF previews are available for files of type DOC, DOCX, PPT, PPTX, TEXT, XLS, and XLSX. SVG files are generated on demand.

If you're concerned that feature-rich SVG previews don't work in your org, choose alternative file previews. To use JPG file previews, enter *general* in the Quick Find box in Setup. Select General Settings, and then select **Display alternative file previews**.

startPageNumber
Type: Integer

The starting page number in the range of file preview URLs.

endPageNumber
Type: Integer

The ending page number in the range of file preview URLs.

result
Type: `ConnectApi.FilePreview`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFilePreview(repositoryId, repositoryFileId, formatType, startPageNumber, endPageNumber)

Apex Developer Guide: Testing ConnectApi Code

setTestGetFilePreview(communityId, repositoryId, repositoryFileId, formatType, result)

Register a `ConnectApi.FilePreview` object to be returned when the matching `getFilePreview(communityId, repositoryId, repositoryFileId, formatType)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetFilePreview(String communityId, String repositoryId, String repositoryFileId, ConnectApi.FilePreviewFormat formatType, ConnectApi.FilePreview result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`repositoryId`
Type: String

The ID of the repository.

`repositoryFileId`
Type: String

The ID of the repository file.

`formatType`
Type: `ConnectApi.FilePreviewFormat`

Specifies the format of the file preview. Values are:

- `Jpg`—Preview format is JPG.
- `Pdf`—Preview format is PDF.
- `Svg`—Preview format is compressed SVG.
- `Thumbnail`—Preview format is 240 x 180 PNG.
- `ThumbnailBig`—Preview format is 720 x 480 PNG.
- `ThumbnailTiny`—Preview format is 120 x 90 PNG.

PDF previews are available for files of type DOC, DOCX, PPT, PPTX, TEXT, XLS, and XLSX. SVG files are generated on demand.

If you're concerned that feature-rich SVG previews don't work in your org, choose alternative file previews. To use JPG file previews, enter *`general`* in the Quick Find box in Setup. Select General Settings, and then select **Display alternative file previews**.

`result`
Type: `ConnectApi.FilePreview`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFilePreview(communityId, repositoryId, repositoryFileId, formatType)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetFilePreview(communityId, repositoryId, repositoryFileId, formatType, startPageNumber, endPageNumber, result)`

Register a `ConnectApi.FilePreview` object to be returned when the matching `getFilePreview(communityId, repositoryId, repositoryFileId, formatType, startPageNumber, endPageNumber)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetFilePreview(String communityId, String repositoryId, String
repositoryFileId, ConnectApi.FilePreviewFormat formatType, Integer startPageNumber,
Integer endPageNumber, ConnectApi.FilePreview result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

repositoryId
Type: String

The ID of the repository.

repositoryFileId
Type: String

The ID of the repository file.

formatType
Type: `ConnectApi.FilePreviewFormat`

Specifies the format of the file preview. Values are:

- `Jpg`—Preview format is JPG.
- `Pdf`—Preview format is PDF.
- `Svg`—Preview format is compressed SVG.
- `Thumbnail`—Preview format is 240 x 180 PNG.
- `ThumbnailBig`—Preview format is 720 x 480 PNG.
- `ThumbnailTiny`—Preview format is 120 x 90 PNG.

PDF previews are available for files of type DOC, DOCX, PPT, PPTX, TEXT, XLS, and XLSX. SVG files are generated on demand.

If you're concerned that feature-rich SVG previews don't work in your org, choose alternative file previews. To use JPG file previews, enter *general* in the Quick Find box in Setup. Select General Settings, and then select **Display alternative file previews**.

startPageNumber
Type: Integer

The starting page number in the range of file preview URLs.

endPageNumber
Type: Integer

The ending page number in the range of file preview URLs.

result
Type: `ConnectApi.FilePreview`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getFilePreview(communityId, repositoryId, repositoryFileId, formatType, startPageNumber, endPageNumber)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetItemType(repositoryId, repositoryItemTypeId, result)`

Register a `ConnectApi.ContentHubItemTypeDetail` object to be returned when the matching `getItemType(repositoryId, repositoryItemTypeId)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetItemType(String repositoryId, String repositoryItemTypeId,
ConnectApi.ContentHubItemTypeDetail result)
```

Parameters

`repositoryId`
Type: String

The ID of the repository.

`repositoryItemTypeId`
Type: String

The ID of the repository item type.

`result`
Type: `ConnectApi.ContentHubItemTypeDetail`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getItemType(repositoryId, repositoryItemTypeId)

Apex Developer Guide: Testing ConnectApi Code

setTestGetItemType(communityId, repositoryId, repositoryItemTypeId, result)

Register a `ConnectApi.ContentHubItemTypeDetail` object to be returned when the matching `getItemType(communityId, repositoryId, repositoryItemTypeId)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetItemType(String communityId, String repositoryId, String repositoryItemTypeId, ConnectApi.ContentHubItemTypeDetail result)
```

Parameters

communityId
: Type: String

  ID for an Experience Cloud site, `internal`, or `null`.

repositoryId
: Type: String

  The ID of the repository.

repositoryItemTypeId
: Type: String

  The ID of the repository item type.

result
: Type: `ConnectApi.ContentHubItemTypeDetail`

  Object containing test data.

Return Value

Type: Void

SEE ALSO:

- getItemType(communityId, repositoryId, repositoryItemTypeId)
- *Apex Developer Guide*: Testing ConnectApi Code

setTestGetPreviews(repositoryId, repositoryFileId, result)

Register a `ConnectApi.FilePreviewCollection` object to be returned when the matching `getPreviews(repositoryId, repositoryFileId)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetPreviews(String repositoryId, String repositoryFileId,
ConnectApi.FilePreviewCollection result)
```

Parameters

repositoryId
Type: String

The ID of the repository.

repositoryFileId
Type: String

The ID of the repository file.

result
Type: `ConnectApi.FilePreviewCollection`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getPreviews(repositoryId, repositoryFileId)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetPreviews(communityId, repositoryId, repositoryFileId, result)`

Register a `ConnectApi.FilePreviewCollection` object to be returned when the matching `getPreviews(communityId, repositoryId, repositoryFileId)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetPreviews(String communityId, String repositoryId, String
repositoryFileId, ConnectApi.FilePreviewCollection result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

repositoryId
Type: String

The ID of the repository.

repositoryFileId
Type: String

The ID of the repository file.

result
Type: `ConnectApi.FilePreviewCollection`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getPreviews(communityId, repositoryId, repositoryFileId)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetRepositories(result)`

Register a `ConnectApi.ContentHubRepositoryCollection` object to be returned when the matching `getRepositories()` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetRepositories(ConnectApi.ContentHubRepositoryCollection
result)
```

Parameters

result
Type: `ConnectApi.ContentHubRepositoryCollection`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getRepositories()

Apex Developer Guide: Testing ConnectApi Code

`setTestGetRepositories(communityId, result)`

Register a `getRepositories(communityId)` object to be returned when the matching `ConnectApi.ContentHubRepositoryCollection` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetRepositories(String communityId,
ConnectApi.ContentHubRepositoryCollection result)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`result`
Type: `ConnectApi.ContentHubRepositoryCollection`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getRepositories(communityId)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetRepositories(pageParam, pageSize, result)`

Register a `ConnectApi.ContentHubRepositoryCollection` object to be returned when the matching `getRepositories(pageParam, pageSize)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetRepositories(Integer pageParam, Integer pageSize,
ConnectApi.ContentHubRepositoryCollection result)
```

Parameters

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default page size is 25.

result
Type: `ConnectApi.ContentHubRepositoryCollection`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getRepositories(pageParam, pageSize)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetRepositories(communityId, pageParam, pageSize, result)`

Register a `ConnectApi.ContentHubRepositoryCollection` object to be returned when the matching `getRepositories(communityId, pageParam, pageSize)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetRepositories(String communityId, Integer pageParam, Integer pageSize, ConnectApi.ContentHubRepositoryCollection result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default page size is 25.

result
Type: ConnectApi.ContentHubRepositoryCollection

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getRepositories(communityId, pageParam, pageSize)

Apex Developer Guide: Testing ConnectApi Code

setTestGetRepository(repositoryId, result)

Register a `ConnectApi.ContentHubRepository` object to be returned when the matching `getRepository(repositoryId)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetRepository(String repositoryId,
ConnectApi.ContentHubRepository result)
```

Parameters

repositoryId
Type: String

The ID of the repository.

result
Type: ConnectApi.ContentHubRepository

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getRepository(repositoryId)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetRepository(communityId, repositoryId, result)`

Register a `ConnectApi.ContentHubRepository` object to be returned when the matching `getRepository(communityId, repositoryId)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetRepository(String communityId, String repositoryId,
ConnectApi.ContentHubRepository result)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`repositoryId`
Type: String

The ID of the repository.

`result`
Type: `ConnectApi.ContentHubRepository`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getRepository(communityId, repositoryId)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetRepositoryFile(repositoryId, repositoryFileId, result)`

Register a `ConnectApi.RepositoryFileDetail` object to be returned when the matching `getRepositoryFile(repositoryId, repositoryFileId)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetRepositoryFile(String repositoryId, String repositoryFileId,
ConnectApi.RepositoryFileDetail result)
```

Parameters

repositoryId
Type: String

The ID of the repository.

repositoryFileId
Type: String

The ID of the repository file.

result
Type: `ConnectApi.RepositoryFileDetail`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getRepositoryFile(repositoryId, repositoryFileId)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetRepositoryFile(repositoryId, repositoryFileId, includeExternalFilePermissionsInfo, result)`

Register a `ConnectApi.RepositoryFileDetail` object to be returned when the matching `getRepositoryFile(repositoryId, repositoryFileId, includeExternalFilePermissionsInfo)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetRepositoryFile(String repositoryId, String repositoryFileId,
Boolean includeExternalFilePermissionsInfo, ConnectApi.RepositoryFileDetail result)
```

Parameters

repositoryId
Type: String

The ID of the repository.

repositoryFileId
Type: String

The ID of the repository file.

includeExternalFilePermissionsInfo
Type: Boolean

Specifies whether to include permission information, such as whether the file is shared and what are the available permission types.

Managing external file permissions is supported for Google Drive, SharePoint Online, and OneDrive for Business.

result
Type: `ConnectApi.RepositoryFileDetail`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getRepositoryFile(repositoryId, repositoryFileId, includeExternalFilePermissionsInfo)

Apex Developer Guide: Testing ConnectApi Code

setTestGetRepositoryFile(communityId, repositoryId, repositoryFileId, result)

Register a `ConnectApi.RepositoryFileDetail` object to be returned when the matching `getRepositoryFile(communityId, repositoryId, repositoryFileId)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetRepositoryFile(String communityId, String repositoryId,
String repositoryFileId, ConnectApi.RepositoryFileDetail result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

repositoryId
Type: String

The ID of the repository.

repositoryFileId
Type: String

The ID of the repository file.

result
Type: `ConnectApi.RepositoryFileDetail`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getRepositoryFile(communityId, repositoryId, repositoryFileId)

Apex Developer Guide: Testing ConnectApi Code

setTestGetRepositoryFile(communityId, repositoryId, repositoryFileId, includeExternalFilePermissionsInfo, result)

Register a `ConnectApi.RepositoryFileDetail` object to be returned when the matching `getRepositoryFile(communityId, repositoryId, repositoryFileId, includeExternalFilePermissionsInfo)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetRepositoryFile(String communityId, String repositoryId,
String repositoryFileId, Boolean includeExternalFilePermissionsInfo,
ConnectApi.RepositoryFileDetail result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

repositoryId
Type: String

The ID of the repository.

repositoryFileId
Type: String

The ID of the repository file.

includeExternalFilePermissionsInfo
Type: Boolean

Specifies whether to include permission information, such as whether the file is shared and what are the available permission types.

Managing external file permissions is supported for Google Drive, SharePoint Online, and OneDrive for Business.

result
Type: ConnectApi.RepositoryFileDetail

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getRepositoryFile(communityId, repositoryId, repositoryFileId, includeExternalFilePermissionsInfo)

Apex Developer Guide: Testing ConnectApi Code

setTestGetRepositoryFolder(repositoryId, repositoryFolderId, result)

Register a `ConnectApi.RepositoryFolderDetail` object to be returned when the matching `getRepositoryFolder(repositoryId, repositoryFolderId)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetRepositoryFolder(String repositoryId, String
repositoryFolderId, ConnectApi.RepositoryFolderDetail result)
```

Parameters

repositoryId
Type: String

The ID of the repository.

repositoryFolderId
Type: String

The ID of the repository folder.

result
Type: ConnectApi.RepositoryFolderDetail

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getRepositoryFolder(repositoryId, repositoryFolderId)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetRepositoryFolder(communityId, repositoryId, repositoryFolderId, result)`

Register a `ConnectApi.RepositoryFolderDetail` object to be returned when the matching `getRepositoryFolder(communityId, repositoryId, repositoryFolderId)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetRepositoryFolder(String communityId, String repositoryId, String repositoryFolderId, ConnectApi.RepositoryFolderDetail result)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`repositoryId`
Type: String

The ID of the repository.

`repositoryFolderId`
Type: String

The ID of the repository folder.

`result`
Type: `ConnectApi.RepositoryFolderDetail`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getRepositoryFolder(communityId, repositoryId, repositoryFolderId)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetRepositoryFolderItems(repositoryId, repositoryFolderId, result)`

Register a `ConnectApi.RepositoryFolderItemsCollection` object to be returned when the matching `getRepositoryFolderItems(repositoryId, repositoryFolderId)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetRepositoryFolderItems(String repositoryId, String
repositoryFolderId, ConnectApi.RepositoryFolderItemsCollection result)
```

Parameters

repositoryId
Type: String

The ID of the repository.

repositoryFolderId
Type: String

The ID of the repository folder.

result
Type: `ConnectApi.RepositoryFolderItemsCollection`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getRepositoryFolderItems(repositoryId, repositoryFolderId)

Apex Developer Guide: Testing ConnectApi Code

setTestGetRepositoryFolderItems(communityId, repositoryId, repositoryFolderId, result)

Register a `ConnectApi.RepositoryFolderItemsCollection` object to be returned when the matching `getRepositoryFolderItems(communityId, repositoryId, repositoryFolderId)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetRepositoryFolderItems(String communityId, String
repositoryId, String repositoryFolderId, ConnectApi.RepositoryFolderItemsCollection
result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

repositoryId
Type: String

The ID of the repository.

repositoryFolderId
Type: String

The ID of the repository folder.

result
Type: `ConnectApi.RepositoryFolderItemsCollection`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getRepositoryFolderItems(communityId, repositoryId, repositoryFolderId)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetRepositoryFolderItems(repositoryId, repositoryFolderId, pageParam, pageSize, result)`

Register a `ConnectApi.RepositoryFolderItemsCollection` object to be returned when the matching `getRepositoryFolderItems(repositoryId, repositoryFolderId, pageParam, pageSize)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetRepositoryFolderItems(String repositoryId, String
repositoryFolderId, Integer pageParam, Integer pageSize,
ConnectApi.RepositoryFolderItemsCollection result)
```

Parameters

repositoryId
Type: String

The ID of the repository.

repositoryFolderId
Type: String

The ID of the repository folder.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default page size is 25.

result
Type: `ConnectApi.RepositoryFolderItemsCollection`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getRepositoryFolderItems(repositoryId, repositoryFolderId, pageParam, pageSize)

Apex Developer Guide: Testing ConnectApi Code

setTestGetRepositoryFolderItems(communityId, repositoryId, repositoryFolderId, pageParam, pageSize, result)

Register a `ConnectApi.RepositoryFolderItemsCollection` object to be returned when the matching `getRepositoryFolderItems(communityId, repositoryId, repositoryFolderId, pageParam, pageSize)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestGetRepositoryFolderItems(String communityId, String
repositoryId, String repositoryFolderId, Integer pageParam, Integer pageSize,
ConnectApi.RepositoryFolderItemsCollection result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

repositoryId
Type: String

The ID of the repository.

repositoryFolderId
Type: String

The ID of the repository folder.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default page size is 25.

result
Type: `ConnectApi.RepositoryFolderItemsCollection`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getRepositoryFolderItems(communityId, repositoryId, repositoryFolderId, pageParam, pageSize)

Apex Developer Guide: Testing ConnectApi Code

setTestUpdateRepositoryFile(communityId, repositoryId, repositoryFileId, file, fileData, result)

Register a `ConnectApi.RepositoryFileDetail` object to be returned when the matching `updateRepositoryFile(communityId, repositoryId, repositoryFileId, file, fileData)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestUpdateRepositoryFile(String communityId, String repositoryId, String repositoryFileId, ConnectApi.ContentHubItemInput file, ConnectApi.BinaryInput fileData, ConnectApi.RepositoryFileDetail result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

repositoryId
Type: String

The ID of the repository.

repositoryFileId
Type: String

The ID of the repository file.

file
Type: `ConnectApi.ContentHubItemInput`

The item type ID and fields of the item type.

fileData
Type: `ConnectApi.BinaryInput`

The binary file.

result
Type: `ConnectApi.RepositoryFileDetail`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

updateRepositoryFile(repositoryId, repositoryFileId, file)

Apex Developer Guide: Testing ConnectApi Code

`setTestUpdateRepositoryFile(repositoryId, repositoryFileId, file, result)`

Register a `ConnectApi.RepositoryFileDetail` object to be returned when the matching `updateRepositoryFile(repositoryId, repositoryFileId, file)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestUpdateRepositoryFile(String repositoryId, String
repositoryFileId, ConnectApi.ContentHubItemInput file, ConnectApi.RepositoryFileDetail
result)
```

Parameters

repositoryId
Type: String

The ID of the repository.

repositoryFileId
Type: String

The ID of the repository file.

file
Type: ConnectApi.ContentHubItemInput

The item type ID and fields of the item type.

result
Type: ConnectApi.RepositoryFileDetail

Object containing test data.

Return Value

Type: Void

SEE ALSO:

updateRepositoryFile(repositoryId, repositoryFileId, file, fileData)

Apex Developer Guide: Testing ConnectApi Code

setTestUpdateRepositoryFile(repositoryId, repositoryFileId, file, fileData, result)

Register a `ConnectApi.RepositoryFileDetail` object to be returned when the matching `updateRepositoryFile(repositoryId, repositoryFileId, file, fileData)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestUpdateRepositoryFile(String repositoryId, String
repositoryFileId, ConnectApi.ContentHubItemInput file, ConnectApi.BinaryInput fileData,
ConnectApi.RepositoryFileDetail result)
```

Parameters

repositoryId
Type: String

The ID of the repository.

repositoryFileId
Type: String

The ID of the repository file.

file
Type: ConnectApi.ContentHubItemInput

The item type ID and fields of the item type.

fileData
Type: ConnectApi.BinaryInput

The binary file.

result
Type: ConnectApi.RepositoryFileDetail

Object containing test data.

Return Value

Type: Void

SEE ALSO:

updateRepositoryFile(communityId, repositoryId, repositoryFileId, file)

Apex Developer Guide: Testing ConnectApi Code

setTestUpdateRepositoryFile(communityId, repositoryId, repositoryFileId, file, result)

Register a `ConnectApi.RepositoryFileDetail` object to be returned when the matching `updateRepositoryFile(communityId, repositoryId, repositoryFileId, file)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

40.0

Signature

```
public static Void setTestUpdateRepositoryFile(String communityId, String repositoryId,
String repositoryFileId, ConnectApi.ContentHubItemInput file,
ConnectApi.RepositoryFileDetail result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

repositoryId
Type: String

The ID of the repository.

repositoryFileId
Type: String

The ID of the repository file.

file
Type: ConnectApi.ContentHubItemInput

The item type ID and fields of the item type.

result
Type: ConnectApi.RepositoryFileDetail

Object containing test data.

Return Value

Type: Void

SEE ALSO:

updateRepositoryFile(communityId, repositoryId, repositoryFileId, file, fileData)

Apex Developer Guide: Testing ConnectApi Code

ConversationApplicationDefinition Class

Access information about a conversation application definition.

Namespace

ConnectApi

ConversationApplicationDefinition Methods

These methods are for `ConversationApplicationDefinition`. All methods are static.

IN THIS SECTION:

getConversationApplicationDefinition(integrationName)
Get information about an integration's conversation application definition and the associated bot.

`getConversationApplicationDefinition(integrationName)`

Get information about an integration's conversation application definition and the associated bot.

API Version

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.ConversationApplicationDefinitionDetailRespresentation
getConversationApplicationDefinition(String integrationName)
```

Parameters

integrationName
Type: String

Name of the conversation application.

Return Value

Type: `ConnectApi.ConversationApplicationDefinitionDetailRespresentation`

Usage

To access this method, enable the bot feature, and the user must be an admin or have the Manage Bots or Manage Bots Training Data user permissions.

Datacloud Class

Purchase Data.com contact or company records, and retrieve purchase information.

Namespace

ConnectApi

Note: When your Data.com Prospector or Data.com Clean contract expires, Data.com features, objects, and fields will be removed from your org.

To support customers' needs around compliance and to remain a leader in trust and privacy, Salesforce removed all contact data from the Data.com service on February 1, 2021.

For more information, see Data.com Prospector and Clean Retirement.

IN THIS SECTION:

Datacloud Methods
These methods are for `Datacloud`. All methods are static.

Datacloud Methods

These methods are for `Datacloud`. All methods are static.

IN THIS SECTION:

getCompaniesFromOrder(orderId, pageSize, page)
Get a list of purchased company records for an order.

getCompany(companyId)
Get a company record.

getContact(contactId)
Get a contact.

getContactsFromOrder(orderId, page, pageSize)
Get a list of purchased contacts for an order.

getOrder(orderId)
Get an order.

getUsage(userId)
Get purchase usage information for a user.

postOrder(orderInput)
Purchase records that are listed in an input file.

getCompaniesFromOrder(orderId, pageSize, page)

Get a list of purchased company records for an order.

API Version

32.0

Requires Chatter

No

Signature

```
public static ConnectApi.DatacloudCcompanies getCompaniesFromOrder(String orderId, String pageSize, String page)
```

Parameters

orderId
Type: String

ID of an order.

page
Type: Integer

Number of the page that you want returned.

pageSize
Type: Integer

Number of companies to show on a page. The default *pageSize* is 25.

Return Value

Type: ConnectApi.DatacloudCompanies

getCompany(companyId)

Get a company record.

API Version

32.0

Requires Chatter

No

Signature

```
public static ConnectApi.DatacloudCompany getCompany(String companyId)
```

Parameters

companyId
Type: String

ID of a company in the Data.com database.

Return Value

Type: ConnectApi.DatacloudCompany

getContact(contactId)

Get a contact.

API Version

32.0

Requires Chatter

No

Signature

```
public static ConnectApi.DatacloudContact getContact(String contactId)
```

Parameters

contactId
Type: String

ID of a contact in the Data.com database.

Return Value

Type: ConnectApi.DatacloudContact

getContactsFromOrder(orderId, page, pageSize)

Get a list of purchased contacts for an order.

API Version

32.0

Requires Chatter

No

Signature

```
public static ConnectApi.DataCloudContacts getContactsFromOrder(String orderId, String page, String pageSize)
```

Parameters

orderId
Type: String

ID of an order.

page
Type: Integer

Number of the page that you want returned.

pageSize
Type: Integer

Number of contacts to show on a page. The default *pageSize* is 25.

Return Value

Type: `ConnectApi.DataCloudContacts`

getOrder(orderId)

Get an order.

API Version

32.0

Requires Chatter

No

Signature

```
public static ConnectApi.DataCloudOrder getOrder(String orderId)
```

Parameters

orderId
Type: String

ID of an order.

Return Value

Type: ConnectApi.DatacloudOrder

getUsage(userId)

Get purchase usage information for a user.

API Version

32.0

Requires Chatter

No

Signature

```
public static ConnectApi.DatacloudPurchaseUsage getUsage(String userId)
```

Parameters

userId
Type: String

ID of a user.

Return Value

Type: ConnectApi.DatacloudPurchaseUsage

postOrder(orderInput)

Purchase records that are listed in an input file.

API Version

32.0

Requires Chatter

No

Signature

```
public static ConnectApi.DatacloudOrder postOrder(ConnectApi.DatacloudOrderInput
orderInput)
```

Parameters

orderInput
Type: ConnectApi.DatacloudOrderInput

A list that contains IDs for the contacts or companies that you want to see.

Return Value

Type: ConnectApi.DatacloudOrder

Example

```
ConnectApi.DatacloudOrderInput inputOrder=new ConnectApi.DatacloudOrderInput();
List<String> ids=new List<String>();
ids.add('1234');
inputOrder.companyIds=ids;
ConnectApi.DatacloudOrder datacloudOrderRep = ConnectApi.Datacloud.postOrder(inputOrder);
```

EinsteinLLM Class

Get a list of prompt templates and generate LLM responses for prompt templates.

Namespace

ConnectApi

EinsteinLLM Methods

These methods are for `EinsteinLLM`. All methods are static.

IN THIS SECTION:

generateMessagesForPromptTemplate(promptTemplateDevName, promptTemplateGenerationsInput)
Generates a response using the specified prompt template and input parameters.

getPromptTemplates(query, sortBy, offset, pageLimit, fields, type, relatedEntity, isActive)
Get a list of prompt templates using the specified filters.

`generateMessagesForPromptTemplate(promptTemplateDevName, promptTemplateGenerationsInput)`

Generates a response using the specified prompt template and input parameters.

API Version

60.0

Requires Chatter

No

Signature

```
public static ConnectApi.EinsteinPromptTemplateGenerationsRepresentation
generateMessagesForPromptTemplate(String promptTemplateDevName,
ConnectApi.EinsteinPromptTemplateGenerationsInput promptTemplateGenerationsInput)
```

Parameters

promptTemplateDevName
Type: String

Developer name or ID of a prompt template record.

promptTemplateGenerationsInput
Type: ConnectApi.EinsteinPromptTemplateGenerationsInput

Input for generating a response using the specified prompt template.

Return Value

Type: ConnectApi.EinsteinPromptTemplateGenerationsRepresentation

Example

In this example, call `generateMessagesForPromptTemplate(promptTemplateDevName, promptTemplateGenerationsInput)` to resolve a Sales Email prompt template. For more examples, see Resolve a Prompt Template.

```
// Create input
ConnectApi.EinsteinPromptTemplateGenerationsInput promptGenerationsInput = new
ConnectApi.EinsteinPromptTemplateGenerationsInput();
promptGenerationsInput.isPreview = false;

// Build input map
Map<String,ConnectApi.WrappedValue> valueMap = new Map<String,ConnectApi.WrappedValue>();

Map<String, String> recipientEntityRecordIdMap = new Map<String, String>();
recipientEntityRecordIdMap.put('id', '00Qxx000002ToPvEAK');

Map<String, String> senderEntityRecordIdMap = new Map<String, String>();
senderEntityRecordIdMap.put('id', '005xx000001XiWLAA0');

ConnectApi.WrappedValue recipientEntityWrappedValue = new ConnectApi.WrappedValue();
recipientEntityWrappedValue.value = recipientEntityRecordIdMap;

ConnectApi.WrappedValue senderEntityWrappedValue = new ConnectApi.WrappedValue();
senderEntityWrappedValue.value = senderEntityRecordIdMap;
```

```
valueMap.put('Input:Account', recipientEntityWrappedValue);
valueMap.put('Input:Recipient', recipientEntityWrappedValue);
valueMap.put('Input:Sender', senderEntityWrappedValue);

promptGenerationsInput.inputParams = valueMap;

// Set additional configuration values
promptGenerationsInput.additionalConfig = new ConnectApi.EinsteinLlmAdditionalConfigInput();
promptGenerationsInput.additionalConfig.applicationName =
'PromptTemplateGenerationsInvocable';

// Call the service
ConnectApi.EinsteinPromptTemplateGenerationsRepresentation generationsOutput =
ConnectApi.EinsteinLLM.generateMessagesForPromptTemplate('0hfxx0000000KTNAA2',
promptGenerationsInput);

// Consume response
System.debug('Prompt Testing: ' + generationsOutput.prompt);
```

`getPromptTemplates(query, sortBy, offset, pageLimit, fields, type, relatedEntity, isActive)`

Get a list of prompt templates using the specified filters.

API Version

62.0

Requires Chatter

No

Signature

```
public static ConnectApi.EinsteinPromptRecordCollectionOutputRepresentation
getPromptTemplates(String query, String sortBy, Integer offset, Integer pageLimit,
List<String> fields, String type, String relatedEntity, Boolean isActive)
```

Parameters

query
Type: String

User-entered search string. If `null`, all prompt template records are returned.

sortBy
Type: String

Field to sort the returned prompt template records by, such as `createdDate`. If `null`, records are returned in the order they're retrieved.

offset
Type: Integer

Used for pagination. Number of rows to skip between returned prompt template records. The default value is `0`.

`pageLimit`
Type: Integer

Used for pagination. Maximum number of prompt template records returned per page. The default value is `50`.

`fields`
Type: List<String>

Comma-separated list of prompt template record fields to return, such as `createdDate`. If `null`, all fields are returned.

`type`
Type: String

Prompt template type to filter records by, such as `einstein_gpt__salesEmail`. If `null`, records of all types are returned.

`relatedEntity`
Type: String

Related entity to filter records by, such as `Contact`. If `null`, all records with all related entities are returned.

`isActive`
Type: Boolean

Specifies whether to return active prompt templates only. The default is `false`.

Return Value

Type: `ConnectApi.EinsteinPromptRecordCollectionOutputRepresentation`

Usage

To get a list of prompt templates, you must have Einstein Generative AI enabled and the Execute Prompt Templates user permission.

Example

```
ConnectApi.EinsteinPromptRecordCollectionOutputRepresentation promptTemplateList =
ConnectApi.EinsteinLLM.getPromptTemplates('Summarize', 'CreatedDate', 0, 5, null,
'einstein_gpt__flex', null, true);

// Get information from the prompt templates in the list
    if (promptTemplateList != null && promptTemplateList.promptRecords != null) {
    for (ConnectApi.EinsteinPromptRecordRepresentation promptTemplate :
promptTemplateList.promptRecords) {
        System.debug('Prompt Template ID: ' + promptTemplate.id);
        for (String fieldName : promptTemplate.fields.keySet()) {
            System.debug('Field Name: ' + fieldName + ', Value: ' +
promptTemplate.fields.get(fieldName).value);
        }
    }
} else {
    System.debug('No prompt templates found.');
}
```

EmailMergeFieldService Class

Extract a list of merge fields for an object. A merge field is a field you can put in an email template, mail merge template, custom link, or formula to incorporate values from a record.

Namespace

ConnectApi

EmailMergeFieldService Methods

These methods are for `EmailMergeFieldService`. All methods are static.

IN THIS SECTION:

getMergeFields(objectApiNames)
Extract the merge fields for a specific object.

`getMergeFields(objectApiNames)`

Extract the merge fields for a specific object.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.EmailMergeFieldInfo getMergeFields(List<String> objectApiNames)
```

Parameters

`objectApiNames`
Type: List<String>

The API names for the objects being referenced.

Return Value

Type: `ConnectApi.EmailMergeFieldInfo`

EmployeeProfiles Class

Get, set and crop, and delete employee banner photos and photos.

Namespace

ConnectApi

EmployeeProfiles Methods

These methods are for `EmployeeProfiles`. All methods are static.

IN THIS SECTION:

deleteBannerPhoto(employeeId)
Delete an employee's banner photo.

deletePhoto(employeeId)
Delete an employee's photo.

getBannerPhoto(employeeId)
Get an employee's banner photo.

getPhoto(employeeId)
Get an employee's photo.

setBannerPhoto(employeeId, fileId, versionNumber)
Set an uploaded file as an employee's banner photo.

setBannerPhoto(employeeId, fileUpload)
Set a file that hasn't been uploaded as an employee's banner photo.

setBannerPhotoWithAttributes(employeeId, bannerPhoto)
Set and crop an uploaded file as an employee's banner photo.

setBannerPhotoWithAttributes(employeeId, bannerPhoto, fileUpload)
Set and crop a file that hasn't been uploaded as an employee's banner photo.

setPhoto(employeeId, fileId, versionNumber)
Set an uploaded file as an employee's photo.

setPhoto(employeeId, fileUpload)
Set a file that hasn't been uploaded as an employee's photo.

setPhotoWithAttributes(employeeId, photo)
Set and crop an uploaded file as an employee's photo.

setPhotoWithAttributes(employeeId, photo, fileUpload)
Set and crop a file that hasn't been uploaded as an employee's photo.

`deleteBannerPhoto(employeeId)`

Delete an employee's banner photo.

API Version

51.0

Requires Chatter

No

Signature

```
public static Void deleteBannerPhoto(String employeeId)
```

Parameters

employeeId
Type: String

ID of the employee.

Return Value

Type: Void

deletePhoto(employeeId)

Delete an employee's photo.

API Version

51.0

Requires Chatter

No

Signature

```
public static Void deletePhoto(String employeeId)
```

Parameters

employeeId
Type: String

ID of the employee.

Return Value

Type: Void

getBannerPhoto(employeeId)

Get an employee's banner photo.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.BannerPhoto getBannerPhoto(String employeeId)
```

Parameters

employeeId
Type: String

ID of the employee.

Return Value

Type: ConnectApi.BannerPhoto

getPhoto(employeeId)

Get an employee's photo.

API Version

51.0

Available to Guest Users

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.Photo getPhoto(String employeeId)
```

Parameters

employeeId
Type: String

ID of the employee.

Return Value

Type: ConnectApi.Photo

setBannerPhoto(employeeId, fileId, versionNumber)

Set an uploaded file as an employee's banner photo.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.BannerPhoto setBannerPhoto(String employeeId, String fileId,
Integer versionNumber)
```

Parameters

employeeId
Type: String

ID of the employee.

fileId
Type: String

ID of the uploaded file to use as the employee banner photo. The file must be an image and be smaller than 2 GB.

versionNumber
Type: Integer

Version number of the file. Specify an existing version number or, to get the latest version, specify `null`.

Return Value

Type: `ConnectApi.BannerPhoto`

setBannerPhoto(employeeId, fileUpload)

Set a file that hasn't been uploaded as an employee's banner photo.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.BannerPhoto setBannerPhoto(String employeeId,
ConnectApi.BinaryInput fileUpload)
```

Parameters

employeeId
Type: String

ID of the employee.

fileUpload
Type: ConnectApi.BinaryInput

File to use as the photo. The content type must be usable as an image.

Return Value

Type: ConnectApi.BannerPhoto

Usage

Photos are processed asynchronously and might not be visible right away.

setBannerPhotoWithAttributes(employeeId, bannerPhoto)

Set and crop an uploaded file as an employee's banner photo.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.BannerPhoto setBannerPhotoWithAttributes(String employeeId,
ConnectApi.BannerPhotoInput bannerPhoto)
```

Parameters

employeeId
Type: String

ID of the employee.

bannerPhoto
Type: ConnectApi.BannerPhotoInput

A `ConnectApi.BannerPhotoInput` object that specifies the ID and version of the file, and how to crop the file.

Return Value

Type: ConnectApi.BannerPhoto

Usage

Photos are processed asynchronously and might not be visible right away.

setBannerPhotoWithAttributes(employeeId, bannerPhoto, fileUpload)

Set and crop a file that hasn't been uploaded as an employee's banner photo.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.BannerPhoto setBannerPhotoWithAttributes(String employeeId,
ConnectApi.BannerPhotoInput bannerPhoto, ConnectApi.BinaryInput fileUpload)
```

Parameters

employeeId
Type: String

ID of the employee.

bannerPhoto
Type: ConnectApi.BannerPhotoInput

A `ConnectApi.BannerPhotoInput` object specifying the cropping parameters.

fileUpload
Type: ConnectApi.BinaryInput

File to use as the photo. The content type must be usable as an image.

Return Value

Type: ConnectApi.BannerPhoto

Usage

Photos are processed asynchronously and might not be visible right away.

setPhoto(employeeId, fileId, versionNumber)

Set an uploaded file as an employee's photo.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.Photo setPhoto(String employeeId, String fileId, Integer versionNumber)
```

Parameters

employeeId
Type: String

ID of the employee.

fileId
Type: String

ID of the uploaded file to use as the employee photo. The file must be an image and be smaller than 2 GB.

versionNumber
Type: Integer

Version number of the file. Specify an existing version number or, to get the latest version, specify `null`.

Return Value

Type: `ConnectApi.Photo`

Usage

Photos are processed asynchronously and might not be visible right away.

setPhoto(employeeId, fileUpload)

Set a file that hasn't been uploaded as an employee's photo.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.Photo setPhoto(String employeeId, ConnectApi.BinaryInput fileUpload)
```

Parameters

employeeId
Type: String

ID of the employee.

fileUpload
Type: ConnectApi.BinaryInput

File to use as the photo. The content type must be usable as an image.

Return Value

Type: ConnectApi.Photo

Usage

Photos are processed asynchronously and might not be visible right away.

setPhotoWithAttributes(employeeId, photo)

Set and crop an uploaded file as an employee's photo.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.Photo setPhotoWithAttributes(String employeeId,
ConnectApi.PhotoInput photo)
```

Parameters

employeeId
Type: String

ID of the employee.

photo
Type: ConnectApi.PhotoInput

A `ConnectApi.PhotoInput` object specifying the file ID, version number, and cropping parameters.

Return Value

Type: ConnectApi.Photo

Usage

Photos are processed asynchronously and might not be visible right away.

`setPhotoWithAttributes(employeeId, photo, fileUpload)`

Set and crop a file that hasn't been uploaded as an employee's photo.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.Photo setPhotoWithAttributes(String employeeId,
ConnectApi.PhotoInput photo, ConnectApi.BinaryInput fileUpload)
```

Parameters

employeeId
Type: String

ID of the employee.

photo
Type: `ConnectApi.PhotoInput`

A `ConnectApi.PhotoInput` object specifying the cropping parameters.

fileUpload
Type: `ConnectApi.BinaryInput`

File to use as the photo. The content type must be usable as an image.

Return Value

Type: `ConnectApi.Photo`

Usage

Photos are processed asynchronously and might not be visible right away.

Exchanges Class

Preview and submit cart to exchange orders.

Namespace

ConnectApi

Exchanges Methods

These methods are for `Exchanges`. All methods are static.

IN THIS SECTION:

previewCartToExchangeOrder(previewCartToExchangeOrderInput)

Retrieves a preview of an exchange order, taking into account the order summary balance and the difference between the return order and the cart that represents the exchange order.

submitCartToExchangeOrder(submitCartToExchangeOrderInput)

Creates an exchange order summary, based on the return order and the cart used for exchanges. The new exchange order summary is attached to the original order summary (created before any exchanges occurred). You can also provide optional payment information and optional information about order summary sequences, which affect the newly created exchange order summary.

previewCartToExchangeOrder(previewCartToExchangeOrderInput)

Retrieves a preview of an exchange order, taking into account the order summary balance and the difference between the return order and the cart that represents the exchange order.

API Version

60.0

Requires Chatter

No

Signature

```
public static ConnectApi.PreviewCartToExchangeOrderOutputRepresentation
previewCartToExchangeOrder(ConnectApi.PreviewCartToExchangeOrderInputRepresentation
previewCartToExchangeOrderInput)
```

Parameters

previewCartToExchangeOrderInput

Type: `ConnectApi.PreviewCartToExchangeOrderInputRepresentation` on page 1909

Information required to preview a cart to exchange order.

Return Value

Type: `ConnectApi.PreviewCartToExchangeOrderOutputRepresentation` on page 2233

submitCartToExchangeOrder(submitCartToExchangeOrderInput)

Creates an exchange order summary, based on the return order and the cart used for exchanges. The new exchange order summary is attached to the original order summary (created before any exchanges occurred). You can also provide optional payment information and optional information about order summary sequences, which affect the newly created exchange order summary.

API Version

60.0

Requires Chatter

No

Signature

```
public static ConnectApi.SubmitCartToExchangeOrderOutputRepresentation
submitCartToExchangeOrder(ConnectApi.SubmitCartToExchangeOrderInputRepresentation
submitCartToExchangeOrderInput)
```

Parameters

submitCartToExchangeOrderInput
Type: ConnectApi.SubmitCartToExchangeOrderInputRepresentation on page 1917

Information required to submit a cart to exchange order.

Return Value

Type: ConnectApi.SubmitCartToExchangeOrderOutputRepresentation on page 2299

ExtendedCommerceDelivery Class

Access information about delivery estimation.

Namespace

ConnectApi

ExtendedCommerceDelivery Methods

This method is for `ExtendedCommerceDelivery`. It is static.

IN THIS SECTION:

estimateDeliveryDate(estimateDeliveryDateInput, externalReference)
Forecast an expected delivery date and time based on delivery estimation settings and the selected shipping carrier method. Provide information on when a package is expected to be shipped and delivered.

`estimateDeliveryDate(estimateDeliveryDateInput, externalReference)`

Forecast an expected delivery date and time based on delivery estimation settings and the selected shipping carrier method. Provide information on when a package is expected to be shipped and delivered.

API Version

63.0

Available to Guest Users

63.0

Requires Chatter

No

Signature

```
public static ConnectApi.EstimateDeliveryDateOutputRepresentation
estimateDeliveryDate(ConnectApi.EstimateDeliveryDateInputRepresentation
estimateDeliveryDateInput, String externalReference)
```

Parameters

estimateDeliveryDateInput
Type: Datetime

ConnectApi.EstimateDeliveryDateInputRepresentation on page 1855

Estimated delivery date.

externalReference
Type: String

Delivery estimation setup external reference ID.

Return Value

Type: ConnectApi.EstimateDeliveryDateOutputRepresentation on page 2084

ExternalEmailServices Class

Access information about integration with external email services, such as sending email within Salesforce through an external email account.

Namespace

ConnectApi

ExternalEmailServices Methods

These methods are for `ExternalEmailService`. All methods are static.

IN THIS SECTION:

getUserOauthInfo(landingPage)
Get information about whether an external email service has been authorized to send email on behalf of a user.

`getUserOauthInfo(landingPage)`

Get information about whether an external email service has been authorized to send email on behalf of a user.

API Version

37.0

Requires Chatter

No

Signature

```
public static getUserOauthInfo(String landingPage)
```

Parameters

landingPage
Type: String

The landing page that the user starts on when they are finished with the OAuth authorization process.

Return Value

Type: `ConnectApi.UserOauthInfo`

SEE ALSO:

Apex Developer Guide: Testing ConnectApi Code

ExternalManagedAccount Class

Get externally managed accounts.

Namespace

ConnectApi

ExternalManagedAccount Methods

These methods are for `ExternalManagedAccount`. All methods are static.

IN THIS SECTION:

getCommunitiesExternalManagedAccounts(communityId)
Get externally managed accounts available to the context user across all Experience Cloud sites.

getCommunitiesExternalManagedAccounts(communityId, includeMyAccount)
Get externally managed accounts available to the context user, including the context user's account, across all Experience Cloud sites.

getExternalManagedAccounts(webstoreId)
Get externally managed accounts for a store.

getExternalManagedAccounts(webstoreId, includeMyAccount)
Get externally managed accounts, including the context user's account, for a store.

getCommunitiesExternalManagedAccounts(communityId)

Get externally managed accounts available to the context user across all Experience Cloud sites.

API Version

50.0

Requires Chatter

No

Signature

```
public static ConnectApi.ExternalManagedAccountCollectionOutput
getCommunitiesExternalManagedAccounts(String communityId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

Note: Regardless of the ID specified, this method returns externally managed accounts available to the context user across all Experience Cloud sites.

Return Value

Type: `ConnectApi.ExternalManagedAccountCollectionOutput`

getCommunitiesExternalManagedAccounts(communityId, includeMyAccount)

Get externally managed accounts available to the context user, including the context user's account, across all Experience Cloud sites.

API Version

53.0

Requires Chatter

No

Signature

```
public static ConnectApi.ExternalManagedAccountCollectionOutput
getCommunitiesExternalManagedAccounts(String communityId, Boolean includeMyAccount)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

Note: Regardless of the ID specified, this method returns externally managed accounts available to the context user across all Experience Cloud sites.

`includeMyAccount`
Type: Boolean

Specifies whether to return the context user's account (`true`) or not (`false`). The default value is `false`.

Return Value

Type: `ConnectApi.ExternalManagedAccountCollectionOutput`

`getExternalManagedAccounts(webstoreId)`

Get externally managed accounts for a store.

API Version

49.0

Requires Chatter

No

Signature

```
public static ConnectApi.ExternalManagedAccountCollectionOutput
getExternalManagedAccounts(String webstoreId)
```

Parameters

`webstoreId`
Type: String

ID of the webstore.

Return Value

Type: `ConnectApi.ExternalManagedAccountCollectionOutput`

`getExternalManagedAccounts(webstoreId, includeMyAccount)`

Get externally managed accounts, including the context user's account, for a store.

API Version

53.0

Requires Chatter

No

Signature

```
public static ConnectApi.ExternalManagedAccountCollectionOutput
getExternalManagedAccounts(String webstoreId, Boolean includeMyAccount)
```

Parameters

webstoreId
Type: String

ID of the webstore.

includeMyAccount
Type: Boolean

Specifies whether to return the context user's account (`true`) or not (`false`). The default value is `false`.

Return Value

Type: `ConnectApi.ExternalManagedAccountCollectionOutput`

FieldService Class

Preview and create shifts from a pattern or filter fields on recordset filter criteria.

Namespace

ConnectApi

FieldService Methods

These methods are for `FieldService`. All methods are static.

IN THIS SECTION:

createShiftsFromPattern(shiftsFromPatternInput, shiftPatternId)
Create up to 2,000 shifts from a pattern.

evaluateRecordsetFilterCriteria(recordsetFilterCriteriaInput)
Filter records on recordset filter criteria.

previewShiftsFromPattern(shiftsFromPatternInput, shiftPatternId)
Preview up to 2,000 shifts from a pattern.

`createShiftsFromPattern(shiftsFromPatternInput, shiftPatternId)`

Create up to 2,000 shifts from a pattern.

API Version

51.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ShiftsFromPattern
createShiftsFromPattern(ConnectApi.ShiftsFromPatternInput shiftsFromPatternInput, String
shiftPatternId)
```

Parameters

shiftsFromPatternInput
Type: ConnectApi.ShiftsFromPatternInput

A `ConnectApi.ShiftsFromPatternInput` object providing the pattern.

shiftPatternId
Type: String

ID of the shift pattern.

Return Value

Type: ConnectApi.ShiftsFromPattern

evaluateRecordsetFilterCriteria(recordsetFilterCriteriaInput)

Filter records on recordset filter criteria.

API Version

53.0

Requires Chatter

No

Signature

```
public static ConnectApi.RecordsetFilterCriteriaOutput
evaluateRecordsetFilterCriteria(ConnectApi.RecordsetFilterCriteriaInput
recordsetFilterCriteriaInput)
```

Parameters

recordsetFilterCriteriaInput
Type: ConnectApi.RecordsetFilterCriteriaInput

An `ConnectApi.RecordsetFilterCriteriaInput` object providing a set of recordset filter criteria and records.

Return Value

Type: ConnectApi.RecordsetFilterCriteriaOutput

Usage

Field service must be enabled.

`previewShiftsFromPattern(shiftsFromPatternInput, shiftPatternId)`

Preview up to 2,000 shifts from a pattern.

API Version

51.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ShiftsFromPattern
previewShiftsFromPattern(ConnectApi.ShiftsFromPatternInput shiftsFromPatternInput,
String shiftPatternId)
```

Parameters

`shiftsFromPatternInput`
Type: `ConnectApi.ShiftsFromPatternInput`

A `ConnectApi.ShiftsFromPatternInput` object providing the pattern.

`shiftPatternId`
Type: String

ID of the shift pattern.

Return Value

Type: `ConnectApi.ShiftsFromPattern`

FulfillmentOrder Class

Fulfill orders in Order Management.

Namespace

ConnectApi

FulfillmentOrder Methods

These methods are for `FulfillmentOrder`. All methods are static.

IN THIS SECTION:

cancelFulfillmentOrderLineItems(fulfillmentOrderId, cancelFulfillmentOrderLineItemsInput)
Cancel FulfillmentOrderLineItems from a FulfillmentOrder. This action doesn't cancel the associated OrderItemSummaries, so reallocate the canceled quantities to a new FulfillmentOrder.

createFulfillmentOrders(fulfillmentOrderInput)
Create one or more FulfillmentOrders and FulfillmentOrderLineItems for an OrderDeliveryGroupSummary, which defines a delivery method and recipient for an OrderSummary. You specify the OrderItemSummaries to allocate, which can be fulfilled from different locations. Specifying multiple fulfillment groups creates one FulfillmentOrder for each location. For each OrderItemSummary, a FulfillmentOrderLineItem is created and assigned to the corresponding FulfillmentOrder.

createInvoice(fulfillmentOrderId, invoiceInput)
Create an invoice for a FulfillmentOrder that doesn't have one.

createMultipleFulfillmentOrder(multipleFulfillmentOrderInput)
Create FulfillmentOrders for multiple OrderDeliveryGroups in a single request.

createMultipleInvoices(invoicesInput)
Create Invoices for multiple FulfillmentOrders.

cancelFulfillmentOrderLineItems(fulfillmentOrderId, cancelFulfillmentOrderLineItemsInput)

Cancel FulfillmentOrderLineItems from a FulfillmentOrder. This action doesn't cancel the associated OrderItemSummaries, so reallocate the canceled quantities to a new FulfillmentOrder.

API Version

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.FulfillmentOrderCancelLineItemsOutputRepresentation
cancelFulfillmentOrderLineItems(String fulfillmentOrderId,
ConnectApi.FulfillmentOrderLineItemsToCancelInputRepresentation
cancelFulfillmentOrderLineItemsInput)
```

Parameters

fulfillmentOrderId
Type: String

ID of the FulfillmentOrder.

cancelFulfillmentOrderLineItemsInput
Type: ConnectApi.FulfillmentOrderLineItemsToCancelInputRepresentation

List of FulfillmentOrderLineItems to cancel.

Return Value

Type: ConnectApi.FulfillmentOrderCancelLineItemsOutputRepresentation

Example

```
String fulfillmentOrderId = '0a3xx0000000085AAA';
List<ConnectApi.FulfillmentOrderLineItemInputRepresentation> itemToCancelList = new
List<ConnectApi.FulfillmentOrderLineItemInputRepresentation>();

for(FulfillmentOrderLineItem fulfillmentOrderLineItem :
fulfillmentOrder.FulfillmentOrderLineItems){
  ConnectApi.FulfillmentOrderLineItemInputRepresentation itemToCancel = new
ConnectApi.FulfillmentOrderLineItemInputRepresentation();
  itemToCancel.fulfillmentOrderLineItemId = fulfillmentOrderLineItem.Id;
  itemToCancel.quantity = 1;
  itemToCancelList.add(itemToCancel);
}

ConnectAPI.FulfillmentOrderLineItemsToCancelInputRepresentation input = new
ConnectAPI.FulfillmentOrderLineItemsToCancelInputRepresentation();
input.fulfillmentOrderLineItemsToCancel = itemToCancelList;

ConnectAPI.FulfillmentOrderCancelLineItemsOutputRepresentation result =
ConnectAPI.FulfillmentOrder.cancelFulfillmentOrderLineItems(fulfillmentOrderId, input);
```

createFulfillmentOrders(fulfillmentOrderInput)

Create one or more FulfillmentOrders and FulfillmentOrderLineItems for an OrderDeliveryGroupSummary, which defines a delivery method and recipient for an OrderSummary. You specify the OrderItemSummaries to allocate, which can be fulfilled from different locations. Specifying multiple fulfillment groups creates one FulfillmentOrder for each location. For each OrderItemSummary, a FulfillmentOrderLineItem is created and assigned to the corresponding FulfillmentOrder.

API Version

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.FulfillmentOrderOutputRepresentation
createFulfillmentOrders(ConnectApi.FulfillmentOrderInputRepresentation
fulfillmentOrderInput)
```

Parameters

fulfillmentOrderInput

Type: ConnectApi.FulfillmentOrderInputRepresentation

OrderItemSummaries to allocate, with location and delivery information.

Return Value

Type: ConnectApi.FulfillmentOrderOutputRepresentation

Example

```
String orderSummaryId = '1Osxx0000004CCG';
String fulfillmentType = 'warehouse';

String warehouseFromLocationId = [SELECT Id from Location WHERE LocationType='Warehouse'
LIMIT 1].Id;

ConnectApi.FulfillmentOrderInputRepresentation fulfillmentOrderInput = new
ConnectApi.FulfillmentOrderInputRepresentation();
fulfillmentOrderInput.orderSummaryId = orderSummaryId;

List<OrderDeliveryGroupSummary> orderDeliveryGroupSummaryList = [SELECT Id FROM
OrderDeliveryGroupSummary WHERE OrderSummaryId =: orderSummaryId];

for (OrderDeliveryGroupSummary orderDeliveryGroupSummary: orderDeliveryGroupSummaryList){


  fulfillmentOrderInput.orderDeliveryGroupSummaryId = orderDeliveryGroupSummary.Id;
  List<ConnectApi.FulfillmentGroupInputRepresentation> fulfillmentGroups = new
List<ConnectApi.FulfillmentGroupInputRepresentation>();
  ConnectApi.FulfillmentGroupInputRepresentation fulfillmentGroup = new
ConnectApi.FulfillmentGroupInputRepresentation();
  fulfillmentGroup.fulfilledFromLocationId = warehouseFromLocationId;
  fulfillmentGroup.fulfillmentType = fulfillmentType;

  List<ConnectApi.OrderItemSummaryInputRepresentation> orderItemSummaries = new
List<ConnectApi.OrderItemSummaryInputRepresentation>();

  List<OrderItemSummary> orderItemSummaryList = [Select Id, quantity FROM OrderItemSummary
 WHERE OrderSummaryId =: orderSummaryId AND OrderDeliveryGroupSummaryId =:
orderDeliveryGroupSummary.Id];
  for(OrderItemSummary orderItemSummary : orderItemSummaryList){
    ConnectApi.OrderItemSummaryInputRepresentation oisInputRepresentation = new
ConnectApi.OrderItemSummaryInputRepresentation();
    oisInputRepresentation.orderItemSummaryId = orderItemSummary.Id;
    oisInputRepresentation.quantity = orderItemSummary.quantity;
    orderItemSummaries.add(oisInputRepresentation);
  }

  fulfillmentGroup.orderItemSummaries = orderItemSummaries;
  fulfillmentGroups.add(fulfillmentGroup);
  fulfillmentOrderInput.fulfillmentGroups = fulfillmentGroups;

}

ConnectApi.FulfillmentOrderOutputRepresentation result =
ConnectAPI.FulfillmentOrder.createFulfillmentOrders(fulfillmentOrderInput);
```

createInvoice(fulfillmentOrderId, invoiceInput)

Create an invoice for a FulfillmentOrder that doesn't have one.

API Version

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.FulfillmentOrderInvoiceOutputRepresentation createInvoice(String
fulfillmentOrderId, ConnectApi.FulfillmentOrderInvoiceInputRepresentation invoiceInput)
```

Parameters

fulfillmentOrderId
Type: String

ID of the FulfillmentOrder.

invoiceInput
Type: ConnectApi.FulfillmentOrderInvoiceInputRepresentation

Required input with no data.

Return Value

Type: ConnectApi.FulfillmentOrderInvoiceOutputRepresentation

Example

```
String fulfillmentOrderId = '0a3xx0000000085AAA';

ConnectApi.FulfillmentOrderInvoiceInputRepresentation input = new
ConnectApi.FulfillmentOrderInvoiceInputRepresentation();
ConnectAPI.FulfillmentOrderInvoiceOutputRepresentation result =
ConnectApi.FulfillmentOrder.createInvoice(fulfillmentOrderId, input);
```

createMultipleFulfillmentOrder(multipleFulfillmentOrderInput)

Create FulfillmentOrders for multiple OrderDeliveryGroups in a single request.

API Version

50.0

Requires Chatter

No

Signature

```
public static ConnectApi.MultipleFulfillmentOrderOutputRepresentation
createMultipleFulfillmentOrder(ConnectApi.MultipleFulfillmentOrderInputRepresentation
multipleFulfillmentOrderInput)
```

Parameters

multipleFulfillmentOrderInput
Type: ConnectApi.MultipleFulfillmentOrderInputRepresentation

Wraps a list of inputs for creating fulfillment orders.

Return Value

Type: ConnectApi.MultipleFulfillmentOrderOutputRepresentation

createMultipleInvoices(invoicesInput)

Create Invoices for multiple FulfillmentOrders.

API Version

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.MultipleFulfillmentOrderInvoicesOutputRepresentation
createMultipleInvoices(ConnectApi.MultipleFulfillmentOrderInvoicesInputRepresentation
invoicesInput)
```

Parameters

invoicesInput
Type: ConnectApi.MultipleFulfillmentOrderInvoicesInputRepresentation

The FulfillmentOrders to create Invoices for.

Return Value

Type: ConnectApi.MultipleFulfillmentOrderInvoicesOutputRepresentation

Knowledge Class

Get information about trending articles in Experience Cloud sites.

Namespace

ConnectApi

Knowledge Methods

These methods are for `Knowledge`. All methods are static.

IN THIS SECTION:

getTopViewedArticlesForTopic(communityId, topicId, maxResults)
Get the top viewed articles for a topic.

getTrendingArticles(communityId, maxResults)
Get trending articles for an Experience Cloud site.

getTrendingArticlesForTopic(communityId, topicId, maxResults)
Get the trending articles for a topic in an Experience Cloud site.

`getTopViewedArticlesForTopic(communityId, topicId, maxResults)`

Get the top viewed articles for a topic.

API Version

41.0

Available to Guest Users

41.0

Requires Chatter

No

Signature

```
public static ConnectApi.KnowledgeArticleVersionCollection
getTopViewedArticlesForTopic(String communityId, String topicId, Integer maxResults)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`topicId`
Type: String

ID of the topic.

`maxResults`
Type: Integer

The maximum number of articles returned for each topic ID. Values can be from 1 to 25. The default value is 5.

Return Value

Type: `ConnectApi.KnowledgeArticleVersionCollection`

`getTrendingArticles(communityId, maxResults)`

Get trending articles for an Experience Cloud site.

API Version

36.0

Available to Guest Users

36.0

Requires Chatter

No

Signature

```
public static ConnectApi.KnowledgeArticleVersionCollection getTrendingArticles(String communityId, Integer maxResults)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`maxResults`
Type: Integer

The maximum number of articles returned. Values can be from 0 to 25. Default is 5.

Return Value

Type: `ConnectApi.KnowledgeArticleVersionCollection`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetTrendingArticles(communityId, maxResults, result)

Apex Developer Guide: Testing ConnectApi Code

`getTrendingArticlesForTopic(communityId, topicId, maxResults)`

Get the trending articles for a topic in an Experience Cloud site.

API Version

36.0

Available to Guest Users

36.0

Requires Chatter

No

Signature

```
public static ConnectApi.KnowledgeArticleVersionCollection
getTrendingArticlesForTopic(String communityId, String topicId, Integer maxResults)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`topicId`
Type: String

ID of the topic.

`maxResults`
Type: Integer

The maximum number of articles returned. Values can be from 0 to 25. Default is 5.

Return Value

Type: `ConnectApi.KnowledgeArticleVersionCollection`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetTrendingArticlesForTopic(communityId, topicId, maxResults, result)

Apex Developer Guide: Testing ConnectApi Code

Knowledge Test Methods

These test methods are for `Knowledge`. All methods are static.

For information about using these methods to test your `ConnectApi` code, see Testing ConnectApi Code.

`setTestGetTrendingArticles(communityId, maxResults, result)`

Register a `ConnectApi.KnowledgeVersionArticleCollection` object to be returned when the matching `ConnectApi.getTrendingArticles` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

36.0

Signature

```
public static Void setTestGetTrendingArticles(String communityId, Integer maxResults,
ConnectApi.KnowledgeArticleVersionCollection result)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`maxResults`
Type: Integer

The maximum number of articles returned. Values can be from 0 to 25. Default is 5.

`result`
Type: `ConnectApi.KnowledgeArticleVersionCollection`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getTrendingArticles(communityId, maxResults)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetTrendingArticlesForTopic(communityId, topicId, maxResults, result)`

Register a `ConnectApi.KnowledgeVersionArticleCollection` object to be returned when the matching `ConnectApi.getTrendingArticlesForTopic` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

36.0

Signature

```
public static Void setTestGetTrendingArticlesForTopic(String communityId, String topicId,
Integer maxResults, ConnectApi.KnowledgeArticleVersionCollection result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

topicId
Type: String

ID of the topic.

maxResults
Type: Integer

The maximum number of articles returned. Values can be from 0 to 25. Default is 5.

result
Type: `ConnectApi.KnowledgeArticleVersionCollection`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getTrendingArticlesForTopic(communityId, topicId, maxResults)

Apex Developer Guide: Testing ConnectApi Code

LightningScheduler Class

Create and update service appointments.

Namespace

ConnectApi

LightningScheduler Methods

These methods are for `LightningScheduler`. All methods are static.

IN THIS SECTION:

createServiceAppointment(createServiceAppointmentInput)
Create a service appointment.

updateServiceAppointment(updateServiceAppointmentInput)
Update a service appointment.

createServiceAppointment(createServiceAppointmentInput)

Create a service appointment.

API Version

53.0

Requires Chatter

No

Signature

```
public static ConnectApi.ServiceAppointmentOutput
createServiceAppointment(ConnectApi.CreateServiceAppointmentInput
createServiceAppointmentInput)
```

Parameters

createServiceAppointmentInput
Type: `ConnectApi.CreateServiceAppointmentInput`

Input parameters to create a service appointment.

Return Value

Type: `ConnectApi.ServiceAppointmentOutput`

Usage

Considerations for using engagement channel types with the `service-appointments` resource:

- Enable **Schedule Appointments Using Engagement Channels** in Salesforce Scheduler Settings in your Salesforce org.
- When you create or modify appointments, shifts must be defined in the scheduling policy. For more information on setting up shifts in the scheduling policy, see Define Shift Rules in Scheduling Policy.

  Note: Engagement channel types are not supported with operating hours rules in the scheduling policy.

- When you use engagement channel type and shifts to create a service appointment, Salesforce Scheduler considers the default value for the Appointment Type (if not specified). However, Salesforce Scheduler only considers the engagement channel type and Appointment Type is ignored.

Example

For an account (existing user):

```
ConnectApi.ExtendedFieldInput extendedFieldEmail = new ConnectApi.ExtendedFieldInput();
extendedFieldEmail.name = 'Email';
extendedFieldEmail.value = 'rachael.adams@salesforce.com';

ConnectApi.ExtendedFieldInput extendedFieldPhone = new ConnectApi.ExtendedFieldInput();
extendedFieldPhone.name = 'Phone';
extendedFieldPhone.value = '1234567890';

List<ConnectApi.ExtendedFieldInput> extendedFieldList = new
List<ConnectApi.ExtendedFieldInput>();
extendedFieldList.add(extendedFieldEmail);
extendedFieldList.add(extendedFieldPhone);

ConnectApi.ServiceAppointmentInput serviceAppInput = new
ConnectApi.ServiceAppointmentInput();
serviceAppInput.extendedFields = extendedFieldList;
serviceAppInput.engagementChannelTypeId = '0eFRM00000000Bv2AI';
serviceAppInput.serviceTerritoryId = '0Hhxx0000004C92CAE';
serviceAppInput.workTypeId = '08qxx0000004C92AAE';
serviceAppInput.parentRecordId = '001xx000003GYR1AAO';
serviceAppInput.schedStartTime = DateTime.valueOf('2021-05-28 12:15:00');
serviceAppInput.schedEndTime = DateTime.valueOf('2021-05-28 12:45:00');
serviceAppInput.appointmentMode = 'Group';
serviceAppInput.attendeeLimit = 20;

ConnectApi.AssignedResourcesInput asResourceInput = new ConnectApi.AssignedResourcesInput();
asResourceInput.serviceResourceId = '0Hnxx0000004CAiCAM';
asResourceInput.isRequiredResource = true;
asResourceInput.isPrimaryResource = true;

List<ConnectApi.AssignedResourcesInput> asResourceInputList = new
List<ConnectApi.AssignedResourcesInput>();
asResourceInputList.add(asResourceInput);

ConnectApi.CreateServiceAppointmentInput createInput = new
ConnectApi.CreateServiceAppointmentInput();
createInput.serviceAppointment = serviceAppInput;
createInput.assignedResources = asResourceInputList;

try{
   ConnectApi.ServiceAppointmentOutput appointmentResult =
ConnectApi.LightningScheduler.createServiceAppointment(createInput);
   String serviceAppointmentId = appointmentResult.result.serviceAppointmentId;
   List<String> assignedResourceIds = appointmentResult.result.assignedResourceIds;
}catch(ConnectApi.ConnectApiException ex){
   //Handle Exception
}
```

For a lead (authenticated guest user):

```
ConnectApi.LeadInput leadInput = new ConnectApi.LeadInput();
leadInput.firstName = 'Rachel';
leadInput.lastName = 'Adams';
leadInput.phone = '012-345-6789';
leadInput.email = 'rachel.adams@salesforce.com';
leadInput.company = 'Salesforce';

ConnectApi.ExtendedFieldInput extendedFieldEmail = new ConnectApi.ExtendedFieldInput();
extendedFieldEmail.name = 'Email';
extendedFieldEmail.value = 'rachael.adams@salesforce.com';

ConnectApi.ExtendedFieldInput extendedFieldPhone = new ConnectApi.ExtendedFieldInput();
extendedFieldPhone.name = 'Phone';
extendedFieldPhone.value = '1234567890';

List<ConnectApi.ExtendedFieldInput> extendedFieldList = new
List<ConnectApi.ExtendedFieldInput>();
extendedFieldList.add(extendedFieldEmail);
extendedFieldList.add(extendedFieldPhone);

ConnectApi.ServiceAppointmentInput serviceAppInput = new
ConnectApi.ServiceAppointmentInput();
serviceAppInput.extendedFields = extendedFieldList;
serviceAppInput.engagementChannelTypeId = '0eFRM00000000Bv2AI';
serviceAppInput.serviceTerritoryId = '0Hhxx0000004C92CAE';
serviceAppInput.workTypeId = '08qxx0000004C92AAE';
serviceAppInput.schedStartTime = DateTime.valueOf('2021-05-28 12:15:00');
serviceAppInput.schedEndTime = DateTime.valueOf('2021-05-28 12:45:00');

ConnectApi.AssignedResourcesInput asResourceInput = new ConnectApi.AssignedResourcesInput();
asResourceInput.serviceResourceId = '0Hnxx0000004CAiCAM';
asResourceInput.isRequiredResource = true;
asResourceInput.isPrimaryResource = true;

List<ConnectApi.AssignedResourcesInput> asResourceInputList = new
List<ConnectApi.AssignedResourcesInput>();
asResourceInputList.add(asResourceInput);

ConnectApi.CreateServiceAppointmentInput createInput = new
ConnectApi.CreateServiceAppointmentInput();
createInput.serviceAppointment = serviceAppInput;
createInput.assignedResources = asResourceInputList;
createInput.lead = leadInput;

try{
   ConnectApi.ServiceAppointmentOutput appointmentResult =
ConnectApi.LightningScheduler.createServiceAppointment(createInput);
   String serviceAppointmentId = appointmentResult.result.serviceAppointmentId;
   List<String> assignedResourceIds = appointmentResult.result.assignedResourceIds;
}catch(ConnectApi.ConnectApiException ex){
```

```
    //Handle Exception
}
```

SEE ALSO:

Service Appointments

updateServiceAppointment(updateServiceAppointmentInput)

Update a service appointment.

API Version

53.0

Requires Chatter

No

Signature

```
public static ConnectApi.ServiceAppointmentOutput
updateServiceAppointment(ConnectApi.UpdateServiceAppointmentInput
updateServiceAppointmentInput)
```

Parameters

updateServiceAppointmentInput

Type: ConnectApi.UpdateServiceAppointmentInput

Input parameters to update a service appointment.

Return Value

Type: ConnectApi.ServiceAppointmentOutput

Usage

Considerations for using engagement channel types with the `service-appointments` resource:

- Enable **Schedule Appointments Using Engagement Channels** in Salesforce Scheduler Settings in your Salesforce org.
- When you create or modify appointments, shifts must be defined in the scheduling policy. For more information on setting up shifts in the scheduling policy, see Define Shift Rules in Scheduling Policy.

  Note: Engagement channel types are not supported with operating hours rules in the scheduling policy.

- When you use engagement channel type and shifts to modify an appointment, Salesforce Scheduler considers the default value for the Appointment Type (if not specified). However, Salesforce Scheduler only considers the engagement channel type and Appointment Type is ignored.

Example

```
ConnectApi.ExtendedFieldInput extendedFieldEmail = new ConnectApi.ExtendedFieldInput();
extendedFieldEmail.name = 'Email';
extendedFieldEmail.value = 'rachel.adams@salesforce.com.example';

ConnectApi.ExtendedFieldInput extendedFieldPhone = new ConnectApi.ExtendedFieldInput();
extendedFieldPhone.name = 'Phone';
extendedFieldPhone.value = '0123456789';

ConnectApi.ExtendedFieldInput extendedFieldStatus = new ConnectApi.ExtendedFieldInput();
extendedFieldStatus.name = 'Status';
extendedFieldStatus.value = 'None';

List<ConnectApi.ExtendedFieldInput> extendedFieldList = new
List<ConnectApi.ExtendedFieldInput>();
extendedFieldList.add(extendedFieldEmail);
extendedFieldList.add(extendedFieldPhone);
extendedFieldList.add(extendedFieldStatus);

ConnectApi.ServiceAppointmentInput serviceAppInput = new
ConnectApi.ServiceAppointmentInput();
serviceAppInput.extendedFields = extendedFieldList;
serviceAppInput.serviceTerritoryId = '0Hhxx0000004C92CAE';
serviceAppInput.workTypeId = '08qxx0000004C92AAE';
serviceAppInput.schedStartTime = DateTime.valueOf('2021-05-28 12:15:00');
serviceAppInput.schedEndTime = DateTime.valueOf('2021-05-28 12:45:00');

ConnectApi.AssignedResourcesInput asResourceInput = new ConnectApi.AssignedResourcesInput();
asResourceInput.serviceResourceId = '0Hnxx0000004CAiCAM';
asResourceInput.isRequiredResource = true;
asResourceInput.isPrimaryResource = true;

//Multi-resource
ConnectApi.AssignedResourcesInput asResourceInputReq = new
ConnectApi.AssignedResourcesInput();
asResourceInputReq.serviceResourceId = '0Hnxx0000004CAgCAM';
asResourceInputReq.isRequiredResource = true;
asResourceInputReq.isPrimaryResource = false;

List<ConnectApi.AssignedResourcesInput> asResourceInputList = new
List<ConnectApi.AssignedResourcesInput>();
asResourceInputList.add(asResourceInput);
asResourceInputList.add(asResourceInputReq);

ConnectApi.UpdateServiceAppointmentInput updateInput = new
ConnectApi.UpdateServiceAppointmentInput();
updateInput.serviceAppointment = serviceAppInput;
updateInput.assignedResources = asResourceInputList;
updateInput.serviceAppointmentId = '08pxx0000004CYqAAM';

try{
   ConnectApi.ServiceAppointmentOutput appointmentResult =
ConnectApi.LightningScheduler.updateServiceAppointment(updateInput);
```

```
    String serviceAppointmentId = appointmentResult.result.serviceAppointmentId;
    List<String> assignedResourceIds = appointmentResult.result.assignedResourceIds;
}catch(ConnectApi.ConnectApiException ex){
    //Handle Exception
}
```

SEE ALSO:

Service Appointments

ManagedContent Class

Clone managed content. Create and get managed content. Delete and replace variants. Get channels. Get a managed content space. Get targets that managed content space folders can be shared with. Get and update targets that managed content space folders are shared with. Publish and unpublish content.

Namespace

ConnectApi

ManagedContent Methods

These methods are for `ManagedContent`. All methods are static.

IN THIS SECTION:

cloneManagedContentDocument(contentKeyOrId, ManagedContentCloneInputParam)
Clone a piece of managed content.

createManagedContent(ManagedContentInputParam)
Create managed content.

createManagedContentWithMedia(ManagedContentInputParam, contentData)
Create managed content with content data.

deleteManagedContentVariant(variantId)
Delete a managed content variant.

getAllContent(channelId, pageParam, pageSize, language, managedContentType, includeMetadata, startDate, endDate)
Get all managed content versions for a channel.

getAllContent(channelId, pageParam, pageSize, language, managedContentType, includeMetadata, startDate, endDate, showAbsoluteUrl)
Get all managed content versions for a channel with absolute URLs.

getAllDeliveryChannels(pageParam, pageSize)
Get managed content delivery channels for the context user.

getAllManagedContent(communityId, pageParam, pageSize, language, managedContentType)
Get all managed content versions for an Experience Cloud site.

getAllManagedContent(communityId, pageParam, pageSize, language, managedContentType, showAbsoluteUrl)
Get all managed content versions for an Experience Cloud site with absolute URLs.

getContentByContentKeys(channelId, contentKeys, pageParam, pageSize, language, managedContentType, includeMetadata, startDate, endDate, showAbsoluteUrl)

Get managed content versions for a channel using a list of content keys.

getContentByIds(channelId, managedContentIds, pageParam, pageSize, language, managedContentType, includeMetadata, startDate, endDate)

Get managed content versions for a channel using a list of managed content IDs.

getContentByIds(channelId, managedContentIds, pageParam, pageSize, language, managedContentType, includeMetadata, startDate, endDate, showAbsoluteUrl)

Get managed content versions for a channel with absolute URLs using a list of managed content IDs.

getManagedContentByContentKeys(communityId, contentKeys, pageParam, pageSize, language, managedContentType, showAbsoluteUrl)

Get managed content versions for an Experience Cloud site using a list of content keys.

getManagedContentByIds(communityId, managedContentIds, pageParam, pageSize, language, managedContentType)

Get managed content versions for an Experience Cloud site using a list of managed content IDs.

getManagedContentByIds(communityId, managedContentIds, pageParam, pageSize, language, managedContentType, showAbsoluteUrl)

Get managed content versions for an Experience Cloud site with absolute URLs using a list of managed content IDs.

getManagedContentByTopics(communityId, topics, pageParam, pageSize, language, managedContentType)

Get managed content versions using a list of content topic names.

getManagedContentByTopics(communityId, topics, pageParam, pageSize, language, managedContentType, showAbsoluteUrl)

Get managed content versions with absolute URLs using a list of content topic names.

getManagedContentByTopicsAndContentKeys(communityId, contentKeys, topics, pageParam, pageSize, language, managedContentType, showAbsoluteUrl)

Get managed content versions using a list of content keys and content topic names.

getManagedContentByTopicsAndIds(communityId, managedContentIds, topics, pageParam, pageSize, language, managedContentType)

Get managed content versions using a list of managed content IDs and content topic names.

getManagedContentByTopicsAndIds(communityId, managedContentIds, topics, pageParam, pageSize, language, managedContentType, showAbsoluteUrl)

Get managed content versions with absolute URLs using a list of managed content IDs and content topic names.

getManagedContentSpace(contentSpaceId)

Get a managed content space.

getMCSFolderShares(folderId)

Get targets that a managed content space folder is shared with.

getMCSFolderShareTargets(folderId)

Get targets that a managed content space folder can be shared with.

patchMCSFolderShares(folderId, mcsFolderShareCollectionUpdateInput)

Update the targets that a managed content space folder is shared with.

publish(publishInput)

Publish content.

replaceManagedContentVariant(variantId, ManagedContentVariantInputParam)

Replace a managed content variant.

replaceManagedContentVariantWithMedia(variantId, ManagedContentVariantInputParam, contentData)
Replace a managed content variant, including content data.

unpublish(unpublishInput)
Unpublish content.

cloneManagedContentDocument(contentKeyOrId, ManagedContentCloneInputParam)

Clone a piece of managed content.

API Version

61.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentDocumentClone cloneManagedContentDocument(String contentKeyOrId, ConnectApi.ManagedContentDocumentCloneInput ManagedContentCloneInputParam)
```

Parameters

contentKeyOrId
Type: String

Content key or ID of the managed content to clone.

ManagedContentCloneInputParam
Type: ConnectApi.ManagedContentDocumentCloneInput

`ConnectApi.ManagedContentDocumentCloneInput` class specifying the details for the cloned content.

Return Value

Type: ConnectApi.ManagedContentDocumentClone

createManagedContent(ManagedContentInputParam)

Create managed content.

API Version

60.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentDocument
createManagedContent(ConnectApi.ManagedContentDocumentInput ManagedContentInputParam)
```

Parameters

ManagedContentInputParam

Type: ConnectApi.ManagedContentDocumentInput

A `ConnectApi.ManagedContentDocumentInput` input class with information to create managed content.

Return Value

Type: ConnectApi.ManagedContentDocument

createManagedContentWithMedia(ManagedContentInputParam, contentData)

Create managed content with content data.

API Version

60.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentDocument
createManagedContentWithMedia(ConnectApi.ManagedContentDocumentInput
ManagedContentInputParam, ConnectApi.BinaryInput contentData)
```

Parameters

ManagedContentInputParam

Type: ConnectApi.ManagedContentDocumentInput

A `ConnectApi.ManagedContentDocumentInput` input class with information to create managed content.

contentData

Type: ConnectApi.BinaryInput

A new binary file of the content data for the managed content.

Return Value

Type: ConnectApi.ManagedContentDocument

deleteManagedContentVariant(variantId)

Delete a managed content variant.

API Version

60.0

Requires Chatter

No

Signature

```
public static Void deleteManagedContentVariant(String variantId)
```

Parameters

variantId
Type: String

ID of the variant to delete.

Return Value

Type: Void

getAllContent(channelId, pageParam, pageSize, language, managedContentType, includeMetadata, startDate, endDate)

Get all managed content versions for a channel.

API Version

48.0

Available to Guest Users

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentVersionCollection getAllContent(String channelId,
Integer pageParam, Integer pageSize, String language, String managedContentType, Boolean
includeMetadata, String startDate, String endDate)
```

Parameters

channelId
Type: String

ID of the channel.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 250. For performance reasons, we recommend 25 or fewer items per page. If you pass in `null`, the default size is 25.

language
Type: String

Language locale for the managed content, for example, `en_US`. If the requested translation isn't available, the language defaults to the context user's language. If the context user's language isn't available, the language defaults to the content type's original language.

managedContentType
Type: String

Developer name of the content type, such as `cms_document` or `cms_image`.

includeMetadata
Type: Boolean

Specifies whether to include metadata in the response (`true`) or not (`false`). The default value is `false`.

startDate
Type: String

Publish start date in ISO 8601 format, for example, 2011-02-25T18:24:31.000Z.

endDate
Type: String

Publish end date in ISO 8601 format, for example, 2011-02-25T18:24:31.000Z.

Return Value

Type: `ConnectApi.ManagedContentVersionCollection`

getAllContent(channelId, pageParam, pageSize, language, managedContentType, includeMetadata, startDate, endDate, showAbsoluteUrl)

Get all managed content versions for a channel with absolute URLs.

API Version

50.0

Available to Guest Users

50.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentVersionCollection getAllContent(String channelId,
Integer pageParam, Integer pageSize, String language, String managedContentType, Boolean
includeMetadata, String startDate, String endDate, Boolean showAbsoluteUrl)
```

Parameters

channelId
Type: String

ID of the channel.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 250. For performance reasons, we recommend 25 or fewer items per page. If you pass in `null`, the default size is 25.

language
Type: String

Language locale for the managed content, for example, `en_US`. If the requested translation isn't available, the language defaults to the context user's language. If the context user's language isn't available, the language defaults to the content type's original language.

managedContentType
Type: String

Developer name of the content type, such as `cms_document` or `cms_image`.

includeMetadata
Type: Boolean

Specifies whether to include metadata in the response (`true`) or not (`false`). The default value is `false`.

startDate
Type: String

Publish start date in ISO 8601 format, for example, 2011-02-25T18:24:31.000Z.

endDate
Type: String

Publish end date in ISO 8601 format, for example, 2011-02-25T18:24:31.000Z.

showAbsoluteUrl
Type: Boolean

Specifies whether to show absolute URLs in the output class (`true`) or not (`false`). The default value is `false`.

Return Value

Type: `ConnectApi.ManagedContentVersionCollection`

`getAllDeliveryChannels(pageParam, pageSize)`

Get managed content delivery channels for the context user.

API Version

48.0–61.0

In version 62.0 and later, use `getChannels(pageParam, pageSize)` in the `ManagedContentDelivery` class to get all delivery channels.

Available to Guest Users

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentChannelCollection getAllDeliveryChannels(Integer pageParam, Integer pageSize)
```

Parameters

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 250. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.ManagedContentChannelCollection`

`getAllManagedContent(communityId, pageParam, pageSize, language, managedContentType)`

Get all managed content versions for an Experience Cloud site.

API Version

47.0

Available to Guest Users

47.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentVersionCollection getAllManagedContent(String communityId, Integer pageParam, Integer pageSize, String language, String managedContentType)
```

Parameters

communityId
Type: String

ID of the Experience Cloud site.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 250. For performance reasons, we recommend 25 or fewer items per page. If you pass in `null`, the default size is 25.

language
Type: String

Language locale for the managed content, for example, `en_US`. If the requested translation isn't available, the language defaults to the context user's language. If the context user's language isn't available, the language defaults to the content type's original language.

managedContentType
Type: String

Developer name of the content type, such as `cms_document` or `cms_image`.

Return Value

Type: `ConnectApi.ManagedContentVersionCollection`

getAllManagedContent(communityId, pageParam, pageSize, language, managedContentType, showAbsoluteUrl)

Get all managed content versions for an Experience Cloud site with absolute URLs.

API Version

50.0

Available to Guest Users

50.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentVersionCollection getAllManagedContent(String communityId, Integer pageParam, Integer pageSize, String language, String managedContentType, Boolean showAbsoluteUrl)
```

Parameters

communityId
Type: String

ID of the Experience Cloud site.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 250. For performance reasons, we recommend 25 or fewer items per page. If you pass in `null`, the default size is 25.

language
Type: String

Language locale for the managed content, for example, `en_US`. If the requested translation isn't available, the language defaults to the context user's language. If the context user's language isn't available, the language defaults to the content type's original language.

managedContentType
Type: String

Developer name of the content type, such as `cms_document` or `cms_image`.

showAbsoluteUrl
Type: Boolean

Specifies whether to show absolute URLs in the output class (`true`) or not (`false`). The default value is `false`.

Return Value

Type: `ConnectApi.ManagedContentVersionCollection`

getContentByContentKeys(channelId, contentKeys, pageParam, pageSize, language, managedContentType, includeMetadata, startDate, endDate, showAbsoluteUrl)

Get managed content versions for a channel using a list of content keys.

API Version

51.0

Available to Guest Users

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentVersionCollection getContentByContentKeys(String channelId, List<String> contentKeys, Integer pageParam, Integer pageSize, String language, String managedContentType, Boolean includeMetadata, String startDate, String endDate, Boolean showAbsoluteUrl)
```

Parameters

channelId
Type: String

ID of the channel.

contentKeys
Type: List<String>

List of up to 50 content keys for the managed content. A content key is a universally unique identifier (UUID) such as MCA4CCV5QS2BAB5H7YRCRPTCWGZQ.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 250. For performance reasons, we recommend 25 or fewer items per page. If you pass in `null`, the default size is 25.

language
Type: String

Language locale for the managed content, for example, `en_US`. If the requested translation isn't available, the language defaults to the context user's language. If the context user's language isn't available, the language defaults to the content type's original language.

managedContentType
Type: String

Developer name of the content type, such as `cms_document` or `cms_image`.

includeMetadata
Type: Boolean

Specifies whether to include metadata in the response (`true`) or not (`false`). The default value is `false`.

startDate
Type: String

Publish start date in ISO 8601 format, for example, 2011-02-25T18:24:31.000Z.

endDate
Type: String

Publish end date in ISO 8601 format, for example, 2011-02-25T18:24:31.000Z.

showAbsoluteUrl
Type: Boolean

Specifies whether to show absolute URLs in the output class (`true`) or not (`false`). The default value is `false`.

Return Value

Type: `ConnectApi.ManagedContentVersionCollection`

getContentByIds(channelId, managedContentIds, pageParam, pageSize, language, managedContentType, includeMetadata, startDate, endDate)

Get managed content versions for a channel using a list of managed content IDs.

API Version

48.0

Available to Guest Users

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentVersionCollection getContentByIds(String channelId, List<String> managedContentIds, Integer pageParam, Integer pageSize, String language, String managedContentType, Boolean includeMetadata, String startDate, String endDate)
```

Parameters

channelId
Type: String

ID of the channel.

managedContentIds
Type: List<String>

Comma-separated list of managed content IDs. HTTP/2 clients support up to 200 IDs. HTTP/1.1 clients don't.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 250. For performance reasons, we recommend 25 or fewer items per page. If you pass in `null`, the default size is 25.

language
Type: String

Language locale for the managed content, for example, `en_US`. If the requested translation isn't available, the language defaults to the context user's language. If the context user's language isn't available, the language defaults to the content type's original language.

managedContentType
Type: String

Developer name of the content type, such as `cms_document` or `cms_image`.

includeMetadata
Type: Boolean

Specifies whether to include metadata in the response (`true`) or not (`false`). The default value is `false`.

startDate
Type: String

Publish start date in ISO 8601 format, for example, 2011-02-25T18:24:31.000Z.

endDate
Type: String

Publish end date in ISO 8601 format, for example, 2011-02-25T18:24:31.000Z.

Return Value

Type: `ConnectApi.ManagedContentVersionCollection`

`getContentByIds(channelId, managedContentIds, pageParam, pageSize, language, managedContentType, includeMetadata, startDate, endDate, showAbsoluteUrl)`

Get managed content versions for a channel with absolute URLs using a list of managed content IDs.

API Version

50.0

Available to Guest Users

50.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentVersionCollection getContentByIds(String
channelId, List<String> managedContentIds, Integer pageParam, Integer pageSize, String
language, String managedContentType, Boolean includeMetadata, String startDate, String
endDate, Boolean showAbsoluteUrl)
```

Parameters

channelId
Type: String

ID of the channel.

managedContentIds
Type: List<String>

Comma-separated list of managed content IDs. HTTP/2 clients support up to 200 IDs. HTTP/1.1 clients don't.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 250. For performance reasons, we recommend 25 or fewer items per page. If you pass in `null`, the default size is 25.

language
Type: String

Language locale for the managed content, for example, `en_US`. If the requested translation isn't available, the language defaults to the context user's language. If the context user's language isn't available, the language defaults to the content type's original language.

managedContentType
Type: String

Developer name of the content type, such as `cms_document` or `cms_image`.

includeMetadata
Type: Boolean

Specifies whether to include metadata in the response (`true`) or not (`false`). The default value is `false`.

startDate
Type: String

Publish start date in ISO 8601 format, for example, 2011-02-25T18:24:31.000Z.

endDate
Type: String

Publish end date in ISO 8601 format, for example, 2011-02-25T18:24:31.000Z.

showAbsoluteUrl
Type: Boolean

Specifies whether to show absolute URLs in the output class (`true`) or not (`false`). The default value is `false`.

Return Value

Type: ConnectApi.ManagedContentVersionCollection

getManagedContentByContentKeys(communityId, contentKeys, pageParam, pageSize, language, managedContentType, showAbsoluteUrl)

Get managed content versions for an Experience Cloud site using a list of content keys.

API Version

51.0

Available to Guest Users

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentVersionCollection
getManagedContentByContentKeys(String communityId, List<String> contentKeys, Integer
pageParam, Integer pageSize, String language, String managedContentType, Boolean
showAbsoluteUrl)
```

Parameters

communityId
Type: String

ID of the Experience Cloud site.

contentKeys
Type: List<String>

List of up to 50 content keys for the managed content. A content key is a universally unique identifier (UUID) such as MCA4CCV5QS2BAB5H7YRCRPTCWGZQ.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 250. For performance reasons, we recommend 25 or fewer items per page. If you pass in `null`, the default size is 25.

language
Type: String

Language locale for the managed content, for example, `en_US`. If the requested translation isn't available, the language defaults to the context user's language. If the context user's language isn't available, the language defaults to the content type's original language.

managedContentType
Type: String

Developer name of the content type, such as `cms_document` or `cms_image`.

showAbsoluteUrl
Type: Boolean

Specifies whether to show absolute URLs in the output class (`true`) or not (`false`). The default value is `false`.

Return Value

Type: `ConnectApi.ManagedContentVersionCollection`

getManagedContentByIds(communityId, managedContentIds, pageParam, pageSize, language, managedContentType)

Get managed content versions for an Experience Cloud site using a list of managed content IDs.

API Version

47.0

Available to Guest Users

47.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentVersionCollection getManagedContentByIds(String communityId, List<String> managedContentIds, Integer pageParam, Integer pageSize, String language, String managedContentType)
```

Parameters

communityId
Type: String

ID of the Experience Cloud site.

managedContentIds
Type: List<String>

Comma-separated list of managed content IDs. HTTP/2 clients support up to 200 IDs. HTTP/1.1 clients don't.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 250. For performance reasons, we recommend 25 or fewer items per page. If you pass in `null`, the default size is 25.

language
Type: String

Language locale for the managed content, for example, `en_US`. If the requested translation isn't available, the language defaults to the context user's language. If the context user's language isn't available, the language defaults to the content type's original language.

managedContentType
Type: String

Developer name of the content type, such as `cms_document` or `cms_image`.

Return Value

Type: `ConnectApi.ManagedContentVersionCollection`

getManagedContentByIds(communityId, managedContentIds, pageParam, pageSize, language, managedContentType, showAbsoluteUrl)

Get managed content versions for an Experience Cloud site with absolute URLs using a list of managed content IDs.

API Version

50.0

Available to Guest Users

50.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentVersionCollection getManagedContentByIds(String communityId, List<String> managedContentIds, Integer pageParam, Integer pageSize, String language, String managedContentType, Boolean showAbsoluteUrl)
```

Parameters

communityId
Type: String

ID of the Experience Cloud site.

managedContentIds
Type: List<String>

Comma-separated list of managed content IDs. HTTP/2 clients support up to 200 IDs. HTTP/1.1 clients don't.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 250. For performance reasons, we recommend 25 or fewer items per page. If you pass in `null`, the default size is 25.

language
Type: String

Language locale for the managed content, for example, `en_US`. If the requested translation isn't available, the language defaults to the context user's language. If the context user's language isn't available, the language defaults to the content type's original language.

managedContentType
Type: String

Developer name of the content type, such as `cms_document` or `cms_image`.

showAbsoluteUrl
Type: Boolean

Specifies whether to show absolute URLs in the output class (`true`) or not (`false`). The default value is `false`.

Return Value

Type: `ConnectApi.ManagedContentVersionCollection`

getManagedContentByTopics(communityId, topics, pageParam, pageSize, language, managedContentType)

Get managed content versions using a list of content topic names.

API Version

47.0

Available to Guest Users

47.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentVersionCollection getManagedContentByTopics(String communityId, List<String> topics, Integer pageParam, Integer pageSize, String language, String managedContentType)
```

Parameters

communityId
Type: String

ID of the Experience Cloud site.

topics
Type: List<String>

Comma-separated list of up to 15 content topic names.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 250. For performance reasons, we recommend 25 or fewer items per page. If you pass in `null`, the default size is 25.

language
Type: String

Language locale for the managed content, for example, `en_US`. If the requested translation isn't available, the language defaults to the context user's language. If the context user's language isn't available, the language defaults to the content type's original language.

managedContentType
Type: String

Developer name of the content type, such as `cms_document` or `cms_image`.

Return Value

Type: `ConnectApi.ManagedContentVersionCollection`

getManagedContentByTopics(communityId, topics, pageParam, pageSize, language, managedContentType, showAbsoluteUrl)

Get managed content versions with absolute URLs using a list of content topic names.

API Version

50.0

Available to Guest Users

50.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentVersionCollection getManagedContentByTopics(String communityId, List<String> topics, Integer pageParam, Integer pageSize, String language, String managedContentType, Boolean showAbsoluteUrl)
```

Parameters

communityId
Type: String

ID of the Experience Cloud site.

topics
Type: List<String>

Comma-separated list of up to 15 content topic names.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 250. For performance reasons, we recommend 25 or fewer items per page. If you pass in `null`, the default size is 25.

language
Type: String

Language locale for the managed content, for example, `en_US`. If the requested translation isn't available, the language defaults to the context user's language. If the context user's language isn't available, the language defaults to the content type's original language.

managedContentType
Type: String

Developer name of the content type, such as `cms_document` or `cms_image`.

showAbsoluteUrl
Type: Boolean

Specifies whether to show absolute URLs in the output class (`true`) or not (`false`). The default value is `false`.

Return Value

Type: `ConnectApi.ManagedContentVersionCollection`

getManagedContentByTopicsAndContentKeys(communityId, contentKeys, topics, pageParam, pageSize, language, managedContentType, showAbsoluteUrl)

Get managed content versions using a list of content keys and content topic names.

API Version

51.0

Available to Guest Users

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentVersionCollection
getManagedContentByTopicsAndContentKeys(String communityId, List<String> contentKeys,
List<String> topics, Integer pageParam, Integer pageSize, String language, String
managedContentType, Boolean showAbsoluteUrl)
```

Parameters

communityId
Type: String

ID of the Experience Cloud site.

contentKeys
Type: List<String>

List of up to 50 content keys for the managed content. A content key is a universally unique identifier (UUID) such as MCA4CCV5QS2BAB5H7YRCRPTCWGZQ.

topics
Type: List<String>

Comma-separated list of up to 15 content topic names.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 250. For performance reasons, we recommend 25 or fewer items per page. If you pass in `null`, the default size is 25.

language
Type: String

Language locale for the managed content, for example, `en_US`. If the requested translation isn't available, the language defaults to the context user's language. If the context user's language isn't available, the language defaults to the content type's original language.

managedContentType
Type: String

Developer name of the content type, such as `cms_document` or `cms_image`.

showAbsoluteUrl
Type: Boolean

Specifies whether to show absolute URLs in the output class (`true`) or not (`false`). The default value is `false`.

Return Value

Type: `ConnectApi.ManagedContentVersionCollection`

getManagedContentByTopicsAndIds(communityId, managedContentIds, topics, pageParam, pageSize, language, managedContentType)

Get managed content versions using a list of managed content IDs and content topic names.

API Version

47.0

Available to Guest Users

47.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentVersionCollection
getManagedContentByTopicsAndIds(String communityId, List<String> managedContentIds,
List<String> topics, Integer pageParam, Integer pageSize, String language, String
managedContentType)
```

Parameters

communityId
Type: String

ID of the Experience Cloud site.

managedContentIds
Type: List<String>

Comma-separated list of managed content IDs. HTTP/2 clients support up to 200 IDs. HTTP/1.1 clients don't.

topics
Type: List<String>

Comma-separated list of up to 15 content topic names.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 250. For performance reasons, we recommend 25 or fewer items per page. If you pass in `null`, the default size is 25.

language
Type: String

Language locale for the managed content, for example, `en_US`. If the requested translation isn't available, the language defaults to the context user's language. If the context user's language isn't available, the language defaults to the content type's original language.

managedContentType
Type: String

Developer name of the content type, such as `cms_document` or `cms_image`.

Return Value

Type: `ConnectApi.ManagedContentVersionCollection`

getManagedContentByTopicsAndIds(communityId, managedContentIds, topics, pageParam, pageSize, language, managedContentType, showAbsoluteUrl)

Get managed content versions with absolute URLs using a list of managed content IDs and content topic names.

API Version

50.0

Available to Guest Users

50.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentVersionCollection
getManagedContentByTopicsAndIds(String communityId, List<String> managedContentIds,
List<String> topics, Integer pageParam, Integer pageSize, String language, String
managedContentType, Boolean showAbsoluteUrl)
```

Parameters

communityId
Type: String

ID of the Experience Cloud site.

managedContentIds
Type: List<String>

Comma-separated list of managed content IDs. HTTP/2 clients support up to 200 IDs. HTTP/1.1 clients don't.

topics
Type: List<String>

Comma-separated list of up to 15 content topic names.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 250. For performance reasons, we recommend 25 or fewer items per page. If you pass in `null`, the default size is 25.

language
Type: String

Language locale for the managed content, for example, `en_US`. If the requested translation isn't available, the language defaults to the context user's language. If the context user's language isn't available, the language defaults to the content type's original language.

managedContentType
Type: String

Developer name of the content type, such as `cms_document` or `cms_image`.

showAbsoluteUrl
Type: Boolean

Specifies whether to show absolute URLs in the output class (`true`) or not (`false`). The default value is `false`.

Return Value

Type: `ConnectApi.ManagedContentVersionCollection`

getManagedContentSpace(contentSpaceId)

Get a managed content space.

API Version

55.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentSpace getManagedContentSpace(String
contentSpaceId)
```

Parameters

contentSpaceId
Type: String

ID of the managed content space.

Return Value

Type: ConnectApi.ManagedContentSpace

getMCSFolderShares(folderId)

Get targets that a managed content space folder is shared with.

To get the targets that a managed content space folder can be shared with, use getMCSFolderShareTargets(folderId).

API Version

63.0

Requires Chatter

No

Signature

```
public static ConnectApi.MCSFolderShareCollection getMCSFolderShares(String folderId)
```

Parameters

folderId
Type: String

ID of the folder.

Return Value

Type: ConnectApi.MCSFolderShareCollection

getMCSFolderShareTargets(folderId)

Get targets that a managed content space folder can be shared with.

To get the targets that a managed content space folder is shared with, use getMCSFolderShares(folderId).

API Version

63.0

Requires Chatter

No

Signature

```
public static ConnectApi.MCSFolderShareTargetCollection getMCSFolderShareTargets(String folderId)
```

Parameters

folderId
Type: String

ID of the folder.

Return Value

Type: ConnectApi.MCSFolderShareTargetCollection

patchMCSFolderShares(folderId, mcsFolderShareCollectionUpdateInput)

Update the targets that a managed content space folder is shared with.

Workspaces are the only supported sharing targets. To get the targets that a managed content space folder is shared with, use `getMCSFolderShares(folderId)`. To get the targets that a managed content space folder can be shared with, use `getMCSFolderShareTargets(folderId)`.

API Version

63.0

Requires Chatter

No

Signature

```
public static ConnectApi.MCSFolderShareCollection patchMCSFolderShares(String folderId, ConnectApi.MCSFolderShareCollectionUpdateInput mcsFolderShareCollectionUpdateInput)
```

Parameters

folderId
Type: String

ID of the folder.

mcsFolderShareCollectionUpdateInput
Type: ConnectApi.MCSFolderShareCollectionUpdateInput

`ConnectApi.MCSFolderShareCollectionUpdateInput` input class with the targets to share and unshare.

Return Value

Type: ConnectApi.MCSFolderShareCollection

publish(publishInput)

Publish content.

API Version

60.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentPublishOutput
publish(ConnectApi.ManagedContentPublishInput publishInput)
```

Parameters

publishInput
Type: ConnectApi.ManagedContentPublishInput

A ConnectApi.ManagedContentPublishInput request body specifying the content to publish.

Return Value

Type: ConnectApi.ManagedContentPublishOutput

replaceManagedContentVariant(variantId, ManagedContentVariantInputParam)

Replace a managed content variant.

API Version

60.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentVariant replaceManagedContentVariant(String
variantId, ConnectApi.ManagedContentVariantUpdateInput ManagedContentVariantInputParam)
```

Parameters

variantId
Type: String

ID of the managed content variant to replace.

ManagedContentVariantInputParam
Type: ConnectApi.ManagedContentVariantUpdateInput

A ConnectApi.ManagedContentVariantUpdateInput input class with information about the managed content variant that is replacing the existing variant.

Return Value

Type: ConnectApi.ManagedContentVariant

replaceManagedContentVariantWithMedia(variantId, ManagedContentVariantInputParam, contentData)

Replace a managed content variant, including content data.

API Version

60.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentVariant
replaceManagedContentVariantWithMedia(String variantId,
ConnectApi.ManagedContentVariantUpdateInput ManagedContentVariantInputParam,
ConnectApi.BinaryInput contentData)
```

Parameters

variantId
Type: String

ID of the managed content variant to replace.

ManagedContentVariantInputParam
Type: ConnectApi.ManagedContentVariantUpdateInput

A ConnectApi.ManagedContentVariantUpdateInput input class with information about the managed content variant that is replacing the existing variant.

contentData
Type: ConnectApi.BinaryInput

A new binary file to replace the content data of the managed content variant.

Return Value

Type: ConnectApi.ManagedContentVariant

`unpublish(unpublishInput)`

Unpublish content.

API Version

60.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentUnpublishOutput
unpublish(ConnectApi.ManagedContentUnpublishInput unpublishInput)
```

Parameters

`unpublishInput`
Type: `ConnectApi.ManagedContentUnpublishInput`

A `ConnectApi,ManagedContentUnpublishInput` request body specifying the content to unpublish.

Return Value

Type: `ConnectApi.ManagedContentUnpublishOutput`

Retired ManagedContent Methods

These methods for `ManagedContent` are retired.

IN THIS SECTION:

getAllDeliveryChannels(pageParam, pageSize)
Get managed content delivery channels for the context user.

`getAllDeliveryChannels(pageParam, pageSize)`

Get managed content delivery channels for the context user.

API Version

48.0–61.0

In version 62.0 and later, use `getChannels(pageParam, pageSize)` in the `ManagedContentDelivery` class to get all delivery channels.

Available to Guest Users

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentChannelCollection getAllDeliveryChannels(Integer pageParam, Integer pageSize)
```

Parameters

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 250. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.ManagedContentChannelCollection`

ManagedContentChannels Class

Get managed content channels. Create, get, update, or delete a managed content channel.

Namespace

ConnectApi

ManagedContentChannels Methods

These methods are for `ManagedContentChannels`. All methods are static.

IN THIS SECTION:

deleteManagedContentChannel(channelId)
Delete a managed content channel.

getManagedContentChannel(channelId)
Get a managed content channel.

getManagedContentChannels(pageParam, pageSize, showDetails)
Get managed content channels.

patchManagedContentChannel(channelId, ManagedContentChannelInput)
Update a managed content channel.

postManagedContentChannel(ManagedContentCreateInputParam)
Create a managed content channel.

deleteManagedContentChannel(channelId)

Delete a managed content channel.

API Version

62.0

Requires Chatter

No

Signature

```
public static Void deleteManagedContentChannel(String channelId)
```

Parameters

channelId
Type: String

ID of the managed content channel to delete.

Return Value

Type: Void

getManagedContentChannel(channelId)

Get a managed content channel.

API Version

62.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentChannel getManagedContentChannel(String channelId)
```

Parameters

channelId
Type: String

ID of the managed content channel.

Return Value

Type: ConnectApi.ManagedContentChannel

getManagedContentChannels(pageParam, pageSize, showDetails)

Get managed content channels.

API Version

62.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentChannelsRepresentation
getManagedContentChannels(Integer pageParam, Integer pageSize, Boolean showDetails)
```

Parameters

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 250. If you pass in `null`, the default size is 25.

showDetails
Type: Boolean

Specifies whether to show the channels' detailed information (`true`) or summary information only (`false`). If you pass in `null`, the default is `false`.

Return Value

Type: `ConnectApi.ManagedContentChannelsRepresentation`

patchManagedContentChannel(channelId, ManagedContentChannelInput)

Update a managed content channel.

API Version

62.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentChannel patchManagedContentChannel(String
channelId, ConnectApi.ManagedContentChannelUpdateRepresentation
ManagedContentChannelInput)
```

Parameters

channelId
Type: String

ID of the managed content channel to update.

ManagedContentChannelInput
Type: ConnectApi.ManagedContentChannelUpdateRepresentation

ConnectApi.ManagedContentChannelUpdateRepresentation input class with the updates.

Return Value

Type: ConnectApi.ManagedContentChannel

postManagedContentChannel(ManagedContentCreateInputParam)

Create a managed content channel.

API Version

62.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentChannel
postManagedContentChannel(ConnectApi.ManagedContentChannelCreateRepresentation
ManagedContentCreateInputParam)
```

Parameters

ManagedContentCreateInputParam
Type: ConnectApi.ManagedContentChannelCreateRepresentation

ConnectApi.ManagedContentChannelCreateRepresentation input class describing the managed content channel.

Return Value

Type: ConnectApi.ManagedContentChannel

ManagedContentDelivery Class

Get collection items. Get a managed content channel. Get managed content.

Namespace

ConnectApi

ManagedContentDelivery Methods

These methods are for `ManagedContentDelivery`. All methods are static.

IN THIS SECTION:

getChannels(pageParam, pageSize)
Get managed content delivery channels for the context user.

getCollectionItemsForChannel(channelId, collectionKeyOrId, language)
Get collection items for a channel.

getCollectionItemsForChannel(channelId, collectionKeyOrId, language, pageToken, pageSize)
Get a page of collection items for a channel.

getCollectionItemsForSite(siteId, collectionKeyOrId, language)
Get collection items for an Experience Cloud site.

getCollectionItemsForSite(siteId, collectionKeyOrId, language, pageToken, pageSize)
Get a page of collection items for an Experience Cloud site.

getManagedContentDeliveryChannel(channelId)
Get a managed content delivery channel.

getManagedContentForChannel(channelId, contentKeyOrId, showAbsoluteUrl)
Get a piece of published content for a channel.

getManagedContentForChannel(channelId, contentKeyOrId, language, showAbsoluteUrl)
Get a piece of published content in a specified language for a channel.

getManagedContentForChannel(channelId, contentKeyOrId, language, showAbsoluteUrl, referenceDepth, expandReferences, referencesAsList)
Get a piece of published content in a specified language with references for a channel.

getManagedContentForSite(siteId, contentKeyOrId, showAbsoluteUrl)
Get a piece of published content for an Experience Cloud site.

getManagedContentForSite(siteId, contentKeyOrId, language, showAbsoluteUrl)
Get a piece of published content in a specified language for an Experience Cloud site.

getManagedContentForSite(siteId, contentKeyOrId, language, showAbsoluteUrl, referenceDepth, expandReferences, referencesAsList)
Get a piece of published content in a specified language with references for an Experience Cloud site.

getManagedContentsForChannel(channelId, managedContentIds, contentKeys, contentTypeFQN, language, publishStartDate, publishEndDate, includeContentBody, referenceDepth, expandReferences, referencesAsList, pageParam, pageSize, showAbsoluteUrl)
Get a collection of published contents for a channel.

getManagedContentsForSite(siteId, managedContentIds, contentKeys, contentTypeFQN, language, publishStartDate, publishEndDate, includeContentBody, referenceDepth, expandReferences, referencesAsList, pageParam, pageSize, showAbsoluteUrl)

Get a collection of published contents for an Experience Cloud site.

`getChannels(pageParam, pageSize)`

Get managed content delivery channels for the context user.

API Version

62.0

Available to Guest Users

62.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentDeliveryChannelsRepresentation getChannels(Integer pageParam, Integer pageSize)
```

Parameters

pageParam

Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize

Type: Integer

Specifies the number of items per page. Valid values are from 1 through 250. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.ManagedContentDeliveryChannelsRepresentation`

`getCollectionItemsForChannel(channelId, collectionKeyOrId, language)`

Get collection items for a channel.

API Version

56.0

Available to Guest Users

56.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentCollectionItems
getCollectionItemsForChannel(String channelId, String collectionKeyOrId, String language)
```

Parameters

channelId
Type: String

ID of the channel.

collectionKeyOrId
Type: String

Collection key or ID of the collection. A collection key is a unique identifier such as MCA4CCV5QS2BAB5H7YRCRPTCWGZQ.

language
Type: String

Language locale for the managed content, for example, `en_US`. If the requested translation isn't available, the language defaults to the channel or site's default language. If the channel or site's default language isn't available, the language defaults to the primary language of the content space.

Return Value

Type: `ConnectApi.ManagedContentCollectionItems`

Example

This example gets a collection of a custom content type that includes references to images and uses the `ConnectApi.ConnectUtilities.unwrapApexWrapper()` utility.

```
ConnectApi.ManagedContentCollectionItems collection =

ConnectApi.ManagedContentDelivery.getCollectionItemsForChannel('0apXXXXXXXXXXXXXX','MCVXXXXXXXXXXXXXXXXXXXXXXXX','en_US');

System.debug(collection.items); //before ApexWrapper is unwrapped

for(ConnectApi.ManagedContentCollectionItem item : collection.items)
{
 //unwrap ApexWrapper
 Map<String,Object> unwrappedItem =
(Map<String,Object>)ConnectApi.ConnectUtilities.unwrapApexWrapper(item.body.get('contentBody'));


 //replace the wrapped object with the unwrapped object
 item.body.put('contentBody', unwrappedItem);
}

System.debug(collection.items); //after ApexWrapper is unwrapped
```

`getCollectionItemsForChannel(channelId, collectionKeyOrId, language, pageToken, pageSize)`

Get a page of collection items for a channel.

API Version

63.0

Available to Guest Users

63.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentCollectionItems
getCollectionItemsForChannel(String channelId, String collectionKeyOrId, String language,
Integer pageToken, Integer pageSize)
```

Parameters

channelId
Type: String

ID of the channel.

collectionKeyOrId
Type: String

Collection key or ID of the collection. A collection key is a unique identifier such as MCA4CCV5QS2BAB5H7YRCRPTCWGZQ.

language
Type: String

Language locale for the managed content, for example, `en_US`. If the requested translation isn't available, the language defaults to the channel or site's default language. If the channel or site's default language isn't available, the language defaults to the primary language of the content space.

pageToken
Type: Integer

Integer specifying a page token of items. If you pass in null, the default value is 0, which returns the first page token.

pageSize
Type: Integer

Number of items per page. Valid values are from 1 through 250. If you pass in null, the default size is 50.

Return Value

Type: `ConnectApi.ManagedContentCollectionItems`

`getCollectionItemsForSite(siteId, collectionKeyOrId, language)`

Get collection items for an Experience Cloud site.

API Version

56.0

Available to Guest Users

56.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentCollectionItems getCollectionItemsForSite(String siteId, String collectionKeyOrId, String language)
```

Parameters

siteId
Type: String

ID for the Experience Cloud site.

collectionKeyOrId
Type: String

Collection key or ID of the collection. A collection key is a unique identifier such as MCA4CCV5QS2BAB5H7YRCRPTCWGZQ.

language
Type: String

Language locale for the managed content, for example, `en_US`. If the requested translation isn't available, the language defaults to the channel or site's default language. If the channel or site's default language isn't available, the language defaults to the primary language of the content space.

Return Value

Type: `ConnectApi.ManagedContentCollectionItems`

Example

This example gets a collection of a custom content type that includes references to images and uses the `ConnectApi.ConnectUtilities.unwrapApexWrapper()` utility.

```
ConnectApi.ManagedContentCollectionItems collection =

ConnectApi.ManagedContentDelivery.getCollectionItemsForSite('0DMXXXXXXXXXXXXXX','MCVXXXXXXXXXXXXXXXXXXXXXXX','en_US');

System.debug(collection.items); //before ApexWrapper is unwrapped
```

```
for(ConnectApi.ManagedContentCollectionItem item : collection.items)
{
 //unwrap ApexWrapper
 Map<String,Object> unwrappedItem =
(Map<String,Object>)ConnectApi.ConnectUtilities.unwrapApexWrapper(item.body.get('contentBody'));


 //replace the wrapped object with the unwrapped object
 item.body.put('contentBody', unwrappedItem);
}

System.debug(collection.items); //after ApexWrapper is unwrapped
```

getCollectionItemsForSite(siteId, collectionKeyOrId, language, pageToken, pageSize)

Get a page of collection items for an Experience Cloud site.

API Version

63.0

Available to Guest Users

63.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentCollectionItems getCollectionItemsForSite(String
siteId, String collectionKeyOrId, String language, Integer pageToken, Integer pageSize)
```

Parameters

siteId
Type: String

ID for the Experience Cloud site.

collectionKeyOrId
Type: String

Collection key or ID of the collection. A collection key is a unique identifier such as MCA4CCV5QS2BAB5H7YRCRPTCWGZQ.

language
Type: String

Language locale for the managed content, for example, `en_US`. If the requested translation isn't available, the language defaults to the channel or site's default language. If the channel or site's default language isn't available, the language defaults to the primary language of the content space.

pageToken
Type: Integer

Integer specifying a page token of items. If you pass in null, the default value is 0, which returns the first page token.

pageSize
Type: Integer

Number of items per page. Valid values are from 1 through 250. If you pass in null, the default size is 50.

Return Value

Type: ConnectApi.ManagedContentCollectionItems

getManagedContentDeliveryChannel(channelId)

Get a managed content delivery channel.

API Version

62.0

Available to Guest Users

62.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentDeliveryChannelRepresentation
getManagedContentDeliveryChannel(String channelId)
```

Parameters

channelId
Type: String

ID of the channel.

Return Value

Type: ConnectApi.ManagedContentDeliveryChannelRepresentation

getManagedContentForChannel(channelId, contentKeyOrId, showAbsoluteUrl)

Get a piece of published content for a channel.

API Version

54.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentDeliveryDocument
getManagedContentForChannel(String channelId, String contentKeyOrId, Boolean
showAbsoluteUrl)
```

Parameters

channelId
Type: String

ID of the channel.

contentKeyOrId
Type: String

Content key or ID of the content. A content key is a unique identifier such as MCA4CCV5QS2BAB5H7YRCRPTCWGZQ.

showAbsoluteUrl
Type: Boolean

For public channels only, specifies whether to return the absolute `unauthenticatedUrl` in the output class. The default value is `false`.

Return Value

Type: `ConnectApi.ManagedContentDeliveryDocument`

Usage

This method returns content only if it's published in the default language of the channel. If you request content that isn't published in the default language of the channel, you get a `ConnectApi.NotFoundException`. To get content for a channel in another language use `getManagedContentForChannel(channelId, contentKeyOrId, language, showAbsoluteUrl)` or `getManagedContentForChannel(channelId, contentKeyOrId, language, showAbsoluteUrl, referenceDepth, expandReferences, referencesAsList)` on page 1511.

Example

This example gets a custom content type with an image reference and uses the `ConnectApi.ConnectUtilities.unwrapApexWrapper()` utility.

```
ConnectApi.ManagedContentDeliveryDocument res =
    ConnectApi.ManagedContentDelivery.getManagedContentForChannel
('0apXXXXXXXXXXXXXX','MCLXXXXXXXXXXXXXXXXXXXXXXXXX',true);

//before contentBody field ApexWrapper is unwrapped
system.debug(res.contentBody);
```

```
//unwrap contentBody field in res
Map<String,Object> contentBody =
(Map<String,Object>)ConnectApi.ConnectUtilities.unwrapApexWrapper(res.contentBody);

//after contentBody field ApexWrapper is unwrapped, but image field still wrapped
system.debug(contentBody);

//before image field ApexWrapper is unwrapped
system.debug(contentBody.get('Image'));

//unwrap Image field in contentBody
Map<String,Object> Image =
(Map<String,Object>)ConnectApi.ConnectUtilities.unwrapApexWrapper(contentBody.get('Image'));

//after image field ApexWrapper is unwrapped
system.debug(Image);

//replace wrapped primary_image in contentBody with unwrapped version
contentBody.put('Image', Image);

//after contentBody field ApexWrapper is unwrapped, with image field also unwrapped
system.debug(contentBody);
```

`getManagedContentForChannel(channelId, contentKeyOrId, language, showAbsoluteUrl)`

Get a piece of published content in a specified language for a channel.

API Version

54.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentDeliveryDocument
getManagedContentForChannel(String channelId, String contentKeyOrId, String language,
Boolean showAbsoluteUrl)
```

Parameters

channelId
Type: String

ID of the channel.

contentKeyOrId
Type: String

Content key or ID of the content. A content key is a unique identifier such as MCA4CCV5QS2BAB5H7YRCRPTCWGZQ.

language
Type: String

Language locale for the managed content, for example, `en_US`. The requested language must be added to the channel, otherwise, you get a `ConnectApi.NotFoundException`. If the requested translation isn't available, the language defaults to the channel or site's default language. If the channel or site's default language isn't available, the language defaults to the primary language of the content space.

showAbsoluteUrl
Type: Boolean

For public channels only, specifies whether to return the absolute `unauthenticatedUrl` in the output class. The default value is `false`.

Return Value

Type: `ConnectApi.ManagedContentDeliveryDocument`

getManagedContentForChannel(channelId, contentKeyOrId, language, showAbsoluteUrl, referenceDepth, expandReferences, referencesAsList)

Get a piece of published content in a specified language with references for a channel.

API Version

54.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentDeliveryDocument
getManagedContentForChannel(String channelId, String contentKeyOrId, String language,
Boolean showAbsoluteUrl, Integer referenceDepth, Boolean expandReferences, Boolean
referencesAsList)
```

Parameters

channelId
Type: String

ID of the channel.

contentKeyOrId
Type: String

Content key or ID of the content. A content key is a unique identifier such as MCA4CCV5QS2BAB5H7YRCRPTCWGZQ.

language
Type: String

Language locale for the managed content, for example, `en_US`. The requested language must be added to the channel, otherwise, you get a `ConnectApi.NotFoundException`. If the requested translation isn't available, the language defaults to the channel or site's default language. If the channel or site's default language isn't available, the language defaults to the primary language of the content space.

showAbsoluteUrl
Type: Boolean

For public channels only, specifies whether to return the absolute `unauthenticatedUrl` in the output class. The default value is `false`.

referenceDepth
Type: Integer

An integer 0–3 specifying the depth of references. If you specify 0, the `references` property of the `ConnectApi.ManagedContentDeliveryDocument` output class is null. If unspecified, the default value is 0.

expandReferences
Type: Boolean

Specifies whether to include details of references (`true`) or summaries of references (`false`) in the output class. If unspecified, the default value is `false`.

referencesAsList
Type: Boolean

Specifies whether to return the references as a list in the `referencesList` property of the `ConnectApi.ManagedContentDeliveryDocument` output class (`true`). If you specify `false`, the references are returned as key value pairs in the `references` property. If unspecified, the default value is `false`.

Return Value

Type: `ConnectApi.ManagedContentDeliveryDocument`

getManagedContentForSite(siteId, contentKeyOrId, showAbsoluteUrl)

Get a piece of published content for an Experience Cloud site.

API Version

54.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentDeliveryDocument getManagedContentForSite(String siteId, String contentKeyOrId, Boolean showAbsoluteUrl)
```

Parameters

siteId
Type: String

ID for the Experience Cloud site.

contentKeyOrId
Type: String

Content key or ID of the content. A content key is a unique identifier such as MCA4CCV5QS2BAB5H7YRCRPTCWGZQ.

showAbsoluteUrl
Type: Boolean

For public channels only, specifies whether to return the absolute `unauthenticatedUrl` in the output class. The default value is `false`.

Return Value

Type: `ConnectApi.ManagedContentDeliveryDocument`

Example

This example gets a custom content type with an image reference and uses the `ConnectApi.ConnectUtilities.unwrapApexWrapper()` utility.

```
ConnectApi.ManagedContentDeliveryDocument res =
    ConnectApi.ManagedContentDelivery.getManagedContentForSite
('0DMXXXXXXXXXXXXXXX','MCLXXXXXXXXXXXXXXXXXXXXXXXXXX',true);

//before contentBody field ApexWrapper is unwrapped
system.debug(res.contentBody);

//unwrap contentBody field in res
Map<String,Object> contentBody =
(Map<String,Object>)ConnectApi.ConnectUtilities.unwrapApexWrapper(res.contentBody);

//after contentBody field ApexWrapper is unwrapped, but image field still wrapped
system.debug(contentBody);

//before image field ApexWrapper is unwrapped
system.debug(contentBody.get('Image'));

//unwrap Image field in contentBody
Map<String,Object> Image =
(Map<String,Object>)ConnectApi.ConnectUtilities.unwrapApexWrapper(contentBody.get('Image'));

//after image field ApexWrapper is unwrapped
system.debug(Image);
```

```
//replace wrapped primary_image in contentBody with unwrapped version
contentBody.put('Image', Image);

//after contentBody field ApexWrapper is unwrapped, with image field also unwrapped
system.debug(contentBody);
```

getManagedContentForSite(siteId, contentKeyOrId, language, showAbsoluteUrl)

Get a piece of published content in a specified language for an Experience Cloud site.

API Version

54.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentDeliveryDocument getManagedContentForSite(String siteId, String contentKeyOrId, String language, Boolean showAbsoluteUrl)
```

Parameters

siteId
Type: String

ID for the Experience Cloud site.

contentKeyOrId
Type: String

Content key or ID of the content. A content key is a unique identifier such as MCA4CCV5QS2BAB5H7YRCRPTCWGZQ.

language
Type: String

Language locale for the managed content, for example, `en_US`. If the requested translation isn't available, the language defaults to the channel or site's default language. If the channel or site's default language isn't available, the language defaults to the primary language of the content space.

showAbsoluteUrl
Type: Boolean

For public channels only, specifies whether to return the absolute `unauthenticatedUrl` in the output class. The default value is `false`.

Return Value

Type: `ConnectApi.ManagedContentDeliveryDocument`

`getManagedContentForSite(siteId, contentKeyOrId, language, showAbsoluteUrl, referenceDepth, expandReferences, referencesAsList)`

Get a piece of published content in a specified language with references for an Experience Cloud site.

API Version

54.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentDeliveryDocument getManagedContentForSite(String siteId, String contentKeyOrId, String language, Boolean showAbsoluteUrl, Integer referenceDepth, Boolean expandReferences, Boolean referencesAsList)
```

Parameters

siteId
Type: String

ID for the Experience Cloud site.

contentKeyOrId
Type: String

Content key or ID of the content. A content key is a unique identifier such as MCA4CCV5QS2BAB5H7YRCRPTCWGZQ.

language
Type: String

Language locale for the managed content, for example, `en_US`. If the requested translation isn't available, the language defaults to the channel or site's default language. If the channel or site's default language isn't available, the language defaults to the primary language of the content space.

showAbsoluteUrl
Type: Boolean

For public channels only, specifies whether to return the absolute `unauthenticatedUrl` in the output class. The default value is `false`.

referenceDepth
Type: Integer

An integer 0–3 specifying the depth of references. If you specify 0, the `references` property of the `ConnectApi.ManagedContentDeliveryDocument` output class is null. If unspecified, the default value is 0.

expandReferences
Type: Boolean

Specifies whether to include details of references (`true`) or summaries of references (`false`) in the output class. If unspecified, the default value is `false`.

referencesAsList
Type: Boolean

Specifies whether to return the references as a list in the `referencesList` property of the `ConnectApi.ManagedContentDeliveryDocument` output class (`true`). If you specify `false`, the references are returned as key value pairs in the `references` property. If unspecified, the default value is `false`.

Return Value

Type: `ConnectApi.ManagedContentDeliveryDocument`

getManagedContentsForChannel(channelId, managedContentIds, contentKeys, contentTypeFQN, language, publishStartDate, publishEndDate, includeContentBody, referenceDepth, expandReferences, referencesAsList, pageParam, pageSize, showAbsoluteUrl)

Get a collection of published contents for a channel.

API Version

58.0

Available to Guest Users

58.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentDeliveryDocumentCollection
getManagedContentsForChannel(String channelId, List<String> managedContentIds,
List<String> contentKeys, String contentTypeFQN, String language, String
publishStartDate, String publishEndDate, Boolean includeContentBody, Integer
referenceDepth, Boolean expandReferences, Boolean referencesAsList, Integer pageParam,
Integer pageSize, Boolean showAbsoluteUrl)
```

Parameters

channelId
Type: String

ID of the channel.

managedContentIds
Type: List<String>

Comma-separated list of up to 100 managed content IDs. Specify either managed content IDs or content keys.

contentKeys
Type: List<String>

Comma-separated list of up to 50 content keys. Specify either managed content IDs or content keys.

contentTypeFQN
Type: String

Fully qualified name of the managed content type.

language
Type: String

Language locale for the managed content, for example, `en_US`. If the requested translation isn't available, the language defaults to the configured fallback language or the channel's default language. If the content isn't available in the fallback language and the channel's default language, we return an error.

publishStartDate
Type: String

ISO 8601 formatted publish start date.

publishEndDate
Type: String

ISO 8601 formatted publish end date.

includeContentBody
Type: Boolean

Specifies whether to return the content body (`true`) or the content summary (`false`). If unspecified, the default value is `false`.

referenceDepth
Type: Integer

An integer 0–3 specifying the depth of references. If you specify 0, the `references` property of the `ConnectApi.ManagedContentDeliveryDocumentCollection` output class is null. If unspecified, the default value is 0.

expandReferences
Type: Boolean

Specifies whether to include details of references (`true`) or summaries of references (`false`) in the output class. If unspecified, the default value is `false`.

referencesAsList
Type: Boolean

Specifies whether to return the references as a list in the `referencesList` property of the `ConnectApi.ManagedContentDeliveryDocumentCollection` output class (`true`). If you specify `false`, the references are returned as key value pairs in the `references` property. If unspecified, the default value is `false`.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 250. If you pass in `null`, the default size is 25. If you specify `true` for *expandReferences* or *includeContentBody*, the maximum page size you can specify is 25.

showAbsoluteUrl
Type: Boolean

Specifies whether to show absolute URLs in the output class (`true`) or not (`false`). The default value is `false`.

Return Value

Type: `ConnectApi.ManagedContentDeliveryDocumentCollection`

Example

This example gets Event custom content type records and uses the `ConnectApi.ConnectUtilities.unwrapApexWrapper()` utility.

```
ConnectApi.ManagedContentDeliveryDocumentCollection resCol =
    ConnectApi.ManagedContentDelivery.getManagedContentsForChannel('0apXXXXXXXXXXXXXXX',
null, new List<String>{'MCLXXXXXXXXXXXXXXXXXXXXXXXXX'}, 'Event', null, null, null, true,
3, true, false, null, 25, true);

Map<String, Object> contentBodyMap = new Map<String,Object>();

for(ConnectApi.AbstractManagedContentDeliveryDocument res1 : resCol.contents)
{
 //cast the abstract object as the ManagedContentDeliveryDocument subclass that contains
contentBody
 ConnectApi.ManagedContentDeliveryDocument res =
(ConnectApi.ManagedContentDeliveryDocument)res1;

 //before contentBody field ApexWrapper is unwrapped
 system.debug(res.contentBody);

 //unwrap contentBody field in res
 Map<String,Object> contentBody =
(Map<String,Object>)ConnectApi.ConnectUtilities.unwrapApexWrapper(res.contentBody);

 //after contentBody field ApexWrapper is unwrapped, but image field still wrapped
 system.debug(contentBody);

 //before image field ApexWrapper is unwrapped
 system.debug(contentBody.get('Image'));

 //unwrap Image field in contentBody
 Map<String,Object> Image =
(Map<String,Object>)ConnectApi.ConnectUtilities.unwrapApexWrapper(contentBody.get('Image'));


 //after image field ApexWrapper is unwrapped
 system.debug(Image);

 //replace wrapped primary_image in contentBody with unwrapped version
 contentBody.put('Image', Image);

 //after contentBody field ApexWrapper is unwrapped, with image field also unwrapped
 system.debug(contentBody);
```

```
 //put unwrapped contentBody in map
 contentBodyMap.put(res.contentKey, contentBody);

}

//check unwrapped contentBody map
System.debug(contentBodyMap);
```

getManagedContentsForSite(siteId, managedContentIds, contentKeys, contentTypeFQN, language, publishStartDate, publishEndDate, includeContentBody, referenceDepth, expandReferences, referencesAsList, pageParam, pageSize, showAbsoluteUrl)

Get a collection of published contents for an Experience Cloud site.

API Version

58.0

Available to Guest Users

58.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentDeliveryDocumentCollection
getManagedContentsForSite(String siteId, List<String> managedContentIds, List<String>
contentKeys, String contentTypeFQN, String language, String publishStartDate, String
publishEndDate, Boolean includeContentBody, Integer referenceDepth, Boolean
expandReferences, Boolean referencesAsList, Integer pageParam, Integer pageSize, Boolean
showAbsoluteUrl)
```

Parameters

siteId
Type: String

ID for the Experience Cloud site.

managedContentIds
Type: List<String>

Comma-separated list of up to 100 managed content IDs. Specify either managed content IDs or content keys.

contentKeys
Type: List<String>

Comma-separated list of up to 50 content keys. Specify either managed content IDs or content keys.

contentTypeFQN
Type: String

Fully qualified name of the managed content type.

language
Type: String

Language locale for the managed content, for example, `en_US`. If the requested translation isn't available, the language defaults to the configured fallback language or the channel's default language. If the content isn't available in the fallback language and the channel's default language, we return an error.

publishStartDate
Type: String

ISO 8601 formatted publish start date.

publishEndDate
Type: String

ISO 8601 formatted publish end date.

includeContentBody
Type: Boolean

Specifies whether to return the content body (`true`) or the content summary (`false`). If unspecified, the default value is `false`.

referenceDepth
Type: Integer

An integer 0–3 specifying the depth of references. If you specify 0, the `references` property of the `ConnectApi.ManagedContentDeliveryDocumentCollection` output class is null. If unspecified, the default value is 0.

expandReferences
Type: Boolean

Specifies whether to include details of references (`true`) or summaries of references (`false`) in the output class. If unspecified, the default value is `false`.

referencesAsList
Type: Boolean

Specifies whether to return the references as a list in the `referencesList` property of the `ConnectApi.ManagedContentDeliveryDocumentCollection` output class (`true`). If you specify `false`, the references are returned as key value pairs in the `references` property. If unspecified, the default value is `false`.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 250. If you pass in `null`, the default size is 25. If you specify `true` for *expandReferences* or *includeContentBody*, the maximum page size you can specify is 25.

showAbsoluteUrl
Type: Boolean

Specifies whether to show absolute URLs in the output class (`true`) or not (`false`). The default value is `false`.

Return Value

Type: ConnectApi.ManagedContentDeliveryDocumentCollection

Example

This example gets custom content type Event records and uses the ConnectApi.ConnectUtilities.unwrapApexWrapper() utility.

```
ConnectApi.ManagedContentDeliveryDocumentCollection resCol =
    ConnectApi.ManagedContentDelivery.getManagedContentsForSite('0DMXXXXXXXXXXXXXX', null,
 new List<String>{'MCLXXXXXXXXXXXXXXXXXXXXXXXXXX'}, 'Event', null, null, null, true, 3,
true, false, null, 25, true);

Map<String, Object> contentBodyMap = new Map<String,Object>();

for(ConnectApi.AbstractManagedContentDeliveryDocument res1 : resCol.contents)
{
 //cast the abstract object as the ManagedContentDeliveryDocument subclass that contains
contentBody
 ConnectApi.ManagedContentDeliveryDocument res =
(ConnectApi.ManagedContentDeliveryDocument)res1;

 //before contentBody field ApexWrapper is unwrapped
 system.debug(res.contentBody);

 //unwrap contentBody field in res
 Map<String,Object> contentBody =
(Map<String,Object>)ConnectApi.ConnectUtilities.unwrapApexWrapper(res.contentBody);

 //after contentBody field ApexWrapper is unwrapped, but image field still wrapped
 system.debug(contentBody);

 //before image field ApexWrapper is unwrapped
 system.debug(contentBody.get('Image'));

 //unwrap Image field in contentBody
 Map<String,Object> Image =
(Map<String,Object>)ConnectApi.ConnectUtilities.unwrapApexWrapper(contentBody.get('Image'));


 //after image field ApexWrapper is unwrapped
 system.debug(Image);

 //replace wrapped primary_image in contentBody with unwrapped version
 contentBody.put('Image', Image);

 //after contentBody field ApexWrapper is unwrapped, with image field also unwrapped
 system.debug(contentBody);

 //put unwrapped contentBody in map
 contentBodyMap.put(res.contentKey, contentBody);

}
```

```
//check unwrapped contentBody map
System.debug(contentBodyMap);
```

Retired ManagedContentDelivery Methods

These methods for `ManagedContentDelivery` are retired.

IN THIS SECTION:

getManagedContentChannel(channelId)
Get a managed content delivery channel.

getManagedContentsForChannel(channelId, managedContentIds, contentKeys, contentTypeFQN, language, publishStartDate, publishEndDate, includeContentBody, referenceDepth, expandReferences, referencesAsList, pageParam, pageSize)
Get a collection of published contents for a channel.

getManagedContentsForSite(siteId, managedContentIds, contentKeys, contentTypeFQN, language, publishStartDate, publishEndDate, includeContentBody, referenceDepth, expandReferences, referencesAsList, pageParam, pageSize)
Get a collection of published contents for an Experience Cloud site.

`getManagedContentChannel(channelId)`

Get a managed content delivery channel.

API Version

54.0–61.0

In version 62.0 and later, use `getManagedContentDeliveryChannel(channelId)` to get a managed content delivery channel.

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentChannelDetail getManagedContentChannel(String channelId)
```

Parameters

`channelId`
Type: String

ID of the channel.

Return Value

Type: ConnectApi.ManagedContentChannelDetail

getManagedContentsForChannel(channelId, managedContentIds, contentKeys, contentTypeFQN, language, publishStartDate, publishEndDate, includeContentBody, referenceDepth, expandReferences, referencesAsList, pageParam, pageSize)

Get a collection of published contents for a channel.

API Version

55.0—57.0

In version 58.0 and later, use getManagedContentsForChannel(channelId, managedContentIds, contentKeys, contentTypeFQN, language, publishStartDate, publishEndDate, includeContentBody, referenceDepth, expandReferences, referencesAsList, pageParam, pageSize, showAbsoluteUrl).

Available to Guest Users

55.0—57.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentDeliveryDocumentCollection
getManagedContentsForChannel(String channelId, List<String> managedContentIds,
List<String> contentKeys, String contentTypeFQN, String language, String
publishStartDate, String publishEndDate, Boolean includeContentBody, Integer
referenceDepth, Boolean expandReferences, Boolean referencesAsList, Integer pageParam,
Integer pageSize)
```

Parameters

channelId
Type: String

ID of the channel.

managedContentIds
Type: List<String>

Comma-separated list of up to 100 managed content IDs. Specify either managed content IDs or content keys.

contentKeys
Type: List<String>

Comma-separated list of up to 50 content keys. Specify either managed content IDs or content keys.

contentTypeFQN
Type: String

Fully qualified name of the managed content type.

language
Type: String

Language locale for the managed content, for example, `en_US`. If the requested translation isn't available, the language defaults to the configured fallback language or the channel's default language. If the content isn't available in the fallback language and the channel's default language, we return an error.

publishStartDate
Type: String

ISO 8601 formatted publish start date.

publishEndDate
Type: String

ISO 8601 formatted publish end date.

includeContentBody
Type: Boolean

Specifies whether to return the content body (`true`) or the content summary (`false`). If unspecified, the default value is `false`.

referenceDepth
Type: Integer

An integer 0–3 specifying the depth of references. If you specify 0, the `references` property of the `ConnectApi.ManagedContentDeliveryDocumentCollection` output class is null. If unspecified, the default value is 0.

expandReferences
Type: Boolean

Specifies whether to include details of references (`true`) or summaries of references (`false`) in the output class. If unspecified, the default value is `false`.

referencesAsList
Type: Boolean

Specifies whether to return the references as a list in the `referencesList` property of the `ConnectApi.ManagedContentDeliveryDocumentCollection` output class (`true`). If you specify `false`, the references are returned as key value pairs in the `references` property. If unspecified, the default value is `false`.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 250. If you pass in `null`, the default size is 25. If you specify `true` for *expandReferences* or *includeContentBody*, the maximum page size you can specify is 25.

Return Value

Type: `ConnectApi.ManagedContentDeliveryDocumentCollection`

getManagedContentsForSite(siteId, managedContentIds, contentKeys, contentTypeFQN, language, publishStartDate, publishEndDate, includeContentBody, referenceDepth, expandReferences, referencesAsList, pageParam, pageSize)

Get a collection of published contents for an Experience Cloud site.

API Version

55.0—57.0

In version 58.0 and later, use `getManagedContentsForSite(siteId, managedContentIds, contentKeys, contentTypeFQN, language, publishStartDate, publishEndDate, includeContentBody, referenceDepth, expandReferences, referencesAsList, pageParam, pageSize, showAbsoluteUrl)`.

Available to Guest Users

55.0—57.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentDeliveryDocumentCollection
getManagedContentsForSite(String siteId, List<String> managedContentIds, List<String>
contentKeys, String contentTypeFQN, String language, String publishStartDate, String
publishEndDate, Boolean includeContentBody, Integer referenceDepth, Boolean
expandReferences, Boolean referencesAsList, Integer pageParam, Integer pageSize)
```

Parameters

siteId
Type: String

ID for the Experience Cloud site.

managedContentIds
Type: List<String>

Comma-separated list of up to 100 managed content IDs. Specify either managed content IDs or content keys.

contentKeys
Type: List<String>

Comma-separated list of up to 50 content keys. Specify either managed content IDs or content keys.

contentTypeFQN
Type: String

Fully qualified name of the managed content type.

language
Type: String

Language locale for the managed content, for example, `en_US`. If the requested translation isn't available, the language defaults to the configured fallback language or the channel's default language. If the content isn't available in the fallback language and the channel's default language, we return an error.

`publishStartDate`
Type: String

ISO 8601 formatted publish start date.

`publishEndDate`
Type: String

ISO 8601 formatted publish end date.

`includeContentBody`
Type: Boolean

Specifies whether to return the content body (`true`) or the content summary (`false`). If unspecified, the default value is `false`.

`referenceDepth`
Type: Integer

An integer 0–3 specifying the depth of references. If you specify 0, the `references` property of the `ConnectApi.ManagedContentDeliveryDocumentCollection` output class is null. If unspecified, the default value is 0.

`expandReferences`
Type: Boolean

Specifies whether to include details of references (`true`) or summaries of references (`false`) in the output class. If unspecified, the default value is `false`.

`referencesAsList`
Type: Boolean

Specifies whether to return the references as a list in the `referencesList` property of the `ConnectApi.ManagedContentDeliveryDocumentCollection` output class (`true`). If you specify `false`, the references are returned as key value pairs in the `references` property. If unspecified, the default value is `false`.

`pageParam`
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

`pageSize`
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 250. If you pass in `null`, the default size is 25. If you specify `true` for *`expandReferences`* or *`includeContentBody`*, the maximum page size you can specify is 25.

Return Value

Type: `ConnectApi.ManagedContentDeliveryDocumentCollection`

ManagedContentSpaces Class

Get channels in a managed content space. Add or remove channels from a managed content space.

Namespace

ConnectApi

ManagedContentSpaces Methods

These methods are for `ManagedContentSpaces`. All methods are static.

IN THIS SECTION:

getManagedContentSpaceChannels(contentSpaceId, pageParam, pageSize)
Get channels for a managed content space.

patchManagedContentSpaceChannels(contentSpaceId, spaceChannels)
Add or remove channels from a managed content space.

`getManagedContentSpaceChannels(contentSpaceId, pageParam, pageSize)`

Get channels for a managed content space.

API Version

62.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentSpaceChannelsRepresentation
getManagedContentSpaceChannels(String contentSpaceId, Integer pageParam, Integer
pageSize)
```

Parameters

contentSpaceId
Type: String

ID of the managed content space.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Number of items per page. Valid values are from 1 through 250. If you pass in `null`, the default size is 25. Items are sorted by the last modified date.

Return Value

Type: ConnectApi.ManagedContentSpaceChannelsRepresentation

patchManagedContentSpaceChannels(contentSpaceId, spaceChannels)

Add or remove channels from a managed content space.

API Version

62.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedContentSpaceChannelsRepresentation
patchManagedContentSpaceChannels(String contentSpaceId,
ConnectApi.ManagedContentSpaceChannelsInputRepresentation spaceChannels)
```

Parameters

contentSpaceId
Type: String

ID of the managed content space.

spaceChannels
Type: ConnectApi.ManagedContentSpaceChannelsInputRepresentation

`ConnectApi.ManagedContentSpaceChannelsInputRepresentation` input class with the channels to add or remove from the managed content space.

Return Value

Type: ConnectApi.ManagedContentSpaceChannelsRepresentation

ManagedTopics Class

Get managed topics in an Experience Cloud site. Create, delete, and reorder managed topics.

Namespace

ConnectApi

ManagedTopics Methods

These methods are for `ManagedTopics`. All methods are static.

IN THIS SECTION:

createManagedTopic(communityId, recordId, managedTopicType)
Create a managed topic of a specific type for an Experience Cloud site.

createManagedTopic(communityId, recordId, managedTopicType, parentId)
Create a child managed topic for an Experience Cloud site.

createManagedTopicByName(communityId, name, managedTopicType)
Create a managed topic of a specific type by name for an Experience Cloud site.

createManagedTopicByName(communityId, name, managedTopicType, parentId)
Create a child managed topic by name for an Experience Cloud site.

deleteManagedTopic(communityId, managedTopicId)
Delete a managed topic from an Experience Cloud site.

getManagedTopic(communityId, managedTopicId)
Get a managed topic in an Experience Cloud site.

getManagedTopic(communityId, managedTopicId, depth)
Get a managed topic, including its parent and children managed topics, in an Experience Cloud site.

getManagedTopics(communityId)
Get the featured and navigational managed topics for an Experience Cloud site.

getManagedTopics(communityId, managedTopicType)
Get managed topics of the specified type for an Experience Cloud site.

getManagedTopics(communityId, managedTopicType, depth)
Get managed topics of the specified type, including their parent and children managed topics, in an Experience Cloud site.

getManagedTopics(communityId, managedTopicType, recordIds, depth)
Get managed topics of the specified type, including their parent and children managed topics, that are associated with topics in an Experience Cloud site.

getManagedTopics(communityId, managedTopicType, pageParam, pageSize)
Get a page of managed topics.

reorderManagedTopics(communityId, managedTopicPositionCollection)
Reorder the relative positions of managed topics in an Experience Cloud site.

`createManagedTopic(communityId, recordId, managedTopicType)`

Create a managed topic of a specific type for an Experience Cloud site.

API Version

32.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedTopic createManagedTopic(String communityId, String
recordId, ConnectApi.ManagedTopicType managedTopicType)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

recordId
Type: String

ID of the topic.

managedTopicType
Type: `ConnectApi.ManagedTopicType`

Specify the type of managed topic.

- `Content`—Topics that are associated with native content.
- `Featured`—Topics that are featured, for example, on the Experience Cloud site home page, but don't provide overall navigation.
- `Navigational`—Topics that display in a navigational menu in the Experience Cloud site.

A topic can be associated with all three managed topic types, so a topic can be a `Featured`, `Navigational`, and `Content` topic.

You can create up to 25 `Featured` and 5,000 `Content` topics. You can create up to eight levels of `Navigational` managed topics with 25 top-level topics and 10 children topics per level for a maximum of 2,775 `Navigational` topics.

Return Value

Type: `ConnectApi.ManagedTopic`

Usage

Only community managers (users with the Create and Set Up Experiences or Manage Experiences permission) can create managed topics.

`createManagedTopic(communityId, recordId, managedTopicType, parentId)`

Create a child managed topic for an Experience Cloud site.

API Version

35.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedTopic createManagedTopic(String communityId, String recordId, ConnectApi.ManagedTopicType managedTopicType, String parentId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

recordId
Type: String

ID of the topic.

managedTopicType
Type: `ConnectApi.ManagedTopicType`

Specify `Navigational` for the type of managed topic to create a child managed topic.

You can create up to 25 `Featured` and 5,000 `Content` topics. You can create up to eight levels of `Navigational` managed topics with 25 top-level topics and 10 children topics per level for a maximum of 2,775 `Navigational` topics.

parentId
Type: String

ID of the parent managed topic.

You can create up to eight levels (parent, direct children, their children, etc.) of managed topics and up to 10 children managed topics per managed topic.

Return Value

Type: `ConnectApi.ManagedTopic`

Usage

Only community managers (users with the Create and Set Up Experiences or Manage Experiences permission) can create managed topics.

`createManagedTopicByName(communityId, name, managedTopicType)`

Create a managed topic of a specific type by name for an Experience Cloud site.

API Version

32.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedTopic createManagedTopicByName(String communityId,
String name, ConnectApi.ManagedTopicType managedTopicType)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

name
Type: String

Name of the topic.

managedTopicType
Type: `ConnectApi.ManagedTopicType`

Specify the type of managed topic.

- `Content`—Topics that are associated with native content.
- `Featured`—Topics that are featured, for example, on the Experience Cloud site home page, but don't provide overall navigation.
- `Navigational`—Topics that display in a navigational menu in the Experience Cloud site.

A topic can be associated with all three managed topic types, so a topic can be a `Featured`, `Navigational`, and `Content` topic.

You can create up to 25 `Featured` and 5,000 `Content` topics. You can create up to eight levels of `Navigational` managed topics with 25 top-level topics and 10 children topics per level for a maximum of 2,775 `Navigational` topics.

Return Value

Type: `ConnectApi.ManagedTopic`

Usage

Only community managers (users with the Create and Set Up Experiences or Manage Experiences permission) can create managed topics.

`createManagedTopicByName(communityId, name, managedTopicType, parentId)`

Create a child managed topic by name for an Experience Cloud site.

API Version

35.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedTopic createManagedTopicByName(String communityId,
String name, ConnectApi.ManagedTopicType managedTopicType, String parentId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

name
Type: String

Name of the topic.

managedTopicType
Type: `ConnectApi.ManagedTopicType`

Specify `Navigational` for the type of managed topic to create a child managed topic.

You can create up to 25 `Featured` and 5,000 `Content` topics. You can create up to eight levels of `Navigational` managed topics with 25 top-level topics and 10 children topics per level for a maximum of 2,775 `Navigational` topics.

parentId
Type: String

ID of the parent managed topic.

You can create up to eight levels (parent, direct children, their children, etc.) of managed topics and up to 10 children managed topics per managed topic.

Return Value

Type: `ConnectApi.ManagedTopic`

Usage

Only community managers (users with the Create and Set Up Experiences or Manage Experiences permission) can create managed topics.

`deleteManagedTopic(communityId, managedTopicId)`

Delete a managed topic from an Experience Cloud site.

API Version

32.0

Requires Chatter

No

Signature

```
public static deleteManagedTopic(String communityId, String managedTopicId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

managedTopicId
Type: String

ID of managed topic.

Return Value

Type: Void

Usage

Only community managers (users with the Create and Set Up Experiences or Manage Experiences permission) can delete managed topics.

`getManagedTopic(communityId, managedTopicId)`

Get a managed topic in an Experience Cloud site.

API Version

32.0

Available to Guest Users

32.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedTopic getManagedTopic(String communityId, String managedTopicId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

managedTopicId
Type: String

ID of managed topic.

Return Value

Type: `ConnectApi.ManagedTopic`

`getManagedTopic(communityId, managedTopicId, depth)`

Get a managed topic, including its parent and children managed topics, in an Experience Cloud site.

API Version

35.0

Available to Guest Users

35.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedTopic getManagedTopic(String communityId, String managedTopicId, Integer depth)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

managedTopicId
Type: String

ID of managed topic.

depth
Type: Integer

Specify an integer 1–8. If you specify 1, the `children` property of the `ConnectApi.ManagedTopic` output class is `null`. If you specify 2, the `children` property of the `ConnectApi.ManagedTopic` output class contains the direct children managed topics, if any, of the managed topic. If you specify 3–8, you get the direct children managed topics and their children managed topics if there are any. If depth isn't specified, it defaults to 1.

Return Value

Type: `ConnectApi.ManagedTopic`

`getManagedTopics(communityId)`

Get the featured and navigational managed topics for an Experience Cloud site.

To get the content topics for an Experience Cloud site, use `getManagedTopics(communityId, managedTopicType)`.

API Version

32.0

Available to Guest Users

32.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedTopicCollection getManagedTopics(String communityId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

Return Value

Type: `ConnectApi.ManagedTopicCollection`

getManagedTopics(communityId, managedTopicType)

Get managed topics of the specified type for an Experience Cloud site.

API Version

32.0

Available to Guest Users

32.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedTopicCollection getManagedTopics(String communityId,
ConnectApi.ManagedTopicType managedTopicType)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

managedTopicType
Type: `ConnectApi.ManagedTopicType`

Type of managed topic.

- `Content`—Topics that are associated with native content.
- `Featured`—Topics that are featured, for example, on the Experience Cloud site home page, but don't provide overall navigation.
- `Navigational`—Topics that display in a navigational menu in the Experience Cloud site.

A topic can be associated with all three managed topic types, so a topic can be a `Featured`, `Navigational`, and `Content` topic.

If you specify `Content`, up to 50 topics are returned. If you want more than 50 `Content` topics, use `getManagedTopics(communityId, managedTopicType, pageParam, pageSize)`.

Return Value

Type: `ConnectApi.ManagedTopicCollection`

getManagedTopics(communityId, managedTopicType, depth)

Get managed topics of the specified type, including their parent and children managed topics, in an Experience Cloud site.

API Version

35.0

Available to Guest Users

35.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedTopicCollection getManagedTopics(String communityId,
ConnectApi.ManagedTopicType managedTopicType, Integer depth)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

managedTopicType
Type: `ConnectApi.ManagedTopicType`

Type of managed topic.

- `Content`—Topics that are associated with native content.

- `Featured`—Topics that are featured, for example, on the Experience Cloud site home page, but don't provide overall navigation.
- `Navigational`—Topics that display in a navigational menu in the Experience Cloud site.

A topic can be associated with all three managed topic types, so a topic can be a `Featured`, `Navigational`, and `Content` topic.

depth
Type: Integer

Specify an integer 1–8. If you specify 1, the `children` property of the `ConnectApi.ManagedTopic` output class is `null`. If you specify 2, the `children` property of the `ConnectApi.ManagedTopic` output class contains the direct children managed topics, if any, of the managed topic. If you specify 3–8, you get the direct children managed topics and their children managed topics if there are any. If depth isn't specified, it defaults to 1.

Return Value

Type: `ConnectApi.ManagedTopicCollection`

getManagedTopics(communityId, managedTopicType, recordIds, depth)

Get managed topics of the specified type, including their parent and children managed topics, that are associated with topics in an Experience Cloud site.

API Version

38.0

Available to Guest Users

38.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedTopicCollection getManagedTopics(String communityId,
ConnectApi.ManagedTopicType managedTopicType, List<String> recordIds, Integer depth)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

managedTopicType
Type: `ConnectApi.ManagedTopicType`

Type of managed topic.

- `Content`—Topics that are associated with native content.
- `Featured`—Topics that are featured, for example, on the Experience Cloud site home page, but don't provide overall navigation.

- `Navigational`—Topics that display in a navigational menu in the Experience Cloud site.

A topic can be associated with all three managed topic types, so a topic can be a `Featured`, `Navigational`, and `Content` topic.

`recordIds`
Type: List<String>

A list of up to 100 topic IDs associated with the managed topics.

If you list more than 10 topic IDs, you can't specify 2–8 for *`depth`*.

`depth`
Type: Integer

Specify an integer 1–8. If you specify 1, the `children` property of the `ConnectApi.ManagedTopic` output class is `null`. If you specify 2, the `children` property of the `ConnectApi.ManagedTopic` output class contains the direct children managed topics, if any, of the managed topic. If you specify 3–8, you get the direct children managed topics and their children managed topics if there are any. If depth isn't specified, it defaults to 1.

Return Value

Type: `ConnectApi.ManagedTopicCollection`

`getManagedTopics(communityId, managedTopicType, pageParam, pageSize)`

Get a page of managed topics.

API Version

44.0

Available to Guest Users

44.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedTopicCollection getManagedTopics(String communityId,
ConnectApi.ManagedTopicType managedTopicType, Integer pageParam, Integer pageSize)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`managedTopicType`
Type: `ConnectApi.ManagedTopicType`

Type of managed topic.

- `Content`—Topics that are associated with native content.
- `Featured`—Topics that are featured, for example, on the Experience Cloud site home page, but don't provide overall navigation.
- `Navigational`—Topics that display in a navigational menu in the Experience Cloud site.

A topic can be associated with all three managed topic types, so a topic can be a `Featured`, `Navigational`, and `Content` topic.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 50.

Return Value

Type: `ConnectApi.ManagedTopicCollection`

reorderManagedTopics(communityId, managedTopicPositionCollection)

Reorder the relative positions of managed topics in an Experience Cloud site.

API Version

32.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedTopicCollection reorderManagedTopics(String communityId,
ConnectApi.ManagedTopicPositionCollectionInput managedTopicPositionCollection)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

managedTopicPositionCollection
Type: `ConnectApi.ManagedTopicPositionCollectionInput`

A collection of relative positions of managed topics. This collection can include only `Featured` and `Navigational` topics and doesn't have to include all managed topics.

Return Value

Type: `ConnectApi.ManagedTopicCollection`

Usage

Only community managers (users with the Create and Set Up Experiences or Manage Experiences permission) can reorder managed topics.

You can reorder parent managed topics or children managed topics with the same parent. If you don't include all managed topics in the `ConnectApi.ManagedTopicPositionCollectionInput`, the managed topics are reordered by respecting the positions indicated in the `ConnectApi.ManagedTopicPositionCollectionInput` and then by pushing down any managed topics that aren't included in the `ConnectApi.ManagedTopicPositionCollectionInput` to the next available position.

Example

If you have these managed topics:

Managed Topic	Position
ManagedTopicA	0
ManagedTopicB	1
ManagedTopicC	2
ManagedTopicD	3
ManagedTopicE	4

And you reorder managed topics by including this information in `ConnectApi.ManagedTopicPositionCollectionInput`:

Managed Topic	Position
ManagedTopicD	0
ManagedTopicE	2

The result is:

Managed Topic	Position
ManagedTopicD	0
ManagedTopicA	1
ManagedTopicE	2
ManagedTopicB	3
ManagedTopicC	4

Retired ManagedTopics Methods

These methods for `ManagedTopics` are retired.

IN THIS SECTION:

getManagedTopics(communityId, managedTopicType, recordId, depth)
Get managed topics of the specified type, including their parent and children managed topics, that are associated with a given topic in an Experience Cloud site.

getManagedTopics(communityId, managedTopicType, recordId, depth)

Get managed topics of the specified type, including their parent and children managed topics, that are associated with a given topic in an Experience Cloud site.

API Version

35.0–37.0

Important: In version 38.0 and later, use `getManagedTopics(communityId, managedTopicType, recordIds, depth)`.

Available to Guest Users

35.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedTopicCollection getManagedTopics(String communityId,
ConnectApi.ManagedTopicType managedTopicType, String recordId, Integer depth)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

managedTopicType
Type: `ConnectApi.ManagedTopicType`

Type of managed topic.

- `Content`—Topics that are associated with native content.
- `Featured`—Topics that are featured, for example, on the Experience Cloud site home page, but don't provide overall navigation.
- `Navigational`—Topics that display in a navigational menu in the Experience Cloud site.

A topic can be associated with all three managed topic types, so a topic can be a `Featured`, `Navigational`, and `Content` topic.

recordId
Type: String

ID of the topic associated with the managed topics.

depth
Type: Integer

Specify an integer 1–8. If you specify 1, the `children` property of the `ConnectApi.ManagedTopic` output class is `null`. If you specify 2, the `children` property of the `ConnectApi.ManagedTopic` output class contains the direct children managed topics, if any, of the managed topic. If you specify 3–8, you get the direct children managed topics and their children managed topics if there are any. If depth isn't specified, it defaults to 1.

Return Value

Type: `ConnectApi.ManagedTopicCollection`

MarketingIntegration Class

Get, save, and submit a microsites marketing integration form for an Experience Cloud site.

Namespace

ConnectApi

MarketingIntegration Methods

The following are methods for `MarketingIntegration`. All methods are static.

`MarketingIntegration` methods make calls to Marketing Cloud Engagement REST APIs to create, query, and insert data to the data extension object. If the API returns errors, ConnectinApex error messages include the error code and message from Marketing Cloud Engagement.

IN THIS SECTION:

getForm(siteId, formId)
Get a marketing integration form for an Experience Cloud site.

saveForm(siteId, formInput)
Save a marketing integration form for an Experience Cloud site.

submitForm(siteId, formId, formSubmissionInput)
Submit a marketing integration form for an Experience Cloud site.

`getForm(siteId, formId)`

Get a marketing integration form for an Experience Cloud site.

API Version

53.0

Requires Chatter

No

Signature

```
public static ConnectApi.Form getForm(String siteId, String formId)
```

Parameters

siteId
Type: String

ID for the Experience Cloud site.

formId
Type: String

ID of the form.

Return Value

Type: ConnectApi.Form

saveForm(siteId, formInput)

Save a marketing integration form for an Experience Cloud site.

API Version

53.0

Requires Chatter

No

Signature

```
public static ConnectApi.Form saveForm(String siteId, ConnectApi.FormInput formInput)
```

Parameters

siteId
Type: String

ID for the Experience Cloud site.

formInput
Type: ConnectApi.FormInput

A ConnectApi.FormInput object to save.

Return Value

Type: ConnectApi.Form

Usage

This method attempts to create a read-only data extension in Marketing Cloud Engagement. A Marketing Cloud Engagement admin can change the read-only setting. We recommend keeping the data extension as read-only to maintain schema consistency with the form.

`submitForm(siteId, formId, formSubmissionInput)`

Submit a marketing integration form for an Experience Cloud site.

API Version

53.0

Available to Guest Users

53.0

Requires Chatter

No

Signature

```
public static ConnectApi.FormSubmission submitForm(String siteId, String formId,
ConnectApi.FormSubmissionInput formSubmissionInput)
```

Parameters

siteId
Type: String

ID for the Experience Cloud site.

formId
Type: String

ID of the form.

formSubmissionInput
Type: `ConnectApi.FormSubmissionInput`

A `ConnectApi.FormSubmissionInput` object to submit.

Return Value

Type: `ConnectApi.FormSubmission`

Mentions Class

Access information about mentions. A mention is an "@" character followed by a user or group name. When a user or group is mentioned, they receive a notification.

Namespace

ConnectApi

Mentions Methods

These methods are for `Mentions`. All methods are static.

All methods in this class require Chatter and are subject to the per user, per namespace, per hour rate limit.

IN THIS SECTION:

getMentionCompletions(communityId, q, contextId)
Get the first page of possible users and groups to mention in a feed item body or comment body.

getMentionCompletions(communityId, q, contextId, type, pageParam, pageSize)
Get a page of possible mention proposals of the specified type.

getMentionValidations(communityId, parentId, recordIds, visibility)
Get information about whether the mentions are valid for the context user.

`getMentionCompletions(communityId, q, contextId)`

Get the first page of possible users and groups to mention in a feed item body or comment body.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.MentionCompletionPage getMentionCompletions (String communityId,
String q, String contextId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

A search term for matching user and group names. Searching for a group requires a minimum of 2 characters. Searching for a user doesn't require a minimum number of characters. This parameter does not support wildcards.

contextId
Type: String

A feed item ID (for a mention in a comment) or a feed subject ID (for a mention in a feed item) that narrows search results, with more useful results listed first. Use a group ID for groups that allow customers to ensure mention completion results include customers.

Return Value

Type: `ConnectApi.MentionCompletionPage`

Usage

Call this method to generate a page of proposed mentions that a user can choose from when they enter characters after @ in a feed item body or a comment body.

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetMentionCompletions(communityId, q, contextId, result)

Apex Developer Guide: Testing ConnectApi Code

`getMentionCompletions(communityId, q, contextId, type, pageParam, pageSize)`

Get a page of possible mention proposals of the specified type.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Mentions getMentionCompletions (String communityId, String q,
String contextId, ConnectApi.MentionCompletionType type, Integer pageParam, Integer
pageSize)
```

Parameters

`communityId`

Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`q`

Type: String

A search term for matching user and group names. Searching for a group requires a minimum of 2 characters. Searching for a user doesn't require a minimum number of characters. This parameter does not support wildcards.

`contextId`

Type: String

A feed item ID (for a mention in a comment) or a feed subject ID (for a mention in a feed item) that narrows search results, with more useful results listed first. Use a group ID for groups that allow customers to ensure mention completion results include customers.

type
Type: `ConnectApi.MentionCompletionType`

Type of mention completion.

- `All`—All mention completions, regardless of the type of record to which the mention refers.
- `Group`—Mention completions for groups.
- `User`—Mention completions for users.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.MentionCompletionPage`

Usage

Call this method to generate a page of proposed mentions that a user can choose from when they enter characters after @ in a feed item body or a comment body.

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetMentionCompletions(communityId, q, contextId, type, pageParam, pageSize, result)

Apex Developer Guide: Testing ConnectApi Code

`getMentionValidations(communityId, parentId, recordIds, visibility)`

Get information about whether the mentions are valid for the context user.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Mentions getMentionValidations(String communityId, String
parentId, List<String> recordIds, ConnectApi.FeedItemVisibilityType visibility)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

parentId
Type: String

The feed item parent ID.

recordIds
Type: List<String>

A comma-separated list of IDs to be mentioned. The maximum value is 25.

visibility
Type: `ConnectApi.FeedItemVisibilityType`

Type of users who can see a feed item.

- `AllUsers`—Visibility is not limited to internal users.
- `InternalUsers`—Visibility is limited to internal users.

Return Value

Type: `ConnectApi.MentionValidations`

Usage

Call this method to check whether the record IDs returned from a call to `ConnectApi.Mentions.getMentionCompletions` are valid for the context user. For example, the context users can't mention private groups they don't belong to. If such a group were included in the list of mention validations, the `ConnectApi.MentionValidations.hasErrors` property would be `true` and the group would have a `ConnectApi.MentionValidation.valdiationStatus` of `Disallowed`.

Mentions Test Methods

These test methods are for `Mentions`. All methods are static.

For information about using these methods to test your `ConnectApi` code, see Testing ConnectApi Code.

`setTestGetMentionCompletions(communityId, q, contextId, result)`

Register a `ConnectApi.MentionCompletionPage` object to be returned when `getMentionCompletions(String, String, String)` is called in a test context.

API Version

29.0

Signature

```
public static Void setTestGetMentionCompletions (String communityId, String q, String contextId, ConnectApi.MentionCompletionPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

A search term for matching user and group names. Searching for a group requires a minimum of 2 characters. Searching for a user doesn't require a minimum number of characters. This parameter does not support wildcards.

contextId
Type: String

A feed item ID (for a mention in a comment) or a feed subject ID (for a mention in a feed item) that narrows search results, with more useful results listed first. Use a group ID for groups that allow customers to ensure mention completion results include customers.

result
Type: `ConnectApi.MentionCompletionPage`

A `ConnectApi.MentionCompletionPage` object containing test data.

Return Value

Type: Void

SEE ALSO:

getMentionCompletions(communityId, q, contextId)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetMentionCompletions(communityId, q, contextId, type, pageParam, pageSize, result)`

Register a `ConnectApi.MentionCompletionPage` object to be returned when `getMentionCompletions(String, String, String, ConnectApi.MentionCompletionType, Integer, Integer)` is called in a test context.

API Version

29.0

Signature

```
public static Void setTestGetMentionCompletions (String communityId, String q, String contextId, ConnectApi.MentionCompletionType type, Integer pageParam, Integer pageSize, ConnectApi.MentionCompletionPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

A search term for matching user and group names. Searching for a group requires a minimum of 2 characters. Searching for a user doesn't require a minimum number of characters. This parameter does not support wildcards.

contextId
Type: String

A feed item ID (for a mention in a comment) or a feed subject ID (for a mention in a feed item) that narrows search results, with more useful results listed first. Use a group ID for groups that allow customers to ensure mention completion results include customers.

type
Type: `ConnectApi.MentionCompletionType`

Type of mention completion.

- `All`—All mention completions, regardless of the type of record to which the mention refers.
- `Group`—Mention completions for groups.
- `User`—Mention completions for users.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

result
Type: `ConnectApi.MentionCompletionPage`

A `ConnectApi.MentionCompletionPage` object containing test data.

Return Value

Type: Void

SEE ALSO:

getMentionCompletions(communityId, q, contextId, type, pageParam, pageSize)

Apex Developer Guide: Testing ConnectApi Code

Missions Class

Export and purge mission activity for users. Get a user's mission progress. Update mission activity counts for users.

Namespace

ConnectApi

Missions Methods

These methods are for `Missions`. All methods are static.

All methods in this class require Chatter and are subject to the per user, per namespace, per hour rate limit.

IN THIS SECTION:

exportUserMissionsActivities(communityId, userId)
Export mission activity for a user.

getUserMissionsProgress(communityId, userId)
Get mission activity progress for a user.

purgeUserMissionsActivities(communityId, userId)
Start a job to purge mission activity for a user.

purgeUserMissionsActivities(communityId)
Start a job to purge mission activity for all users.

updateUserMissionActivityCount(activityType, activityCount, communityId, userId)
Update the mission activity count for a user.

`exportUserMissionsActivities(communityId, userId)`

Export mission activity for a user.

API Version

45.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserMissionActivitiesJob exportUserMissionsActivities(String communityId, String userId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID of the user.

Return Value

Type: `ConnectApi.UserMissionActivitiesJob`

Usage

You can export these activities with this method.

- `FeedItemAnswerAQuestion`—User answered a question.
- `FeedItemLikeSomething`—User liked a post or comment.
- `FeedItemMarkAnswerAsBest`—User marked an answer as the best answer.
- `FeedItemPostQuestion`—User posted a question.
- `FeedItemReceiveAComment`—User received a comment on a post.
- `FeedItemReceiveALike`—User received a like on a post or comment.
- `FeedItemReceiveAnAnswer`—User received an answer to a question.
- `FeedItemWriteAComment`—User commented on a post.
- `FeedItemWriteAPost`—User made a post.
- `FeedItemYourAnswerMarkedBest`—User's answer was marked as the best answer.

`getUserMissionsProgress(communityId, userId)`

Get mission activity progress for a user.

API Version

46.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserMissionActivityCollection getUserMissionsProgress(String communityId, String userId)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`userId`
Type: String

ID of the user.

Return Value

Type: `ConnectApi.UserMissionActivityCollection`

`purgeUserMissionsActivities(communityId, userId)`

Start a job to purge mission activity for a user.

API Version

45.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserMissionActivitiesJob purgeUserMissionsActivities(String communityId, String userId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID of the user.

Return Value

Type: `ConnectApi.UserMissionActivitiesJob`

Usage

This method purges these activities.

- `FeedItemAnswerAQuestion`—User answered a question.
- `FeedItemLikeSomething`—User liked a post or comment.
- `FeedItemMarkAnswerAsBest`—User marked an answer as the best answer.
- `FeedItemPostQuestion`—User posted a question.
- `FeedItemReceiveAComment`—User received a comment on a post.
- `FeedItemReceiveALike`—User received a like on a post or comment.
- `FeedItemReceiveAnAnswer`—User received an answer to a question.
- `FeedItemWriteAComment`—User commented on a post.
- `FeedItemWriteAPost`—User made a post.
- `FeedItemYourAnswerMarkedBest`—User's answer was marked as the best answer.

`purgeUserMissionsActivities(communityId)`

Start a job to purge mission activity for all users.

API Version

49.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserMissionActivitiesJob purgeUserMissionsActivities(String communityId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

Return Value

Type: `ConnectApi.UserMissionActivitiesJob`

Usage

This method purges these activities.

- `FeedItemAnswerAQuestion`—User answered a question.
- `FeedItemLikeSomething`—User liked a post or comment.
- `FeedItemMarkAnswerAsBest`—User marked an answer as the best answer.
- `FeedItemPostQuestion`—User posted a question.
- `FeedItemReceiveAComment`—User received a comment on a post.
- `FeedItemReceiveALike`—User received a like on a post or comment.
- `FeedItemReceiveAnAnswer`—User received an answer to a question.
- `FeedItemWriteAComment`—User commented on a post.
- `FeedItemWriteAPost`—User made a post.
- `FeedItemYourAnswerMarkedBest`—User's answer was marked as the best answer.

`updateUserMissionActivityCount(activityType, activityCount, communityId, userId)`

Update the mission activity count for a user.

API Version

45.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserMissionActivityStatus
updateUserMissionActivityCount(ConnectApi.UserMissionActivityType activityType, Integer
activityCount, String communityId, String userId)
```

Parameters

activityType
Type: ConnectApi.UserMissionActivityType

Type of mission activity for a user. Values are:

- FeedItemAnswerAQuestion—User answered a question.
- FeedItemLikeSomething—User liked a post or comment.
- FeedItemMarkAnswerAsBest—User marked an answer as the best answer.
- FeedItemPostQuestion—User posted a question.
- FeedItemReceiveAComment—User received a comment on a post.
- FeedItemReceiveALike—User received a like on a post or comment.
- FeedItemReceiveAnAnswer—User received an answer to a question.
- FeedItemWriteAComment—User commented on a post.
- FeedItemWriteAPost—User made a post.
- FeedItemYourAnswerMarkedBest—User's answer was marked as the best answer.

activityCount
Type: Integer

Number of mission activities of the specified type for the user.

communityId
Type: String

ID for an Experience Cloud site, internal, or null.

userId
Type: String

ID of the user.

Return Value

Type: ConnectApi.UserMissionActivityStatus

NamedCredentials Class

Create, refresh, get, delete, replace, and update credentials. Create and get external credentials. Create and get named credentials. Create, get, delete, and update external auth identity providers. Get the URL for the OAuth token flow for an external credential.

Note: Managed packages can access only the named credentials and external credentials that are included in or created from the package's Apex code. If a managed package tries to access non-packaged named credentials and external credentials that a Salesforce admin created in the org, an error occurs.

Namespace

ConnectApi

NamedCredentials Methods

These methods are for `NamedCredentials`. All methods are static.

IN THIS SECTION:

createCredential(requestBody)
Create a credential.

createCredential(requestBody, action)
Refresh an OAuth or AWS Roles Anywhere credential.

createExternalAuthIdentityProvider(requestBody)
Create an external auth identity provider.

createExternalAuthIdentityProviderCredentials(fullName, requestBody)
Create external auth identity provider credentials.

createExternalCredential(requestBody)
Create an external credential.

createNamedCredential(requestBody)
Create a named credential.

deleteCredential(externalCredential, principalName, principalType)
Delete a credential.

deleteCredential(externalCredential, principalName, principalType, authenticationParameters)
Delete a credential with authentication parameters.

deleteExternalAuthIdentityProvider(fullName)
Delete an external auth identity provider.

getCredential(externalCredential, principalName, principalType)
Get a credential.

getExternalAuthIdentityProvider(fullName)
Get an external auth identity provider.

getExternalAuthIdentityProviderCredentials(fullName)
Get external auth identity provider credentials.

getExternalAuthIdentityProviders()
Get a list of external auth identity providers in the org.

getExternalCredential(developerName)
Get an external credential, including the named credentials and principals associated with it and the type and status of each principal.

getExternalCredentials()
Get external credentials that the user can authenticate to.

getNamedCredential(developerName)
Get a named credential.

getNamedCredentials()
Get a list of named credentials for the org.

getOAuthCredentialAuthUrl(requestBody)
Get the URL for the OAuth token flow for an external credential.

patchCredential(requestBody)
Update custom credentials.

updateCredential(requestBody)
Replace a credential.

updateExternalAuthIdentityProvider(fullName, requestBody)
Update an external auth identity provider.

updateExternalAuthIdentityProviderCredentials(fullName, requestBody)
Replace external auth identity provider credentials.

SEE ALSO:

Salesforce Help: Named Credentials

Named Credentials Developer Guide: Get Started with Named Credentials

Named Credentials Developer Guide: Named Credential API Links

Apex Developer Guide: Invoking Callouts Using Apex

Apex Developer Guide: Named Credentials as Callout Endpoints

`createCredential(requestBody)`

Create a credential.

API Version

56.0

Requires Chatter

No

Signature

```
public static ConnectApi.Credential createCredential(ConnectApi.CredentialInput
requestBody)
```

Parameters

`requestBody`
Type: `ConnectApi.CredentialInput`

A `ConnectApi.CredentialInput` class.

Return Value

Type: `ConnectApi.Credential`

createCredential(requestBody, action)

Refresh an OAuth or AWS Roles Anywhere credential.

API Version

58.0

Requires Chatter

No

Signature

```
public static ConnectApi.Credential createCredential(ConnectApi.CredentialInput
requestBody, ConnectApi.CreateCredentialAction action)
```

Parameters

requestBody
Type: ConnectApi.CredentialInput

A ConnectApi.CredentialInput class.

action
Type: ConnectApi.CreateCredentialAction

Action to take when creating the credential. Value is:

- Refresh

Return Value

Type: ConnectApi.Credential

createExternalAuthIdentityProvider(requestBody)

Create an external auth identity provider.

API Version

62.0

Requires Chatter

No

Signature

```
public static ConnectApi.ExternalAuthIdentityProvider
createExternalAuthIdentityProvider(ConnectApi.ExternalAuthIdentityProviderInput
requestBody)
```

Parameters

requestBody
Type: ConnectApi.ExternalAuthIdentityProviderInput on page 1858

A ConnectApi.ExternalAuthIdentityProviderInput input class.

Return Value

Type: ConnectApi.ExternalAuthIdentityProvider on page 2086

createExternalAuthIdentityProviderCredentials(fullName, requestBody)

Create external auth identity provider credentials.

API Version

62.0

Requires Chatter

No

Signature

```
public static ConnectApi.ExternalAuthIdentityProviderCredentials
createExternalAuthIdentityProviderCredentials(String fullName,
ConnectApi.ExternalAuthIdentityProviderCredentialsInput requestBody)
```

Parameters

fullName
Type: String

Full name of the external auth identity provider to create credentials for.

requestBody
Type: ConnectApi.ExternalAuthIdentityProviderCredentialsInput on page 1857

A ConnectApi.ExternalAuthIdentityProviderCredentialsInput input class

Return Value

Type: ConnectApi.ExternalAuthIdentityProviderCredentials on page 2088

createExternalCredential(requestBody)

Create an external credential.

API Version

58.0

Requires Chatter

No

Signature

```
public static ConnectApi.ExternalCredential
createExternalCredential(ConnectApi.ExternalCredentialInput requestBody)
```

Parameters

requestBody
Type: ConnectApi.ExternalCredentialInput

Input used to create or update an external credential.

Return Value

Type: ConnectApi.ExternalCredential

createNamedCredential(requestBody)

Create a named credential.

API Version

58.0

Requires Chatter

No

Signature

```
public static ConnectApi.NamedCredential
createNamedCredential(ConnectApi.NamedCredentialInput requestBody)
```

Parameters

requestBody
Type: ConnectApi.NamedCredentialInput

Input used to create or update a named credential.

Return Value

Type: ConnectApi.NamedCredential

deleteCredential(externalCredential, principalName, principalType)

Delete a credential.

This method deletes the user external credentials that store the encrypted access tokens used for named credential callouts, not the external credential itself. You can delete an external credential only in the UI or by using REST API.

API Version

56.0

Requires Chatter

No

Signature

```
public static Void deleteCredential(String externalCredential, String principalName,
ConnectApi.CredentialPrincipalType principalType)
```

Parameters

externalCredential
Type: String

Fully qualified developer name of the external credential.

principalName
Type: String

Name of the external credential named principal.

principalType
Type: `ConnectApi.CredentialPrincipalType`

Type of credential principal. Values are:

- `AwsStsPrincipal`
- `NamedPrincipal`
- `PerUserPrincipal`

Return Value

Type: Void

`deleteCredential(externalCredential, principalName, principalType, authenticationParameters)`

Delete a credential with authentication parameters.

This method deletes the user external credentials that store the encrypted access tokens used for named credential callouts, not the external credential itself. You can delete an external credential only in the UI or by using REST API.

API Version

59.0

Requires Chatter

No

Signature

```
public static Void deleteCredential(String externalCredential, String principalName,
ConnectApi.CredentialPrincipalType principalType, List<String> authenticationParameters)
```

Parameters

externalCredential
Type: String

Fully qualified developer name of the external credential.

principalName
Type: String

Name of the external credential named principal.

principalType
Type: ConnectApi.CredentialPrincipalType

Type of credential principal. Values are:

- `AwsStsPrincipal`
- `NamedPrincipal`
- `PerUserPrincipal`

authenticationParameters
Type: List<String>

List of authentication parameters only for custom protocols, for example `myApiKey,myApiSecret`. If unspecified, all credentials are deleted.

Return Value

Type: Void

deleteExternalAuthIdentityProvider(fullName)

Delete an external auth identity provider.

API Version

62.0

Requires Chatter

No

Signature

```
public static Void deleteExternalAuthIdentityProvider(String fullName)
```

Parameters

fullName
Type: String

Full name of the external auth identity provider to delete.

Return Value

Type: Void

getCredential(externalCredential, principalName, principalType)

Get a credential.

API Version

56.0

Requires Chatter

No

Signature

```
public static ConnectApi.Credential getCredential(String externalCredential, String
principalName, ConnectApi.CredentialPrincipalType principalType)
```

Parameters

externalCredential
Type: String

Fully qualified developer name of the external credential.

principalName
Type: String

Name of the external credential named principal.

principalType
Type: ConnectApi.CredentialPrincipalType

Type of credential principal. Values are:

- AwsStsPrincipal
- NamedPrincipal
- PerUserPrincipal

Return Value

Type: ConnectApi.Credential

getExternalAuthIdentityProvider(fullName)

Get an external auth identity provider.

API Version

62.0

Requires Chatter

No

Signature

```
public static ConnectApi.ExternalAuthIdentityProvider
getExternalAuthIdentityProvider(String fullName)
```

Parameters

fullName
Type: String

Full name of the external auth identity provider.

Return Value

Type: ConnectApi.ExternalAuthIdentityProvider on page 2086

getExternalAuthIdentityProviderCredentials(fullName)

Get external auth identity provider credentials.

API Version

62.0

Requires Chatter

No

Signature

```
public static ConnectApi.ExternalAuthIdentityProviderCredentials
getExternalAuthIdentityProviderCredentials(String fullName)
```

Parameters

fullName
Type: String

Full name of the external auth identity provider.

Return Value

Type: `ConnectApi.ExternalAuthIdentityProviderCredentials` on page 2088

`getExternalAuthIdentityProviders()`

Get a list of external auth identity providers in the org.

API Version

62.0

Requires Chatter

No

Signature

```
public static ConnectApi.ExternalAuthIdentityProviderList
getExternalAuthIdentityProviders()
```

Return Value

Type: `ConnectApi.ExternalAuthIdentityProviderList` on page 2088

`getExternalCredential(developerName)`

Get an external credential, including the named credentials and principals associated with it and the type and status of each principal.

API Version

56.0

Requires Chatter

No

Signature

```
public static ConnectApi.ExternalCredential getExternalCredential(String developerName)
```

Parameters

`developerName`
Type: String

Fully qualified developer name of the external credential.

Return Value

Type: `ConnectApi.ExternalCredential`

getExternalCredentials()

Get external credentials that the user can authenticate to.

API Version

56.0

Requires Chatter

No

Signature

```
public static ConnectApi.ExternalCredentialList getExternalCredentials()
```

Return Value

Type: ConnectApi.ExternalCredentialList

getNamedCredential(developerName)

Get a named credential.

API Version

59.0

Requires Chatter

No

Signature

```
public static ConnectApi.NamedCredential getNamedCredential(String developerName)
```

Parameters

developerName
Type: String

Fully qualified developer name of the named credential.

Return Value

Type: ConnectApi.NamedCredential

getNamedCredentials()

Get a list of named credentials for the org.

API Version

59.0

Requires Chatter

No

Signature

```
public static ConnectApi.NamedCredentialList getNamedCredentials()
```

Return Value

Type: ConnectApi.NamedCredentialList

getOAuthCredentialAuthUrl(requestBody)

Get the URL for the OAuth token flow for an external credential.

API Version

56.0

Requires Chatter

No

Signature

```
public static ConnectApi.OAuthCredentialAuthUrl
getOAuthCredentialAuthUrl(ConnectApi.OAuthCredentialAuthUrlInput requestBody)
```

Parameters

requestBody
Type: ConnectApi.OAuthCredentialAuthUrlInput

A `ConnectApi.OAuthCredentialAuthUrlInput` class indicating the OAuth authentication flow.

Return Value

Type: ConnectApi.OAuthCredentialAuthUrl

Usage

Accepts input parameters representing a specific external credential and, optionally, a named principal. Returns the URL a user must visit to begin the authentication flow, ultimately returning authentication tokens to Salesforce. Use this method as part of building a customized or branded user interface to help users initiate authentication.

Example

```
ConnectApi.OAuthCredentialAuthUrlInput input = new ConnectApi.OAuthCredentialAuthUrlInput();

input.externalCredential = 'MyExternalCredentialDeveloperName';
input.principalType = ConnectApi.CredentialPrincipalType.PerUserPrincipal;
input.principalName = 'MyPrincipal'; // Only required when principalType = NamedPrincipal

ConnectApi.OAuthCredentialAuthUrl output =
ConnectApi.NamedCredentials.getOAuthCredentialAuthUrl(input);

String authenticationUrl = output.authenticationUrl; // Redirect users to this URL to
authenticate in the browser
```

`patchCredential(requestBody)`

Update custom credentials.

This method updates custom credentials. To replace a credential, use `updateCredential(requestBody)`.

API Version

59.0

Requires Chatter

No

Signature

```
public static ConnectApi.Credential patchCredential(ConnectApi.CredentialInput
requestBody)
```

Parameters

requestBody
Type: `ConnectApi.CredentialInput`

A `ConnectApi.CredentialInput` class. Only the custom credentials in the input class are updated.

Return Value

Type: `ConnectApi.Credential`

`updateCredential(requestBody)`

Replace a credential.

This method uses the `ConnectApi.CredentialInput` and the `ConnectApi.CredentialValueInput` input classes to replace a credential's values. In the UI, these values appear as the credential's authentication parameters. To update a credential, use `patchCredential(requestBody)`.

API Version

56.0

Requires Chatter

No

Signature

```
public static ConnectApi.Credential updateCredential(ConnectApi.CredentialInput
requestBody)
```

Parameters

requestBody
Type: ConnectApi.CredentialInput

A ConnectApi.CredentialInput class.

Return Value

Type: ConnectApi.Credential

updateExternalAuthIdentityProvider(fullName, requestBody)

Update an external auth identity provider.

API Version

62.0

Requires Chatter

No

Signature

```
public static ConnectApi.ExternalAuthIdentityProvider
updateExternalAuthIdentityProvider(String fullName,
ConnectApi.ExternalAuthIdentityProviderInput requestBody)
```

Parameters

fullName
Type: String

Full name of the external auth identity provider to update.

requestBody
Type: ConnectApi.ExternalAuthIdentityProviderInput on page 1858

A ConnectApi.ExternalAuthIdentityProviderInput input class.

Return Value

Type: `ConnectApi.ExternalAuthIdentityProvider` on page 2086

`updateExternalAuthIdentityProviderCredentials(fullName, requestBody)`

Replace external auth identity provider credentials.

API Version

62.0

Requires Chatter

No

Signature

```
public static ConnectApi.ExternalAuthIdentityProviderCredentials
updateExternalAuthIdentityProviderCredentials(String fullName,
ConnectApi.ExternalAuthIdentityProviderCredentialsInput requestBody)
```

Parameters

`fullName`
Type: String

The external auth identity provider credentials to replace.

`requestBody`
Type: `ConnectApi.ExternalAuthIdentityProviderCredentialsInput` on page 1857

A `ConnectApi.ExternalAuthIdentityProviderCredentialsInput` input class.

Return Value

Type: `ConnectApi.ExternalAuthIdentityProviderCredentials` on page 2088

NavigationMenu Class

Get navigation menu items for an Experience Cloud site.

Namespace

ConnectApi

NavigationMenu Methods

These methods are for `NavigationMenu`. All methods are static.

IN THIS SECTION:

getCommunityNavigationMenu(communityId, navigationLinkSetId, navigationLinkSetDeveloperName, publishStatus, includeImageUrl, addHomeMenuItem, menuItemTypesToSkip)

Get navigation menu items for an Experience Cloud site.

getCommunityNavigationMenu(communityId, navigationLinkSetId, navigationLinkSetDeveloperName, publishStatus, includeImageUrl, addHomeMenuItem, menuItemTypesToSkip, effectiveAccountId)

Get navigation menu items for an Experience Cloud based on an effective account.

getCommunityNavigationMenu(communityId, navigationLinkSetId, navigationLinkSetDeveloperName, publishStatus, includeImageUrl, addHomeMenuItem, menuItemTypesToSkip)

Get navigation menu items for an Experience Cloud site.

API Version

52.0

Available to Guest Users

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.NavigationMenuItemCollection getCommunityNavigationMenu(String communityId, String navigationLinkSetId, String navigationLinkSetDeveloperName, ConnectApi.PublishStatus publishStatus, Boolean includeImageUrl, Boolean addHomeMenuItem, List<ConnectApi.NavigationMenuItemType> menuItemTypesToSkip)
```

Parameters

communityId
Type: String

ID of an Experience Cloud site.

navigationLinkSetId
Type: String

ID of the navigation link set.

navigationLinkSetDeveloperName
Type: String

Developer name of the navigation link set.

publishStatus
Type: ConnectApi.PublishStatus

Publish status of the navigation menu item. Values are:

- `Draft`
- `Live`

`includeImageUrl`
Type: Boolean

Specifies whether to include the image URL with the menu item (`true`) or not (`false`).

`addHomeMenuItem`
Type: Boolean

Specifies whether to add the Home menu item (`true`) or not (`false`).

`menuItemTypesToSkip`
Type: List<`ConnectApi.NavigationMenuItemType`>

List of menu item types to filter out of the results. Values are:

- `DataSourceDriven`—Menu items dynamically added from a data source.
- `Event`—Event, such as logging in, logging out, or switching accounts.
- `ExternalLink`—URL outside of your site.
- `GlobalAction`—Lets users create records that aren't related to other records.
- `InternalLink`—Relative URL inside your site.
- `MenuLabel`—Menu label.
- `Modal`—Modal, such as Account Switcher.
- `NavigationalTopic`—Dropdown list with links to the navigational topics in your site.
- `SalesforceObject`—Objects such as accounts, cases, contacts, and custom objects.
- `SystemLink`—System link, such as a link to Builder, Workspaces, or Setup.

Return Value

Type: `ConnectApi.NavigationMenuItemCollection`

Usage

Supported navigation menu item types are:

- `DataSourceDriven`—Menu items dynamically added from a data source.
- `Event`—Event, such as logging in, logging out, or switching accounts.
- `ExternalLink`—URL outside of your site.
- `GlobalAction`—Lets users create records that aren't related to other records.
- `InternalLink`—Relative URL inside your site.
- `MenuLabel`—Menu label.
- `Modal`—Modal, such as Account Switcher.
- `NavigationalTopic`—Dropdown list with links to the navigational topics in your site.
- `SalesforceObject`—Objects such as accounts, cases, contacts, and custom objects.
- `SystemLink`—System link, such as a link to Builder, Workspaces, or Setup.

`getCommunityNavigationMenu(communityId, navigationLinkSetId, navigationLinkSetDeveloperName, publishStatus, includeImageUrl, addHomeMenuItem, menuItemTypesToSkip, effectiveAccountId)`

Get navigation menu items for an Experience Cloud based on an effective account.

API Version

54.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.NavigationMenuItemCollection getCommunityNavigationMenu(String communityId, String navigationLinkSetId, String navigationLinkSetDeveloperName, ConnectApi.PublishStatus publishStatus, Boolean includeImageUrl, Boolean addHomeMenuItem, List<ConnectApi.NavigationMenuItemType> menuItemTypesToSkip, String effectiveAccountId)
```

Parameters

communityId
Type: String

ID of an Experience Cloud site.

navigationLinkSetId
Type: String

ID of the navigation link set.

navigationLinkSetDeveloperName
Type: String

Developer name of the navigation link set.

publishStatus
Type: `ConnectApi.PublishStatus`

Publish status of the navigation menu item. Values are:

- `Draft`
- `Live`

includeImageUrl
Type: Boolean

Specifies whether to include the image URL with the menu item (`true`) or not (`false`).

addHomeMenuItem
Type: Boolean

Specifies whether to add the Home menu item (`true`) or not (`false`).

`menuItemTypesToSkip`
Type: List<`ConnectApi.NavigationMenuItemType`>

List of menu item types to filter out of the results. Values are:

- `DataSourceDriven`—Menu items dynamically added from a data source.
- `Event`—Event, such as logging in, logging out, or switching accounts.
- `ExternalLink`—URL outside of your site.
- `GlobalAction`—Lets users create records that aren't related to other records.
- `InternalLink`—Relative URL inside your site.
- `MenuLabel`—Menu label.
- `Modal`—Modal, such as Account Switcher.
- `NavigationalTopic`—Dropdown list with links to the navigational topics in your site.
- `SalesforceObject`—Objects such as accounts, cases, contacts, and custom objects.
- `SystemLink`—System link, such as a link to Builder, Workspaces, or Setup.

`effectiveAccountId`
Type: String

ID of the account for which the request is made. If unspecified, defaults to the account ID for the context user.

Return Value

Type: `ConnectApi.NavigationMenuItemCollection`

Usage

Supported navigation menu item types are:

- `DataSourceDriven`—Menu items dynamically added from a data source.
- `Event`—Event, such as logging in, logging out, or switching accounts.
- `ExternalLink`—URL outside of your site.
- `GlobalAction`—Lets users create records that aren't related to other records.
- `InternalLink`—Relative URL inside your site.
- `MenuLabel`—Menu label.
- `Modal`—Modal, such as Account Switcher.
- `NavigationalTopic`—Dropdown list with links to the navigational topics in your site.
- `SalesforceObject`—Objects such as accounts, cases, contacts, and custom objects.
- `SystemLink`—System link, such as a link to Builder, Workspaces, or Setup.

NextBestAction Class

Execute recommendation strategies, get recommendations, manage recommendation reactions.

Namespace

ConnectApi

Usage

Community users can't access this class. Portal and guest users can access strategies only through the Suggested Actions component.

NextBestAction Methods

These methods are for `NextBestAction`. All methods are static.

IN THIS SECTION:

deleteRecommendationReaction(reactionId)
Delete a recommendation reaction.

executeStrategy(strategyName, maxResults, contextRecordId)
Execute a strategy.

executeStrategy(strategyName, maxResults, contextRecordId, debugTrace)
Execute a strategy and request a trace.

executeStrategy(strategyName, strategyInput)
Execute a strategy using an input class.

getRecommendation(recommendationId)
Get a recommendation.

getRecommendationReaction(reactionId)
Get a recommendation reaction.

getRecommendationReactions(onBehalfOfId, createdById, targetId, contextRecordId, pageParam, pageSize)
Get recommendation reactions.

setRecommendationReaction(reaction)
Record user reactions to recommendations.

`deleteRecommendationReaction(reactionId)`

Delete a recommendation reaction.

API Version

45.0

Requires Chatter

No

Signature

```
public static Void deleteRecommendationReaction(String reactionId)
```

Parameters

`reactionId`
Type: String

ID of the recommendation reaction or other sObject that caused the user to react.

Return Value

Type: Void

Usage

Users with the Manage Next Best Action Recommendations or Modify All Data permission can delete recommendation reactions.

executeStrategy(strategyName, maxResults, contextRecordId)

Execute a strategy.

API Version

45.0

Available to Guest Users

45.0

Requires Chatter

No

Signature

```
public static ConnectApi.NBARecommendations executeStrategy(String strategyName, Integer
maxResults, String contextRecordId)
```

Parameters

strategyName
Type: String

Name of the strategy.

maxResults
Type: Integer

Maximum number of results. Valid values are from 1 to 25. The default is 3.

contextRecordId
Type: String

ID of the context record. For example, if the next best action is on a case detail page, the ID of the case.

Return Value

Type: `ConnectApi.NBARecommendations`

executeStrategy(strategyName, maxResults, contextRecordId, debugTrace)

Execute a strategy and request a trace.

API Version

45.0

Available to Guest Users

45.0

Requires Chatter

No

Signature

```
public static ConnectApi.NBARecommendations executeStrategy(String strategyName, Integer maxResults, String contextRecordId, Boolean debugTrace)
```

Parameters

strategyName
Type: String

Name of the strategy.

maxResults
Type: Integer

Maximum number of results. Valid values are from 1 to 25. The default is 3.

contextRecordId
Type: String

ID of the context record. For example, if the next best action is on a case detail page, the ID of the case.

debugTrace
Type: Boolean

Specifies whether to return trace and debug information in the response (`true`) or not (`false`).

Return Value

Type: `ConnectApi.NBARecommendations`

executeStrategy(strategyName, strategyInput)

Execute a strategy using an input class.

API Version

45.0

Available to Guest Users

45.0

Requires Chatter

No

Signature

```
public static ConnectApi.NBARecommendations executeStrategy(String strategyName,
ConnectApi.NBAStrategyInput strategyInput)
```

Parameters

strategyName
Type: String

Name of the strategy.

strategyInput
Type: ConnectApi.NBAStrategyInput

A ConnectApi.NBAStrategyInput body.

Return Value

Type: ConnectApi.NBARecommendations

getRecommendation(recommendationId)

Get a recommendation.

API Version

45.0

Requires Chatter

No

Signature

```
public static ConnectApi.Recommendation getRecommendation(String recommendationId)
```

Parameters

recommendationId
Type: String

ID of the recommendation.

Return Value

Type: ConnectApi.Recommendation

getRecommendationReaction(reactionId)

Get a recommendation reaction.

API Version

45.0

Requires Chatter

No

Signature

```
public static ConnectApi.RecommendationReaction getRecommendationReaction(String reactionId)
```

Parameters

reactionId
Type: String

ID of the recommendation reaction or other sObject that caused the user to react.

Return Value

Type: ConnectApi.RecommendationReaction

Usage

Users with the Manage Next Best Action Recommendations or Modify All Data permission can get recommendation reactions.

getRecommendationReactions(onBehalfOfId, createdById, targetId, contextRecordId, pageParam, pageSize)

Get recommendation reactions.

API Version

45.0

Requires Chatter

No

Signature

```
public static ConnectApi.RecommendationReactions getRecommendationReactions(String onBehalfOfId, String createdById, String targetId, String contextRecordId, Integer pageParam, Integer pageSize)
```

Parameters

onBehalfOfId
Type: String

Use the ID of the user who is indirectly reacting to the recommendation to filter the results.

createdById
Type: String

Use the ID of the user or record that created the recommendation reaction to filter the results.

targetId
Type: String

Use the ID of the target to filter the results.

contextRecordId
Type: String

Use the ID of a context record to filter the results.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.RecommendationReactions`

Usage

Users with the Manage Next Best Action Recommendations or Modify All Data permission can get recommendation reactions.

`setRecommendationReaction(reaction)`

Record user reactions to recommendations.

API Version

45.0

Available to Guest Users

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.RecommendationReaction
setRecommendationReaction(ConnectApi.RecommendationReactionInput reaction)
```

Parameters

reaction
Type: ConnectApi.RecommendationReactionInput

A ConnectApi.RecommendationReactionInput object representing a reaction to a recommendation produced by a recommendation strategy.

Return Value

Type: ConnectApi.RecommendationReaction

OmnichannelInventoryService Class

Route orders to inventory locations in Order Management.

Namespace

ConnectApi

OmnichannelInventoryService Methods

These methods are for OmnichannelInventoryService. All methods are static.

IN THIS SECTION:

createReservation(createReservationInputRepresentation)
Create an inventory reservation in Omnichannel Inventory.

fulfillReservation(fulfillReservationInputRepresentation)
Fulfill one or more inventory reservations.

getInventoryAvailability(inventoryAvailabilityInputRepresentation)
Retrieve inventory availability data for one or more products at one or more inventory locations or location groups.

getInventoryAvailabilityUploadStatus(uploadId)
Retrieve the status of an inventory availability upload job.

getPublishLocationStructureStatus(uploadId)
Retrieve the status of a publish location structure job.

publishLocationStructure()

Asynchronously publish information about your inventory locations and location groups to Omnichannel Inventory. The publish includes records whose IsEnabled and ShouldSyncWithOci fields are both `true`. This method returns an ID that you can use to retrieve the status of the publish job.

releaseReservation(releaseReservationInputRepresentation)

Release one or more existing inventory reservations to free up that inventory.

submitInventoryAvailabilityUpload(fileUpload)

Upload an inventory availability data file to Omnichannel Inventory.

transferReservation(transferReservationInputRepresentation)

Transfer one or more inventory reservations between locations or location groups. This API doesn't change physical quantities, but reduces the reserved quantity at the source and increases it at the destination.

updateReservation(updateReservationInputRepresentation)

Updates an existing reservation in Omnichannel Inventory. Add, remove, and update quantities for existing SKUs in the reservation.

createReservation(createReservationInputRepresentation)

Create an inventory reservation in Omnichannel Inventory.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OCICreateReservationOutputRepresentation
createReservation(ConnectApi.OCICreateReservationInputRepresentation
createReservationInputRepresentation)
```

Parameters

createReservationInputRepresentation

Type: ConnectApi.OCICreateReservationInputRepresentation

Data to reserve inventory at one or more Omnichannel Inventory locations or location groups.

Return Value

Type: ConnectApi.OCICreateReservationOutputRepresentation

fulfillReservation(fulfillReservationInputRepresentation)

Fulfill one or more inventory reservations.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OCIFulfillReservationOutputRepresentation
fulfillReservation(ConnectApi.OCIFulfillReservationInputRepresentation
fulfillReservationInputRepresentation)
```

Parameters

fulfillReservationInputRepresentation
Type: ConnectApi.OCIFulfillReservationInputRepresentation

Wraps a list of inventory reservations to fulfill.

Return Value

Type: ConnectApi.OCIFulfillReservationOutputRepresentation

getInventoryAvailability(inventoryAvailabilityInputRepresentation)

Retrieve inventory availability data for one or more products at one or more inventory locations or location groups.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OCIGetInventoryAvailabilityOutputRepresentation
getInventoryAvailability(ConnectApi.OCIGetInventoryAvailabilityInputRepresentation
inventoryAvailabilityInputRepresentation)
```

Parameters

inventoryAvailabilityInputRepresentation
Type: ConnectApi.OCIGetInventoryAvailabilityInputRepresentation

Details of a request to retrieve inventory availability.

Return Value

Type: ConnectApi.OCIGetInventoryAvailabilityOutputRepresentation

getInventoryAvailabilityUploadStatus(uploadId)

Retrieve the status of an inventory availability upload job.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OCIUploadInventoryAvailabilityStatusOutputRepresentation
getInventoryAvailabilityUploadStatus(String uploadId)
```

Parameters

uploadId
Type: String

The upload ID of the upload job.

Return Value

Type: ConnectApi.OCIUploadInventoryAvailabilityStatusOutputRepresentation

getPublishLocationStructureStatus(uploadId)

Retrieve the status of a publish location structure job.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OCIPublishLocationStructureStatusOutputRepresentation
getPublishLocationStructureStatus(String uploadId)
```

Parameters

uploadId
Type: String

The upload ID of the publish job.

Return Value

Type: ConnectApi.OCIPublishLocationStructureStatusOutputRepresentation

publishLocationStructure()

Asynchronously publish information about your inventory locations and location groups to Omnichannel Inventory. The publish includes records whose IsEnabled and ShouldSyncWithOci fields are both *true*. This method returns an ID that you can use to retrieve the status of the publish job.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OCIPublishLocationStructureOutputRepresentation
publishLocationStructure()
```

Return Value

Type: ConnectApi.OCIPublishLocationStructureOutputRepresentation

releaseReservation(releaseReservationInputRepresentation)

Release one or more existing inventory reservations to free up that inventory.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OCIReleaseReservationOutputRepresentation
releaseReservation(ConnectApi.OCIReleaseReservationInputRepresentation
releaseReservationInputRepresentation)
```

Parameters

releaseReservationInputRepresentation
Type: ConnectApi.OCIReleaseReservationInputRepresentation

Details of one or more inventory reservations to release.

Return Value

Type: `ConnectApi.OCIReleaseReservationOutputRepresentation`

`submitInventoryAvailabilityUpload(fileUpload)`

Upload an inventory availability data file to Omnichannel Inventory.

API Version

51.0 (NDJSON), 63.0 (CSV)

Requires Chatter

No

Signature

```
public static ConnectApi.OCIUploadInventoryAvailabilityOutputRepresentation
submitInventoryAvailabilityUpload(ConnectApi.BinaryInput fileUpload)
```

Parameters

`fileUpload`
Type: `ConnectApi.BinaryInput`

NDJSON or CSV file containing inventory availability data.

Return Value

Type: `ConnectApi.OCIUploadInventoryAvailabilityOutputRepresentation`

Usage

To create an inventory data file, format the data as a series of NDJSON or CSV entries that represent locations and individual inventory records.

Inventory Import Data Considerations:

- Separate the top-level entries with line feeds, not commas. Each entry must be on a single line.
- When the system reads a location entry, it assigns the subsequent inventory entries to that location until it reads another location entry.
- Legacy NDJSON requires that you specify a header record for each location entry (`{"location":"wickenburg","mode":"UPDATE"}`). The header isn't required for a high-performance NDJSON layout or CSV file.
- Each inventory record entry requires a unique recordId. Best practice is to use a UUID. The recordId protects against importing duplicate data. The recordId is provided in NDJSON and automatically generated for CSV.
- Each inventory record entry requires an effectiveDate.
- If provided, each futures entry requires a nonzero quantity and a future expectedDate.

Note: The file must be in NDJSON or CSV format. For larger collections, use the Commerce API or split the data into multiple files. The Commerce API accepts GZIP, NDJSON, or CSV files up to 100 MB.

This example illustrates the data format:

Note: For readability, this example shows the first few entries on multiple lines. In the import file, each location and inventory record entry must be on a single line.

```
{
  "location":"Warehouse-A",  // location identifier
  "mode":"UPDATE"  // must be UPDATE (other operations might be available in future releases)
}
{
  "recordId":"0a87539d-f3dd-47bc-91c7-9c752e39dbe0",  // unique identifier for the inventory
 record
  "onHand":10,
  "sku":"12389156",
  "effectiveDate":"2020-12-08T14:05:22.790896-07:00",
  "futures":[  // list of future restocks
    {
    "quantity":1,
    "expectedDate":"2021-04-18T14:05:22.781-07:00"
    },
    {
    "quantity":5,
    "expectedDate":"2021-05-18T14:05:22.781-07:00"
    }
  ],
  "safetyStockCount":0
}
{
  "recordId":"0a87539d-f3dd-47bc-91c7-9c752e312345",
  "onHand":10,
  "sku":"9485728",
  "effectiveDate":"2020-12-08T14:05:22.790896-07:00",
  "futures":[
    {
    "quantity":10,
    "expectedDate":"2021-04-18T14:05:22.781-07:00"
    }
  ],
  "safetyStockCount":0
}
{"location":"Warehouse-B","mode":"UPDATE"}
{"recordId":"0a87539d-f3dd-47bc-91c7-9c752e312345","onHand":10,"sku":"12389156","effectiveDate":"2020-12-08T14:05:22.790896-07:00","futures":[{"quantity":1,"expectedDate":"2021-04-18T14:05:22.781-07:00"},"safetyStockCount":0}
{"recordId":"0a87539d-f3dd-47bc-91c7-9c752e312345","onHand":10,"sku":"9485728","effectiveDate":"2020-12-08T14:05:22.790896-07:00","futures":[{"quantity":5,"expectedDate":"2021-04-18T14:05:22.781-07:00"},"safetyStockCount":0}
```

`transferReservation(transferReservationInputRepresentation)`

Transfer one or more inventory reservations between locations or location groups. This API doesn't change physical quantities, but reduces the reserved quantity at the source and increases it at the destination.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.OCITransferReservationOutputRepresentation
transferReservation(ConnectApi.OCITransferReservationInputRepresentation
transferReservationInputRepresentation)
```

Parameters

transferReservationInputRepresentation
Type: ConnectApi.OCITransferReservationInputRepresentation

Wraps a list of inventory reservation transfers and specifies whether a single failure cancels the entire list.

Return Value

Type: ConnectApi.OCITransferReservationOutputRepresentation

updateReservation(updateReservationInputRepresentation)

Updates an existing reservation in Omnichannel Inventory. Add, remove, and update quantities for existing SKUs in the reservation.

API Version

61.0

Requires Chatter

No

Signature

```
public static ConnectApi.OCIUpdateReservationOutputRepresentation
updateReservation(ConnectApi.OCIUpdateReservationInputRepresentation
updateReservationInputRepresentation)
```

Parameters

updateReservationInputRepresentation
Type: ConnectApi.OCIUpdateReservationInputRepresentation on page 1901

Data to update one or more Omnichannel Inventory item reservations.

Return Value

Type: ConnectApi.OCIUpdateReservationOutputRepresentation on page 2202

OMSAnalytics Class

Get products with return rates, get text classified into different classifications using text analysis, and capture the return reasons from external sources based on the product ids.

Namespace

ConnectApi

IN THIS SECTION:

OMSAnalytics Class
These methods are for `OMSAnalytics`. All methods are static.

OMSAnalytics Class

These methods are for `OMSAnalytics`. All methods are static.

Namespace

ConnectApi

IN THIS SECTION:

getTextClassificationsBulkResults(ids)
Gets text classification results for request IDs.

productsExpand(scope, products, expand)
Fetches expanded details of a product that aren't found in the sObject. The expanded variable fields, such as return reasons, are added as output. This method supports an extensibility framework that lets the context user override the existing implementation so they can fetch the data from third-party apps. The application doesn't require two separate APIs to get return reasons.

productsReturnRate(pageParam, pageSize)
Gets pages of data showing the return rates of products that are calculated by the Customer Data Platform. Return data is paginated in descending order.

productsReturnRate(pageParam, pageSize)
Get a page of products and their return rates.

productsReturnRate(pageParam, pageSize, products)
Get a page of products and their return rates for a list of product IDs.

submitTextClassificationsRequest(textClassificationsRequestInput, llmType)
Submits a text classification request to Einstein

`getTextClassificationsBulkResults(ids)`

Gets text classification results for request IDs.

API Version

59.0

Requires Chatter

No

Signature

```
public static ConnectApi.TextClassificationsBulkResultsOutputRepresentation
getTextClassificationsBulkResults(List<String> ids)
```

Parameters

ids
Type: List(String)

List of request IDs.

Return Value

Type: ConnectApi.TextClassificationsBulkResultsOutputRepresentation

Example

```
List <String> requestIds = new List <String> ();
        requestIds.add(requestId);
        ConnectApi.TextClassificationsBulkResultsOutputRepresentation output =
ConnectApi.OMSAnalytics.getTextClassificationsBulkResults(requestIds);
```

productsExpand(scope, products, expand)

Fetches expanded details of a product that aren't found in the sObject. The expanded variable fields, such as return reasons, are added as output. This method supports an extensibility framework that lets the context user override the existing implementation so they can fetch the data from third-party apps. The application doesn't require two separate APIs to get return reasons.

API Version

59.0

Requires Chatter

No

Signature

```
public static ConnectApi.ProductsListOutputRepresentation productsExpand(String scope,
List<String> products, List<ConnectApi.ProductExpandType> expand)
```

Parameters

scope
Type: String

The scope for the extensibility framework. Requires a web store ID.

products
Type: List)String)

A list of IDs to fetch details for.

expand
Type: List)String)

Output representation for expand feature.

Return Value

Type: `ConnectApi.ProductsListOutputRepresentation`

Example

```
String scope = 'webstoreId eq ' + ws.Id;
        ConnectApi.ProductsListOutputRepresentation output =
ConnectApi.OMSAnalytics.productsExpand(scope, productIds, new List
<ConnectApi.ProductExpandType> {
        ConnectApi.ProductExpandType.ReturnReasons
        });
```

productsReturnRate(pageParam, pageSize)

Gets pages of data showing the return rates of products that are calculated by the Customer Data Platform. Return data is paginated in descending order.

API Version

59.0

Requires Chatter

No

Signature

```
public static ConnectApi.ProductReturnRateListOutputRepresentation
productsReturnRate(Integer pageParam, Integer pageSize)
```

Parameters

page
Type: String

The page number for the list of products. Starts at 0.

products
Type: List)String)

A list of IDs to fetch details for.

pageSize
Type: List)Integer)

The number of products that are returned on each page.

Return Value

Type: ConnectApi.ProductReturnRateListOutputRepresentation

Example

```
ConnectApi.ProductReturnRateListOutputRepresentation output =
ConnectApi.OMSAnalytics.productsReturnRate(page, pageSize);
```

productsReturnRate(pageParam, pageSize)

Get a page of products and their return rates.

API Version

59.0

Requires Chatter

No

Signature

```
public static ConnectApi.ProductReturnRateListOutputRepresentation
productsReturnRate(Integer pageParam, Integer pageSize)
```

Parameters

pageParam
Type: Integer

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.ProductReturnRateListOutputRepresentation`

productsReturnRate(pageParam, pageSize, products)

Get a page of products and their return rates for a list of product IDs.

API Version

60.0

Requires Chatter

No

Signature

```
public static ConnectApi.ProductReturnRateListOutputRepresentation
productsReturnRate(Integer pageParam, Integer pageSize, List<String> products)
```

Parameters

pageParam
Type: Integer

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

products
Type: List<String>

List of product IDs.

Return Value

Type: `ConnectApi.ProductReturnRateListOutputRepresentation`

`submitTextClassificationsRequest(textClassificationsRequestInput, llmType)`

Submits a text classification request to Einstein

API Version

59.0

Requires Chatter

No

Signature

```
public static ConnectApi.TextClassificationsOutputRepresentation
submitTextClassificationsRequest(ConnectApi.TextClassificationsInputRepresentation
textClassificationsRequestInput, String llmType)
```

Parameters

textClassificationsRequestInput
Type: ConnectApi.TextClassificationsInputRepresentation

Text classification containing a list of text strings and classifiers. Each text string is classified into classifiers based on analysis.

llmType
Type: List)String)

The large language model that's used for analysis. Supports Open AI only.

Return Value

Type: ConnectApi.TextClassificationsOutputRepresentation

Example

```
ConnectApi.TextClassificationsInputRepresentation textClassificationsInputRepresentation
= new ConnectApi.TextClassificationsInputRepresentation();
textClassificationsInputRepresentation.textList = textList;
textClassificationsInputRepresentation.classifiers = classifiers;
List < String > requestIds = new List < String > ();
```

Orchestration Class

Get orchestration instances.

Namespace

ConnectApi

Orchestration Methods

These methods are for `Orchestration`. All methods are static.

IN THIS SECTION:

getOrchestrationInstance(instanceId)
Get an orchestration instance associated with an orchestration instance ID.

getOrchestrationInstanceCollection(relatedRecordId)
Get orchestration instances associated with a Salesforce record that's configured as a context record for orchestration interactive steps.

`getOrchestrationInstance(instanceId)`

Get an orchestration instance associated with an orchestration instance ID.

API Version

63.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrchestrationInstance getOrchestrationInstance(String
instanceId)
```

Parameters

instanceId
Type: String

The ID of orchestration instance to get details for.

Return Value

Type: ConnectApi.OrchestrationInstance

getOrchestrationInstanceCollection(relatedRecordId)

Get orchestration instances associated with a Salesforce record that's configured as a context record for orchestration interactive steps.

API Version

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrchestrationInstanceCollection
getOrchestrationInstanceCollection(String relatedRecordId)
```

Parameters

relatedRecordId
Type: String

The ID of a record configured as a context record for orchestration interactive steps.

Return Value

Type: ConnectApi.OrchestrationInstanceCollection

OrderPaymentSummary Class

Work with payments in Order Management.

Namespace

ConnectApi

OrderPaymentSummary Methods

These methods are for `OrderPaymentSummary`. All methods are static.

IN THIS SECTION:

createOrderPaymentSummary(orderPaymentSummaryInput)
Create an OrderPaymentSummary for an OrderSummary. Specify a payment authorization or payments that share the same payment method. In an org with the multicurrency feature enabled, the OrderPaymentSummary inherits the CurrencyIsoCode value from the OrderSummary.

createOrderPaymentSummary(orderPaymentSummaryInput)

Create an OrderPaymentSummary for an OrderSummary. Specify a payment authorization or payments that share the same payment method. In an org with the multicurrency feature enabled, the OrderPaymentSummary inherits the CurrencyIsoCode value from the OrderSummary.

API Version

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.CreateOrderPaymentSummaryOutputRepresentation
createOrderPaymentSummary(ConnectApi.CreateOrderPaymentSummaryInputRepresentation
orderPaymentSummaryInput)
```

Parameters

orderPaymentSummaryInput
Type: `ConnectApi.CreateOrderPaymentSummaryInputRepresentation`

The OrderSummary and payment authorization or payments.

Return Value

Type: `ConnectApi.CreateOrderPaymentSummaryOutputRepresentation`

Example

```
String orderSummaryId = '1Osxx0000004CCG';
String paymentId1 = '0a3xx0000000085AAA';
String paymentId2 = '0a3xx0000000085BBB';

ConnectApi.CreateOrderPaymentSummaryInputRepresentation orderPaymentSummaryInput = new
ConnectApi.CreateOrderPaymentSummaryInputRepresentation();
orderPaymentSummaryInput.orderSummaryId = orderSummaryId;
List<String> paymentList = new List<String>();
```

```
paymentList.add(paymentId1);
paymentList.add(paymentId2);
orderPaymentSummaryInput.paymentIds = paymentList;

ConnectApi.CreateOrderPaymentSummaryOutputRepresentation result =
ConnectAPI.OrderPaymentSummary.createOrderPaymentSummary(orderPaymentSummaryInput);
```

OrderSummary Class

Work with orders in Order Management.

Namespace

ConnectApi

OrderSummary Methods

These methods are for `OrderSummary`. All methods are static.

IN THIS SECTION:

adjustPreview(orderSummaryId, adjustInput)

Retrieve the expected results of adjusting the price of one or more OrderItemSummaries from an OrderSummary, without actually executing the adjustment. The response data contains the financial changes that would result from submitting the proposed adjustment.

adjustSubmit(orderSummaryId, adjustInput)

Adjust the price of one or more OrderItemSummaries from an OrderSummary, and create corresponding change orders.

createCreditMemo(orderSummaryId, creditMemoInput)

Create a credit memo to represent the refund for one or more change orders associated with an OrderSummary.

createMultipleInvoices(invoicesInput)

Create Invoices to represent the charges for one or more change orders. Create Invoices for change orders that increase order amounts, such as for return fees. When you ensure the refund for a return, include the invoices for any associated return fees in the request.

ensureFundsAsync(orderSummaryId, ensureFundsInput)

Ensure funds for an Invoice and apply them to it. If needed, capture authorized funds by sending a request to a payment provider. This method inserts a background operation into an asynchronous job queue and returns the ID of that operation so you can track its status. Payment gateway responses appear in the payment gateway log and do not affect the background operation status.

ensureRefundsAsync(orderSummaryId, ensureRefundsInput)

Ensure refunds for a CreditMemo or excess funds by sending a request to a payment provider. This method inserts a background operation into an asynchronous job queue and returns the ID of that operation so you can track its status. Payment gateway responses appear in the payment gateway log and don't affect the background operation status.

multipleEnsureFundsAsync(multipleEnsureFundsInput)

Ensure and apply funds for one or more Invoices. If needed, capture authorized funds by sending a request to a payment provider. This method inserts a background operation into an asynchronous job queue and returns the ID of that operation so you can track its status. Payment gateway responses appear in the payment gateway log and do not affect the background operation status.

previewCancel(orderSummaryId, changeInput)

Retrieve the expected change order values for canceling one or more OrderItemSummaries from an OrderSummary, without actually executing the cancel.

previewReturn(orderSummaryId, changeInput)

Retrieve the expected change order values for a simple return of one or more OrderItemSummaries from an OrderSummary, without actually executing the return.

submitCancel(orderSummaryId, changeInput)

Cancel one or more OrderItemSummaries from an OrderSummary, and create a corresponding change order.

submitReturn(orderSummaryId, changeInput)

Return one or more OrderItemSummaries from an OrderSummary, and create a corresponding change order. This return is a simple return that creates a change order but not a ReturnOrder.

`adjustPreview(orderSummaryId, adjustInput)`

Retrieve the expected results of adjusting the price of one or more OrderItemSummaries from an OrderSummary, without actually executing the adjustment. The response data contains the financial changes that would result from submitting the proposed adjustment.

API Version

49.0

Requires Chatter

No

Signature

```
public static ConnectApi.AdjustOrderSummaryOutputRepresentation adjustPreview(String
orderSummaryId, ConnectApi.AdjustOrderItemSummaryInputRepresentation adjustInput)
```

Parameters

orderSummaryId

Type: String

ID of the OrderSummary.

adjustInput

Type: `ConnectApi.AdjustOrderItemSummaryInputRepresentation`

Price adjustments to order item summaries that together make up a price adjustment to an order, with options for adjusting items in the process of being fulfilled.

Return Value

Type: `ConnectApi.AdjustOrderSummaryOutputRepresentation`

Usage

When a price adjustment is applied to an OrderItemSummary, its quantities are considered in three groups:

Pre-fulfillment
QuantityAvailableToFulfill, which is equal to QuantityOrdered - QuantityCanceled - QuantityAllocated

In-fulfillment
QuantityAllocated - QuantityFulfilled

Post-fulfillment
QuantityAvailableToReturn, which is equal to QuantityFulfilled - QuantityReturnInitiated

You can apply adjustments to these groups in three different ways, controlled by the `allocatedItemsChangeOrderType` input property:

- Distribute the adjustment evenly between pre-fulfillment and post-fulfillment quantities. Ignore in-fulfillment quantities. Submitting the adjustment would create one change order for the adjustments to pre-fulfillment quantities and one change order for the adjustments to post-fulfillment quantities.
- Distribute the adjustment evenly between pre-fulfillment, in-fulfillment, and post-fulfillment quantities. Submitting the adjustment would create one change order for the adjustments to both pre-fulfillment and in-fulfillment quantities, and one change order for the adjustments to post-fulfillment quantities.
- Distribute the adjustment evenly between pre-fulfillment, in-fulfillment, and post-fulfillment quantities. Submitting the adjustment would create one change order for the adjustments to pre-fulfillment quantities, one change order for the adjustments to in-fulfillment quantities, and one change order for the adjustments to post-fulfillment quantities.

SEE ALSO:

createCreditMemo(orderSummaryId, creditMemoInput)

ensureRefundsAsync(orderSummaryId, ensureRefundsInput)

adjustSubmit(orderSummaryId, adjustInput)

`adjustSubmit(orderSummaryId, adjustInput)`

Adjust the price of one or more OrderItemSummaries from an OrderSummary, and create corresponding change orders.

API Version

49.0

Requires Chatter

No

Signature

```
public static ConnectApi.AdjustOrderSummaryOutputRepresentation adjustSubmit(String
orderSummaryId, ConnectApi.AdjustOrderItemSummaryInputRepresentation adjustInput)
```

Parameters

`orderSummaryId`
Type: String

ID of the OrderSummary.

adjustInput
Type: ConnectApi.AdjustOrderItemSummaryInputRepresentation

Price adjustments to order item summaries that together make up a price adjustment to an order, with options for adjusting items in the process of being fulfilled.

Return Value

Type: ConnectApi.AdjustOrderSummaryOutputRepresentation

Usage

When a price adjustment is applied to an OrderItemSummary, its quantities are considered in three groups:

Pre-fulfillment
QuantityAvailableToFulfill, which is equal to QuantityOrdered - QuantityCanceled - QuantityAllocated

In-fulfillment
QuantityAllocated - QuantityFulfilled

Post-fulfillment
QuantityAvailableToReturn, which is equal to QuantityFulfilled - QuantityReturnInitiated

You can apply adjustments to these groups in three different ways, controlled by the *allocatedItemsChangeOrderType* input property:

- Distribute the adjustment evenly between pre-fulfillment and post-fulfillment quantities. Ignore in-fulfillment quantities. Create one change order for the adjustments to pre-fulfillment quantities and one change order for the adjustments to post-fulfillment quantities.
- Distribute the adjustment evenly between pre-fulfillment, in-fulfillment, and post-fulfillment quantities. Create one change order for the adjustments to both pre-fulfillment and in-fulfillment quantities, and one change order for the adjustments to post-fulfillment quantities.
- Distribute the adjustment evenly between pre-fulfillment, in-fulfillment, and post-fulfillment quantities. Create one change order for the adjustments to pre-fulfillment quantities, one change order for the adjustments to in-fulfillment quantities, and one change order for the adjustments to post-fulfillment quantities.

After submitting a price adjustment, process refunds as appropriate:

- If the discount only applied to OrderItemSummaries for which payment hasn't been captured, it doesn't require a refund. This situation normally applies to OrderItemSummaries in the US that haven't been fulfilled.
- If the discount applied to OrderItemSummaries that haven't been fulfilled and for which payment has been captured, process a refund. In this case, pass the *totalExcessFundsAmount* from the output representation to the ensureRefundsAsync() method.
- If the discount applied to OrderItemSummaries that have been fulfilled, process a refund. Pass the *postFulfillmentChangeOrderId* from the output representation to the createCreditMemo() method, then pass the CreditMemo to the ensureRefundsAsync() method.
- If the discount applied to both fulfilled and unfulfilled OrderItemSummaries for which payment has been captured, process both refunds. Pass the *postFulfillmentChangeOrderId* from the output representation to the createCreditMemo()

method, then pass the credit memo and the *totalExcessFundsAmount* from the output representation to the `ensureRefundsAsync()` method.

SEE ALSO:

createCreditMemo(orderSummaryId, creditMemoInput)

ensureRefundsAsync(orderSummaryId, ensureRefundsInput)

adjustPreview(orderSummaryId, adjustInput)

createCreditMemo(orderSummaryId, creditMemoInput)

Create a credit memo to represent the refund for one or more change orders associated with an OrderSummary.

API Version

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.CreateCreditMemoOutputRepresentation createCreditMemo(String
orderSummaryId, ConnectApi.CreateCreditMemoInputRepresentation creditMemoInput)
```

Parameters

orderSummaryId
Type: String

ID of the OrderSummary.

creditMemoInput
Type: `ConnectApi.CreateCreditMemoInputRepresentation`

The list of change order IDs.

Return Value

Type: `ConnectApi.CreateCreditMemoOutputRepresentation`

createMultipleInvoices(invoicesInput)

Create Invoices to represent the charges for one or more change orders. Create Invoices for change orders that increase order amounts, such as for return fees. When you ensure the refund for a return, include the invoices for any associated return fees in the request.

API Version

56.0

Requires Chatter

No

Signature

```
public static ConnectApi.CreateMultipleInvoicesFromChangeOrdersOutputRepresentation
createMultipleInvoices(ConnectApi.CreateMultipleInvoicesFromChangeOrdersInputRepresentation
invoicesInput)
```

Parameters

invoicesInput
Type: ConnectApi.CreateMultipleInvoicesFromChangeOrdersInputRepresentation

Data about the change orders to create Invoices for.

Return Value

Type: ConnectApi.CreateMultipleInvoicesFromChangeOrdersOutputRepresentation

SEE ALSO:

ensureRefundsAsync(orderSummaryId, ensureRefundsInput)

createReturnOrder(returnOrderInput)

returnItems(returnOrderId, returnItemsInput)

ensureFundsAsync(orderSummaryId, ensureFundsInput)

Ensure funds for an Invoice and apply them to it. If needed, capture authorized funds by sending a request to a payment provider. This method inserts a background operation into an asynchronous job queue and returns the ID of that operation so you can track its status. Payment gateway responses appear in the payment gateway log and do not affect the background operation status.

API Version

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.EnsureFundsAsyncOutputRepresentation ensureFundsAsync(String
orderSummaryId, ConnectApi.EnsureFundsAsyncInputRepresentation ensureFundsInput)
```

Parameters

isConsiderReservedBalanceAmount
Type: Boolean

If true, the reserved balance amount is used for the Order Summary to fund the invoice. If not enough reserved balance amount, any available balance that isn't reserved by another Order Summary is used. If false, any available balance is used.

orderSummaryId
Type: String

ID of the OrderSummary.

ensureFundsInput
Type: `ConnectApi.EnsureFundsAsyncInputRepresentation`

The ID of the Invoice.

Return Value

Type: `ConnectApi.EnsureFundsAsyncOutputRepresentation`

Usage

This method checks the OrderPaymentSummaries associated with the specified OrderSummary for funds to apply to the Invoice balance following this logic:

Note: If multiple OrderPaymentSummaries have equal `BalanceAmount` values, their order of selection is random.

1. Verify that the Invoice balance doesn't exceed the total `BalanceAmount` of all the OrderPaymentSummaries associated with the OrderSummary.
2. If an OrderPaymentSummary has a `BalanceAmount` equal to the Invoice balance, apply the funds from that OrderPaymentSummary.
3. If no exact match was found, apply funds from the OrderPaymentSummary with the largest `BalanceAmount`.
4. If the Invoice still has a balance to ensure, repeat steps 2 and 3 until the full balance is ensured or no captured funds remain.
5. If the Invoice still has a balance, look for an OrderPaymentSummary with an authorized amount equal to the remaining Invoice balance. If one exists, capture and apply the funds from that OrderPaymentSummary.
6. If no exact match was found, capture and apply funds from the OrderPaymentSummary with the largest authorized amount.
7. If the Invoice still has a balance to ensure, repeat steps 5 and 6 until the full balance is ensured.

Note: If the method creates a payment, the payment record's ClientContext value isn't predictable. Don't use it in custom logic.

SEE ALSO:

multipleEnsureFundsAsync(multipleEnsureFundsInput)

`ensureRefundsAsync(orderSummaryId, ensureRefundsInput)`

Ensure refunds for a CreditMemo or excess funds by sending a request to a payment provider. This method inserts a background operation into an asynchronous job queue and returns the ID of that operation so you can track its status. Payment gateway responses appear in the payment gateway log and don't affect the background operation status.

API Version

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.EnsureRefundsAsyncOutputRepresentation ensureRefundsAsync(String orderSummaryId, ConnectApi.EnsureRefundsAsyncInputRepresentation ensureRefundsInput)
```

Parameters

isConsiderReservedBalanceAmount
Type: Boolean

If true, the refundable amount is used to open the payment balance for the reservedBalanceAmount in the Order Payment Summaries. The remaining refundable amount considers the sequence of order payment summaries, if provided. If false, any reserved balance amount for exchanges is refunded.

orderSummaryId
Type: String

ID of the OrderSummary.

ensureRefundsInput
Type: `ConnectApi.EnsureRefundsAsyncInputRepresentation`

ID of a credit memo to ensure refunds for, an amount of excess funds to refund, or both. At least one is required. Also includes any invoices for fees that reduce the refund amount, such as return fees. If multiple payment methods are available, you can specify how to distribute the refund.

Return Value

Type: `ConnectApi.EnsureRefundsAsyncOutputRepresentation`

Usage

This method applies the refund to the OrderPaymentSummaries associated with the specified OrderSummary following this logic.

Note: If multiple OrderPaymentSummaries have equal `AvailableToRefund` amounts, their order of selection is random.

1. Verify that the CreditMemo balance and excess funds amount don't exceed the total `AvailableToRefund` amount of all the OrderPaymentSummaries associated with the OrderSummary.
2. If `sequences` is specified, follow these steps.
   a. Traverse the `sequences` list in order and apply the specified refund amounts to the specified OrderPaymentSummaries.
   b. If the specified CreditMemo and excess funds are fully refunded, or if `isAllowPartial` is true, then the action stops here.
3. If a CreditMemo is specified, follow these steps.
   a. If an OrderPaymentSummary has an `AvailableToRefund` amount matching the CreditMemo's remaining balance, apply the refund to that payment.
   b. If no exact match was found but one or more OrderPaymentSummary has a large enough `AvailableToRefund` amount to cover the balance, use the OrderPaymentSummary with the smallest `AvailableToRefund` amount.
   c. If no single OrderPaymentSummary has a large enough `AvailableToRefund` amount, use multiple OrderPaymentSummaries in descending order of `AvailableToRefund` amount. This ensures the fewest OrderPaymentSummaries are used.

4. If only one OrderPaymentSummary is specified but has multiple payments, follow these steps.
   a. If a payment has an amount matching the CreditMemo's remaining balance, apply the refund to that payment.
   b. If no exact match was found but one or more payment has a large enough amount to cover the balance, use the payment with the smallest amount.
   c. If no single payment has a large enough amount, use multiple payments in descending order of amount. This ensures the fewest payments are used.
5. If an excess funds amount is specified, follow these steps.
   a. Examine those OrderPaymentSummaries. If one has an `AvailableToRefund` amount matching the excess funds amount, apply the refund to that OrderPaymentSummary.
   b. If no exact match was found but one or more OrderPaymentSummary has a large enough `AvailableToRefund` amount to cover the balance, use the OrderPaymentSummary with the smallest `AvailableToRefund` amount.
   c. If no single OrderPaymentSummary has a large enough `AvailableToRefund` amount, use multiple OrderPaymentSummaries in descending order of `AvailableToRefund` amount. This ensures the fewest OrderPaymentSummaries are used.

Note: If the method creates a refund, the refund record's ClientContext value isn't predictable. Don't use it in custom logic.

SEE ALSO:

createReturnOrder(returnOrderInput)

returnItems(returnOrderId, returnItemsInput)

createMultipleInvoices(invoicesInput)

`multipleEnsureFundsAsync(multipleEnsureFundsInput)`

Ensure and apply funds for one or more Invoices. If needed, capture authorized funds by sending a request to a payment provider. This method inserts a background operation into an asynchronous job queue and returns the ID of that operation so you can track its status. Payment gateway responses appear in the payment gateway log and do not affect the background operation status.

API Version

56.0

Requires Chatter

No

Signature

```
public static ConnectApi.MultipleAsyncOutputRepresentation
multipleEnsureFundsAsync(ConnectApi.MultipleEnsureFundsAsyncInputRepresentation
multipleEnsureFundsInput)
```

Parameters

`multipleEnsureFundsInput`
Type: `ConnectApi.MultipleEnsureFundsAsyncInputRepresentation`

List of Invoices and the associated OrderSummaries.

Return Value

Type: `ConnectApi.MultipleAsyncOutputRepresentation`

Usage

For each Invoice in the request, this method checks the OrderPaymentSummaries associated with the specified OrderSummary for funds to apply to the Invoice balance following this logic.

Note: If multiple OrderPaymentSummaries have equal `BalanceAmount` values, their order of selection is random.

1. Verify that the Invoice balance doesn't exceed the total `BalanceAmount` of all the OrderPaymentSummaries associated with the OrderSummary.
2. If an OrderPaymentSummary has a `BalanceAmount` equal to the invoice balance, apply the funds from that OrderPaymentSummary.
3. If no exact match was found, apply funds from the OrderPaymentSummary with the largest `BalanceAmount`.
4. If the Invoice still has a balance to ensure, repeat steps 2 and 3 until the full balance is ensured or no captured funds remain.
5. If the Invoice still has a balance, look for an OrderPaymentSummary with an authorized amount equal to the remaining Invoice balance. If one exists, capture and apply the funds from that OrderPaymentSummary.
6. If no exact match was found, capture and apply funds from the OrderPaymentSummary with the largest authorized amount.
7. If the Invoice still has a balance to ensure, repeat steps 5 and 6 until the full balance is ensured.

Note: If the method creates a payment, the payment record's ClientContext value isn't predictable. Don't use it in custom logic.

SEE ALSO:

ensureFundsAsync(orderSummaryId, ensureFundsInput)

`previewCancel(orderSummaryId, changeInput)`

Retrieve the expected change order values for canceling one or more OrderItemSummaries from an OrderSummary, without actually executing the cancel.

API Version

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.PreviewCancelOutputRepresentation previewCancel(String
orderSummaryId, ConnectApi.ChangeInputRepresentation changeInput)
```

Parameters

orderSummaryId
Type: String

ID of the OrderSummary.

changeInput
Type: ConnectApi.ChangeInputRepresentation

A list of changes to OrderItemSummaries that make up an order change, such as a cancel or return.

Return Value

Type: ConnectApi.PreviewCancelOutputRepresentation

SEE ALSO:

createCreditMemo(orderSummaryId, creditMemoInput)

ensureRefundsAsync(orderSummaryId, ensureRefundsInput)

submitCancel(orderSummaryId, changeInput)

previewReturn(orderSummaryId, changeInput)

Retrieve the expected change order values for a simple return of one or more OrderItemSummaries from an OrderSummary, without actually executing the return.

API Version

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.PreviewReturnOutputRepresentation previewReturn(String
orderSummaryId, ConnectApi.ChangeInputRepresentation changeInput)
```

Parameters

orderSummaryId
Type: String

ID of the OrderSummary.

changeInput
Type: ConnectApi.ChangeInputRepresentation

A list of changes to OrderItemSummaries that make up an order change, such as a cancel or return.

Return Value

Type: `ConnectApi.PreviewReturnOutputRepresentation`

SEE ALSO:

createCreditMemo(orderSummaryId, creditMemoInput)

ensureRefundsAsync(orderSummaryId, ensureRefundsInput)

submitReturn(orderSummaryId, changeInput)

`submitCancel(orderSummaryId, changeInput)`

Cancel one or more OrderItemSummaries from an OrderSummary, and create a corresponding change order.

API Version

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.SubmitCancelOutputRepresentation submitCancel(String
orderSummaryId, ConnectApi.ChangeInputRepresentation changeInput)
```

Parameters

`orderSummaryId`
Type: String

ID of the OrderSummary.

`changeInput`
Type: `ConnectApi.ChangeInputRepresentation`

A list of changes to OrderItemSummaries that make up an order change, such as a cancel or return.

Return Value

Type: `ConnectApi.SubmitCancelOutputRepresentation`

SEE ALSO:

createCreditMemo(orderSummaryId, creditMemoInput)

ensureRefundsAsync(orderSummaryId, ensureRefundsInput)

previewCancel(orderSummaryId, changeInput)

`submitReturn(orderSummaryId, changeInput)`

Return one or more OrderItemSummaries from an OrderSummary, and create a corresponding change order. This return is a simple return that creates a change order but not a ReturnOrder.

API Version

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.SubmitReturnOutputRepresentation submitReturn(String
orderSummaryId, ConnectApi.ChangeInputRepresentation changeInput)
```

Parameters

orderSummaryId
Type: String

ID of the OrderSummary.

changeInput
Type: `ConnectApi.ChangeInputRepresentation`

A list of changes to OrderItemSummaries that make up an order change, such as a cancel or return.

Return Value

Type: `ConnectApi.SubmitReturnOutputRepresentation`

Usage

After submitting a return, process a refund. Pass the *changeOrderId* from the output representation to the `createCreditMemo()` method, then pass the credit memo to the `ensureRefundsAsync()` method.

SEE ALSO:

createCreditMemo(orderSummaryId, creditMemoInput)

ensureRefundsAsync(orderSummaryId, ensureRefundsInput)

previewReturn(orderSummaryId, changeInput)

OrderSummaryCreation Class

Create Order Summaries in Order Management.

Namespace

ConnectApi

OrderSummaryCreation Methods

These methods are for `OrderSummaryCreation`. All methods are static.

IN THIS SECTION:

createOrderSummary(orderSummaryInput)
Create an OrderSummary based on an order. That order is considered the original order for the OrderSummary. Subsequent change orders that apply to the OrderSummary are also represented as orders. You can specify whether the order is managed in Salesforce Order Management or by an external system. Most Salesforce Order Management APIs can run only on orders that it manages.

`createOrderSummary(orderSummaryInput)`

Create an OrderSummary based on an order. That order is considered the original order for the OrderSummary. Subsequent change orders that apply to the OrderSummary are also represented as orders. You can specify whether the order is managed in Salesforce Order Management or by an external system. Most Salesforce Order Management APIs can run only on orders that it manages.

API Version

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.OrderSummaryOutputRepresentation
createOrderSummary(ConnectApi.OrderSummaryInputRepresentation orderSummaryInput)
```

Parameters

`orderSummaryInput`
Type: `ConnectApi.OrderSummaryInputRepresentation`

Input object that wraps the ID of the source order.

Return Value

Type: `ConnectApi.OrderSummaryOutputRepresentation`

Organization Class

Access information about an org.

Namespace

ConnectApi

Organization Methods

These methods are for `Organization`. All methods are static.

IN THIS SECTION:

getSettings()
Get information about the context user and the org, including which features are enabled.

`getSettings()`

Get information about the context user and the org, including which features are enabled.

API Version

28.0

Requires Chatter

No

Signature

```
public static ConnectApi. OrganizationSettings getSettings()
```

Return Value

Type: `ConnectApi.OrganizationSettings`

PardotBusinessUnitContext Class

Get the Pardot business units the context user has access to.

Namespace

ConnectApi

PardotBusinessUnitContext Methods

These methods are for `PardotBusinessUnitContext`. All methods are static.

IN THIS SECTION:

getBusinessUnitContext()
Get the Pardot business units the context user has access to.
getBusinessUnitContextByIsCurrentStatus(isCurrent)
Get the Pardot business units the context user has access to by specifying the current status.

`getBusinessUnitContext()`

Get the Pardot business units the context user has access to.

API Version

55.0

Requires Chatter

No

Signature

```
public static ConnectApi.PardotBusinessUnitContextOutput getBusinessUnitContext()
```

Return Value

Type: ConnectApi.PardotBusinessUnitContextOutput

getBusinessUnitContextByIsCurrentStatus(isCurrent)

Get the Pardot business units the context user has access to by specifying the current status.

API Version

55.0

Requires Chatter

No

Signature

```
public static ConnectApi.PardotBusinessUnitContextOutput
getBusinessUnitContextByIsCurrentStatus(Boolean isCurrent)
```

Parameters

isCurrent
Type: Boolean

Specifies whether to return only the business unit that's selected as the context user's current business unit context in the business unit switcher of the Pardot Lightning app (`true`) or to return only the business units that aren't selected as the context user's current business unit context (`false`).

Return Value

Type: ConnectApi.PardotBusinessUnitContextOutput

Payments Class

Authorize a payment, capture an authorized payment, and refund an authorized payment.

Namespace

ConnectApi

Payments Methods

These methods are for `Payments`. All methods are static.

To access Payments methods, you need these permissions.

- Salesforce Order Management License
- PaymentsAPIUser user permission. This permission is available with the Salesforce Order Management License. Your Salesforce admin assigns it to your profile.

IN THIS SECTION:

authorize(authorizePayment)
Authorize a payment.

postAuth(postAuthorizePayment)
Confirms that the merchant is ready to capture payment of an existing pre-authorized transaction.

reverseAuthorization(AuthReversalInput, authorizationId)
Reverses a payment authorization.

capture(AuthCaptureInput, authorizationId)
Capture an authorized payment.

refund(ReferencedRefundInput, paymentId)
Refund an authorized payment.

sale(sale)
Captures a payment without any prior authorization and creates a payment entity. The payment sale transaction is a combination of an Authorize transaction and Capture transaction. This payment sale method allows merchants to request that the funds are transferred to the merchant account in a single command, with no further action (such as charging a credit card) from the merchant.

tokenizePaymentMethod(tokenizePaymentMethodInput)
Method to take the input parameters of the payment method you want to tokenize and then pass them to the payment gateway's tokenization service. The results of the tokenization request are returned as a response from the payment gateway.

`authorize(authorizePayment)`

Authorize a payment.

API Version

51.0

Requires Chatter

No

Signature

```
global static ConnectApi.AuthorizationResponse authorize(ConnectApi.AuthorizationRequest
authorizePayment)
```

Parameters

authorizePayment
Type: ConnectApi.AuthorizationRequest

Represents a payment authorization.

Return Value

Type: ConnectApi.AuthorizationResponse

postAuth(postAuthorizePayment)

Confirms that the merchant is ready to capture payment of an existing pre-authorized transaction.

API Version

54.0

Requires Chatter

No

Signature

```
global static ConnectApi.PostAuthorizationResponse postAuth(ConnectApi.PostAuthRequest
postAuthorizePayment)
```

Parameters

postAuthorizePayment
Type: ConnectApi.PostAuthRequest

Information about the payment, payment method, and payment gateway from the original payment authorization.

Return Value

Type: ConnectApi.PostAuthorizationResponse

reverseAuthorization(AuthReversalInput, authorizationId)

Reverses a payment authorization.

API Version

51.0

Requires Chatter

No

Signature

```
global static ConnectApi.AuthorizationReversalResponse
reverseAuthorization(ConnectApi.AuthorizationReversalRequest AuthReversalInput, String
authorizationId)
```

Parameters

AuthReversalInput
Type: ConnectApi.AuthorizationReversalRequest

Input information for the payment authorization reversal.

authorizationId
Type: String

The ID of the payment authorization to be reversed.

Return Value

Type: ConnectApi.AuthorizationReversalResponse

capture(AuthCaptureInput, authorizationId)

Capture an authorized payment.

API Version

50.0

Requires Chatter

No

Signature

```
global static ConnectApi.CaptureResponse capture(ConnectApi.CaptureRequest
AuthCaptureInput, String authorizationId)
```

Parameters

AuthCaptureInput
Type: ConnectApi.CaptureRequest

A `ConnectApi.CaptureRequest` object with information about the payment capture.

authorizationId
Type: String

ID of the payment authorization. Required.

Return Value

Type: ConnectApi.CaptureResponse

refund(ReferencedRefundInput, paymentId)

Refund an authorized payment.

To access Payments methods, you need these permissions.

- Salesforce Order Management License
- PaymentsAPIUser user permission. This permission is available with the Salesforce Order Management License. Your Salesforce admin assigns it to your profile.

API Version

50.0

Requires Chatter

No

Signature

```
global static ConnectApi.ReferencedRefundResponse
refund(ConnectApi.ReferencedRefundRequest ReferencedRefundInput, String paymentId)
```

Parameters

ReferencedRefundInput
Type: ConnectApi.ReferencedRefundRequest

A ConnectApi.ReferencedRefundRequest object with information about the refund.

paymentId
Type: String

ID of the payment to be refunded. Required.

Return Value

Type: ConnectApi.ReferencedRefundResponse

sale(sale)

Captures a payment without any prior authorization and creates a payment entity. The payment sale transaction is a combination of an Authorize transaction and Capture transaction. This payment sale method allows merchants to request that the funds are transferred to the merchant account in a single command, with no further action (such as charging a credit card) from the merchant.

API Version

54.0

Requires Chatter

No

Signature

```
global static ConnectApi.SaleResponse sale(ConnectApi.SaleRequest sale)
```

Parameters

sale
Type: ConnectApi.SaleRequest

Payment sale input class.

Return Value

Type: ConnectApi.SaleResponse

tokenizePaymentMethod(tokenizePaymentMethodInput)

Method to take the input parameters of the payment method you want to tokenize and then pass them to the payment gateway's tokenization service. The results of the tokenization request are returned as a response from the payment gateway.

API Version

52.0

Requires Chatter

No

Signature

```
global static ConnectApi.PaymentMethodTokenizationResponse
tokenizePaymentMethod(ConnectApi.PaymentMethodTokenizationRequest
tokenizePaymentMethodInput)
```

Parameters

tokenizePaymentMethodInput
Type: ConnectApi.PaymentMethodTokenizationRequest

Information about the payment method to be tokenized.

Return Value

Type: ConnectApi.PaymentMethodTokenizationResponse

Usage

Accepts input parameters representing a payment method and passes them in a tokenization request to the payment gateway. The results of the tokenization request are returned as a response from the payment gateway. If the tokenization was successful, the response

contains the tokenized value and details about the tokenization process. Otherwise, the response contains an error message and details about the error.

Example

```
ConnectApi.PaymentMethodTokenizationRequest request = new
ConnectApi.PaymentMethodTokenizationRequest();
request.paymentGatewayId = '0b0xx0000001Ja5AAE';
ConnectApi.CardPaymentMethodRequest cpmRequest = new ConnectApi.CardPaymentMethodRequest();
cpmRequest.cardHolderName = 'Jo Manager';
cpmRequest.expiryMonth = 11;
cpmRequest.expiryYear = 2222;
cpmRequest.cardNumber = '4111111111111111';
cpmRequest.cvv = '111';
cpmRequest.cardCategory = ConnectApi.CardCategory.CreditCard;
cpmRequest.cardType = ConnectApi.CardType.Visa.name();
request.cardPaymentMethod = cpmRequest;
ConnectApi.PaymentMethodTokenizationResponse response =
ConnectApi.Payments.tokenizePaymentMethod(request);
```

Personalization Class

Get assigned personalization audiences that match the user context. Create, get, update, and delete an audience. Get personalization targets that match the user context, based on the assigned audiences that include the user. Create and update targets. Get and delete a target.

Namespace

ConnectApi

Note: Personalization varies what the user can see in the browser but doesn't secure data in any way. To prevent users accessing sensitive data, use standard Salesforce security features, such as sharing rules and permission sets.

Personalization Methods

These methods are for `Personalization`. All methods are static.

IN THIS SECTION:

createAudience(communityId, audience)
Create an audience.

createTargets(communityId, target)
Create targets.

deleteAudience(communityId, audienceId)
Delete an audience.

deleteTarget(communityId, targetId)
Delete a target.

getAudience(communityId, audienceId, includeAudienceCriteria)
Get an audience.

getAudienceBatch(communityId, audienceIds)
Get audience information for a comma-separated list of audience IDs.

getAudiences(communityId, ipAddress, domain, userId, publishStatus, includeAudienceCriteria, targetTypes, recordId)
Get a list of assigned audiences that match the user context and record information.

getAudiences(communityId, ipAddress, domain, userId, publishStatus, includeAudienceCriteria, targetTypes)
Get a list of assigned audiences that match the user context.

getTarget(communityId, targetId)
Get a target.

getTargetBatch(communityId, targetIds)
Get target information for a comma-separated list of target IDs.

getTargets(communityId, ipAddress, domain, userId, publishStatus, recordId, targetTypes, includeAudience, includeAllMatchingTargetsWithinGroup, groupNames)
Get a list of targets that match the user context, based on the assigned audiences that include the user.

updateAudience(communityId, audienceId, audience)
Update an audience.

updateTargets(communityId, target)
Update targets.

`createAudience(communityId, audience)`

Create an audience.

API Version

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.Audience createAudience(String communityId,
ConnectApi.AudienceInput audience)
```

Parameters

communityId
Type: String

ID of the Experience Cloud site.

audience
Type: `ConnectApi.AudienceInput`

A `ConnectApi.AudienceInput` object that defines the audience.

Return Value

Type: ConnectApi.Audience

createTargets(communityId, target)

Create targets.

API Version

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.TargetCollection createTargets(String communityId,
ConnectApi.TargetCollectionInput target)
```

Parameters

communityId
Type: String

ID of the Experience Cloud site.

target
Type: ConnectApi.TargetCollectionInput

A `ConnectApi.TargetCollectionInput` object that defines the targets.

Return Value

Type: ConnectApi.TargetCollection

deleteAudience(communityId, audienceId)

Delete an audience.

API Version

48.0

Requires Chatter

No

Signature

```
public static Void deleteAudience(String communityId, String audienceId)
```

Parameters

communityId
Type: String

ID of the Experience Cloud site.

audienceId
Type: String

ID of the audience.

Return Value

Type: Void

deleteTarget(communityId, targetId)

Delete a target.

API Version

48.0

Requires Chatter

No

Signature

```
public static Void deleteTarget(String communityId, String targetId)
```

Parameters

communityId
Type: String

ID of the Experience Cloud site.

targetId
Type: String

ID of the target.

Return Value

Type: Void

getAudience(communityId, audienceId, includeAudienceCriteria)

Get an audience.

API Version

48.0

Available to Guest Users

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.Audience getAudience(String communityId, String audienceId,
Boolean includeAudienceCriteria)
```

Parameters

communityId
Type: String

ID of the Experience Cloud site.

audienceId
Type: String

ID of the audience.

includeAudienceCriteria
Type: Boolean

Specifies whether to include audience criteria (`true`) or not (`false`). If unspecified, defaults to `false`.

Return Value

Type: `ConnectApi.Audience`

getAudienceBatch(communityId, audienceIds)

Get audience information for a comma-separated list of audience IDs.

API Version

48.0

Available to Guest Users

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.BatchResult[] getAudienceBatch(String communityId, List<String>
audienceIds)
```

Parameters

communityId
Type: String

ID of the Experience Cloud site.

audienceIds
Type: List<String>

Comma-separated list of audience IDs.

Return Value

Type: `ConnectApi.BatchResult`[]

The `ConnectApi.BatchResult.getResult()` method returns a `ConnectApi.Audience` object and errors for audiences that didn't load.

getAudiences(communityId, ipAddress, domain, userId, publishStatus, includeAudienceCriteria, targetTypes, recordId)

Get a list of assigned audiences that match the user context and record information.

API Version

51.0

Available to Guest Users

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.AudienceCollection getAudiences(String communityId, String ipAddress, String domain, String userId, ConnectApi.PublishStatus publishStatus, Boolean includeAudienceCriteria, List<String> targetTypes, String recordId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

ipAddress
Type: String

IP address of the user. If `null`, no audiences with location criteria are returned.

domain
Type: String

Name of the user's Salesforce custom domain. If `null`, no audiences with domain criteria are returned.

`userId`
Type: String

ID of the user. If `null`, defaults to the ID of the context user.

`publishStatus`
Type: `ConnectApi.PublishStatus`

Publish status of the audience. Values are:

- `Draft`
- `Live`

If `null`, defaults to `Live`.

`includeAudienceCriteria`
Type: Boolean

Specifies whether to include audience criteria (`true`) or not (`false`). If unspecified, defaults to `false`.

`targetTypes`
Type: List<String>

Comma-separated list of target types to filter the results. Supported values include:

- `ExperienceVariation` (version 48.0 and later)
- Custom object API names, such as ***`CustomObjectName`***`__c` (version 48.0 and later)
- `NavigationLinkSet` (version 49.0 and later)
- `Topic` (version 49.0 and later)
- `CollaborationGroup` (version 49.0 and later)
- `KnowledgeArticle` (version 49.0 and later)
- `ContentDocument` (version 49.0 and later)
- `ManagedContent` (version 49.0 and later)
- `Report` (version 49.0 and later)
- `Dashboard` (version 49.0 and later)

If `null`, all target types are returned.

`recordId`
Type: String

ID of the record for field based criteria. If `null`, all applicable audiences with field based criteria are returned.

Return Value

Type: `ConnectApi.AudienceCollection`

`getAudiences(communityId, ipAddress, domain, userId, publishStatus, includeAudienceCriteria, targetTypes)`

Get a list of assigned audiences that match the user context.

API Version

48.0

Available to Guest Users

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.AudienceCollection getAudiences(String communityId, String
ipAddress, String domain, String userId, ConnectApi.PublishStatus publishStatus, Boolean
includeAudienceCriteria, List<String> targetTypes)
```

Parameters

communityId
Type: String

ID of the Experience Cloud site.

ipAddress
Type: String

IP address of the user. If `null`, no audiences with location criteria are returned.

domain
Type: String

Name of the user's Salesforce custom domain. If `null`, no audiences with domain criteria are returned.

userId
Type: String

ID of the user. If `null`, defaults to the ID of the context user.

publishStatus
Type: `ConnectApi.PublishStatus`

Publish status of the audience. Values are:

- `Draft`
- `Live`

If `null`, defaults to `Live`.

includeAudienceCriteria
Type: Boolean

Specifies whether to include audience criteria (`true`) or not (`false`). If unspecified, defaults to `false`.

targetTypes
Type: List<String>

Comma-separated list of target types to filter the results. Supported values include:

- `ExperienceVariation` (version 48.0 and later)

- Custom object API names, such as ***CustomObjectName***__c (version 48.0 and later)
- NavigationLinkSet (version 49.0 and later)
- Topic (version 49.0 and later)
- CollaborationGroup (version 49.0 and later)
- KnowledgeArticle (version 49.0 and later)
- ContentDocument (version 49.0 and later)
- ManagedContent (version 49.0 and later)
- Report (version 49.0 and later)
- Dashboard (version 49.0 and later)

If null, all target types are returned.

Return Value

Type: ConnectApi.AudienceCollection

getTarget(communityId, targetId)

Get a target.

API Version

48.0

Available to Guest Users

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.Target getTarget(String communityId, String targetId)
```

Parameters

communityId
Type: String

ID of the Experience Cloud site.

targetId
Type: String

ID of the target.

Return Value

Type: ConnectApi.Target

`getTargetBatch(communityId, targetIds)`

Get target information for a comma-separated list of target IDs.

API Version

48.0

Available to Guest Users

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.BatchResult[] getTargetBatch(String communityId, List<String>
targetIds)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

targetIds
Type: List<String>

Comma-separated list of target IDs.

Return Value

Type: `ConnectApi.BatchResult`

The `ConnectApi.BatchResult.getResult()` method returns a `ConnectApi.Target` object and errors for targets that didn't load.

`getTargets(communityId, ipAddress, domain, userId, publishStatus, recordId, targetTypes, includeAudience, includeAllMatchingTargetsWithinGroup, groupNames)`

Get a list of targets that match the user context, based on the assigned audiences that include the user.

API Version

48.0

Available to Guest Users

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.TargetCollection getTargets(String communityId, String
ipAddress, String domain, String userId, ConnectApi.PublishStatus publishStatus, String
recordId, List<String> targetTypes, Boolean includeAudience, Boolean
includeAllMatchingTargetsWithinGroup, List<String> groupNames)
```

Parameters

communityId
Type: String

ID of the Experience Cloud site.

ipAddress
Type: String

IP address of the user. If `null`, no audiences with location criteria are returned.

domain
Type: String

Name of the user's Salesforce custom domain. If `null`, no audiences with domain criteria are returned.

userId
Type: String

ID of the user. If `null`, the default is the ID of the context user.

publishStatus
Type: `ConnectApi.PublishStatus`

Publish status of the target. Values are:

- `Draft`
- `Live`

recordId
Type: String

ID of the record, if you want to specify field based criteria in audiences.

targetTypes
Type: List<String>

Comma-separated list of target types to filter the results. Supported values include:

- `ExperienceVariation` (version 48.0 and later)
- Custom object API names, such as ***`CustomObjectName`*****`__c`** (version 48.0 and later)
- `NavigationLinkSet` (version 49.0 and later)
- `Topic` (version 49.0 and later)
- `CollaborationGroup` (version 49.0 and later)
- `KnowledgeArticle` (version 49.0 and later)
- `ContentDocument` (version 49.0 and later)

- `ManagedContent` (version 49.0 and later)
- `Report` (version 49.0 and later)
- `Dashboard` (version 49.0 and later)

If `null`, all target types are returned.

`includeAudience`
Type: Boolean

Specifies whether to include the matching audience (`true`) or not (`false`). If `null`, the default is `false`.

`includeAllMatchingTargetsWithinGroup`
Type: Boolean

Specifies whether to include all the matching targets within a target group (`true`) or not (`false`). If `null`, the default is `false`. If `false`, the first matching target within each group, based on priority within the group, is returned.

`groupNames`
Type: List<String>

A comma-separated list of group names. Groups bundle related target and audience pairs.

Return Value

Type: `ConnectApi.TargetCollection`

`updateAudience(communityId, audienceId, audience)`

Update an audience.

API Version

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.Audience updateAudience(String communityId, String audienceId,
ConnectApi.AudienceInput audience)
```

Parameters

`communityId`
Type: String

ID of the Experience Cloud site.

`audienceId`
Type: String

ID of the audience.

`audience`
Type: `ConnectApi.AudienceInput`

A `ConnectApi.AudienceInput` object that defines the updates to the audience.

Return Value

Type: `ConnectApi.Audience`

`updateTargets(communityId, target)`

Update targets.

API Version

48.0

Requires Chatter

No

Signature

```
public static ConnectApi.TargetCollection updateTargets(String communityId,
ConnectApi.TargetCollectionUpdateInput target)
```

Parameters

`communityId`
Type: String

ID of the Experience Cloud site.

`target`
Type: `ConnectApi.TargetCollectionUpdateInput`

A `ConnectApi.TargetCollectionUpdateInput` object that defines the updates for the targets.

Return Value

Type: `ConnectApi.TargetCollection`

PickTicket Class

Create tickets to fulfill orders.

Namespace

ConnectApi

PickTicket Methods

These methods are for `PickTicket`. All methods are static.

IN THIS SECTION:

distributePickedQuantities(distributePickedQuantitiesInput)
Distribute picked quantities among orders in a pick ticket.

distributePickedQuantities(distributePickedQuantitiesInput)

Distribute picked quantities among orders in a pick ticket.

API Version

58.0

Requires Chatter

No

Signature

```
public static ConnectApi.DistributePickedQuantitiesOutputRepresentation
distributePickedQuantities(ConnectApi.DistributePickedQuantitiesInputRepresentation
distributePickedQuantitiesInput)
```

Parameters

distributePickedQuantitiesInput
Type: `ConnectApi.DistributePickedQuantitiesInputRepresentation`

Input to distribute picked quantities.

Return Value

Type: `ConnectApi.DistributePickedQuantitiesOutputRepresentation`

QuestionAndAnswers Class

Access question and answers suggestions.

Namespace

ConnectApi

IN THIS SECTION:

QuestionAndAnswers Methods
These methods are for `QuestionAndAnswers`. All methods are static.

QuestionAndAnswers Methods

These methods are for `QuestionAndAnswers`. All methods are static.

All methods in this class require Chatter and are subject to the per user, per namespace, per hour rate limit.

IN THIS SECTION:

getSuggestions(communityId, q, subjectId, includeArticles, maxResults)
Get question and answers suggestions.

setTestGetSuggestions(communityId, q, subjectId, includeArticles, maxResults, result)
Register a `ConnectApi.QuestionAndAnswersSuggestions` object to be returned when `getSuggestions` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

updateQuestionAndAnswers(communityId, feedElementId, questionAndAnswersCapability)
Choose or change the best answer for a question.

`getSuggestions(communityId, q, subjectId, includeArticles, maxResults)`

Get question and answers suggestions.

API Version

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.QuestionAndAnswersSuggestions getSuggestions(String communityId,
String q, String subjectId, Boolean includeArticles, Integer maxResults)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`q`
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

`subjectId`
Type: String

Specify a subject ID to search only questions on that object. If the ID is a topic or a user, the ID is ignored.

`includeArticles`
Type: Boolean

Specify `true` to include knowledge articles in the search results. To return only questions, specify `false`.

`maxResults`
Type: Integer

The maximum number of results to return for each type of item. Possible values are 1–10. The default value is 5.

Return Value

Type: `ConnectApi.QuestionAndAnswersSuggestions`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetSuggestions(communityId, q, subjectId, includeArticles, maxResults, result)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetSuggestions(communityId, q, subjectId, includeArticles, maxResults, result)`

Register a `ConnectApi.QuestionAndAnswersSuggestions` object to be returned when `getSuggestions` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

32.0

Signature

```
public static Void setTestGetSuggestions(String communityId, String q, String subjectId,
Boolean includeArticles, Integer maxResults, ConnectApi.QuestionAndAnswersSuggestions
result)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`q`
Type: String

Required and can't be `null`. Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

`subjectId`
Type: String

Specify a subject ID to search only questions on that object. If the ID is a topic or a user, the ID is ignored.

`includeArticles`
Type: Boolean

Specify `true` to include knowledge articles in the search results. To return only questions, specify `false`.

maxResults
Type: Integer

The maximum number of results to return for each type of item. Possible values are 1–10. The default value is 5.

result
Type: `ConnectApi.QuestionAndAnswersSuggestions`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getSuggestions(communityId, q, subjectId, includeArticles, maxResults)

Apex Developer Guide: Testing ConnectApi Code

updateQuestionAndAnswers(communityId, feedElementId, questionAndAnswersCapability)

Choose or change the best answer for a question.

API Version

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.QuestionAndAnswersCapability updateQuestionAndAnswers(String communityId, String feedElementId, ConnectApi.QuestionAndAnswersCapabilityInput questionAndAnswersCapability)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

feedElementId
Type: String

ID of the feed element.

questionAndAnswersCapability
Type: `ConnectApi.QuestionAndAnswersCapabilityInput`

Specify the best answer (comment ID) for the question.

Return Value

Type: `ConnectApi.QuestionAndAnswersCapability`

If the feed element doesn't support this capability, the return value is `ConnectApi.NotFoundException`.

Example

```
ConnectApi.QuestionAndAnswersCapabilityInput qaInput = new
ConnectApi.QuestionAndAnswersCapabilityInput();
qaInput.bestAnswerId = '0D7D000000001MAKAY';

ConnectApi.QuestionAndAnswersCapability qa =
ConnectApi.QuestionAndAnswers.updateQuestionAndAnswers(null, '0D5D0000000XZjJ', qaInput);
```

Recommendations Class

Get and reject Chatter, custom, and static recommendations. Create, get, update, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

For Next Best Action recommendations, see NextBestAction Class.

Namespace

ConnectApi

Recommendations Methods

These methods are for `Recommendations`. All methods are static.

All methods in this class require Chatter and are subject to the per user, per namespace, per hour rate limit.

IN THIS SECTION:

createRecommendationAudience(communityId, recommendationAudience)
Create an audience for a custom recommendation.

createRecommendationAudience(communityId, name)
Create an audience for a custom recommendation.

createRecommendationDefinition(communityId, recommendationDefinition)
Create a custom recommendation definition.

createRecommendationDefinition(communityId, name, title, actionUrl, actionUrlName, explanation)
Create a custom recommendation definition with the specified parameters.

createScheduledRecommendation(communityId, scheduledRecommendation)
Create a scheduled custom recommendation.

createScheduledRecommendation(communityId, recommendationDefinitionId, rank, enabled, recommendationAudienceId, channel)
Create a scheduled custom recommendation with the specified parameters.

deleteRecommendationAudience(communityId, recommendationAudienceId)
Delete a custom recommendation audience.

deleteRecommendationDefinition(communityId, recommendationDefinitionId)
Delete a custom recommendation definition.

deleteRecommendationDefinitionPhoto(communityId, recommendationDefinitionId)
Delete a custom recommendation definition photo.

deleteScheduledRecommendation(communityId, scheduledRecommendationId, deleteDefinitionIfLast)
Delete a scheduled custom recommendation.

getRecommendationAudience(communityId, recommendationAudienceId)
Get information about a custom recommendation audience.

getRecommendationAudienceMembership(communityId, recommendationAudienceId)
Get the members of a custom recommendation audience.

getRecommendationAudienceMembership(communityId, recommendationAudienceId, pageParam, pageSize)
Get a page of custom recommendation audience members.

getRecommendationAudiences(communityId)
Get custom recommendation audiences.

getRecommendationAudiences(communityId, pageParam, pageSize)
Get a page of custom recommendation audiences.

getRecommendationDefinition(communityId, recommendationDefinitionId)
Get a custom recommendation definition.

getRecommendationDefinitionPhoto(communityId, recommendationDefinitionId)
Get a custom recommendation definition photo.

getRecommendationDefinitions(communityId)
Get custom recommendation definitions.

getRecommendationForUser(communityId, userId, action, objectId)
Get the Chatter, custom, or static recommendation for the context user for the specified action and object ID.

getRecommendationsForUser(communityId, userId, contextAction, contextObjectId, channel, maxResults)
Get the Chatter recommendations, such as user, group, file, article, record, and topic recommendations for the context user. Get the custom and static recommendations for the context user.

getRecommendationsForUser(communityId, userId, action, contextAction, contextObjectId, channel, maxResults)
Get the Chatter, custom, and static recommendations for the context user for the specified action.

getRecommendationsForUser(communityId, userId, action, objectCategory, contextAction, contextObjectId, channel, maxResults)
Get the Chatter, custom, and static recommendations for the context user for the specified action and object category.

getScheduledRecommendation(communityId, scheduledRecommendationId)
Get a scheduled custom recommendation.

getScheduledRecommendations(communityId, channel)
Get scheduled custom recommendations.

rejectRecommendationForUser(communityId, userId, action, objectId)
Reject a Chatter, custom, or static recommendation for the context user for the specified action and object ID.

rejectRecommendationForUser(communityId, userId, action, objectEnum)
Reject a static recommendation for the context user.

updateRecommendationAudience(communityId, recommendationAudienceId, recommendationAudience)
Update a custom recommendation audience.

updateRecommendationDefinition(communityId, recommendationDefinitionId, name, title, actionUrl, actionUrlName, explanation)
Update a custom recommendation definition with the specified parameters.

updateRecommendationDefinition(communityId, recommendationDefinitionId, recommendationDefinition)
Update a custom recommendation definition.

updateRecommendationDefinitionPhoto(communityId, recommendationDefinitionId, fileUpload)
Update a custom recommendation definition photo with a file that hasn't been uploaded.

updateRecommendationDefinitionPhoto(communityId, recommendationDefinitionId, fileId, versionNumber)
Update a custom recommendation definition photo with an uploaded file.

updateRecommendationDefinitionPhotoWithAttributes(communityId, recommendationDefinitionId, photo)
Update a custom recommendation definition photo with an uploaded file that requires cropping.

updateRecommendationDefinitionPhotoWithAttributes(communityId, recommendationDefinitionId, photo, fileUpload)
Update a custom recommendation definition photo with a file that hasn't been uploaded and requires cropping.

updateScheduledRecommendation(communityId, scheduledRecommendationId, scheduledRecommendation)
Update a scheduled custom recommendation.

updateScheduledRecommendation(communityId, scheduledRecommendationId, rank, enabled, recommendationAudienceId)
Update a scheduled custom recommendation with the specified parameters.

`createRecommendationAudience(communityId, recommendationAudience)`

Create an audience for a custom recommendation.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.RecommendationAudience createRecommendationAudience(String
communityId, ConnectApi.RecommendationAudienceInput recommendationAudience)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

recommendationAudience
Type: `ConnectApi.RecommendationAudienceInput`

A `ConnectApi.RecommendationAudienceInput` object.

Return Value

Type: `ConnectApi.RecommendationAudience`

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

createRecommendationAudience(communityId, name)

Create an audience for a custom recommendation.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.RecommendationAudience createRecommendationAudience(String communityId, String name)
```

Parameters

communityId
: Type: String

  ID for an Experience Cloud site, `internal`, or `null`.

name
: Type: String

  Name of the audience.

Return Value

Type: `ConnectApi.RecommendationAudience`

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

createRecommendationDefinition(communityId, recommendationDefinition)

Create a custom recommendation definition.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.RecommendationDefinition createRecommendationDefinition(String communityId, ConnectApi.RecommendationDefinitionInput recommendationDefinition)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

recommendationDefinition
Type: `ConnectApi.RecommendationDefinitionInput`

A `ConnectApi.RecommendationDefinitionInput` object.

Return Value

Type: `ConnectApi.RecommendationDefinition`

Usage

Recommendation definitions allow you to create custom recommendations that appear in Experience Cloud sites, encouraging users to watch videos, take training and more.

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

These recommendations appear by default on the Customer Service template. They appear on the home and question detail pages and in the feed in Salesforce mobile web. They also appear anywhere community managers add recommendations using Experience Builder in the Customer Service template.

So that users don't see the same recommendations all the time, Salesforce periodically removes and brings back custom recommendations that haven't been accepted or dismissed.

createRecommendationDefinition(communityId, name, title, actionUrl, actionUrlName, explanation)

Create a custom recommendation definition with the specified parameters.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.RecommendationDefinition createRecommendationDefinition(String communityId, String name, String title, String actionUrl, String actionUrlName, String explanation)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

name
Type: String

Name of the custom recommendation definition. The name is displayed in Setup.

title
Type: String

Title of the custom recommendation definition.

actionUrl
Type: String

URL for acting on the custom recommendation, for example, the URL to join a group.

actionUrlName
Type: String

Text label for the action URL in the user interface, for example, "Launch."

explanation
Type: String

Explanation, or body, of the custom recommendation.

Return Value

Type: `ConnectApi.RecommendationDefinition`

Usage

Recommendation definitions allow you to create custom recommendations that appear in Experience Cloud sites, encouraging users to watch videos, take training and more.

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

These recommendations appear by default on the Customer Service template. They appear on the home and question detail pages and in the feed in Salesforce mobile web. They also appear anywhere community managers add recommendations using Experience Builder in the Customer Service template.

So that users don't see the same recommendations all the time, Salesforce periodically removes and brings back custom recommendations that haven't been accepted or dismissed.

createScheduledRecommendation(communityId, scheduledRecommendation)

Create a scheduled custom recommendation.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ScheduledRecommendation createScheduledRecommendation(String communityId, ConnectApi.ScheduledRecommendationInput scheduledRecommendation)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

scheduledRecommendation
Type: `ConnectApi.ScheduledRecommendationInput`

A `ConnectApi.ScheduledRecommendationInput` object.

Return Value

Type: `ConnectApi.ScheduledRecommendation`

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

createScheduledRecommendation(communityId, recommendationDefinitionId, rank, enabled, recommendationAudienceId, channel)

Create a scheduled custom recommendation with the specified parameters.

API Version

36.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ScheduledRecommendation createScheduledRecommendation(String communityId, String recommendationDefinitionId, Integer rank, Boolean enabled, String recommendationAudienceId, ConnectApi.RecommendationChannel channel)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

recommendationDefinitionId
Type: String

ID of the custom recommendation definition.

rank
Type: Integer

Relative rank of the scheduled custom recommendation indicated by ascending whole numbers starting with 1.

Setting the rank is comparable to an insertion into an ordered list. The scheduled custom recommendation is inserted into the position specified by the `rank`. The `rank` of all the scheduled custom recommendations after it is pushed down. See Ranking scheduled custom recommendations example.

If the specified `rank` is larger than the size of the list, the scheduled custom recommendation is put at the end of the list. The `rank` of the scheduled custom recommendation is the size of the list, instead of the one specified.

If a `rank` is not specified, the scheduled custom recommendation is put at the end of the list.

enabled
Type: Boolean

Indicates whether scheduling is enabled. If `true`, the custom recommendation is enabled and appears in Experience Cloud sites. If `false`, custom recommendations in feeds in Salesforce mobile web aren't removed, but no new custom recommendations appear. In Customer Service and Partner Central sites, disabled custom recommendations no longer appear.

recommendationAudienceId
Type: String

ID of the custom recommendation definition that this scheduled recommendation schedules.

channel
Type: `ConnectApi.RecommendationChannel`

A way to tie custom recommendations together. For example, display recommendations in specific places in the UI or show recommendations based on time of day or geographic locations. Values are:

- `CustomChannel1`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels. For example, community managers can use Experience Builder to determine where recommendations appear.
- `CustomChannel2`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.

- `CustomChannel3`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `CustomChannel4`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `CustomChannel5`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `DefaultChannel`—Default recommendation channel. Recommendations appear by default on the Home and Question Detail pages of Customer Service and Partner Central Experience Builder templates. They also appear in the feed in the Salesforce mobile web and anywhere community managers add recommendations using Experience Builder.

Use these channel values; you can't rename or create other channels.

Return Value

Type: `ConnectApi.ScheduledRecommendation`

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

Ranking scheduled custom recommendations example

If you have these scheduled custom recommendations:

Scheduled Recommendations	Rank
ScheduledRecommendationA	1
ScheduledRecommendationB	2
ScheduledRecommendationC	3

And you include this information in the Scheduled Custom Recommendation Input:

Scheduled Recommendation	Rank
ScheduledRecommendationD	2

The result is:

Scheduled Recommendation	Rank
ScheduledRecommendationA	1
ScheduledRecommendationD	2
ScheduledRecommendationB	3
ScheduledRecommendationC	4

`deleteRecommendationAudience(communityId, recommendationAudienceId)`

Delete a custom recommendation audience.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static Void deleteRecommendationAudience(String communityId, String
recommendationAudienceId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

recommendationAudienceId
Type: String

ID of the custom recommendation audience.

Return Value

Type: Void

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

`deleteRecommendationDefinition(communityId, recommendationDefinitionId)`

Delete a custom recommendation definition.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static Void deleteRecommendationDefinition(String communityId, String
recommendationDefinitionId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

recommendationDefinitionId
Type: String

ID of the custom recommendation definition.

Return Value

Type: Void

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

`deleteRecommendationDefinitionPhoto(communityId, recommendationDefinitionId)`

Delete a custom recommendation definition photo.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static Void deleteRecommendationDefinitionPhoto(String communityId, String
recommendationDefinitionId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

recommendationDefinitionId
Type: String

ID of the custom recommendation definition.

Return Value

Type: Void

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

`deleteScheduledRecommendation(communityId, scheduledRecommendationId, deleteDefinitionIfLast)`

Delete a scheduled custom recommendation.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static Void deleteScheduledRecommendation(String communityId, String scheduledRecommendationId, Boolean deleteDefinitionIfLast)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

scheduledRecommendationId
Type: String

ID of the scheduled custom recommendation.

deleteDefinitionIfLast
Type: Boolean

If `true` and if this is the last scheduled custom recommendation of a custom recommendation definition, deletes the custom recommendation definition. Default is `false`.

Return Value

Type: Void

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

Deleting a scheduled custom recommendation is comparable to a deletion in an ordered list. All scheduled custom recommendations after the deleted scheduled custom recommendation receive a new, higher rank automatically.

`getRecommendationAudience(communityId, recommendationAudienceId)`

Get information about a custom recommendation audience.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.RecommendationAudience getRecommendationAudience(String communityId, String recommendationAudienceId)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`recommendationAudienceId`
Type: String

ID of the custom recommendation audience.

Return Value

Type: `ConnectApi.RecommendationAudience`

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

`getRecommendationAudienceMembership(communityId, recommendationAudienceId)`

Get the members of a custom recommendation audience.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserReferencePage getRecommendationAudienceMembership(String communityId, String recommendationAudienceId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

recommendationAudienceId
Type: String

ID of the custom recommendation audience.

Return Value

Type: `ConnectApi.UserReferencePage`

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

`getRecommendationAudienceMembership(communityId, recommendationAudienceId, pageParam, pageSize)`

Get a page of custom recommendation audience members.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserReferencePage getRecommendationAudienceMembership(String communityId, String recommendationAudienceId, Integer pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

recommendationAudienceId
Type: String

ID of the custom recommendation audience.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of members per page.

Return Value

Type: `ConnectApi.UserReferencePage`

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

`getRecommendationAudiences(communityId)`

Get custom recommendation audiences.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.RecommendationAudiencePage getRecommendationAudiences(String
communityId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

Return Value

Type: `ConnectApi.RecommendationAudiencePage`

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

`getRecommendationAudiences(communityId, pageParam, pageSize)`

Get a page of custom recommendation audiences.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.RecommendationAudiencePage getRecommendationAudiences(String communityId, Integer pageParam, Integer pageSize)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`pageParam`
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

`pageSize`
Type: Integer

Specifies the number of audiences per page.

Return Value

Type: `ConnectApi.RecommendationAudiencePage`

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission

can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

`getRecommendationDefinition(communityId, recommendationDefinitionId)`

Get a custom recommendation definition.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.RecommendationDefinition getRecommendationDefinition(String communityId, String recommendationDefinitionId)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`recommendationDefinitionId`
Type: String

ID of the custom recommendation definition.

Return Value

Type: `ConnectApi.RecommendationDefinition`

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

`getRecommendationDefinitionPhoto(communityId, recommendationDefinitionId)`

Get a custom recommendation definition photo.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Photo getRecommendationDefinitionPhoto(String communityId,
String recommendationDefinitionId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

recommendationDefinitionId
Type: String

ID of the custom recommendation definition.

Return Value

Type: `ConnectApi.Photo`

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

`getRecommendationDefinitions(communityId)`

Get custom recommendation definitions.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.RecommendationDefinitionPage getRecommendationDefinitions(String
communityId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

Return Value

Type: `ConnectApi.RecommendationDefinitionPage`

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

`getRecommendationForUser(communityId, userId, action, objectId)`

Get the Chatter, custom, or static recommendation for the context user for the specified action and object ID.

API Version

33.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.RecommendationCollection getRecommendationForUser(String
communityId, String userId, ConnectApi.RecommendationActionType action, String objectId)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`userId`
Type: String

ID for the context user or the keyword `me`.

`action`
Type: `ConnectApi.RecommendationActionType`

Specifies the action to take on a recommendation.

- `follow`—Follow a file, record, topic, or user.
- `join`—Join a group.
- `view`—View a file, group, article, record, user, custom, or static recommendation.

`objectId`
Type: String

Specifies the object to act on.

- If *action* is follow, *objectId* is a user ID, file ID, record ID, or topic ID (version 36.0 and later).
- If *action* is join, *objectId* is a group ID.
- If *action* is view, *objectId* is a user ID, file ID, group ID, record ID, custom recommendation ID (version 34.0 and later), the enum Today for static recommendations (version 35.0 and later), or an article ID (version 37.0 and later).

Return Value

Type: ConnectApi.RecommendationCollection

Usage

To test code that uses this method, use the matching set test method (prefix the method name with setTest). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetRecommendationForUser(communityId, userId, action, objectId, result)

Apex Developer Guide: Testing ConnectApi Code

getRecommendationsForUser(communityId, userId, contextAction, contextObjectId, channel, maxResults)

Get the Chatter recommendations, such as user, group, file, article, record, and topic recommendations for the context user. Get the custom and static recommendations for the context user.

API Version

36.0

Available to Guest Users

38.0

Note: Only article and file recommendations are available to guest users.

Requires Chatter

Yes

Signature

```
public static ConnectApi.RecommendationCollection getRecommendationsForUser(String communityId, String userId, ConnectApi.RecommendationActionType contextAction, String contextObjectId, ConnectApi.RecommendationChannel channel, Integer maxResults)
```

Parameters

communityId

Type: String

ID for an Experience Cloud site, internal, or null.

userId
Type: String

ID for the context user or the keyword `me`.

contextAction
Type: `ConnectApi.RecommendationActionType`

Action that the context user just performed. Supported values are:

- `follow`
- `view`

Use *contextAction* and *contextObjectId* together to get new recommendations based on the action just performed. If you don't want recommendations based on a recent action, specify `null`.

contextObjectId
Type: String

ID of the object that the context user just performed an action on.

- If *contextAction* is `follow`, *contextObjectId* is a user ID, file ID, record ID, or topic ID.
- If *contextAction* is `view`, *contextObjectId* is a user ID, file ID, group ID, record ID, or article ID (version 37.0 and later).

Use *contextAction* and *contextObjectId* together to get new recommendations based on the action just performed. If you don't want recommendations based on a recent action, specify `null`.

channel
Type: `ConnectApi.RecommendationChannel`

A way to tie custom recommendations together. For example, display recommendations in specific places in the UI or show recommendations based on time of day or geographic locations. Values are:

- `CustomChannel1`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels. For example, community managers can use Experience Builder to determine where recommendations appear.
- `CustomChannel2`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `CustomChannel3`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `CustomChannel4`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `CustomChannel5`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `DefaultChannel`—Default recommendation channel. Recommendations appear by default on the Home and Question Detail pages of Customer Service and Partner Central Experience Builder templates. They also appear in the feed in the Salesforce mobile web and anywhere community managers add recommendations using Experience Builder.

maxResults
Type: Integer

Maximum number of recommendation results; default is 10. Values must be from 1 to 99.

Return Value

Type: `ConnectApi.RecommendationCollection`

Usage

If you want to get recommendations based on a recent action performed, such as following a user, use *contextAction* and *contextObjectId* together. For example, if you just followed Pam, you specify `follow` for *contextAction* and Pam's user ID for *contextObjectId*.

This method only recommends users who are followed by people who follow Pam. In this example, John follows Pam so the method returns a recommendation for you to follow Suzanne since John also follows Suzanne.

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetRecommendationsForUser(communityId, userId, contextAction, contextObjectId, channel, maxResults, result)

Apex Developer Guide: Testing ConnectApi Code

`getRecommendationsForUser(communityId, userId, action, contextAction, contextObjectId, channel, maxResults)`

Get the Chatter, custom, and static recommendations for the context user for the specified action.

API Version

36.0

Available to Guest Users

38.0

Note: Only article and file recommendations are available to guest users.

Requires Chatter

Yes

Signature

```
public static ConnectApi.RecommendationCollection getRecommendationsForUser(String
communityId, String userId, ConnectApi.RecommendationActionType action,
ConnectApi.RecommendationActionType contextAction, String contextObjectId,
ConnectApi.RecommendationChannel channel, Integer maxResults)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for the context user or the keyword `me`.

action
Type: `ConnectApi.RecommendationActionType`

Specifies the action to take on a recommendation.

- `follow`—Follow a file, record, topic, or user.
- `join`—Join a group.
- `view`—View a file, group, article, record, user, custom, or static recommendation.

contextAction
Type: `ConnectApi.RecommendationActionType`

Action that the context user just performed. Supported values are:

- `follow`
- `view`

Use *contextAction* and *contextObjectId* together to get new recommendations based on the action just performed. If you don't want recommendations based on a recent action, specify `null`.

contextObjectId
Type: String

ID of the object that the context user just performed an action on.

- If *contextAction* is `follow`, *contextObjectId* is a user ID, file ID, record ID, or topic ID.
- If *contextAction* is `view`, *contextObjectId* is a user ID, file ID, group ID, record ID, or article ID (version 37.0 and later).

Use *contextAction* and *contextObjectId* together to get new recommendations based on the action just performed. If you don't want recommendations based on a recent action, specify `null`.

channel
Type: `ConnectApi.RecommendationChannel`

A way to tie custom recommendations together. For example, display recommendations in specific places in the UI or show recommendations based on time of day or geographic locations. Values are:

- `CustomChannel1`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels. For example, community managers can use Experience Builder to determine where recommendations appear.
- `CustomChannel2`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `CustomChannel3`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `CustomChannel4`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `CustomChannel5`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `DefaultChannel`—Default recommendation channel. Recommendations appear by default on the Home and Question Detail pages of Customer Service and Partner Central Experience Builder templates. They also appear in the feed in the Salesforce mobile web and anywhere community managers add recommendations using Experience Builder.

maxResults
Type: Integer

Maximum number of recommendation results; default is 10. Values must be from 1 to 99.

Return Value

Type: `ConnectApi.RecommendationCollection`

Usage

If you want to get recommendations based on a recent action performed, such as following a user, use *`contextAction`* and *`contextObjectId`* together. For example, if you just followed Pam, you specify `follow` for *`contextAction`* and Pam's user ID for *`contextObjectId`*.

This method only recommends users who are followed by people who follow Pam. In this example, John follows Pam so the method returns a recommendation for you to follow Suzanne since John also follows Suzanne.

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetRecommendationsForUser(communityId, userId, action, contextAction, contextObjectId, channel, maxResults, result)

Apex Developer Guide: Testing ConnectApi Code

`getRecommendationsForUser(communityId, userId, action, objectCategory, contextAction, contextObjectId, channel, maxResults)`

Get the Chatter, custom, and static recommendations for the context user for the specified action and object category.

API Version

36.0

Available to Guest Users

38.0

Note: Only article and file recommendations are available to guest users.

Requires Chatter

Yes

Signature

```
public static ConnectApi.RecommendationCollection getRecommendationsForUser(String communityId, String userId, ConnectApi.RecommendationActionType action, String objectCategory, ConnectApi.RecommendationActionType contextAction, String contextObjectId, ConnectApi.RecommendationChannel channel, Integer maxResults)
```

Parameters

`communityId`

Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for the context user or the keyword `me`.

action
Type: `ConnectApi.RecommendationActionType`

Specifies the action to take on a recommendation.

- `follow`—Follow a file, record, topic, or user.
- `join`—Join a group.
- `view`—View a file, group, article, record, user, custom, or static recommendation.

objectCategory
Type: String

- If *action* is `follow`, *objectCategory* is `users`, `files`, `topics`, or `records`.
- If *action* is `join`, *objectCategory* is `groups`.
- If *action* is `view`, *objectCategory* is `users`, `files`, `groups`, `records`, `custom`, `apps`, or `articles` (version 37.0 and later).

You can also specify a key prefix, the first three characters of the object ID, as the *objectCategory*. Valid values are:

- If *action* is `follow`, *objectCategory* is `005` (users), `069` (files), `0TO` (topics), or `001` (accounts), for example.
- If *action* is `join`, *objectCategory* is `0F9` (groups).
- If *action* is `view`, *objectCategory* is `005` (users), `069` (files), `0F9` (groups), `0RD` (custom recommendations), `T` (static recommendations), `001` (accounts), or `kA0` (articles), for example, (version 370 and later).

contextAction
Type: `ConnectApi.RecommendationActionType`

Action that the context user just performed. Supported values are:

- `follow`
- `view`

Use *contextAction* and *contextObjectId* together to get new recommendations based on the action just performed. If you don't want recommendations based on a recent action, specify `null`.

contextObjectId
Type: String

ID of the object that the context user just performed an action on.

- If *contextAction* is `follow`, *contextObjectId* is a user ID, file ID, record ID, or topic ID.
- If *contextAction* is `view`, *contextObjectId* is a user ID, file ID, group ID, record ID, or article ID (version 37.0 and later).

Use *contextAction* and *contextObjectId* together to get new recommendations based on the action just performed. If you don't want recommendations based on a recent action, specify `null`.

channel
Type: `ConnectApi.RecommendationChannel`

A way to tie custom recommendations together. For example, display recommendations in specific places in the UI or show recommendations based on time of day or geographic locations. Values are:

- `CustomChannel1`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels. For example, community managers can use Experience Builder to determine where recommendations appear.
- `CustomChannel2`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `CustomChannel3`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `CustomChannel4`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `CustomChannel5`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `DefaultChannel`—Default recommendation channel. Recommendations appear by default on the Home and Question Detail pages of Customer Service and Partner Central Experience Builder templates. They also appear in the feed in the Salesforce mobile web and anywhere community managers add recommendations using Experience Builder.

`maxResults`
Type: Integer

Maximum number of recommendation results; default is 10. Values must be from 1 to 99.

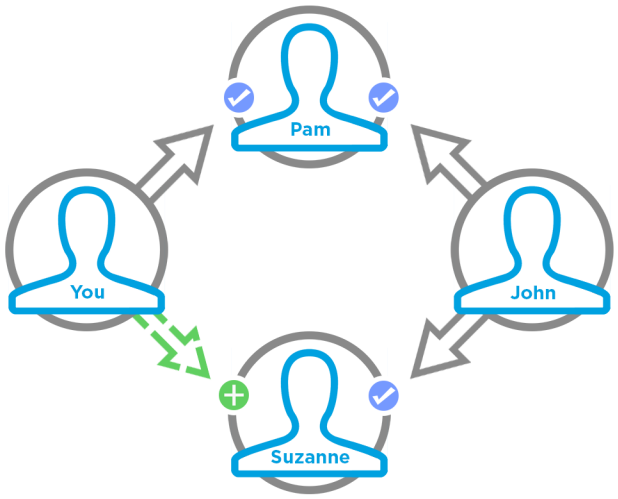
Return Value

Type: `ConnectApi.RecommendationCollection`

Usage

If you want to get recommendations based on a recent action performed, such as following a user, use *`contextAction`* and *`contextObjectId`* together. For example, if you just followed Pam, you specify `follow` for *`contextAction`* and Pam's user ID for *`contextObjectId`*.

This method only recommends users who are followed by people who follow Pam. In this example, John follows Pam so the method returns a recommendation for you to follow Suzanne since John also follows Suzanne.



To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetRecommendationsForUser(communityId, userId, action, objectCategory, contextAction, contextObjectId, channel, maxResults, result)

Apex Developer Guide: Testing ConnectApi Code

`getScheduledRecommendation(communityId, scheduledRecommendationId)`

Get a scheduled custom recommendation.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ScheduledRecommendation getScheduledRecommendation(String communityId, String scheduledRecommendationId)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`scheduledRecommendationId`
Type: String

ID of the scheduled custom recommendation.

Return Value

Type: `ConnectApi.ScheduledRecommendation`

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

`getScheduledRecommendations(communityId, channel)`

Get scheduled custom recommendations.

API Version

36.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ScheduledRecommendationPage getScheduledRecommendations(String communityId, ConnectApi.RecommendationChannel channel)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

channel
Type: `ConnectApi.RecommendationChannel`

A way to tie custom recommendations together. For example, display recommendations in specific places in the UI or show recommendations based on time of day or geographic locations. Values are:

- `CustomChannel1`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels. For example, community managers can use Experience Builder to determine where recommendations appear.
- `CustomChannel2`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `CustomChannel3`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `CustomChannel4`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `CustomChannel5`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `DefaultChannel`—Default recommendation channel. Recommendations appear by default on the Home and Question Detail pages of Customer Service and Partner Central Experience Builder templates. They also appear in the feed in the Salesforce mobile web and anywhere community managers add recommendations using Experience Builder.

Return Value

Type: `ConnectApi.ScheduledRecommendationPage`

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

`rejectRecommendationForUser(communityId, userId, action, objectId)`

Reject a Chatter, custom, or static recommendation for the context user for the specified action and object ID.

API Version

33.0

Requires Chatter

Yes

Signature

```
public static rejectRecommendationForUser(String communityId, String userId,
ConnectApi.RecommendationActionType action, String objectId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for the context user or the keyword `me`.

action
Type: `ConnectApi.RecommendationActionType`

Specifies the action to take on a recommendation. Supported values are:

- `follow`—Follow a file, record, topic, or user.
- `join`—Join a group.
- `view`—View a file, group, article, record, user, custom, or static recommendation.

objectId
Type: String

Specifies the object to take action on.

- If *`action`* is `follow`, *`objectId`* is a user ID, file ID, record ID, or topic ID (version 36.0 and later).
- If *`action`* is `join`, *`objectId`* is a group ID.
- If *`action`* is `view`, *`objectId`* is a custom recommendation ID, the enum `Today` for static recommendations, or an article ID (version 37.0 and later).

Return Value

Type: Void

`rejectRecommendationForUser(communityId, userId, action, objectEnum)`

Reject a static recommendation for the context user.

API Version

34.0

Requires Chatter

Yes

Signature

```
public static rejectRecommendationForUser(String communityId, String userId,
ConnectApi.RecommendationActionType action, ConnectApi.RecommendedObjectType objectEnum)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for the context user or the keyword `me`.

action
Type: `ConnectApi.RecommendationActionType`

Specifies the action to take on a recommendation. Supported values are:

- `view`—View a static recommendation.

objectEnum
Type: `ConnectApi.RecommendedObjectType`

Specifies the object type to take action on.

- `Today`—Static recommendations that don't have an ID, for example, the Today app recommendation.

Return Value

Type: Void

updateRecommendationAudience(communityId, recommendationAudienceId, recommendationAudience)

Update a custom recommendation audience.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.RecommendationAudience updateRecommendationAudience(String communityId, String recommendationAudienceId, ConnectApi.RecommendationAudienceInput recommendationAudience)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

recommendationAudienceId
Type: String

ID of the custom recommendation audience.

recommendationAudience
Type: `ConnectApi.RecommendationAudienceInput`

A `ConnectApi.RecommendationAudienceInput` object.

Return Value

Type: `ConnectApi.RecommendationAudience`

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

updateRecommendationDefinition(communityId, recommendationDefinitionId, name, title, actionUrl, actionUrlName, explanation)

Update a custom recommendation definition with the specified parameters.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.RecommendationDefinition updateRecommendationDefinition(String communityId, String recommendationDefinitionId, String name, String title, String actionUrl, String actionUrlName, String explanation recommendationDefinition)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

recommendationDefinitionId
Type: String

ID of the custom recommendation definition.

name
Type: String

Name of the custom recommendation definition. The name is displayed in Setup.

title
Type: String

Title of the custom recommendation definition.

actionUrl
Type: String

URL for acting on the custom recommendation, for example, the URL to join a group.

actionUrlName
Type: String

Text label for the action URL in the user interface, for example, "Launch."

explanation
Type: String

Explanation, or body, of the custom recommendation.

Return Value

Type: `ConnectApi.RecommendationDefinition`

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

`updateRecommendationDefinition(communityId, recommendationDefinitionId, recommendationDefinition)`

Update a custom recommendation definition.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.RecommendationDefinition updateRecommendationDefinition(String communityId, String recommendationDefinitionId, ConnectApi.RecommendationDefinitionInput recommendationDefinition)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

recommendationDefinitionId
Type: String

ID of the custom recommendation definition.

recommendationDefinition
Type: `ConnectApi.RecommendationDefinitionInput`

A `ConnectApi.RecommendationDefinitionInput` object containing the properties to update.

Return Value

Type: `ConnectApi.RecommendationDefinition`

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

`updateRecommendationDefinitionPhoto(communityId, recommendationDefinitionId, fileUpload)`

Update a custom recommendation definition photo with a file that hasn't been uploaded.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Photo updateRecommendationDefinitionPhoto(String communityId,
String recommendationDefinitionId, ConnectApi.BinaryInput fileUpload)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

recommendationDefinitionId
Type: String

ID of the custom recommendation definition.

fileUpload
Type: `ConnectApi.BinaryInput`

File to use as the photo. The content type must be usable as an image.

Return Value

Type: `ConnectApi.Photo`

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

`updateRecommendationDefinitionPhoto(communityId, recommendationDefinitionId, fileId, versionNumber)`

Update a custom recommendation definition photo with an uploaded file.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Photo updateRecommendationDefinitionPhoto(String communityId,
String recommendationDefinitionId, String fileId, Integer versionNumber)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

recommendationDefinitionId
Type: String

ID of the custom recommendation definition.

fileId
Type: String

ID of a file already uploaded. The file must be an image, and be smaller than 2 GB.

versionNumber
Type: Integer

Version number of the existing file. Specify either an existing version number, or `null` to get the latest version.

Return Value

Type: `ConnectApi.Photo`

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

updateRecommendationDefinitionPhotoWithAttributes(communityId, recommendationDefinitionId, photo)

Update a custom recommendation definition photo with an uploaded file that requires cropping.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Photo updateRecommendationDefinitionPhotoWithAttributes(String communityId, String recommendationDefinitionId, ConnectApi.PhotoInput photo)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

recommendationDefinitionId
Type: String

ID of the custom recommendation definition.

photo
Type: `ConnectApi.PhotoInput`

A `ConnectApi.PhotoInput` object specifying the file ID, version number, and cropping parameters.

Return Value

Type: `ConnectApi.Photo`

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

updateRecommendationDefinitionPhotoWithAttributes(communityId, recommendationDefinitionId, photo, fileUpload)

Update a custom recommendation definition photo with a file that hasn't been uploaded and requires cropping.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Photo updateRecommendationDefinitionPhotoWithAttributes(String communityId, String recommendationDefinitionId, ConnectApi.PhotoInput photo, ConnectApi.BinaryInput fileUpload)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

recommendationDefinitionId
Type: String

ID of the custom recommendation definition.

photo
Type: ConnectApi.PhotoInput

A ConnectApi.PhotoInput object specifying the cropping parameters.

fileUpload
Type: ConnectApi.BinaryInput

File to use as the photo. The content type must be usable as an image.

Return Value

Type: ConnectApi.Photo

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

updateScheduledRecommendation(communityId, scheduledRecommendationId, scheduledRecommendation)

Update a scheduled custom recommendation.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ScheduledRecommendation updateScheduledRecommendation(String communityId, String scheduledRecommendationId, ConnectApi.ScheduledRecommendationInput scheduledRecommendation)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, internal, or null.

scheduledRecommendationId
Type: String

ID of the scheduled custom recommendation.

scheduledRecommendation
Type: ConnectApi.ScheduledRecommendationInput

A `ConnectApi.ScheduledRecommendationInput` object containing the properties to update.

Return Value

Type: `ConnectApi.ScheduledRecommendation`

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

Ranking scheduled custom recommendations example

If you have these scheduled custom recommendations:

Scheduled Recommendations	Rank
ScheduledRecommendationA	1
ScheduledRecommendationB	2
ScheduledRecommendationC	3

And you include this information in the Scheduled Custom Recommendation Input:

Scheduled Recommendation	Rank
ScheduledRecommendationD	2

The result is:

Scheduled Recommendation	Rank
ScheduledRecommendationA	1
ScheduledRecommendationD	2
ScheduledRecommendationB	3
ScheduledRecommendationC	4

`updateScheduledRecommendation(communityId, scheduledRecommendationId, rank, enabled, recommendationAudienceId)`

Update a scheduled custom recommendation with the specified parameters.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ScheduledRecommendation updateScheduledRecommendation(String communityId, String scheduledRecommendationId, Integer rank, Boolean enabled, String recommendationAudienceId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

scheduledRecommendationId
Type: String

ID of the scheduled custom recommendation.

rank
Type: Integer

Relative rank of the scheduled custom recommendation indicated by ascending whole numbers starting with 1.

Setting the rank is comparable to an insertion into an ordered list. The scheduled custom recommendation is inserted into the position specified by the `rank`. The `rank` of all the scheduled custom recommendations after it is pushed down. See Ranking scheduled custom recommendations example.

If the specified `rank` is larger than the size of the list, the scheduled custom recommendation is put at the end of the list. The `rank` of the scheduled custom recommendation is the size of the list, instead of the one specified.

If a `rank` is not specified, the scheduled custom recommendation is put at the end of the list.

enabled
Type: Boolean

Indicates whether scheduling is enabled. If `true`, the custom recommendation is enabled and appears in Experience Cloud sites. If `false`, custom recommendations in feeds in Salesforce mobile web aren't removed, but no new custom recommendations appear. In Customer Service and Partner Central sites, disabled custom recommendations no longer appear.

recommendationAudienceId
Type: String

ID of the custom recommendation definition that this scheduled recommendation schedules.

Return Value

Type: `ConnectApi.ScheduledRecommendation`

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

Ranking scheduled custom recommendations example

If you have these scheduled custom recommendations:

Scheduled Recommendations	Rank
ScheduledRecommendationA	1
ScheduledRecommendationB	2
ScheduledRecommendationC	3

And you include this information in the Scheduled Custom Recommendation Input:

Scheduled Recommendation	Rank
ScheduledRecommendationD	2

The result is:

Scheduled Recommendation	Rank
ScheduledRecommendationA	1
ScheduledRecommendationD	2
ScheduledRecommendationB	3
ScheduledRecommendationC	4

Recommendations Test Methods

These test methods are for `Recommendations`. All methods are static.

For information about using these methods to test your `ConnectApi` code, see Testing ConnectApi Code.

IN THIS SECTION:

setTestGetRecommendationForUser(communityId, userId, action, objectId, result)

Register a `ConnectApi.RecommendationCollection` object to be returned when `getRecommendationForUser` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetRecommendationsForUser(communityId, userId, contextAction, contextObjectId, channel, maxResults, result)

Register a `ConnectApi.RecommendationCollection` object to be returned when `getRecommendationsForUser` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetRecommendationsForUser(communityId, userId, action, contextAction, contextObjectId, channel, maxResults, result)

Register a `ConnectApi.RecommendationCollection` object to be returned when `getRecommendationsForUser` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetRecommendationsForUser(communityId, userId, action, objectCategory, contextAction, contextObjectId, channel, maxResults, result)

Register a `ConnectApi.RecommendationCollection` object to be returned when `getRecommendationsForUser` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetRecommendationForUser(communityId, userId, action, objectId, result)

Register a `ConnectApi.RecommendationCollection` object to be returned when `getRecommendationForUser` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

33.0

Requires Chatter

Yes

Signature

```
public static Void setTestGetRecommendationForUser(String communityId, String userId,
ConnectApi.RecommendationActionType action, String objectId,
ConnectApi.RecommendationCollection result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for the context user or the keyword `me`.

action
Type: `ConnectApi.RecommendationActionType`

Specifies the action to take on a recommendation.

- `follow`—Follow a file, record, topic, or user.
- `join`—Join a group.
- `view`—View a file, group, article, record, user, custom, or static recommendation.

objectId
Type: String

Specifies the object to take action on.

- If *`action`* is `follow`, *`objectId`* is a user ID, file ID, record ID, or topic ID (version 36.0 and later).
- If *`action`* is `join`, *`objectId`* is a group ID.
- If *`action`* is `view`, *`objectId`* is a user ID, file ID, group ID, record ID, custom recommendation ID, the enum `Today` for static recommendations, or an article ID (version 37.0 and later).

result
Type: ConnectApi.RecommendationCollection

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getRecommendationForUser(communityId, userId, action, objectId)

Apex Developer Guide: Testing ConnectApi Code

setTestGetRecommendationsForUser(communityId, userId, contextAction, contextObjectId, channel, maxResults, result)

Register a `ConnectApi.RecommendationCollection` object to be returned when `getRecommendationsForUser` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

36.0

Requires Chatter

Yes

Signature

```
public static Void setTestGetRecommendationsForUser(String communityId, String userId,
ConnectApi.RecommendationActionType contextAction, String contextObjectId,
ConnectApi.RecommendationChannel channel, Integer maxResults,
ConnectApi.RecommendationCollection result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for the context user or the keyword `me`.

contextAction
Type: `ConnectApi.RecommendationActionType`

Action that the context user just performed. Supported values are:

- `follow`
- `view`

Use *contextAction* and *contextObjectId* together to get new recommendations based on the action just performed. If you don't want recommendations based on a recent action, specify `null`.

contextObjectId
Type: String

ID of the object that the context user just performed an action on.

- If *contextAction* is `follow`, *contextObjectId* is a user ID, file ID, record ID, or topic ID.
- If *contextAction* is `view`, *contextObjectId* is a user ID, file ID, group ID, record ID, or article ID (version 37.0 and later).

Use *contextAction* and *contextObjectId* together to get new recommendations based on the action just performed. If you don't want recommendations based on a recent action, specify `null`.

channel
Type: `ConnectApi.RecommendationChannel`

A way to tie custom recommendations together. For example, display recommendations in specific places in the UI or show recommendations based on time of day or geographic locations. Values are:

- `CustomChannel1`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels. For example, community managers can use Experience Builder to determine where recommendations appear.
- `CustomChannel2`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `CustomChannel3`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `CustomChannel4`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `CustomChannel5`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `DefaultChannel`—Default recommendation channel. Recommendations appear by default on the Home and Question Detail pages of Customer Service and Partner Central Experience Builder templates. They also appear in the feed in the Salesforce mobile web and anywhere community managers add recommendations using Experience Builder.

maxResults
Type: Integer

Maximum number of recommendation results; default is 10. Values must be from 1 to 99.

result
Type: `ConnectApi.RecommendationCollection`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getRecommendationsForUser(communityId, userId, contextAction, contextObjectId, channel, maxResults)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetRecommendationsForUser(communityId, userId, action, contextAction, contextObjectId, channel, maxResults, result)`

Register a `ConnectApi.RecommendationCollection` object to be returned when `getRecommendationsForUser` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

36.0

Requires Chatter

Yes

Signature

```
public static Void setTestGetRecommendationsForUser(String communityId, String userId,
ConnectApi.RecommendationActionType action, ConnectApi.RecommendationActionType
contextAction, String contextObjectId, ConnectApi.RecommendationChannel channel, Integer
maxResults, ConnectApi.RecommendationCollection result)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`userId`
Type: String

ID for the context user or the keyword `me`.

`action`
Type: `ConnectApi.RecommendationActionType`

Specifies the action to take on a recommendation.

- `follow`—Follow a file, record, topic, or user.
- `join`—Join a group.
- `view`—View a file, group, article, record, user, custom, or static recommendation.

`contextAction`
Type: `ConnectApi.RecommendationActionType`

Action that the context user just performed. Supported values are:

- `follow`
- `view`

Use *`contextAction`* and *`contextObjectId`* together to get new recommendations based on the action just performed. If you don't want recommendations based on a recent action, specify `null`.

`contextObjectId`
Type: String

ID of the object that the context user just performed an action on.

- If *contextAction* is follow, *contextObjectId* is a user ID, file ID, record ID, or topic ID.
- If *contextAction* is view, *contextObjectId* is a user ID, file ID, group ID, record ID, or article ID (version 37.0 and later).

Use *contextAction* and *contextObjectId* together to get new recommendations based on the action just performed. If you don't want recommendations based on a recent action, specify null.

channel
Type: ConnectApi.RecommendationChannel

A way to tie custom recommendations together. For example, display recommendations in specific places in the UI or show recommendations based on time of day or geographic locations. Values are:

- CustomChannel1—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels. For example, community managers can use Experience Builder to determine where recommendations appear.
- CustomChannel2—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- CustomChannel3—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- CustomChannel4—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- CustomChannel5—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- DefaultChannel—Default recommendation channel. Recommendations appear by default on the Home and Question Detail pages of Customer Service and Partner Central Experience Builder templates. They also appear in the feed in the Salesforce mobile web and anywhere community managers add recommendations using Experience Builder.

maxResults
Type: Integer

Maximum number of recommendation results; default is 10. Values must be from 1 to 99.

result
Type: ConnectApi.RecommendationCollection

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getRecommendationsForUser(communityId, userId, action, contextAction, contextObjectId, channel, maxResults)

Apex Developer Guide: Testing ConnectApi Code

setTestGetRecommendationsForUser(communityId, userId, action, objectCategory, contextAction, contextObjectId, channel, maxResults, result)

Register a ConnectApi.RecommendationCollection object to be returned when getRecommendationsForUser is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

36.0

Requires Chatter

Yes

Signature

```
public static Void setTestGetRecommendationsForUser(String communityId, String userId,
ConnectApi.RecommendationActionType action, String objectCategory,
ConnectApi.RecommendationActionType contextAction, String contextObjectId,
ConnectApi.RecommendationChannel channel, Integer maxResults,
ConnectApi.RecommendationCollection result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for the context user or the keyword `me`.

action
Type: `ConnectApi.RecommendationActionType`

Specifies the action to take on a recommendation.

- `follow`—Follow a file, record, topic, or user.
- `join`—Join a group.
- `view`—View a file, group, article, record, user, custom, or static recommendation.

objectCategory
Type: String

- If *action* is `follow`, *objectCategory* is `users`, `files`, `records`, or `topics`.
- If *action* is `join`, *objectCategory* is `groups`.
- If *action* is `view`, *objectCategory* is `users`, `files`, `groups`, `records`,`custom`, `apps`, or `articles` (version 37.0 and later).

You can also specify a key prefix, the first three characters of the object ID, as the *objectCategory*. Valid values are:

- If *action* is `follow`, *objectCategory* is `005` (users), `069` (files), `0TO` (topics), or `001` (accounts), for example.
- If *action* is `join`, *objectCategory* is `0F9` (groups).
- If *action* is `view`, *objectCategory* is `005` (users), `069` (files), `0F9` (groups), `0RD` (custom recommendations), `T` (static recommendations), `001` (accounts), or `kA0` (articles), for example, (version 370 and later).

contextAction
Type: `ConnectApi.RecommendationActionType`

Action that the context user just performed. Supported values are:

- `follow`
- `view`

Use *`contextAction`* and *`contextObjectId`* together to get new recommendations based on the action just performed. If you don't want recommendations based on a recent action, specify `null`.

`contextObjectId`
Type: String

ID of the object that the context user just performed an action on.

- If *`contextAction`* is `follow`, *`contextObjectId`* is a user ID, file ID, record ID, or topic ID.
- If *`contextAction`* is `view`, *`contextObjectId`* is a user ID, file ID, group ID, record ID, or article ID (version 37.0 and later).

Use *`contextAction`* and *`contextObjectId`* together to get new recommendations based on the action just performed. If you don't want recommendations based on a recent action, specify `null`.

`channel`
Type: `ConnectApi.RecommendationChannel`

A way to tie custom recommendations together. For example, display recommendations in specific places in the UI or show recommendations based on time of day or geographic locations. Values are:

- `CustomChannel1`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels. For example, community managers can use Experience Builder to determine where recommendations appear.
- `CustomChannel2`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `CustomChannel3`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `CustomChannel4`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `CustomChannel5`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.
- `DefaultChannel`—Default recommendation channel. Recommendations appear by default on the Home and Question Detail pages of Customer Service and Partner Central Experience Builder templates. They also appear in the feed in the Salesforce mobile web and anywhere community managers add recommendations using Experience Builder.

`maxResults`
Type: Integer

Maximum number of recommendation results; default is 10. Values must be from 1 to 99.

`result`
Type: `ConnectApi.RecommendationCollection`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getRecommendationsForUser(communityId, userId, action, objectCategory, contextAction, contextObjectId, channel, maxResults)

Apex Developer Guide: Testing ConnectApi Code

Retired Recommendations Methods

These methods for `Recommendations` are retired.

IN THIS SECTION:

createScheduledRecommendation(communityId, recommendationDefinitionId, rank, enabled, recommendationAudienceId)
Create a scheduled custom recommendation with the specified parameters.

getRecommendationsForUser(communityId, userId, contextAction, contextObjectId, maxResults)
Get the Chatter recommendations, such as user, group, file, and record recommendations for the context user. Get the custom and static recommendations for the context user.

getRecommendationsForUser(communityId, userId, action, contextAction, contextObjectId, maxResults)
Get the Chatter, custom, and static recommendations for the context user for the specified action.

getRecommendationsForUser(communityId, userId, action, objectCategory, contextAction, contextObjectId, maxResults)
Get the Chatter, custom, and static recommendations for the context user for the specified action and object category.

getScheduledRecommendations(communityId)
Get scheduled custom recommendations.

setTestGetRecommendationsForUser(communityId, userId, contextAction, contextObjectId, maxResults, result)
Register a `ConnectApi.RecommendationCollection` object to be returned when `getRecommendationsForUser` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetRecommendationsForUser(communityId, userId, action, contextAction, contextObjectId, maxResults, result)
Register a `ConnectApi.RecommendationCollection` object to be returned when `getRecommendationsForUser` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

setTestGetRecommendationsForUser(communityId, userId, action, objectCategory, contextAction, contextObjectId, maxResults, result)
Register a `ConnectApi.RecommendationCollection` object to be returned when `getRecommendationsForUser` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

`createScheduledRecommendation(communityId, recommendationDefinitionId, rank, enabled, recommendationAudienceId)`

Create a scheduled custom recommendation with the specified parameters.

API Version

35.0 only

Important: In version 36.0 and later, use `createScheduledRecommendation(communityId, recommendationDefinitionId, rank, enabled, recommendationAudienceId, channel)`.

Requires Chatter

Yes

Signature

```
public static ConnectApi.ScheduledRecommendation createScheduledRecommendation(String communityId, String recommendationDefinitionId, Integer rank, Boolean enabled, String recommendationAudienceId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

recommendationDefinitionId
Type: String

ID of the custom recommendation definition.

rank
Type: Integer

Relative rank of the scheduled custom recommendation indicated by ascending whole numbers starting with 1.

Setting the rank is comparable to an insertion into an ordered list. The scheduled custom recommendation is inserted into the position specified by the `rank`. The `rank` of all the scheduled custom recommendations after it is pushed down. See Ranking scheduled custom recommendations example.

If the specified `rank` is larger than the size of the list, the scheduled custom recommendation is put at the end of the list. The `rank` of the scheduled custom recommendation is the size of the list, instead of the one specified.

If a `rank` is not specified, the scheduled custom recommendation is put at the end of the list.

enabled
Type: Boolean

Indicates whether scheduling is enabled. If `true`, the custom recommendation is enabled and appears in Experience Cloud sites. If `false`, custom recommendations in feeds in Salesforce mobile web aren't removed, but no new custom recommendations appear. In Customer Service and Partner Central sites, disabled custom recommendations no longer appear.

recommendationAudienceId
Type: String

ID of the custom recommendation definition that this scheduled recommendation schedules.

Return Value

Type: `ConnectApi.ScheduledRecommendation`

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

Ranking scheduled custom recommendations example

If you have these scheduled custom recommendations:

Scheduled Recommendations	Rank
ScheduledRecommendationA	1
ScheduledRecommendationB	2
ScheduledRecommendationC	3

And you include this information in the Scheduled Custom Recommendation Input:

Scheduled Recommendation	Rank
ScheduledRecommendationD	2

The result is:

Scheduled Recommendation	Rank
ScheduledRecommendationA	1
ScheduledRecommendationD	2
ScheduledRecommendationB	3
ScheduledRecommendationC	4

`getRecommendationsForUser(communityId, userId, contextAction, contextObjectId, maxResults)`

Get the Chatter recommendations, such as user, group, file, and record recommendations for the context user. Get the custom and static recommendations for the context user.

API Version

33.0–35.0

Important: In version 36.0 and later, use `getRecommendationsForUser(communityId, userId, contextAction, contextObjectId, channel, maxResults)`.

Requires Chatter

Yes

Signature

```
public static ConnectApi.RecommendationCollection getRecommendationsForUser(String communityId, String userId, ConnectApi.RecommendationActionType contextAction, String contextObjectId, Integer maxResults)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for the context user or the keyword `me`.

contextAction
Type: `ConnectApi.RecommendationActionType`

Action that the context user just performed. Supported values are:

- `follow`
- `view`

Use *contextAction* and *contextObjectId* together to get new recommendations based on the action just performed. If you don't want recommendations based on a recent action, specify `null`.

contextObjectId
Type: String

ID of the object that the context user just performed an action on.

- If *contextAction* is `follow`, *contextObjectId* is a user ID, file ID, or record ID.
- If *contextAction* is `view`, *contextObjectId* is a user ID, file ID, group ID, or record ID.

Use *contextAction* and *contextObjectId* together to get new recommendations based on the action just performed. If you don't want recommendations based on a recent action, specify `null`.

maxResults
Type: Integer

Maximum number of recommendation results; default is 10. Values must be from 1 to 99.

Return Value

Type: `ConnectApi.RecommendationCollection`

Usage

If you want to get recommendations based on a recent action performed, such as following a user, use *contextAction* and *contextObjectId* together. For example, if you just followed Pam, you specify `follow` for *contextAction* and Pam's user ID for *contextObjectId*.

This method only recommends users who are followed by people who follow Pam. In this example, John follows Pam so the method returns a recommendation for you to follow Suzanne since John also follows Suzanne.

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetRecommendationsForUser(communityId, userId, contextAction, contextObjectId, maxResults, result)

Apex Developer Guide: Testing ConnectApi Code

`getRecommendationsForUser(communityId, userId, action, contextAction, contextObjectId, maxResults)`

Get the Chatter, custom, and static recommendations for the context user for the specified action.

API Version

33.0–35.0

Important: In version 36.0 and later, use `getRecommendationsForUser(communityId, userId, action, contextAction, contextObjectId, channel, maxResults)`.

Requires Chatter

Yes

Signature

```
public static ConnectApi.RecommendationCollection getRecommendationsForUser(String communityId, String userId, ConnectApi.RecommendationActionType action, ConnectApi.RecommendationActionType contextAction, String contextObjectId, Integer maxResults)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for the context user or the keyword `me`.

action
Type: `ConnectApi.RecommendationActionType`

Specifies the action to take on a recommendation.

- `follow`—Follow a file, record, topic, or user.
- `join`—Join a group.
- `view`—View a file, group, article, record, user, custom, or static recommendation.

contextAction
Type: `ConnectApi.RecommendationActionType`

Action that the context user just performed. Supported values are:

- `follow`
- `view`

Use *contextAction* and *contextObjectId* together to get new recommendations based on the action just performed. If you don't want recommendations based on a recent action, specify `null`.

contextObjectId
Type: String

ID of the object that the context user just performed an action on.

- If *contextAction* is `follow`, *contextObjectId* is a user ID, file ID, or record ID.
- If *contextAction* is `view`, *contextObjectId* is a user ID, file ID, group ID, or record ID.

Use *contextAction* and *contextObjectId* together to get new recommendations based on the action just performed. If you don't want recommendations based on a recent action, specify `null`.

maxResults
Type: Integer

Maximum number of recommendation results; default is 10. Values must be from 1 to 99.

Return Value

Type: `ConnectApi.RecommendationCollection`

Usage

If you want to get recommendations based on a recent action performed, such as following a user, use *contextAction* and *contextObjectId* together. For example, if you just followed Pam, you specify `follow` for *contextAction* and Pam's user ID for *contextObjectId*.

This method only recommends users who are followed by people who follow Pam. In this example, John follows Pam so the method returns a recommendation for you to follow Suzanne since John also follows Suzanne.

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetRecommendationsForUser(communityId, userId, action, contextAction, contextObjectId, maxResults, result)

Apex Developer Guide: Testing ConnectApi Code

`getRecommendationsForUser(communityId, userId, action, objectCategory, contextAction, contextObjectId, maxResults)`

Get the Chatter, custom, and static recommendations for the context user for the specified action and object category.

API Version

33.0–35.0

Important: In version 36.0 and later, use `getRecommendationsForUser(communityId, userId, action, objectCategory, contextAction, contextObjectId, channel, maxResults)`.

Requires Chatter

Yes

Signature

```
public static ConnectApi.RecommendationCollection getRecommendationsForUser(String
communityId, String userId, ConnectApi.RecommendationActionType action, String
```

```
objectCategory, ConnectApi.RecommendationActionType contextAction, String
contextObjectId, Integer maxResults)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for the context user or the keyword `me`.

action
Type: `ConnectApi.RecommendationActionType`

Specifies the action to take on a recommendation.

- `follow`—Follow a file, record, topic, or user.
- `join`—Join a group.
- `view`—View a file, group, article, record, user, custom, or static recommendation.

objectCategory
Type: String

- If *action* is `follow`, *objectCategory* is `users`, `files`, or `records`.
- If *action* is `join`, *objectCategory* is `groups`.
- If *action* is `view`, *objectCategory* is `users`, `files`, `groups`, `records`, `custom`, or `apps`.

You can also specify a key prefix, the first three characters of the object ID, as the *objectCategory*. Valid values are:

- If *action* is `follow`, *objectCategory* is `005` (users), `069` (files), or `001` (accounts), for example.
- If *action* is `join`, *objectCategory* is `0F9` (groups).
- If *action* is `view`, *objectCategory* is `005` (users), `069` (files), `0F9` (groups), `0RD` (custom recommendations), `T` (static recommendations), or `001` (accounts), for example.

contextAction
Type: `ConnectApi.RecommendationActionType`

Action that the context user just performed. Supported values are:

- `follow`
- `view`

Use *contextAction* and *contextObjectId* together to get new recommendations based on the action just performed. If you don't want recommendations based on a recent action, specify `null`.

contextObjectId
Type: String

ID of the object that the context user just performed an action on.

- If *contextAction* is `follow`, *contextObjectId* is a user ID, file ID, or record ID.
- If *contextAction* is `view`, *contextObjectId* is a user ID, file ID, group ID, or record ID.

Use *contextAction* and *contextObjectId* together to get new recommendations based on the action just performed. If you don't want recommendations based on a recent action, specify `null`.

maxResults
Type: Integer

Maximum number of recommendation results; default is 10. Values must be from 1 to 99.

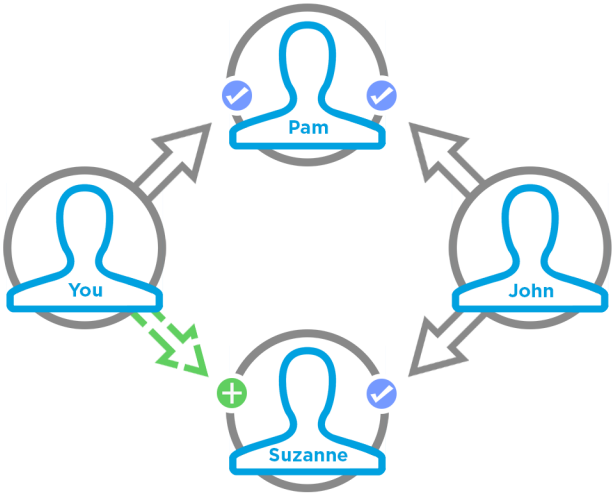
Return Value

Type: `ConnectApi.RecommendationCollection`

Usage

If you want to get recommendations based on a recent action performed, such as following a user, use *contextAction* and *contextObjectId* together. For example, if you just followed Pam, you specify `follow` for *contextAction* and Pam's user ID for *contextObjectId*.

This method only recommends users who are followed by people who follow Pam. In this example, John follows Pam so the method returns a recommendation for you to follow Suzanne since John also follows Suzanne.



To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetRecommendationsForUser(communityId, userId, action, objectCategory, contextAction, contextObjectId, maxResults, result)

Apex Developer Guide: Testing ConnectApi Code

`getScheduledRecommendations(communityId)`

Get scheduled custom recommendations.

API Version

35.0 only

Important: In version 36.0 and later, use `getScheduledRecommendations(communityId, channel)`.

Requires Chatter

Yes

Signature

```
public static ConnectApi.ScheduledRecommendationPage getScheduledRecommendations(String communityId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

Return Value

Type: `ConnectApi.ScheduledRecommendationPage`

Usage

Community managers can access, create, and delete audiences, definitions, and schedules for custom recommendations. (Community managers are users with the Create and Set Up Experiences or Manage Experiences permission.) Users with the Modify All Data permission can also access, create, and delete custom recommendation audiences, custom recommendation definitions, and scheduled custom recommendations.

setTestGetRecommendationsForUser(communityId, userId, contextAction, contextObjectId, maxResults, result)

Register a `ConnectApi.RecommendationCollection` object to be returned when `getRecommendationsForUser` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

33.0–35.0

Requires Chatter

Yes

Signature

```
public static Void setTestGetRecommendationsForUser(String communityId, String userId, ConnectApi.RecommendationActionType contextAction, String contextObjectId, Integer maxResults, ConnectApi.RecommendationCollection result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for the context user or the keyword `me`.

contextAction
Type: `ConnectApi.RecommendationActionType`

Action that the context user just performed. Supported values are:

- `follow`
- `view`

Use *contextAction* and *contextObjectId* together to get new recommendations based on the action just performed. If you don't want recommendations based on a recent action, specify `null`.

contextObjectId
Type: String

ID of the object that the context user just performed an action on.

- If *contextAction* is `follow`, *contextObjectId* is a user ID, file ID, or record ID.
- If *contextAction* is `view`, *contextObjectId* is a user ID, file ID, group ID, or record ID.

Use *contextAction* and *contextObjectId* together to get new recommendations based on the action just performed. If you don't want recommendations based on a recent action, specify `null`.

maxResults
Type: Integer

Maximum number of recommendation results; default is 10. Values must be from 1 to 99.

result
Type: `ConnectApi.RecommendationCollection`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getRecommendationsForUser(communityId, userId, contextAction, contextObjectId, maxResults)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetRecommendationsForUser(communityId, userId, action, contextAction, contextObjectId, maxResults, result)`

Register a `ConnectApi.RecommendationCollection` object to be returned when `getRecommendationsForUser` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

33.0–35.0

Requires Chatter

Yes

Signature

```
public static Void setTestGetRecommendationsForUser(String communityId, String userId,
ConnectApi.RecommendationActionType action, ConnectApi.RecommendationActionType
contextAction, String contextObjectId, Integer maxResults,
ConnectApi.RecommendationCollection result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for the context user or the keyword `me`.

action
Type: `ConnectApi.RecommendationActionType`

Specifies the action to take on a recommendation.

- `follow`—Follow a file, record, topic, or user.
- `join`—Join a group.
- `view`—View a file, group, article, record, user, custom, or static recommendation.

contextAction
Type: `ConnectApi.RecommendationActionType`

Action that the context user just performed. Supported values are:

- `follow`
- `view`

Use *contextAction* and *contextObjectId* together to get new recommendations based on the action just performed. If you don't want recommendations based on a recent action, specify `null`.

contextObjectId
Type: String

ID of the object that the context user just performed an action on.

- If *contextAction* is `follow`, *contextObjectId* is a user ID, file ID, or record ID.
- If *contextAction* is `view`, *contextObjectId* is a user ID, file ID, group ID, or record ID.

Use *contextAction* and *contextObjectId* together to get new recommendations based on the action just performed. If you don't want recommendations based on a recent action, specify `null`.

maxResults
Type: Integer

Maximum number of recommendation results; default is 10. Values must be from 1 to 99.

result
Type: `ConnectApi.RecommendationCollection`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getRecommendationsForUser(communityId, userId, action, contextAction, contextObjectId, maxResults)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetRecommendationsForUser(communityId, userId, action, objectCategory, contextAction, contextObjectId, maxResults, result)`

Register a `ConnectApi.RecommendationCollection` object to be returned when `getRecommendationsForUser` is called with matching parameters in a test context. Use the method with the same parameters or the code throws an exception.

API Version

33.0–35.0

Requires Chatter

Yes

Signature

```
public static Void setTestGetRecommendationsForUser(String communityId, String userId,
ConnectApi.RecommendationActionType action, String objectCategory,
ConnectApi.RecommendationActionType contextAction, String contextObjectId, Integer
maxResults, ConnectApi.RecommendationCollection result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for the context user or the keyword `me`.

action
Type: `ConnectApi.RecommendationActionType`

Specifies the action to take on a recommendation.

- `follow`—Follow a file, record, topic, or user.
- `join`—Join a group.
- `view`—View a file, group, article, record, user, custom, or static recommendation.

`objectCategory`
Type: String

- If *`action`* is `follow`, *`objectCategory`* is `users`, `files`, or `records`.
- If *`action`* is `join`, *`objectCategory`* is `groups`.
- If *`action`* is `view`, *`objectCategory`* is `users`, `files`, `groups`, `records`,`custom`, or `apps`.

You can also specify a key prefix, the first three characters of the object ID, as the *`objectCategory`*. Valid values are:

- If *`action`* is `follow`, *`objectCategory`* is `005` (users), `069` (files), or `001` (accounts), for example.
- If *`action`* is `join`, *`objectCategory`* is `0F9` (groups).
- If *`action`* is `view`, *`objectCategory`* is `005` (users), `069` (files), `0F9` (groups), `0RD` (custom recommendations), `T` (static recommendations), or `001` (accounts), for example.

`contextAction`
Type: `ConnectApi.RecommendationActionType`

Action that the context user just performed. Supported values are:

- `follow`
- `view`

Use *`contextAction`* and *`contextObjectId`* together to get new recommendations based on the action just performed. If you don't want recommendations based on a recent action, specify `null`.

`contextObjectId`
Type: String

ID of the object that the context user just performed an action on.

- If *`contextAction`* is `follow`, *`contextObjectId`* is a user ID, file ID, or record ID.
- If *`contextAction`* is `view`, *`contextObjectId`* is a user ID, file ID, group ID, or record ID.

Use *`contextAction`* and *`contextObjectId`* together to get new recommendations based on the action just performed. If you don't want recommendations based on a recent action, specify `null`.

`maxResults`
Type: Integer

Maximum number of recommendation results; default is 10. Values must be from 1 to 99.

`result`
Type: `ConnectApi.RecommendationCollection`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getRecommendationsForUser(communityId, userId, action, objectCategory, contextAction, contextObjectId, maxResults)

Apex Developer Guide: Testing ConnectApi Code

Records Class

Access information about record motifs, which are small icons used to distinguish record types in the Salesforce UI.

Namespace

ConnectApi

Records Methods

These methods are for `Records`. All methods are static.

IN THIS SECTION:

getMotif(communityId, idOrPrefix)
Get a motif that contains the URLs for a set of small, medium, and large motif icons for a record. It can also contain a base color for the record.

getMotifBatch(communityId, idOrPrefixList)
Get a motif for a list of objects.

`getMotif(communityId, idOrPrefix)`

Get a motif that contains the URLs for a set of small, medium, and large motif icons for a record. It can also contain a base color for the record.

API Version

28.0

Requires Chatter

No

Signature

```
public static ConnectApi.Motif getMotif(String communityId, String idOrPrefix)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

idOrPrefix
Type: String

An ID or key prefix.

Return Value

Type: `ConnectApi.Motif`

Usage

Each Salesforce record type has its own set of motif icons.

getMotifBatch(communityId, idOrPrefixList)

Get a motif for a list of objects.

API Version

31.0

Requires Chatter

No

Signature

```
public static ConnectApi.BatchResult[] getMotifBatch(String communityId, List<String>
idOrPrefixList)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

idOrPrefixList
Type: List<String>

A list of object IDs or prefixes.

Return Value

Type: `ConnectApi.BatchResult`[]

The `ConnectApi.BatchResult.getResult()` method returns a `ConnectApi.Motif` object and errors for motifs that didn't load.

Example

```
String communityId = null;
List<String> prefixIds = new List<String> { '001', '01Z', '069' };

// Get info about the motifs of all records in the list.
ConnectApi.BatchResult[] batchResults = ConnectApi.Records.getMotifBatch(communityId,
prefixIds);

for (ConnectApi.BatchResult batchResult : batchResults) {
    if (batchResult.isSuccess()) {
        // Operation was successful.
        // Print the color of each motif.
        ConnectApi.Motif motif;
        if(batchResult.getResult() instanceof ConnectApi.Motif) {
            motif = (ConnectApi.Motif) batchResult.getResult();
        }
        System.debug('SUCCESS');
        System.debug(motif.color);
    }
    else {
        // Operation failed. Print errors.
        System.debug('FAILURE');
        System.debug(batchResult.getErrorMessage());
    }
}
```

RegisterGuestBuyer Class

Register a guest buyer for a webstore using an account ID, enabling a guest buyer to order on behalf of another buyer.

Namespace

ConnectApi

RegisterGuestBuyer Methods

These methods are for `RegisterGuestBuyer`. All methods are static. Your org must have the Order Management Growth license or Order Management as part of Connected Commerce.

IN THIS SECTION:

registerGuestBuyer(webstoreId, accountId)
Register a guest buyer for a webstore using an account ID. This method enables a guest buyer to order on behalf of another buyer.

`registerGuestBuyer(webstoreId, accountId)`

Register a guest buyer for a webstore using an account ID. This method enables a guest buyer to order on behalf of another buyer.

API Version

60.0

Requires Chatter

No

Signature

```
public static ConnectApi.RegisterGuestBuyerOutputRepresentation registerGuestBuyer(String webstoreId, String accountId)
```

Parameters

webstoreId
Type: String

ID of the webstore.

accountId
Type: String

ID of the account for which the request is made.

Return Value

Type: `ConnectApi.RegisterGuestBuyerOutputRepresentation` on page 2275

Repricing Class

Perform functions related to repricing orders in Order Management.

Namespace

ConnectApi

Repricing Methods

These methods are for `Repricing`. All methods are static.

IN THIS SECTION:

productDetails(webstoreId, skuOrProductId, effectiveAccountId, currencyCode, locale)
Get details of a product in a web store.

searchProducts(webstoreId, searchTerm, pageParam, pageSize, effectiveAccountId, facets)
Search products in a webstore.

`productDetails(webstoreId, skuOrProductId, effectiveAccountId, currencyCode, locale)`

Get details of a product in a web store.

API Version

55.0

Requires Chatter

No

Signature

```
public static ConnectApi.ProductDetailsOutputRepresentation productDetails(String webstoreId, String skuOrProductId, String effectiveAccountId, String currencyCode, String locale)
```

Parameters

webstoreId
Type: String

ID of the WebStore.

skuOrProductId
Type: String

SKU or ID of the Product.

effectiveAccountId
Type: String

Effective Account ID. Required for B2B stores. For other stores, pass null.

currencyCode
Type: String

ISO currency code. If you pass null, the default store value is used.

locale
Type: String

Locale. If you pass null, the default store value is used.

excludeAttributeSetInfo
Type: String

Specifies whether the attribute set information for the product is returned.

excludeMedia
Type: String

Specifies whether the media groups and default images of the product are returned.

excludeQuantityRule
Type: String

Specifies whether the quantity rule information for the product is returned.

excludeVariationInfo
Type: String

Specifies whether the variation information for the product is returned.

excludePrices
Type: String

Specifies whether the prices for the product is returned.

Return Value

Type: `ConnectApi.ProductDetailsOutputRepresentation`

searchProducts(webstoreId, searchTerm, pageParam, pageSize, effectiveAccountId, facets)

Search products in a webstore.

API Version

59.0

Requires Chatter

No

Signature

```
public static ConnectApi.ProductSearchOutputRepresentation searchProducts(String webstoreId, String searchTerm, Integer pageParam, Integer pageSize, String effectiveAccountId, String facets)
```

Parameters

webstoreId
Type: String

ID of the webstore.

searchTerm
Type: String

Term used for the search.

pageParam
Type: Integer

Maximum number of search results pages to return. If you don't specify a value, the default is 1.

pageSize
Type: Integer

Number of items per page. Valid values are from 1 through 100. If you don’t specify a value, the default size is 20.

effectiveAccountId
Type: String

ID of the account for which the request is made. If unspecified, defaults to the account ID for the context user.

facets
Type: String

A list of facet names to filter the search.For example, `["size_medium", "color_red"]` is encoded to `WyJzaXplX21lZGl1bSIsICJjb2xvcl9yZWQiXQ==`

Return Value

Type: `ConnectApi.ProductSearchOutputRepresentation` on page 2252

ReturnOrder Class

Process ReturnOrders in Order Management, limited to 2,000 requests per hour.

Namespace

ConnectApi

ReturnOrder Methods

These methods are for `ReturnOrder`. All methods are static.

IN THIS SECTION:

createReturnOrder(returnOrderInput)
Create a ReturnOrder and ReturnOrderLineItems for items belonging to an OrderSummary.

returnItems(returnOrderId, returnItemsInput)
Process ReturnOrderLineItems belonging to a ReturnOrder. Processing a ReturnOrderLineItem generates a change Order and makes that ReturnOrderLineItem read-only. The change order for a returned item or delivery charge has a positive amount and should be used to create a credit memo. The change order for a return fee has a negative amount and should be used to create an invoice. If a processed ReturnOrderLineItem has any remaining expected quantity, then the API creates a separate ReturnOrderLineItem representing that quantity.

`createReturnOrder(returnOrderInput)`

Create a ReturnOrder and ReturnOrderLineItems for items belonging to an OrderSummary.

API Version

50.0

Requires Chatter

No

Signature

```
public static ConnectApi.ReturnOrderOutputRepresentation
createReturnOrder(ConnectApi.ReturnOrderInputRepresentation returnOrderInput)
```

Parameters

returnOrderInput
Type: ConnectApi.ReturnOrderInputRepresentation

Data for creating a ReturnOrder and ReturnOrderLineItems.

Return Value

Type: ConnectApi.ReturnOrderOutputRepresentation

SEE ALSO:

returnItems(returnOrderId, returnItemsInput)

returnItems(returnOrderId, returnItemsInput)

Process ReturnOrderLineItems belonging to a ReturnOrder. Processing a ReturnOrderLineItem generates a change Order and makes that ReturnOrderLineItem read-only. The change order for a returned item or delivery charge has a positive amount and should be used to create a credit memo. The change order for a return fee has a negative amount and should be used to create an invoice. If a processed ReturnOrderLineItem has any remaining expected quantity, then the API creates a separate ReturnOrderLineItem representing that quantity.

API Version

52.0

Requires Chatter

No

Signature

```
public static ConnectApi.ReturnItemsOutputRepresentation returnItems(String
returnOrderId, ConnectApi.ReturnItemsInputRepresentation returnItemsInput)
```

Parameters

returnOrderId
Type: String

ID of the ReturnOrder.

returnItemsInput
Type: ConnectApi.ReturnItemsInputRepresentation

Data about products and delivery charges to return, as well as associated return fees.

Return Value

Type: `ConnectApi.ReturnItemsOutputRepresentation`

SEE ALSO:

createMultipleInvoices(invoicesInput)

ensureRefundsAsync(orderSummaryId, ensureRefundsInput)

createReturnOrder(returnOrderInput)

Routing Class

Route orders to inventory locations in Order Management.

Namespace

ConnectApi

Routing Methods

These methods are for `Routing`. All methods are static.

IN THIS SECTION:

confirmHeldFOCapacity(confirmHeldFOCapacityInput)

Confirm held fulfillment order capacity at one or more locations. This call decreases a location's held capacity and increases its assigned fulfillment order count. Confirm held capacity when you assign a fulfillment order to a location.

findRoutesWithFewestSplits(findRoutesWithFewestSplitsInputRepresentation)

Returns combinations of inventory locations that can fulfill an order within a specified limit of shipment splits. By default, checks up to 1,000,000 potential routes, returning a maximum of 10,000 results.

findRoutesWithFewestSplitsUsingOCI(findRoutesWithFewestSplitsUsingOCIInput)

For one or more order summaries, find inventory availability using Omnichannel Inventory and identify the fulfillment routes with fewest splits. By default, checks up to 1,000,000 potential routes, returning a maximum of 10,000 results. This method combines the functionality of the `getInventoryAvailability()` and `findRoutesWithFewestSplits()` methods.

getFOCapacityValues(getFOCapacityValuesInput)

Get information about the current fulfillment order capacity of one or more locations.

holdFOCapacity(holdFOCapacityInput)

Hold fulfillment order capacity at a location. Holding capacity at a location reserves a space for a fulfillment order that you'll assign to it.

rankAverageDistance(rankAverageDistanceInputRepresentation)

Calculates the average distance from sets of inventory locations to an order recipient, and ranks them. Use this method to compare the average shipping distances for different sets of locations that can fulfill an order. While this method is executing, you can't invoke another Apex callout.

releaseHeldFOCapacity(releaseHeldFOCapacityInput)

Release held fulfillment order capacity at one or more locations. This call decreases a location's held capacity without changing its assigned fulfillment order count. Release held capacity when you cancel the assignment of a fulfillment order to a location.

confirmHeldFOCapacity(confirmHeldFOCapacityInput)

Confirm held fulfillment order capacity at one or more locations. This call decreases a location's held capacity and increases its assigned fulfillment order count. Confirm held capacity when you assign a fulfillment order to a location.

API Version

55.0

Requires Chatter

No

Signature

```
public static ConnectApi.ConfirmHeldFOCapacityOutputRepresentation
confirmHeldFOCapacity(ConnectApi.ConfirmHeldFOCapacityInputRepresentation
confirmHeldFOCapacityInput)
```

Parameters

confirmHeldFOCapacityInput
Type: ConnectApi.ConfirmHeldFOCapacityInputRepresentation

The input includes, for each fulfillment order, the location where capacity is held for it.

Return Value

Type: ConnectApi.ConfirmHeldFOCapacityOutputRepresentation

findRoutesWithFewestSplits(findRoutesWithFewestSplitsInputRepresentation)

Returns combinations of inventory locations that can fulfill an order within a specified limit of shipment splits. By default, checks up to 1,000,000 potential routes, returning a maximum of 10,000 results.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.FindRoutesWithFewestSplitsOutputRepresentation
findRoutesWithFewestSplits(ConnectApi.FindRoutesWithFewestSplitsInputRepresentation
findRoutesWithFewestSplitsInputRepresentation)
```

Parameters

findRoutesWithFewestSplitsInputRepresentation
Type: ConnectApi.FindRoutesWithFewestSplitsInputRepresentation

The input includes the ordered item quantities, data about available inventory, and, optionally, a maximum allowable number of shipment splits.

Return Value

Type: ConnectApi.FindRoutesWithFewestSplitsOutputRepresentation

findRoutesWithFewestSplitsUsingOCI(findRoutesWithFewestSplitsUsingOCIInput)

For one or more order summaries, find inventory availability using Omnichannel Inventory and identify the fulfillment routes with fewest splits. By default, checks up to 1,000,000 potential routes, returning a maximum of 10,000 results. This method combines the functionality of the `getInventoryAvailability()` and `findRoutesWithFewestSplits()` methods.

API Version

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.FindRoutesWithFewestSplitsUsingOCIOutputRepresentation
findRoutesWithFewestSplitsUsingOCI(ConnectApi.FindRoutesWithFewestSplitsUsingOCIInputRepresentation
findRoutesWithFewestSplitsUsingOCIInput)
```

Parameters

findRoutesWithFewestSplitsUsingOCIInput
Type: ConnectApi.FindRoutesWithFewestSplitsUsingOCIInputRepresentation

The input includes, for each order, the ordered item quantities, the assigned location group or locations, and, optionally, a maximum allowable number of shipment splits and a list of locations to exclude from the calculation.

Return Value

Type: ConnectApi.FindRoutesWithFewestSplitsUsingOCIOutputRepresentation

SEE ALSO:

getInventoryAvailability(inventoryAvailabilityInputRepresentation)

findRoutesWithFewestSplits(findRoutesWithFewestSplitsInputRepresentation)

getFOCapacityValues(getFOCapacityValuesInput)

Get information about the current fulfillment order capacity of one or more locations.

API Version

55.0

Requires Chatter

No

Signature

```
public static ConnectApi.GetFOCapacityValuesOutputRepresentation
getFOCapacityValues(ConnectApi.GetFOCapacityValuesRequestInputRepresentation
getFOCapacityValuesInput)
```

Parameters

getFOCapacityValuesInput
Type: ConnectApi.GetFOCapacityValuesRequestInputRepresentation

Locations to get fulfillment order capacity information about.

Return Value

Type: ConnectApi.GetFOCapacityValuesOutputRepresentation

holdFOCapacity(holdFOCapacityInput)

Hold fulfillment order capacity at a location. Holding capacity at a location reserves a space for a fulfillment order that you'll assign to it.

API Version

55.0

Requires Chatter

No

Signature

```
public static ConnectApi.HoldFOCapacityOutputRepresentation
holdFOCapacity(ConnectApi.HoldFOCapacityInputRepresentation holdFOCapacityInput)
```

Parameters

holdFOCapacityInput
Type: ConnectApi.HoldFOCapacityInputRepresentation

The input includes, for each fulfillment order, the location to hold capacity for it.

Return Value

Type: ConnectApi.HoldFOCapacityOutputRepresentation

rankAverageDistance(rankAverageDistanceInputRepresentation)

Calculates the average distance from sets of inventory locations to an order recipient, and ranks them. Use this method to compare the average shipping distances for different sets of locations that can fulfill an order. While this method is executing, you can't invoke another Apex callout.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.RankAverageDistanceOutputRepresentation
rankAverageDistance(ConnectApi.RankAverageDistanceInputRepresentation
rankAverageDistanceInputRepresentation)
```

Parameters

rankAverageDistanceInputRepresentation
Type: ConnectApi.RankAverageDistanceInputRepresentation

An order recipient's geographic location and information about sets of inventory locations that can fulfill the order.

Return Value

Type: ConnectApi.RankAverageDistanceOutputRepresentation

releaseHeldFOCapacity(releaseHeldFOCapacityInput)

Release held fulfillment order capacity at one or more locations. This call decreases a location's held capacity without changing its assigned fulfillment order count. Release held capacity when you cancel the assignment of a fulfillment order to a location.

API Version

55.0

Requires Chatter

No

Signature

```
public static ConnectApi.ReleaseHeldFOCapacityOutputRepresentation
releaseHeldFOCapacity(ConnectApi.ReleaseHeldFOCapacityInputRepresentation
releaseHeldFOCapacityInput)
```

Parameters

releaseHeldFOCapacityInput
Type: ConnectApi.ReleaseHeldFOCapacityInputRepresentation

The input includes, for each fulfillment order, the location that holds the capacity to release.

Return Value

Type: ConnectApi.ReleaseHeldFOCapacityOutputRepresentation

SalesforceInbox Class

Access information about Automated Activity Capture, which is available in Einstein and Salesforce Inbox.

Namespace

ConnectApi

SalesforceInbox Methods

These methods are for `SalesforceInbox`. All methods are static.

IN THIS SECTION:

shareActivity(activityId, sharingInfo)
Share emails or events with certain groups of users.

shareActivity(activityId, sharingInfo)

Share emails or events with certain groups of users.

API Version

39.0

Requires Chatter

No

Signature

```
public static ConnectApi.ActivitySharingResult shareActivity(String activityId,
ConnectApi.ActivitySharingInput sharingInfo)
```

Parameters

activityId
Type: String

The ID of the activity.

sharingInfo
Type: ConnectApi.ActivitySharingInput

A ConnectApi.ActivitySharingInput object.

Return Value

Type: ConnectApi.ActivitySharingResult

Usage

This method is a feature of both Sales Cloud Einstein and Inbox. It lets users connect their email and calendar to Salesforce. Then, their emails and events are automatically added to related Salesforce records. Users can specify who their individual emails and events are shared with.

Search Class

Search objects using keywords or a natural language query.

Namespace

ConnectApi

Search Methods

These methods are for Search. All methods are static.

IN THIS SECTION:

answer(q)
Search objects using a natural language query and return an answer.

answer(q, objectApiName)
Search an object using a natural language query and return an answer.

answer(q, objectApiName, displayFields)
Search an object using a natural language query and display fields.

findAndGroup(q)
Search objects using keyword search and return result groups.

findAndGroup(q, configurationName)
Search objects using keyword search and a configuration. The search returns result groups.

findAndGroup(q, configurationName, highlights)
Search objects using keyword search, a configuration, and highlights. The search returns result groups.

find(objectApiName, request)
Search an object using keywords and return results.

`answer(q)`

Search objects using a natural language query and return an answer.

API Version

63.0

Available to Guest Users

63.0

Requires Chatter

No

Signature

```
public static ConnectApi.SearchAnswer answer(String q)
```

Parameters

q
Type: String

Natural language query to search for in the org.

Return Value

Type: `ConnectApi.SearchAnswer`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

`answer(q, objectApiName)`

Search an object using a natural language query and return an answer.

API Version

63.0

Available to Guest Users

63.0

Requires Chatter

No

Signature

```
public static ConnectApi.SearchAnswer answer(String q, String objectApiName)
```

Parameters

q
Type: String

Natural language query to search for in the org.

objectApiName
Type: String

API name of the object.

Return Value

Type: ConnectApi.SearchAnswer

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

answer(q, objectApiName, displayFields)

Search an object using a natural language query and display fields.

API Version

63.0

Available to Guest Users

63.0

Requires Chatter

No

Signature

```
public static ConnectApi.SearchAnswer answer(String q, String objectApiName, List<String>
displayFields)
```

Parameters

q
Type: String

Natural language query to search for in the org.

objectApiName
Type: String

API name of the object.

displayFields
Type: List<String>

List of fields to display and return in the search answer. The default is the citation field.

Return Value

Type: `ConnectApi.SearchAnswer`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

`findAndGroup(q)`

Search objects using keyword search and return result groups.

API Version

63.0

Available to Guest Users

63.0

Requires Chatter

No

Signature

```
public static ConnectApi.SearchResultGroups findAndGroup(String q)
```

Parameters

q

Type: String

One or more keywords to search for in the org.

Return Value

Type: `ConnectApi.SearchResultGroups`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

`findAndGroup(q, configurationName)`

Search objects using keyword search and a configuration. The search returns result groups.

API Version

63.0

Available to Guest Users

63.0

Requires Chatter

No

Signature

```
public static ConnectApi.SearchResultGroups findAndGroup(String q, String
configurationName)
```

Parameters

q

Type: String

One or more keywords to search for in the org.

configurationName

Type: String

Search configuration to apply.

Return Value

Type: `ConnectApi.SearchResultGroups`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

`findAndGroup(q, configurationName, highlights)`

Search objects using keyword search, a configuration, and highlights. The search returns result groups.

API Version

63.0

Available to Guest Users

63.0

Requires Chatter

No

Signature

```
public static ConnectApi.SearchResultGroups findAndGroup(String q, String
configurationName, Boolean highlights)
```

Parameters

q
Type: String

One or more keywords to search for in the org.

configurationName
Type: String

Search configuration to apply.

highlights
Type: Boolean

Specifies whether search generates a text highlight (`true`) or not (`false`).

Return Value

Type: `ConnectApi.SearchResultGroups`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

`find(objectApiName, request)`

Search an object using keywords and return results.

API Version

63.0

Available to Guest Users

63.0

Requires Chatter

No

Signature

```
public static ConnectApi.ScopedSearchResults find(String objectApiName,
ConnectApi.SearchRequest request)
```

Parameters

objectApiName
Type: String

API name of the object to search.

request
Type: `ConnectApi.SearchRequest`

`ConnectApi.SearchRequest` input class with more information about what and how to search.

Return Value

Type: `ConnectApi.ScopedSearchResults`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

Search Test Methods

These test methods are for `Search`. All methods are static.

For information about using these methods to test your `ConnectApi` code, see Testing ConnectApi Code.

`setTestAnswer(q, result)`

Registers a `ConnectApi.SearchAnswer` object to be returned when the matching `answer(q)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

63.0

Signature

```
public static Void setTestAnswer(String q, ConnectApi.SearchAnswer result)
```

Parameters

q
Type: String

Natural language query to search for in the org.

result
Type: `ConnectApi.SearchAnswer`

Object containing test data.

Return Value

Type: Void

setTestAnswer(q, objectApiName, result)

Registers a `ConnectApi.SearchAnswer` object to be returned when the matching `answer(q, objectApiName)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

63.0

Signature

```
public static Void setTestAnswer(String q, String objectApiName, ConnectApi.SearchAnswer result)
```

Parameters

q
Type: String

Natural language query to search for in the org.

objectApiName
Type: String

API name of the object.

result
Type: `ConnectApi.SearchAnswer`

Object containing test data.

Return Value

Type: Void

setTestAnswer(q, objectApiName, displayFields, result)

Registers a `ConnectApi.SearchAnswer` object to be returned when the matching `answer(q, objectApiName, displayFields)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

62.0

Signature

```
public static Void setTestAnswer(String q, String objectApiName, List<String> displayFields, ConnectApi.SearchAnswer result)
```

Parameters

q
Type: String

Natural language query to search for in the org.

objectApiName
Type: String

API name of the object.

displayFields
Type: List<String>

List of fields to display and return in the search results. By default, the fields displayed are defined by the search layout.

result
Type: `ConnectApi.SearchAnswer`

Object containing test data.

Return Value

Type: Void

setTestFindAndGroup(q, result)

Registers a `ConnectApi.SearchResultGroups` object to be returned when the matching `findAndGroup(q)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

63.0

Signature

```
public static Void setTestFindAndGroup(String q, ConnectApi.SearchResultGroups result)
```

Parameters

q
Type: String

One or more keywords to search for in the org.

result
Type: `ConnectApi.SearchResultGroups`

Object containing test data.

Return Value

Type: Void

setTestFindAndGroup(q, configurationName, result)

Registers a `ConnectApi.SearchResultGroups` object to be returned when the matching `findAndGroup(q, configurationName)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

63.0

Signature

```
public static Void setTestFindAndGroup(String q, String configurationName,
ConnectApi.SearchResultGroups result)
```

Parameters

q
Type: String

One or more keywords to search for in the org.

configurationName
Type: String

Search configuration to apply.

result
Type: `ConnectApi.SearchResultGroups`

Object containing test data.

Return Value

Type: Void

`setTestFindAndGroup(q, configurationName, highlights, result)`

Registers a `ConnectApi.SearchResultGroups` object to be returned when the matching `findAndGroup(q, configurationName, highlights)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

63.0

Signature

```
public static Void setTestFindAndGroup(String q, String configurationName, Boolean
highlights, ConnectApi.SearchResultGroups result)
```

Parameters

q
Type: String

One or more keywords to search for in the org.

configurationName
Type: String

Search configuration to apply.

highlights
Type: Boolean

Specifies whether search generates a text highlight (`true`) or not (`false`).

result
Type: `ConnectApi.SearchResultGroups`

Object containing test data.

Return Value

Type: Void

`setTestFind(objectApiName, request, result)`

Registers a `ConnectApi.ScopedSearchResults` object to be returned when the matching `find(objectApiName, request)` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

63.0

Signature

```
public static Void setTestFind(String objectApiName, ConnectApi.SearchRequest request,
ConnectApi.ScopedSearchResults result)
```

Parameters

objectApiName
Type: String

API name of the object to search.

request
Type: `ConnectApi.SearchRequest`

`ConnectApi.SearchRequest` input class with more information about what and how to search.

result
Type: `ConnectApi.ScopedSearchResults`

Object containing test data.

Return Value

Type: Void

Sites Class

Search an Experience Cloud site.

Namespace

ConnectApi

Sites Methods

These methods are for `Sites`. All methods are static.

IN THIS SECTION:

searchSite(siteId, queryTerm, pageToken, pageSize, language)
Search an Experience Cloud site.

`searchSite(siteId, queryTerm, pageToken, pageSize, language)`

Search an Experience Cloud site.

API Version

54.0

Available to Guest Users

54.0

Requires Chatter

No

Signature

```
public static ConnectApi.SiteSearchResult searchSite(String siteId, String queryTerm,
String pageToken, Integer pageSize, String language)
```

Parameters

siteId
Type: String

ID for the Experience Cloud site.

queryTerm
Type: String

White-space separated words used to search for relevant content. Provide a maximum of 1024 characters, composed of up to 32 words and white spaces. Logical operators aren't supported.

pageToken
Type: String

Specifies the base64 encoded page token. Page tokens are returned as part of the response. If unspecified, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 240. If you pass in `null`, the default size is 25.

`language`
Type: String

Language locale for the context user. If unspecified or if the specified language locale isn't available, the default value is `en_US`.

Return Value

Type: `ConnectApi.SiteSearchResult`

SmartDataDiscovery Class

Get predictions on Salesforce objects.

Use the `ConnectApi.SmartDataDiscovery.predict` method to get predictions on Salesforce objects. For more information, see Get Predictions in Apex.

SocialEngagement Class

Manage information about social accounts or fan pages for social networks.

Namespace

ConnectApi

SocialEngagement Methods

These methods are for `SocialEngagement`. All methods are static.

IN THIS SECTION:

deleteSocialPost(socialPostId, socialAccountId)
Delete a social post from its social network.

followSocialPersona(socialPersonaId, socialAccountId)
Follow a social persona in its social network.

followSocialPostPersona(socialPostId, socialAccountId)
Follow a social persona on a social post in its social network.

getIntents(socialPostId)
Get available intents for a social post.

getManagedSocialAccount(id)
Get a managed social account that is in the org and assigned to the user.

getManagedSocialAccounts()
Gets a list of managed social accounts that are in the org and assigned to the user.

getManagedSocialAccounts(socialNetwork)
Get a list of managed social accounts that are in the org and assigned to the user.

getRelationship(id, socialPersonaId)
Get the follow relationship between a managed social account and a social persona.

hideSocialPost(socialPostId, socialAccountId)
Hide a social post in its social network.

likeSocialPost(socialPostId, socialAccountId)
Like a social post in its social network.

massApprove(massApproval)
Approve or reject the publishing of a large number of social posts.

recallApproval(socialPostId)
Recall an approval request to publish a social post.

unfollowSocialPersona(socialPersonaId, socialAccountId)
Stop following a social persona in its social network.

unfollowSocialPostPersona(socialPostId, socialAccountId)
Stop following a social persona of a social post in its social network.

unhideSocialPost(socialPostId, socialAccountId)
Unhide a social post in its social network.

unlikeSocialPost(socialPostId, socialAccountId)
Unlike a social post in its social network.

`deleteSocialPost(socialPostId, socialAccountId)`

Delete a social post from its social network.

Note: Deleting a social post from its social network doesn't delete the record from Salesforce.

API Version

46.0

Requires Chatter

No

Signature

```
public static ConnectApi.DeleteSocialPostIntent deleteSocialPost(String socialPostId,
String socialAccountId)
```

Parameters

socialPostId
Type: String

ID of the social post to delete.

socialAccountId
Type: String

ID of the social account that deletes the post.

Return Value

Type: ConnectApi.DeleteSocialPostIntent

followSocialPersona(socialPersonaId, socialAccountId)

Follow a social persona in its social network.

API Version

45.0

Requires Chatter

No

Signature

```
public static ConnectApi.FollowSocialPersonaIntent followSocialPersona(String
socialPersonaId, String socialAccountId)
```

Parameters

socialPersonaId
Type: String

ID of the social persona to follow.

socialAccountId
Type: String

ID of the social account that follows the social persona.

Return Value

Type: ConnectApi.FollowSocialPersonaIntent

followSocialPostPersona(socialPostId, socialAccountId)

Follow a social persona on a social post in its social network.

API Version

45.0

Requires Chatter

No

Signature

```
public static ConnectApi.FollowSocialPersonaIntent followSocialPostPersona(String
socialPostId, String socialAccountId)
```

Parameters

socialPostId
Type: String

ID of the social post authored by the social persona to follow.

socialAccountId
Type: String

ID of the social account that follows the social persona.

Return Value

Type: ConnectApi.FollowSocialPersonaIntent

getIntents(socialPostId)

Get available intents for a social post.

API Version

45.0

Requires Chatter

No

Signature

```
public static ConnectApi.SocialPostIntents getIntents(String socialPostId)
```

Parameters

socialPostId
Type: String

ID of a social post.

Return Value

Type: ConnectApi.SocialPostIntents

getManagedSocialAccount(id)

Get a managed social account that is in the org and assigned to the user.

API Version

44.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedSocialAccount getManagedSocialAccount(String id)
```

Parameters

id
Type: String

Description: Internal SFDC ID for this managed social account.

Return Value

Type: ConnectApi.ManagedSocialAccount

getManagedSocialAccounts()

Gets a list of managed social accounts that are in the org and assigned to the user.

API Version

44.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedSocialAccounts getManagedSocialAccounts()
```

Return Value

Type: ConnectApi.ManagedSocialAccounts

getManagedSocialAccounts(socialNetwork)

Get a list of managed social accounts that are in the org and assigned to the user.

API Version

44.0

Requires Chatter

No

Signature

```
public static ConnectApi.ManagedSocialAccounts
getManagedSocialAccounts(ConnectApi.SocialNetworkProvider socialNetwork)
```

Parameters

socialNetwork
Type: ConnectApi.SocialNetworkProvider

Description: Filters results based on the social network. Values are:

- Facebook
- GooglePlus
- Instagram
- InstagramBusiness
- KakaoTalk
- Kik
- Line
- LinkedIn
- Messenger
- Other
- Pinterest
- QQ
- Rypple
- SinaWeibo
- SMS
- Snapchat
- Telegram
- Twitter
- VKontakte
- WeChat
- WhatsApp
- YouTube

Return Value

Type: ConnectApi.ManagedSocialAccounts

getRelationship(id, socialPersonaId)

Get the follow relationship between a managed social account and a social persona.

API Version

46.0

Requires Chatter

No

Signature

```
public static ConnectApi.SocialAccountRelationship getRelationship(String id, String socialPersonaId)
```

Parameters

id
Type: String

ID of the managed social account.

socialPersonaId
Type: String

ID of the social persona.

Return Value

Type: `ConnectApi.SocialAccountRelationship`

hideSocialPost(socialPostId, socialAccountId)

Hide a social post in its social network.

API Version

46.0

Requires Chatter

No

Signature

```
public static ConnectApi.HideSocialPostIntent hideSocialPost(String socialPostId, String socialAccountId)
```

Parameters

socialPostId
Type: String

ID of the social post to hide.

socialAccountId
Type: String

ID of the social account that hides the post.

Return Value

Type: ConnectApi.HideSocialPostIntent

likeSocialPost(socialPostId, socialAccountId)

Like a social post in its social network.

API Version

46.0

Requires Chatter

No

Signature

```
public static ConnectApi.LikeSocialPostIntent likeSocialPost(String socialPostId, String socialAccountId)
```

Parameters

socialPostId
Type: String

ID of the social post to like.

socialAccountId
Type: String

ID of the social account that likes the post.

Return Value

Type: ConnectApi.LikeSocialPostIntent

massApprove(massApproval)

Approve or reject the publishing of a large number of social posts.

API Version

46.0

Requires Chatter

No

Signature

```
public static ConnectApi.SocialPostMassApprovalOutput
massApprove(ConnectApi.SocialPostMassApprovalInput massApproval)
```

Parameters

massApproval
Type: ConnectApi.SocialPostMassApprovalInput

A ConnectApi.SocialPostMassApprovalInput body that includes a list of social post IDs and the action to approve or reject publishing them.

Return Value

Type: ConnectApi.SocialPostMassApprovalOutput

recallApproval(socialPostId)

Recall an approval request to publish a social post.

API Version

46.0

Requires Chatter

No

Signature

```
public static Void recallApproval(String socialPostId)
```

Parameters

socialPostId
Type: String

ID of the social post.

Return Value

Type: Void

unfollowSocialPersona(socialPersonaId, socialAccountId)

Stop following a social persona in its social network.

API Version

45.0

Requires Chatter

No

Signature

```
public static Void unfollowSocialPersona(String socialPersonaId, String socialAccountId)
```

Parameters

socialPersonaId
Type: String

ID of the social persona to stop following.

socialAccountId
Type: String

ID of the social account that stops following the social persona.

Return Value

Type: Void

unfollowSocialPostPersona(socialPostId, socialAccountId)

Stop following a social persona of a social post in its social network.

API Version

45.0

Requires Chatter

No

Signature

```
public static Void unfollowSocialPostPersona(String socialPostId, String socialAccountId)
```

Parameters

socialPostId
Type: String

ID of the social post authored by the social persona to stop following.

socialAccountId
Type: String

ID of the social account that stops following the social persona.

Return Value

Type: Void

unhideSocialPost(socialPostId, socialAccountId)

Unhide a social post in its social network.

API Version

46.0

Requires Chatter

No

Signature

```
public static Void unhideSocialPost(String socialPostId, String socialAccountId)
```

Parameters

socialPostId
Type: String

ID of the social post to unhide.

socialAccountId
Type: String

ID of the social account that unhides the post.

Return Value

Type: Void

unlikeSocialPost(socialPostId, socialAccountId)

Unlike a social post in its social network.

API Version

46.0

Requires Chatter

No

Signature

```
public static Void unlikeSocialPost(String socialPostId, String socialAccountId)
```

Parameters

socialPostId
Type: String

ID of the social post to unlike.

socialAccountId
Type: String

ID of the social account that unlikes the post.

Return Value

Type: Void

Surveys Class

Send survey invitations by email.

Namespace

ConnectApi

Surveys Methods

These methods are for `Surveys`. All methods are static.

IN THIS SECTION:

sendSurveyInvitationEmail(surveyID, SurveyEmailInput)
Email survey invitations to up to 300 participants. You can email either leads, contacts, or users in your org. Either a link to launch the survey or a question can be embedded in the email invitations.

`sendSurveyInvitationEmail(surveyID, SurveyEmailInput)`

Email survey invitations to up to 300 participants. You can email either leads, contacts, or users in your org. Either a link to launch the survey or a question can be embedded in the email invitations.

API Version

50.0

Requires Chatter

No

Signature

```
public static ConnectApi.SurveyInvitationEmailOutput sendSurveyInvitationEmail(String
surveyID, ConnectApi.SurveyInvitationEmailInput SurveyEmailInput)
```

Parameters

surveyID
Type: String

ID of the survey.

SurveyEmailInput
Type: ConnectApi.SurveyInvitationEmailInput

A ConnectApi.SurveyInvitationEmailInput object.

Return Value

Type: ConnectApi.SurveyInvitationEmailOutput

TaxPlatform Class

Apply or cancel tax.

Namespace

ConnectApi

TaxPlatform Methods

These methods are for TaxPlatform. All methods are static.

IN THIS SECTION:

calculateTax(calculateTax)
Apply tax or cancel tax.

calculateTax(calculateTax)

Apply tax or cancel tax.

API Version

55.0

Requires Chatter

No

Signature

```
global static ConnectApi.CalculateTaxResponse calculateTax(ConnectApi.CalculateTaxRequest
calculateTax)
```

Parameters

calculateTax
Type: ConnectApi.CalculateTaxRequest

Represents a request to calculate tax for one or more line items.

Return Value

Type: `ConnectApi.CalculateTaxResponse`

Topics Class

Access information about topics, such as their descriptions, the number of people talking about them, related topics, and information about groups contributing to the topic. Update a topic's name or description, merge topics, and add and remove topics from records and feed items.

Namespace

ConnectApi

Topics Methods

These methods are for `Topics`. All methods are static.

These methods in this class require Chatter and are subject to the per user, per namespace, per hour rate limit.

- `getGroupsRecentlyTalkingAboutTopic(communityId, topicId)`
- `getRecentlyTalkingAboutTopicsForGroup(communityId, groupId)`
- `getRecentlyTalkingAboutTopicsForUser(communityId, userId)`

All other methods in this class count toward the Salesforce Platform total API request allocations, which are per org and span a 24-hour period.

IN THIS SECTION:

assignTopic(communityId, recordId, topicId)
Assign a topic to a record or feed item.

assignTopicByName(communityId, recordId, topicName)
Assign a topic to a record or feed item.

createTopic(communityId, name, description)
Create a topic.

createTopicDataCategoryRules(communityId, dataCategoryGroup, dataCategory, topicNames)
Create topic and article assignment rules by data category.

deleteTopic(communityId, topicId)
Delete a topic.

getGroupsRecentlyTalkingAboutTopic(communityId, topicId)
Get information about the five groups that most recently contributed to a topic.

getRecentlyTalkingAboutTopicsForGroup(communityId, groupId)
Get up to five topics most recently used in a group.

getRecentlyTalkingAboutTopicsForUser(communityId, userId)
Get up to five topics most recently used by a user.

getRelatedTopics(communityId, topicId)
Get up to five topics most closely related to a topic.

getTopic(communityId, topicId)
Get a topic.

getTopicDataCategoryRules(communityId, dataCategoryGroup, dataCategory)
Get topic and article assignment rules by data category.

getTopics(communityId, recordId)
Get the first page of topics assigned to a record or feed item.

getTopics(communityId)
Get the first page of topics for the org or Experience Cloud site.

getTopics(communityId, sortParam)
Get the first page of sorted topics for the org or community.

getTopics(communityId, pageParam, pageSize)
Get a page of topics.

getTopics(communityId, pageParam, pageSize, sortParam)
Get a page of sorted topics.

getTopics(communityId, q, sortParam)
Get the sorted topics that match the search criteria.

getTopics(communityId, q, pageParam, pageSize)
Get a page of topics that match the search criteria.

getTopics(communityId, q, pageParam, pageSize, sortParam)
Get a page of sorted topics that match the search criteria.

getTopics(communityId, q, exactMatch)
Get the topic that matches the exact, case-insensitive name.

getTopicsOrFallBackToRenamedTopics(communityId, q, exactMatch, fallBackToRenamedTopics)
Get the most recent renamed topic match, if there isn't an exact match.

getTopicSuggestions(communityId, recordId, maxResults)
Get up to a specified number of suggested topics for a record or feed item.

getTopicSuggestions(communityId, recordId)
Get suggested topics for a record or feed item.

getTopicSuggestionsForText(communityId, text, maxResults)
Get up to a specified number of suggested topics for a string of text.

getTopicSuggestionsForText(communityId, text)
Get suggested topics for a string of text.

getTrendingTopics(communityId)
Get trending topics for the org or Experience Cloud site.

getTrendingTopics(communityId, maxResults)
Get up to a specified number of trending topics for the org or Experience Cloud site.

mergeTopics(communityId, topicId, idsToMerge)
Merge up to five secondary topics with a primary topic.

reassignTopicDataCategoryRules(communityId, dataCategoryGroup, dataCategory, topicNames)
Reassign topic and article assignment rules by data category by deleting the existing rules and creating new rules.

reassignTopicsByName(communityId, recordId, topicNames)

Reassign all the topics on a record or feed item, that is, remove all the assigned topics on a record or feed item and add topics. Optionally, provide a list of suggested topics to assign to a record or feed item to improve future topic suggestions.

unassignTopic(communityId, recordId, topicId)

Remove a topic from a record or feed item.

updateTopic(communityId, topicId, topic)

Update the description or name of a topic or merge up to five secondary topics with a primary topic.

updateTopicsForArticlesInDataCategory(communityId, dataCategoryGroup, dataCategory, articleTopicAssignmentJob)

Assign topics to articles and unassign topics from articles in a data category.

`assignTopic(communityId, recordId, topicId)`

Assign a topic to a record or feed item.

API Version

29.0

Requires Chatter

No

Signature

```
public static ConnectApi.Topic assignTopic(String communityId, String recordId, String topicId)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`recordId`
Type: String

ID for a record or feed item.

`topicId`
Type: String

ID for a topic.

Return Value

Type: `ConnectApi.Topic`

Usage

Only users with the Assign Topics permission can add existing topics to records or feed items. Administrators must enable topics for objects before users can add topics to records of that object type.

assignTopicByName(communityId, recordId, topicName)

Assign a topic to a record or feed item.

API Version

29.0

Requires Chatter

No

Signature

```
public static ConnectApi.Topic assignTopicByName(String communityId, String recordId,
String topicName)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

recordId
Type: String

The ID of the record or feed item to which to assign the topic.

topicName
Type: String

The name of a new or existing topic.

Return Value

Type: `ConnectApi.Topic`

Usage

Only users with the Assign Topics permission can add existing topics to records or feed items. Only users with the Create Topics permission can add new topics to records or feed items. Administrators must enable topics for objects before users can add topics to records of that object type.

createTopic(communityId, name, description)

Create a topic.

API Version

36.0

Requires Chatter

No

Signature

```
public static ConnectApi.Topic createTopic(String communityId, String name, String
description)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

name
Type: String

The name of the topic.

description
Type: String

The description of the topic.

Return Value

Type: `ConnectApi.Topic`

Usage

Only users with the Create Topics permission can create a topic.

createTopicDataCategoryRules(communityId, dataCategoryGroup, dataCategory, topicNames)

Create topic and article assignment rules by data category.

API Version

40.0

Requires Chatter

No

Signature

```
public static ConnectApi.TopicPage createTopicDataCategoryRules(String communityId,
String dataCategoryGroup, String dataCategory, ConnectApi.TopicNamesInput topicNames)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

dataCategoryGroup
Type: String

The data category group used by articles.

dataCategory
Type: String

The data category used by articles.

topicNames
Type: `ConnectApi.TopicNamesInput`

A `ConnectApi.TopicNamesInput` object with the names of topics to assign to articles in a data category.

Return Value

Type: `ConnectApi.TopicPage`

`deleteTopic(communityId, topicId)`

Delete a topic.

API Version

29.0

Requires Chatter

No

Signature

```
public static Void deleteTopic(String communityId, String topicId)
```

Parameters

communityId
Type: String,

ID for an Experience Cloud site, `internal`, or `null`.

topicId
Type: String

ID for a topic.

Return Value

Type: Void

Usage

Only users with the Delete Topics or Modify All Data permission can delete topics.

Topic deletion is asynchronous. If a topic is requested before the deletion completes, the response is successful and the `isBeingDeleted` property of `ConnectApi.Topic` is `true` in version 33.0 and later. If a topic is requested after the deletion completes, the response is `ConnectApi.NotFoundException`.

`getGroupsRecentlyTalkingAboutTopic(communityId, topicId)`

Get information about the five groups that most recently contributed to a topic.

API Version

29.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ChatterGroupSummaryPage
getGroupsRecentlyTalkingAboutTopic(String communityId, String topicId)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`topicId`
Type: String

ID for a topic.

Return Value

Type: `ConnectApi.ChatterGroupSummaryPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetGroupsRecentlyTalkingAboutTopic(communityId, topicId, result)

Apex Developer Guide: Testing ConnectApi Code

`getRecentlyTalkingAboutTopicsForGroup(communityId, groupId)`

Get up to five topics most recently used in a group.

API Version

29.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.TopicPage getRecentlyTalkingAboutTopicsForGroup(String communityId, String groupId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

ID for a group.

Return Value

Type: `ConnectApi.TopicPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetRecentlyTalkingAboutTopicsForGroup(communityId, groupId, result)

Apex Developer Guide: Testing ConnectApi Code

`getRecentlyTalkingAboutTopicsForUser(communityId, userId)`

Get up to five topics most recently used by a user.

API Version

29.0

Available to Guest Users

32.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.TopicPage getRecentlyTalkingAboutTopicsForUser(String
communityId, String userId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for a user.

Return Value

Type: `ConnectApi.TopicPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetRecentlyTalkingAboutTopicsForUser(communityId, userId, result)

Apex Developer Guide: Testing ConnectApi Code

`getRelatedTopics(communityId, topicId)`

Get up to five topics most closely related to a topic.

Two topics that are assigned to the same feed item at least three times are related.

API Version

29.0

Available to Guest Users

32.0

Requires Chatter

No

Signature

```
public static ConnectApi.TopicPage getRelatedTopics(String communityId, String topicId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

topicId
Type: String

ID for a topic.

Return Value

Type: `ConnectApi.TopicPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetRelatedTopics(communityId, topicId, result)

Apex Developer Guide: Testing ConnectApi Code

`getTopic(communityId, topicId)`

Get a topic.

API Version

29.0

Available to Guest Users

32.0

Requires Chatter

No

Signature

```
public static ConnectApi.Topic getTopic(String communityId, String topicId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

topicId
Type: String

ID for a topic.

Return Value

Type: `ConnectApi.Topic`

getTopicDataCategoryRules(communityId, dataCategoryGroup, dataCategory)

Get topic and article assignment rules by data category.

API Version

40.0

Requires Chatter

No

Signature

```
public static ConnectApi.TopicPage getTopicDataCategoryRules(String communityId, String
dataCategoryGroup, String dataCategory)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

dataCategoryGroup
Type: String

The data category group used by articles.

dataCategory
Type: String

The data category used by articles.

Return Value

Type: ConnectApi.TopicPage

getTopics(communityId, recordId)

Get the first page of topics assigned to a record or feed item.

API Version

29.0

Available to Guest Users

32.0

Requires Chatter

No

Signature

```
public static ConnectApi.TopicPage getTopics(String communityId, String recordId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

recordId
Type: String

ID for a record or feed item.

Return Value

Type: ConnectApi.TopicPage

Usage

Administrators must enable topics for objects before users can add topics to records of that object type.

getTopics(communityId)

Get the first page of topics for the org or Experience Cloud site.

API Version

29.0

Available to Guest Users

32.0

Requires Chatter

No

Signature

```
public static ConnectApi.TopicPage getTopics(String communityId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

Return Value

Type: `ConnectApi.TopicPage`

`getTopics(communityId, sortParam)`

Get the first page of sorted topics for the org or community.

API Version

29.0

Available to Guest Users

32.0

Requires Chatter

No

Signature

```
public static ConnectApi.TopicPage getTopics(String communityId, ConnectApi.TopicSort
sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

sortParam
Type: `ConnectApi.TopicSort`

Values are:

- `popularDesc`—Sorts topics by popularity with the most popular first. This value is the default.
- `alphaAsc`—Sorts topics alphabetically.

Return Value

Type: `ConnectApi.TopicPage`

`getTopics(communityId, pageParam, pageSize)`

Get a page of topics.

API Version

29.0

Available to Guest Users

32.0

Requires Chatter

No

Signature

```
public static ConnectApi.TopicPage getTopics(String communityId, Integer pageParam,
Integer pageSize)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`pageParam`
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

`pageSize`
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.TopicPage`

`getTopics(communityId, pageParam, pageSize, sortParam)`

Get a page of sorted topics.

API Version

29.0

Available to Guest Users

32.0

Requires Chatter

No

Signature

```
public static ConnectApi.TopicPage getTopics(String communityId, Integer pageParam,
Integer pageSize, ConnectApi.TopicSort sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.TopicSort`

Values are:

- `popularDesc`—Sorts topics by popularity with the most popular first. This value is the default.
- `alphaAsc`—Sorts topics alphabetically.

Return Value

Type: `ConnectApi.TopicPage`

`getTopics(communityId, q, sortParam)`

Get the sorted topics that match the search criteria.

API Version

29.0

Available to Guest Users

32.0

Requires Chatter

No

Signature

```
public static ConnectApi.TopicPage getTopics(String communityId, String q,
ConnectApi.TopicSort sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Specifies the string to search. The string must contain at least two characters, not including wildcards.

sortParam
Type: `ConnectApi.TopicSort`

Values are:

- `popularDesc`—Sorts topics by popularity with the most popular first. This value is the default.
- `alphaAsc`—Sorts topics alphabetically.

Return Value

Type: `ConnectApi.TopicPage`

`getTopics(communityId, q, pageParam, pageSize)`

Get a page of topics that match the search criteria.

API Version

29.0

Available to Guest Users

32.0

Requires Chatter

No

Signature

```
public static ConnectApi.TopicPage getTopics(String communityId, String q, Integer pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Specifies the string to search. The string must contain at least two characters, not including wildcards.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.TopicPage`

`getTopics(communityId, q, pageParam, pageSize, sortParam)`

Get a page of sorted topics that match the search criteria.

API Version

29.0

Available to Guest Users

32.0

Requires Chatter

No

Signature

```
public static ConnectApi.TopicPage getTopics(String communityId, String q, Integer pageParam, Integer pageSize, ConnectApi.TopicSort sortParam)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Specifies the string to search. The string must contain at least two characters, not including wildcards.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

sortParam
Type: `ConnectApi.TopicSort`

Values are:

- `popularDesc`—Sorts topics by popularity with the most popular first. This value is the default.
- `alphaAsc`—Sorts topics alphabetically.

Return Value

Type: `ConnectApi.TopicPage`

`getTopics(communityId, q, exactMatch)`

Get the topic that matches the exact, case-insensitive name.

API Version

33.0

Available to Guest Users

33.0

Requires Chatter

No

Signature

```
public static ConnectApi.TopicPage getTopics(String communityId, String q, Boolean
exactMatch)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Specifies the string to search. The string must contain at least two characters, not including wildcards.

exactMatch
Type: Boolean

Specify `true` to find a topic by its exact, case-insensitive name.

Return Value

Type: `ConnectApi.TopicPage`

getTopicsOrFallBackToRenamedTopics(communityId, q, exactMatch, fallBackToRenamedTopics)

Get the most recent renamed topic match, if there isn't an exact match.

API Version

35.0

Available to Guest Users

35.0

Requires Chatter

No

Signature

```
public static ConnectApi.TopicPage getTopicsOrFallBackToRenamedTopics(String communityId,
String q, Boolean exactMatch, Boolean fallBackToRenamedTopics)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

q
Type: String

Specifies the string to search. The string must contain at least two characters, not including wildcards.

exactMatch
Type: Boolean

Specify `true` to find a topic by its exact, case-insensitive name or to find the most recent renamed topic match if there isn't an exact match.

fallBackToRenamedTopics
Type: Boolean

Specify `true` and if there isn't an exact match, the most recent renamed topic match is returned. If there are multiple renamed topic matches, only the most recent is returned. If there are no renamed topic matches, an empty collection is returned.

Return Value

Type: `ConnectApi.TopicPage`

`getTopicSuggestions(communityId, recordId, maxResults)`

Get up to a specified number of suggested topics for a record or feed item.

API Version

29.0

Requires Chatter

No

Signature

```
public static ConnectApi.TopicSuggestionPage getTopicSuggestions(String communityId,
String recordId, Integer maxResults)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`recordId`
Type: String

ID for a record or feed item.

`maxResults`
Type: Integer

Maximum number of topic suggestions that get returned. The default is 5. Value must be greater than 0 and less than or equal to 25.

Return Value

Type: `ConnectApi.TopicSuggestionPage`

Usage

Administrators must enable topics for objects before users can see suggested topics for records of that object type.

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetTopicSuggestions(communityId, recordId, maxResults, result)

Apex Developer Guide: Testing ConnectApi Code

`getTopicSuggestions(communityId, recordId)`

Get suggested topics for a record or feed item.

API Version

29.0

Requires Chatter

No

Signature

```
public static ConnectApi.TopicSuggestionPage getTopicSuggestions(String communityId,
String recordId)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`recordId`
Type: String

ID for a record or feed item.

Return Value

Type: `ConnectApi.TopicSuggestionPage`

Usage

Administrators must enable topics for objects before users can see suggested topics for records of that object type.

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetTopicSuggestions(communityId, recordId, result)

Apex Developer Guide: Testing ConnectApi Code

getTopicSuggestionsForText(communityId, text, maxResults)

Get up to a specified number of suggested topics for a string of text.

API Version

29.0

Requires Chatter

No

Signature

```
public static ConnectApi.TopicSuggestionPage getTopicSuggestionsForText(String
communityId, String text, Integer maxResults)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

text
Type: String

String of text.

maxResults
Type: Integer

Maximum number of topic suggestions that get returned. The default is 5. Value must be greater than 0 and less than or equal to 25.

Return Value

Type: `ConnectApi.TopicSuggestionPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetTopicSuggestionsForText(communityId, text, maxResults, result)

Apex Developer Guide: Testing ConnectApi Code

getTopicSuggestionsForText(communityId, text)

Get suggested topics for a string of text.

API Version

29.0

Requires Chatter

No

Signature

```
public static ConnectApi.TopicSuggestionPage getTopicSuggestionsForText(String communityId, String text)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

text
Type: String

String of text.

Return Value

Type: `ConnectApi.TopicSuggestionPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetTopicSuggestionsForText(communityId, text, result)

Apex Developer Guide: Testing ConnectApi Code

`getTrendingTopics(communityId)`

Get trending topics for the org or Experience Cloud site.

API Version

29.0

Available to Guest Users

32.0

Requires Chatter

No

Signature

```
public static ConnectApi.TopicPage getTrendingTopics(String communityId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

Return Value

Type: `ConnectApi.TopicPage`

Usage

The more frequently people add a specific topic to their posts and comments and comment on or like posts with the same topic over a short period, the more likely it is to become a trending topic. For example, if your coworkers are attending the upcoming Dreamforce conference and have started discussing it in Chatter, you might see a trending topic for Dreamforce. A trending topic is not solely based on popularity and usually relates to a one-time or infrequent event that has a spike in activity, such as a conference or a project deadline.

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetTrendingTopics(communityId, result)

Apex Developer Guide: Testing ConnectApi Code

`getTrendingTopics(communityId, maxResults)`

Get up to a specified number of trending topics for the org or Experience Cloud site.

API Version

29.0

Available to Guest Users

32.0

Requires Chatter

No

Signature

```
public static ConnectApi.TopicPage getTrendingTopics(String communityId, Integer
maxResults)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

maxResults
Type: Integer

Maximum number of topic suggestions that get returned. The default is 5. Value must be greater than 0 and less than or equal to 25.

Return Value

Type: `ConnectApi.TopicPage`

Usage

The more frequently people add a specific topic to their posts and comments and comment on or like posts with the same topic over a short period, the more likely it is to become a trending topic. For example, if your coworkers are attending the upcoming Dreamforce conference and have started discussing it in Chatter, you might see a trending topic for Dreamforce. A trending topic is not solely based on popularity and usually relates to a one-time or infrequent event that has a spike in activity, such as a conference or a project deadline.

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestGetTrendingTopics(communityId, maxResults, result)

Apex Developer Guide: Testing ConnectApi Code

`mergeTopics(communityId, topicId, idsToMerge)`

Merge up to five secondary topics with a primary topic.

API Version

33.0

Requires Chatter

No

Signature

```
public static ConnectApi.Topic mergeTopics(String communityId, String topicId,
List<String> idsToMerge)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

topicId
Type: String

The ID for the primary topic for the merge. If this topic is a managed topic, it retains its topic type, topic images, and children topics.

idsToMerge
Type: List<String>

A list of up to five comma-separated secondary topic IDs to merge with the primary topic. If any of the secondary topics are navigational or featured topics, they lose their topic type, topic images, and children topics. Their feed items are reassigned to the primary topic. If you merge a topic with a content topic, the content associations are preserved. If you merge a topic with an inactive endorsee, the endorsement isn't mapped to the primary topic.

Return Value

Type: `ConnectApi.Topic`

Usage

Only users with the Delete Topics or Modify All Data permission can merge topics.

`reassignTopicDataCategoryRules(communityId, dataCategoryGroup, dataCategory, topicNames)`

Reassign topic and article assignment rules by data category by deleting the existing rules and creating new rules.

API Version

40.0

Requires Chatter

No

Signature

```
public static ConnectApi.TopicPage reassignTopicDataCategoryRules(String communityId,
String dataCategoryGroup, String dataCategory, ConnectApi.TopicNamesInput topicNames)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

dataCategoryGroup
Type: String

The data category group used by articles.

dataCategory
Type: String

The data category used by articles.

topicNames
Type: ConnectApi.TopicNamesInput

A ConnectApi.TopicNamesInput object with the names of topics to reassign to articles in a data category.

Return Value

Type: ConnectApi.TopicPage

reassignTopicsByName(communityId, recordId, topicNames)

Reassign all the topics on a record or feed item, that is, remove all the assigned topics on a record or feed item and add topics. Optionally, provide a list of suggested topics to assign to a record or feed item to improve future topic suggestions.

API Version

35.0

Requires Chatter

No

Signature

```
public static ConnectApi.TopicPage reassignTopicsByName(String communityId, String recordId, ConnectApi.TopicNamesInput topicNames)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, internal, or null.

recordId
Type: String

The ID of the record or feed item to which to assign the topic.

topicNames
Type: ConnectApi.TopicNamesInput

A list of topics to replace the currently assigned topics. Optionally, a list of suggested topics to assign to improve future topic suggestions.

Return Value

Type: ConnectApi.TopicPage

Usage

Only users with the Assign Topics permission can remove topics from records or feed items and add existing topics to records or feed items. Only users with the Create Topics permission can add new topics to records or feed items. Administrators must enable topics for objects before users can add topics to records of that object type.

unassignTopic(communityId, recordId, topicId)

Remove a topic from a record or feed item.

API Version

29.0

Requires Chatter

No

Signature

```
public static Void unassignTopic(String communityId, String recordId, String topicId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

recordId
Type: String

ID for a record or feed item.

topicId
Type: String

ID for a topic.

Return Value

Type: Void

Usage

Only users with the Assign Topics permission can remove topics from feed items or records. Administrators must enable topics for objects before users can add topics to records of that object type.

updateTopic(communityId, topicId, topic)

Update the description or name of a topic or merge up to five secondary topics with a primary topic.

API Version

29.0

Requires Chatter

No

Signature

```
public static ConnectApi.Topic updateTopic(String communityId, String topicId,
ConnectApi.TopicInput topic)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

topicId
Type: String

ID for a topic.

topic
Type: `ConnectApi.TopicInput`

A `ConnectApi.TopicInput` object containing the name and description of the topic or up to five comma-separated secondary topic IDs to merge with the primary topic.

Return Value

Type: `ConnectApi.Topic`

Usage

Only users with the Edit Topics or Modify All Data permission can update topic names and descriptions. Only users with the Delete Topics or Modify All Data permission can merge topics.

updateTopicsForArticlesInDataCategory(communityId, dataCategoryGroup, dataCategory, articleTopicAssignmentJob)

Assign topics to articles and unassign topics from articles in a data category.

API Version

40.0

Requires Chatter

No

Signature

```
public static ConnectApi.TopicPage updateTopicsForArticlesInDataCategory(String communityId, String dataCategoryGroup, String dataCategory, ConnectApi.ArticleTopicAssignmentJobInput articleTopicAssignmentJob)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

dataCategoryGroup
Type: String

The data category group used by articles.

dataCategory
Type: String

The data category used by articles.

articleTopicAssignmentJob
Type: `ConnectApi.ArticleTopicAssignmentJobInput`

A `ConnectApi.ArticleTopicAssignmentJobInput` object that indicates the operation to take on which topics.

Return Value

Type: `ConnectApi.TopicPage`

Topics Test Methods

These test methods are for `Topics`. All methods are static.

For information about using these methods to test your `ConnectApi` code, see Testing ConnectApi Code.

`setTestGetGroupsRecentlyTalkingAboutTopic(communityId, topicId, result)`

Register a `ConnectApi.ChatterGroupSummaryPage` object to be returned when the matching `ConnectApi.getGroupsRecentlyTalkingAboutTopic` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

29.0

Signature

```
public static Void setTestGetGroupsRecentlyTalkingAboutTopic(String communityId, String topicId, ConnectApi.ChatterGroupSummaryPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

topicId
Type: String

ID for a topic.

result
Type: `ConnectApi.ChatterGroupSummaryPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getGroupsRecentlyTalkingAboutTopic(communityId, topicId)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetRecentlyTalkingAboutTopicsForGroup(communityId, groupId, result)`

Register a `ConnectApi.TopicPage` object to be returned when the matching `ConnectApi.getRecentlyTalkingAboutTopicsForGroup` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

29.0

Signature

```
public static Void setTestGetRecentlyTalkingAboutTopicsForGroup(String communityId,
String groupId, ConnectApi.TopicPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

groupId
Type: String

ID for a group.

result
Type: `ConnectApi.TopicPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getRecentlyTalkingAboutTopicsForGroup(communityId, groupId)

Apex Developer Guide: Testing ConnectApi Code

setTestGetRecentlyTalkingAboutTopicsForUser(communityId, userId, result)

Register a `ConnectApi.TopicPage` object to be returned when the matching `ConnectApi.getRecentlyTalkingAboutTopicsForUser` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

29.0

Signature

```
public static Void setTestGetRecentlyTalkingAboutTopicsForUser(String communityId,
String userId, ConnectApi.TopicPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for a user.

result
Type: `ConnectApi.TopicPage`

Specify the test topics page.

Return Value

Type: Void

SEE ALSO:

getRecentlyTalkingAboutTopicsForUser(communityId, userId)

Apex Developer Guide: Testing ConnectApi Code

setTestGetRelatedTopics(communityId, topicId, result)

Register a `ConnectApi.TopicPage` object to be returned when the matching `ConnectApi.getRelatedTopics` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

29.0

Signature

```
public static Void setTestGetRelatedTopics(String communityId, String topicId,
ConnectApi.TopicPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

topicId
Type: String

ID for a topic.

result
Type: `ConnectApi.TopicPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getRelatedTopics(communityId, topicId)

Apex Developer Guide: Testing ConnectApi Code

setTestGetTopicSuggestions(communityId, recordId, maxResults, result)

Register a `ConnectApi.TopicSuggestionPage` object to be returned when the matching `ConnectApi.getTopicSuggestions` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

29.0

Signature

```
public static Void setTestGetTopicSuggestions(String communityId, String recordId,
Integer maxResults, ConnectApi.TopicSuggestionPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

recordId
Type: String

ID for a record or feed item.

maxResults
Type: Integer

Maximum number of topic suggestions that get returned. The default is 5. Value must be greater than 0 and less than or equal to 25.

result
Type: `ConnectApi.TopicSuggestionPage`

Specify the test topic suggestions page.

Return Value

Type: Void

SEE ALSO:

getTopicSuggestions(communityId, recordId, maxResults)

Apex Developer Guide: Testing ConnectApi Code

setTestGetTopicSuggestions(communityId, recordId, result)

Register a `ConnectApi.TopicSuggestionPage` object to be returned when the matching `ConnectApi.getTopicSuggestions` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

29.0

Signature

```
public static Void setTestGetTopicSuggestions(String communityId, String recordId,
ConnectApi.TopicSuggestionPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

recordId
Type: String

ID for a record or feed item.

result
Type: `ConnectApi.TopicSuggestionPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getTopicSuggestions(communityId, recordId)

Apex Developer Guide: Testing ConnectApi Code

setTestGetTopicSuggestionsForText(communityId, text, maxResults, result)

Register a `ConnectApi.TopicSuggestionPage` object to be returned when the matching `ConnectApi.getTopicSuggestionsForText` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

29.0

Signature

```
public static Void setTestGetTopicSuggestionsForText(String communityId, String text,
Integer maxResults, ConnectApi.TopicSuggestionPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

text
Type: String

String of text.

maxResults
Type: Integer

Maximum number of topic suggestions that get returned. The default is 5. Value must be greater than 0 and less than or equal to 25.

result
Type: `ConnectApi.TopicSuggestionPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getTopicSuggestionsForText(communityId, text, maxResults)

Apex Developer Guide: Testing ConnectApi Code

setTestGetTopicSuggestionsForText(communityId, text, result)

Register a `ConnectApi.TopicSuggestionPage` object to be returned when the matching `ConnectApi.getTopicSuggestionsForText` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

29.0

Signature

```
public static Void setTestGetTopicSuggestionsForText(String communityId, String text,
ConnectApi.TopicSuggestionPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

text
Type: String

String of text.

result
Type: `ConnectApi.TopicSuggestionPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getTopicSuggestionsForText(communityId, text)

Apex Developer Guide: Testing ConnectApi Code

setTestGetTrendingTopics(communityId, result)

Register a `ConnectApi.TopicPage` object to be returned when the matching `ConnectApi.getTrendingTopics` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

29.0

Signature

```
public static Void setTestGetTrendingTopics(String communityId, ConnectApi.TopicPage
result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

result
Type: `ConnectApi.TopicPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getTrendingTopics(communityId)

Apex Developer Guide: Testing ConnectApi Code

`setTestGetTrendingTopics(communityId, maxResults, result)`

Register a `ConnectApi.TopicPage` object to be returned when the matching `ConnectApi.getTrendingTopics` method is called in a test context. Use the method with the same parameters or you receive an exception.

API Version

29.0

Signature

```
public static Void setTestGetTrendingTopics(String communityId, Integer maxResults,
ConnectApi.TopicPage result)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

maxResults
Type: Integer

Maximum number of topic suggestions that get returned. The default is 5. Value must be greater than 0 and less than or equal to 25.

result
Type: `ConnectApi.TopicPage`

Object containing test data.

Return Value

Type: Void

SEE ALSO:

getTrendingTopics(communityId, maxResults)

Apex Developer Guide: Testing ConnectApi Code

UserProfiles Class

Access user profile data. The user profile data populates the profile page (also called the Chatter profile page). This data includes user information (such as address, manager, and phone number), some user capabilities (permissions), and a set of subtab apps, which are custom tabs on the profile page.

Namespace

ConnectApi

UserProfiles Methods

These methods are for `UserProfiles`. All methods are static.

All methods in this class require Chatter and are subject to the per user, per namespace, per hour rate limit.

IN THIS SECTION:

deleteBannerPhoto(communityId, userId)
Delete a user's banner photo.

deletePhoto(communityId, userId)
Delete a user's photo.

getBannerPhoto(communityId, userId)
Get a user's banner photo.

getPhoto(communityId, userId)
Get a user's photo.

getUserProfile(communityId, userId)
Get the user profile of the context user.

setBannerPhoto(communityId, userId, fileId, versionNumber)
Set an uploaded file as a user's banner photo.

setBannerPhoto(communityId, userId, fileUpload)
Set a file that hasn't been uploaded as a user's banner photo.

setBannerPhotoWithAttributes(communityId, userId, bannerPhoto)
Set and crop an uploaded file as a user's banner photo.

setBannerPhotoWithAttributes(communityId, userId, bannerPhoto, fileUpload)
Set and crop a file that hasn't been uploaded as a user's banner photo.

setPhoto(communityId, userId, fileId, versionNumber)
Set an uploaded file as a user's photo.

setPhoto(communityId, userId, fileUpload)
Set a file that hasn't been uploaded as a user's photo.

setPhotoWithAttributes(communityId, userId, photo)
Set and crop an uploaded file as a user's photo.

setPhotoWithAttributes(communityId, userId, photo, fileUpload)
Set and crop a file that hasn't been uploaded as a user's photo.

`deleteBannerPhoto(communityId, userId)`

Delete a user's banner photo.

API Version

36.0

Requires Chatter

Yes

Signature

```
public static Void deleteBannerPhoto(String communityId, String userId)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`userId`
Type: String

ID of the user.

Return Value

Type: Void

deletePhoto(communityId, userId)

Delete a user's photo.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static Void deletePhoto(String communityId, String userId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for the context user or the keyword `me`.

Return Value

Type: Void

getBannerPhoto(communityId, userId)

Get a user's banner photo.

API Version

36.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.BannerPhoto getBannerPhoto(String communityId, String userId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID of the user.

Return Value

Type: `ConnectApi.BannerPhoto`

getPhoto(communityId, userId)

Get a user's photo.

API Version

35.0

Available to Guest Users

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Photo getPhoto(String communityId, String userId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for a user.

Return Value

Type: `ConnectApi.Photo`

getUserProfile(communityId, userId)

Get the user profile of the context user.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.UserProfile getUserProfile(String communityId, String userId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for a user.

Return Value

Type: `ConnectApi.UserProfile`

setBannerPhoto(communityId, userId, fileId, versionNumber)

Set an uploaded file as a user's banner photo.

API Version

36.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.BannerPhoto setBannerPhoto(String communityId, String userId,
String fileId, Integer versionNumber)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID of the user.

fileId
Type: String

ID of the uploaded file to use as the user banner. The key prefix must be 069, and the image must be smaller than 8 MB.

versionNumber
Type: Integer

Version number of the file. Specify an existing version number or, to get the latest version, specify `null`.

Return Value

Type: `ConnectApi.BannerPhoto`

Usage

Photos are processed asynchronously and might not be visible right away.

setBannerPhoto(communityId, userId, fileUpload)

Set a file that hasn't been uploaded as a user's banner photo.

API Version

36.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.BannerPhoto setBannerPhoto(String communityId, String userId,
ConnectApi.BinaryInput fileUpload)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID of the user.

fileUpload
Type: `ConnectApi.BinaryInput`

File to use as the photo. The content type must be usable as an image.

Return Value

Type: `ConnectApi.BannerPhoto`

Usage

Photos are processed asynchronously and might not be visible right away.

setBannerPhotoWithAttributes(communityId, userId, bannerPhoto)

Set and crop an uploaded file as a user's banner photo.

API Version

36.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.BannerPhoto setBannerPhotoWithAttributes(String communityId,
String userId, ConnectApi.BannerPhotoInput bannerPhoto)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID of the user.

bannerPhoto
Type: `ConnectApi.BannerPhotoInput`

A `ConnectApi.BannerPhotoInput` object that specifies the ID and version of the file, and how to crop the file.

Return Value

Type: `ConnectApi.BannerPhoto`

Usage

Photos are processed asynchronously and might not be visible right away.

setBannerPhotoWithAttributes(communityId, userId, bannerPhoto, fileUpload)

Set and crop a file that hasn't been uploaded as a user's banner photo.

API Version

36.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.BannerPhoto setBannerPhotoWithAttributes(String communityId,
String userId, ConnectApi.BannerPhotoInput bannerPhoto, ConnectApi.BinaryInput
fileUpload)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID of the user.

bannerPhoto
Type: `ConnectApi.BannerPhotoInput`

A `ConnectApi.BannerPhotoInput` object specifying the cropping parameters.

fileUpload
Type: `ConnectApi.BinaryInput`

File to use as the photo. The content type must be usable as an image.

Return Value

Type: `ConnectApi.BannerPhoto`

Usage

Photos are processed asynchronously and might not be visible right away.

setPhoto(communityId, userId, fileId, versionNumber)

Set an uploaded file as a user's photo.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Photo setPhoto(String communityId, String userId, String
fileId, Integer versionNumber)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for the context user or the keyword `me`.

fileId
Type: String

ID of an uploaded file. The file must be an image, and be smaller than 2 GB.

versionNumber
Type: Integer

Version number of the existing file. Specify either an existing version number, or `null` to get the latest version.

Return Value

Type: `ConnectApi.Photo`

Usage

Photos are processed asynchronously and might not be visible right away.

setPhoto(communityId, userId, fileUpload)

Set a file that hasn't been uploaded as a user's photo.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Photo setPhoto(String communityId, String userId,
ConnectApi.BinaryInput fileUpload)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for the context user or the keyword `me`.

`fileUpload`
Type: `ConnectApi.BinaryInput`

File to use as the photo. The content type must be usable as an image.

Return Value

Type: `ConnectApi.Photo`

Usage

Photos are processed asynchronously and might not be visible right away.

`setPhotoWithAttributes(communityId, userId, photo)`

Set and crop an uploaded file as a user's photo.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Photo setPhotoWithAttributes(String communityId, String userId,
ConnectApi.PhotoInput photo)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`userId`
Type: String

ID for the context user or the keyword `me`.

`photo`
Type: `ConnectApi.PhotoInput`

A `ConnectApi.PhotoInput` object specifying the file ID, version number, and cropping parameters.

Return Value

Type: `ConnectApi.Photo`

Usage

Photos are processed asynchronously and might not be visible right away.

`setPhotoWithAttributes(communityId, userId, photo, fileUpload)`

Set and crop a file that hasn't been uploaded as a user's photo.

API Version

35.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Photo setPhotoWithAttributes(String communityId, String userId,
ConnectApi.PhotoInput photo, ConnectApi.BinaryInput fileUpload)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

userId
Type: String

ID for the context user or the keyword `me`.

photo
Type: `ConnectApi.PhotoInput`

A `ConnectApi.PhotoInput` object specifying the cropping parameters.

fileUpload
Type: `ConnectApi.BinaryInput`

File to use as the photo. The content type must be usable as an image.

Return Value

Type: `ConnectApi.Photo`

Usage

Photos are processed asynchronously and might not be visible right away.

Zones Class

Access information about Chatter Answers zones in your organization. Zones organize questions into logical groups, with each zone having its own focus and unique questions.

Note: With the Spring '18 release, Salesforce no longer supports Chatter Answers. Users of Chatter Answers can post, answer, comment, or view existing Chatter Answers data, but support and updates are scheduled to end. We recommend transitioning to Chatter Questions. For more information, see End of Support for Chatter Answers in Spring '18.

Namespace

ConnectApi

Zones Methods

These methods are for `Zones`. All methods are static.

All methods in this class require Chatter and are subject to the per user, per namespace, per hour rate limit.

IN THIS SECTION:

getZone(communityId, zoneId)
Get a zone.

getZones(communityId)
Get a list of zones.

getZones(communityId, pageParam, pageSize)
Get a page of zones.

searchInZone(communityId, zoneId, q, filter)
Search articles or questions in a zone.

searchInZone(communityId, zoneId, q, filter, pageParam, pageSize)
Search a page of articles or questions in a zone.

searchInZone(communityId, zoneId, q, filter, language)
Search articles or questions in a zone, and specify the language of the results.

`getZone(communityId, zoneId)`

Get a zone.

Note: With the Spring '18 release, Salesforce no longer supports Chatter Answers. Users of Chatter Answers can post, answer, comment, or view existing Chatter Answers data, but support and updates are scheduled to end. We recommend transitioning to Chatter Questions. For more information, see End of Support for Chatter Answers in Spring '18.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Zone getZone(String communityId, String zoneId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

zoneId
Type: String

The ID of a zone.

Return Value

Type: `ConnectApi.Zone`

getZones(communityId)

Get a list of zones.

Note: With the Spring '18 release, Salesforce no longer supports Chatter Answers. Users of Chatter Answers can post, answer, comment, or view existing Chatter Answers data, but support and updates are scheduled to end. We recommend transitioning to Chatter Questions. For more information, see End of Support for Chatter Answers in Spring '18.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ZonePage getZones(String communityId)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

Return Value

Type: `ConnectApi.ZonePage`

getZones(communityId, pageParam, pageSize)

Get a page of zones.

Note: With the Spring '18 release, Salesforce no longer supports Chatter Answers. Users of Chatter Answers can post, answer, comment, or view existing Chatter Answers data, but support and updates are scheduled to end. We recommend transitioning to Chatter Questions. For more information, see End of Support for Chatter Answers in Spring '18.

API Version

29.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.Zone getZones(String communityId, Integer pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

pageParam
Type: Integer

Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.ZonePage`

searchInZone(communityId, zoneId, q, filter)

Search articles or questions in a zone.

Note: With the Spring '18 release, Salesforce no longer supports Chatter Answers. Users of Chatter Answers can post, answer, comment, or view existing Chatter Answers data, but support and updates are scheduled to end. We recommend transitioning to Chatter Questions. For more information, see End of Support for Chatter Answers in Spring '18.

API Version

29.0

Available to Guest Users

37.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ZoneSearchPage searchInZone(String communityId, String zoneId,
String q, ConnectApi.ZoneSearchResultType filter)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

zoneId
Type: String

ID of a zone.

q
Type: String

Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

filter
Type: `ConnectApi.ZoneSearchResultType`

A `ZoneSearchResultType` enum value. One of the following:

- `Article`—Search results contain only articles.
- `Question`—Search results contain only questions.

Return Value

Type: `ConnectApi.ZoneSearchPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchInZone(communityId, zoneId, q, filter, result)

Apex Developer Guide: Testing ConnectApi Code

`searchInZone(communityId, zoneId, q, filter, pageParam, pageSize)`

Search a page of articles or questions in a zone.

Note: With the Spring '18 release, Salesforce no longer supports Chatter Answers. Users of Chatter Answers can post, answer, comment, or view existing Chatter Answers data, but support and updates are scheduled to end. We recommend transitioning to Chatter Questions. For more information, see End of Support for Chatter Answers in Spring '18.

API Version

29.0

Available to Guest Users

37.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ZoneSearchPage searchInZone(String communityId, String zoneId,
String q, ConnectApi.ZoneSearchResultType filter, String pageParam, Integer pageSize)
```

Parameters

communityId
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

zoneId
Type: String

ID of a zone.

q
Type: String

Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

filter
Type: `ConnectApi.ZoneSearchResultType`

A `ZoneSearchResultType` enum value. One of the following:

- `Article`—Search results contain only articles.
- `Question`—Search results contain only questions.

pageParam
Type: String

Specifies the page token to use to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

Return Value

Type: `ConnectApi.ZoneSearchPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchInZone(communityId, zoneId, q, filter, pageParam, pageSize, result)

Apex Developer Guide: Testing ConnectApi Code

`searchInZone(communityId, zoneId, q, filter, language)`

Search articles or questions in a zone, and specify the language of the results.

Note: With the Spring '18 release, Salesforce no longer supports Chatter Answers. Users of Chatter Answers can post, answer, comment, or view existing Chatter Answers data, but support and updates are scheduled to end. We recommend transitioning to Chatter Questions. For more information, see End of Support for Chatter Answers in Spring '18.

API Version

36.0

Available to Guest Users

37.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.ZoneSearchPage searchInZone(String communityId, String zoneId,
String q, ConnectApi.ZoneSearchResultType filter, String language)
```

Parameters

`communityId`
Type: String

ID for an Experience Cloud site, `internal`, or `null`.

`zoneId`
Type: String

ID of a zone.

`q`
Type: String

Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

`filter`
Type: `ConnectApi.ZoneSearchResultType`

- `Article`—Search results contain only articles.

- `Question`—Search results contain only questions.

`language`
Type: String

The language of the articles or questions. The value must be a Salesforce supported locale code.

Return Value

Type: `ConnectApi.ZoneSearchPage`

Usage

To test code that uses this method, use the matching set test method (prefix the method name with `setTest`). Use the set test method with the same parameters or the code throws an exception.

SEE ALSO:

setTestSearchInZone(communityId, zoneId, q, filter, language, result)

Zones Test Methods

These test methods are for `Zones`. All methods are static.

For information about using these methods to test your `ConnectApi` code, see Testing ConnectApi Code.

`setTestSearchInZone(communityId, zoneId, q, filter, result)`

Register a `ConnectApi.ZoneSearchPage` object to be returned when `searchInZone(communityId, zoneId, q, filter)` is called in a test context. Use the method with the same parameters or you receive an exception.

Note: With the Spring '18 release, Salesforce no longer supports Chatter Answers. Users of Chatter Answers can post, answer, comment, or view existing Chatter Answers data, but support and updates are scheduled to end. We recommend transitioning to Chatter Questions. For more information, see End of Support for Chatter Answers in Spring '18.

API Version

29.0

Signature

```
public static Void setTestSearchInZone(String communityId, String zoneId, String q,
ConnectApi.ZoneSearchResultType filter, ConnectApi.ZoneSearchPage result)
```

Parameters

`communityId`
Type: String

Use either the ID for an Experience Cloud site, `internal`, or `null`.

`zoneId`
Type: String

The ID of a zone.

q
Type: String

Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

filter
Type: `ConnectApi.ZoneSearchResultType`

A `ZoneSearchResultType` enum value. One of the following:

- `Article`—Search results contain only articles.
- `Question`—Search results contain only questions.

result
Type: `ConnectApi.ZoneSearchPage`

The object containing test data.

Return Value

Type: Void

SEE ALSO:

searchInZone(communityId, zoneId, q, filter)

Apex Developer Guide: Testing ConnectApi Code

`setTestSearchInZone(communityId, zoneId, q, filter, pageParam, pageSize, result)`

Register a `ConnectApi.ZoneSearchPage` object to be returned when `searchInZone(communityId, zoneId, q, filter, pageParam, pageSize)` is called in a test context. Use the method with the same parameters or you receive an exception.

Note: With the Spring '18 release, Salesforce no longer supports Chatter Answers. Users of Chatter Answers can post, answer, comment, or view existing Chatter Answers data, but support and updates are scheduled to end. We recommend transitioning to Chatter Questions. For more information, see End of Support for Chatter Answers in Spring '18.

API Version

29.0

Signature

```
public static Void setTestSearchInZone(String communityId, String zoneId, String q,
ConnectApi.ZoneSearchResultType filter, String pageParam, Integer pageSize,
ConnectApi.ZoneSearchPage result)
```

Parameters

communityId
Type: String

Use either the ID for an Experience Cloud site, `internal`, or `null`.

zoneId
Type: String

The ID of a zone.

q
Type: String

Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

filter
Type: `ConnectApi.ZoneSearchResultType`

A `ZoneSearchResultType` enum value. One of the following:

- `Article`—Search results contain only articles.
- `Question`—Search results contain only questions.

pageParam
Type: String

Specifies the page token to be used to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.

pageSize
Type: Integer

Specifies the number of items per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.

result
Type: `ConnectApi.ZoneSearchPage`

The object containing test data.

Return Value

Type: Void

SEE ALSO:

searchInZone(communityId, zoneId, q, filter, pageParam, pageSize)

Apex Developer Guide: Testing ConnectApi Code

`setTestSearchInZone(communityId, zoneId, q, filter, language, result)`

Register a `ConnectApi.ZoneSearchPage` object to be returned when `searchInZone(communityId, zoneId, q, filter, language)` is called in a test context. Use the method with the same parameters or you receive an exception.

Note: With the Spring '18 release, Salesforce no longer supports Chatter Answers. Users of Chatter Answers can post, answer, comment, or view existing Chatter Answers data, but support and updates are scheduled to end. We recommend transitioning to Chatter Questions. For more information, see End of Support for Chatter Answers in Spring '18.

API Version

36.0

Signature

```
public static Void setTestSearchInZone(String communityId, String zoneId, String q,
ConnectApi.ZoneSearchResultType filter, String language, ConnectApi.ZoneSearchPage
result)
```

Parameters

communityId
Type: String

Use either the ID for an Experience Cloud site, `internal`, or `null`.

zoneId
Type: String

The ID of a zone.

q
Type: String

Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.

filter
Type: `ConnectApi.ZoneSearchResultType`

- `Article`—Search results contain only articles.
- `Question`—Search results contain only questions.

language
Type: String

The language of the articles or questions. The value must be a Salesforce supported locale code. In an `<apex:page>`, the default value is the language of the page. Otherwise, the default value is the user's locale.

result
Type: `ConnectApi.ZoneSearchPage`

The object containing test data.

Return Value

Type: Void

SEE ALSO:

searchInZone(communityId, zoneId, q, filter, language)

Apex Developer Guide: Testing ConnectApi Code

`ConnectApi` Input Classes

Some `ConnectApi` methods take arguments that are instances of `ConnectApi` input classes.

Input classes are concrete unless marked abstract in this documentation. Concrete input classes have public constructors that have no parameters.

Some methods have parameters that are typed with an abstract class. You must pass in an instance of a concrete child class for these parameters.

Most input class properties can be set. Read-only properties are noted in this documentation.

ConnectApi.AbstractCheckoutAddressInput

A checkout address.

This class is abstract.

Superclass of:

- `ConnectApi.CartShippingAddressInput`

Property	Type	Description	Required or Optional	Available Version
`city`	String	City of the address.	Optional	53.0
`companyName`	String	Company name of the address.	Optional	59.0
`country`	String	ISO code of the address country. Must match one of the valid ISO codes defined within the org's State-Country picklist.	Required	53.0
`firstName`	String	First name of the contact.	Optional	57.0
`id`	String	ID of the address.	Required	53.0
`lastName`	String	Last name of the contact.	Optional	57.0
`name`	String	Name of the contact.	Required	53.0
`postalCode`	String	ZIP code of the address.	Optional	53.0
`region`	String	ISO code of the address region. Must match one of the valid ISO codes defined within the org's State-Country picklist.	Optional	53.0
`shipToPhoneNumber`	String	Phone number of the contact.	Optional	63.0
`street`	String	Street of the address.	Required	53.0

ConnectApi.AbstractList

Primitive list input.

This class is abstract.

Superclass of:

- ConnectApi.BooleanList
- ConnectApi.DoubleList
- ConnectApi.LongList
- ConnectApi.StringList

No additional properties.

SEE ALSO:

ConnectApi.SearchFilter

ConnectApi.ActionLinkDefinitionInput

The definition of an action link. An action link is a button on a feed element. Clicking an action link can take a user to a Web page, initiate a file download, or invoke an API call to Salesforce or to an external server. An action link includes a URL and an HTTP method, and can include a request body and header information, such as an OAuth token for authentication. Use action links to integrate Salesforce and third-party services into the feed so that users can drive productivity and accelerate innovation.

Usage

You can use context variables in the `actionUrl`, `headers`, and `requestBody` properties. Use context variables to pass information about the user who executed the action link to your server-side code. Salesforce substitutes the value when the action link is executed.

The available context variables are:

Context Variable	Description
`{!actionLinkId}`	The ID of the action link the user executed.
`{!actionLinkGroupId}`	The ID of the action link group containing the action link the user executed.
`{!communityId}`	The ID of the site in which the user executed the action link. The value for your internal org is the empty key `"000000000000000000"`.
`{!communityUrl}`	The URL of the site in which the user executed the action link. The value for your internal org is empty string `""`.
`{!orgId}`	The ID of the org in which the user executed the action link.
`{!userId}`	The ID of the user that executed the action link.

Property	Type	Description	Required or Optional	Available Version
`actionType`	`ConnectApi.ActionLinkType`	Defines the type of action link. Values are: • `Api`—The action link calls a synchronous API at the action URL. Salesforce sets the status to `SuccessfulStatus` or `FailedStatus` based on the HTTP status code returned by your server. • `ApiAsync`—The action link calls an asynchronous API at the action URL. The action remains in a `PendingStatus`	Required Can be defined in an action link template.	33.0

Property	Type	Description	Required or Optional	Available Version
		state until a third party makes a request to `/connect/action-links/actionLinkId` to set the status to `SuccessfulStatus` or `FailedStatus` when the asynchronous operation is complete. • `Download`—The action link downloads a file from the action URL. • `Ui`—The action link takes the user to a web page at the action URL. Use `Ui` if you need to load a page before the user performs an action, for example, to have the user provide input or view something before the action happens. Note: Invoking `ApiAsync` action links from an app requires a call to set the status. However, there isn't currently a way to set the status of an action link using Apex. To set the status, use Connect REST API. See the Action Link resource in the Connect REST API Developer Guidefor more information.		
`actionUrl`	String	The action link URL. For example, a `Ui` action link URL is a Web page. A `Download` action link URL is a link to the file to download. `Ui` and `Download` action link URLs are provided to clients. An `Api` or `ApiAsync` action link URL is a REST resource. `Api` and `ApiAsync` action link URLs aren't provided to clients. Links to Salesforce can be relative. All other links must be absolute and start with `https://`. Tip: To avoid issues due to upgrades or changing functionality in your API, we recommend using a versioned API for `actionUrl`, for example, `https://www.example.com/api/v1/exampleResource`. If your API isn't versioned, you can use the `expirationDate`	Required Can be defined in an action link template.	33.0

Property	Type	Description	Required or Optional	Available Version
		property of the `ConnectApi.ActionLinkGroupDefinitionInput` class to avoid issues due to upgrades or changing functionality in your API.		
`excludedUserId`	String	ID of a single user to exclude from performing the action. If you specify an `excludedUserId`, you can't specify a `userId`.	Optional Can be defined in an action link template using the `User Visibility` and `Custom User Alias` fields.	33.0
`groupDefault`	Boolean	`true` if this action is the default action link in the action link group; `false` otherwise. There can be only one default action link per action link group. The default action link gets distinct styling in the Salesforce UI.	Optional Can be defined in an action link template.	33.0
`headers`	List<`ConnectApi.RequestHeaderInput`>	The request headers for the `Api` and `ApiAsync` action link types. See Action Links Overview, Authentication, and Security.	Optional Can be defined in an action link template.	33.0
`labelKey`	String	Key for the set of labels to show in the user interface. A set includes labels for these states: NewStatus, PendingStatus, SuccessStatus, FailedStatus. For example, if you use the `Approve` key, you get these labels: Approve, Pending, Approved, Failed. For a complete list of keys and labels, see Action Links Labels. If none of the predefined labels work for your action link, use a custom label. To use a custom label, create an action link template. See Create Action Link Templates.	Required Can be defined in an action link template.	33.0
`method`	`ConnectApi.HttpRequestMethod`	One of these HTTP methods: • `HttpDelete`—Returns HTTP 204 on success. Response body or output class is empty. • `HttpGet`—Returns HTTP 200 on success.	Required Can be defined in an action link template.	33.0

Property	Type	Description	Required or Optional	Available Version
		• `HttpHead`—Returns HTTP 200 on success. Response body or output class is empty. • `HttpPatch`—Returns HTTP 200 on success or HTTP 204 if the response body or output class is empty. • `HttpPost`—Returns HTTP 201 on success or HTTP 204 if the response body or output class is empty. Exceptions are the batch posting resources and methods, which return HTTP 200 on success. • `HttpPut`—Return HTTP 200 on success or HTTP 204 if the response body or output class is empty.		
`requestBody`	String	The request body for `Api` action links. Note: Escape quotation mark characters in the `requestBody` value.	Optional Can be defined in an action link template.	33.0
`requires Confirmation`	Boolean	`true` to require the user to confirm the action; `false` otherwise.	Required Can be defined in an action link template.	33.0
`userId`	String	The ID of the user who can execute the action. If not specified or `null`, any user can execute the action. If you specify a `userId`, you can't specify an `excludedUserId`.	Optional Can be defined in an action link template using the `User Visibility` and `Custom User Alias` fields.	33.0

SEE ALSO:

ConnectApi.ActionLinkGroupDefinitionInput

ConnectApi.ActionLinkGroupDefinitionInput

The definition of an action link group. All action links must belong to a group. Action links in a group are mutually exclusive and share some properties. Define standalone actions in their own action group.

Action link definition can be sensitive to a third party (for example, OAuth bearer token headers). For this reason, only calls made from the Apex namespace that created the action link definition can read, modify, or delete the definition. In addition, the user making the call must have created the definition or have View All Data permission.

Property	Type	Description	Required or Optional	Available Version
`actionLinks`	`List<ConnectApi.ActionLinkDefinitionInput>`	The action links that make up this group. Within an action link group, action links are displayed in the order listed in the `actionLinks` property of the `ConnectApi.ActionLinkGroupDefinitionInput` class. Within a feed item, action link groups are displayed in the order specified in the `actionLinkGroupIds` property of the `ConnectApi.AssociatedActionsCapabilityInput` class. You can create up to three action links in a `Primary` group and up to four in an `Overflow` group.	Required to instantiate this action link group without a template. To instantiate from a template, don't specify a value.	33.0
`category`	`ConnectApi.PlatformActionGroupCategory`	Indicates the priority and relative locations of action links in an associated feed item. Values are: • `Primary`—The action link group is displayed in the body of the feed element. • `Overflow`—The action link group is displayed in the overflow menu of the feed element.	Required to instantiate this action link group without a template. To instantiate from a template, don't specify a value.	33.0
`executionsAllowed`	`ConnectApi.ActionLinkExecutionsAllowed`	Defines the number of times an action link can be executed. Values are: • `Once`—An action link can be executed only one time across all users. • `OncePerUser`—An action link can be executed only one time for each user. • `Unlimited`—An action link can be executed an unlimited number of times by each user. If the action link's `actionType` is `Api` or `ApiAsync`, you can't use this value.	Required to instantiate this action link group without a template. To instantiate from a template, don't specify a value.	33.0
`expirationDate`	`Datetime`	ISO 8601 date string, for example, 2011-02-25T18:24:31.000Z, that represents	Required to instantiate this action link group without a template.	33.0

Property	Type	Description	Required or Optional	Available Version
		the date and time this action link group is removed from associated feed items and can no longer be executed. The `expirationDate` must be within one year of the creation date. If the action link group definition includes an OAuth token, it is a good idea to set the expiration date of the action link group to the same value as the expiration date of the OAuth token so that users can't execute the action link and get an OAuth error. To set a date when instantiating from a template, see Set the Action Link Group Expiration Time in Design Action Link Templates.	Optional to instantiate from a template.	
`templateBindings`	[illegible]	A collection of key-value pairs to fill in binding variable values or a custom user alias from an action link template. To instantiate this action link group from an action link template that uses binding variables, you must provide values for all the variables. See Define Binding Variables in Design Action Link Templates.	To instantiate without a template, don't specify a value. Required to instantiate this action link group from a template that uses binding variables.	33.0
`templateId`	`String`	The ID of the action link group template from which to instantiate this action link group.	To instantiate without a template, don't specify a value. Required to instantiate this action link group from a template.	33.0

SEE ALSO:

- Define an Action Link and Post with a Feed Element
- Define an Action Link in a Template and Post with a Feed Element
- createActionLinkGroupDefinition(communityId, actionLinkGroup)

ConnectApi.ActionLinkTemplateBindingInput

A key-value pair to fill in a binding variable value from an action link template.

Property	Type	Description	Required or Optional	Available Version
`key`	String	The name of the binding variable key specified in the action link template in Setup. For example, if the binding variable in the template is `{!Binding.firstName}`, the key is `firstName`	Required	33.0
`value`	String	The value of the binding variable key. For example, if the key is `firstName`, this value could be `Joan`.	Required	33.0

SEE ALSO:

ConnectApi.ActionLinkGroupDefinitionInput

ConnectApi.ActivitySharingInput

Defines who a captured email or event is shared with.

Property	Type	Description	Required or Optional	Available Version
`groupsToShareWith`	List<`String`>	List of IDs for the groups that you share the activity with. Valid only if `sharingType` is `MyGroups`.	Optional	39.0
`sharingType`	`ConnectApi.ActivitySharingType`	Type of sharing operation. Values are: • `Everyone`—The activity is shared with everyone. • `MyGroups`—The activity is shared only with a selection of the context user's groups. • `OnlyMe`—The activity is private.	Required	39.0

ConnectApi.AddressRequest

Address input representation for a payment method or card payment method.

Name	Type	Description	Required or Optional	Available Version
`city`	String	Payment method city.	Optional	51.0
`companyName`	String	Payment method company name.	Optional	51.0
`country`	String	Payment method country.	Optional	51.0

Name	Type	Description	Required or Optional	Available Version
`postalCode`	String	Payment method postal code.	Optional	51.0
`state`	String	Payment method state.	Optional	51.0
`street`	String	Payment method street.	Optional	51.0

ConnectApi.AdjustItemInputRepresentation

A price adjustment to an OrderItemSummary. It only supports discounts, not increases.

Property	Type	Description	Required or Optional	Available Version
`adjustmentType`	String	Describes how the amount is calculated. It can have one of these values: • `AmountWithTax`—Value of amount is the adjustment, including tax. • `AmountWithoutTax`—Value of amount is the adjustment, not including tax. Tax is calculated on the value and added. • `Percentage`—Value of amount is a percentage discount. It is divided by 100, and then multiplied by the TotalPrice and TotalTaxAmount of the OrderItemSummary to determine the adjustment amount.	Required	49.0
`amount`	Double	Value used to calculate the adjustment amount, as described by the adjustmentType. It must be a negative value.	Required	49.0
`description`	String	Description of the adjustment.	Optional	49.0
`orderItemSummaryId`	String	ID of the OrderItemSummary.	Required	49.0
`reason`	String	Reason for the adjustment. The value must match one of the picklist values on the Reason field of the OrderItemSummaryChange object.	Required	49.0

ConnectApi.AdjustOrderItemSummaryInputRepresentation

Price adjustments to order item summaries that together make up a price adjustment to an order, with options for adjusting items in the process of being fulfilled.

Property	Type	Description	Required or Optional	Available Version
`adjustItems`	List<`ConnectApi.AdjustItemInputRepresentation`>	List of price adjustments to order item summaries.	Required	49.0
`allocatedItemsChangeOrderType`	String	Process to use for order item summary quantities that are currently being fulfilled, defined as `QuantityAllocated - QuantityFulfilled`. Values are: • `Disallowed`—When distributing the adjustment, ignore any quantities being fulfilled. If an order item summary's entire quantity is being fulfilled, return an error. This is the default value. • `InFulfillment`—When distributing the adjustment, include quantities being fulfilled. Create a separate change order for the adjustments made to those quantities. • `PreFulfillment`—When distributing the adjustment, include quantities being fulfilled. Include the adjustments made to those quantities in the change order for pre-fulfillment quantity adjustments.	Optional	55.0
individualItemTaxAdjustments	Boolean	Specifies whether to create multiple lines for each tax adjustment or one line with all tax adjustments. The default value is false.	Optional	59.0

ConnectApi.AlternativeInput

Alternative representation for an extension on a feed element.

Property	Type	Description	Required or Optional	Available Version
`textRepresentation`	String	Text representation of the extension.	Required	40.0
`thumbnailUrl`	String	Thumbnail URL to the extension.	Optional	40.0
`title`	String	Title of the extension.	Optional	40.0

ConnectApi.AlternativePaymentMethod

A payment method that doesn't have a defined Salesforce entity such as CardPaymentMethod or DigitalWallet. Common examples of alternative payment methods include CashOnDeliver, Klarna, and Direct Debit. AlternativePaymentMethod functions the same as any other type of payment method for processing transactions in the payment gateway.

Subclass of ConnectApi.BasePaymentMethodRequest

Property	Type	Description	Required or Optional	Available Version
`accountId`	String	Salesforce Payments account to which this payment method is linked.	Required	54.0
`comments`	String	Details about a record added by a user. Maximum of 1,000 characters.	Optional	54.0
`email`	String	Email address of the card holder.	Optional	54.0
`gatewayToken`	String	A unique, alphanumeric ID, called a token, that a payment gateway generates when it first processes a payment. The token replaces the actual payment data so that the data is kept secure. This token is stored as encrypted text, and can be used for recurring payments.	Required	54.0
`gatewayTokenDetails`	String	Detailed information about the gateway token.	Required	54.0
`name`	String	Name that you assign to the payment method object.	Optional	54.0

ConnectApi.AnnouncementInput

An announcement.

Property	Type	Description	Required or Optional	Available
`body`	ConnectApi.MessageBodyInput	Text of the announcement.	Required for creating an announcement if `feedItemId` isn't specified Don't specify for updating an announcement.	31.0

Property	Type	Description	Required or Optional	Available
`expirationDate`	Datetime	The Salesforce UI displays an announcement until 11:59 p.m. on this date unless another announcement is posted first. The Salesforce UI ignores the time value in the `expirationDate`. However, you can use the time value to create your own display logic in your own UI.	Required for creating an announcement Optional for updating an announcement	31.0
`feedItemId`	String	ID of an `AdvancedTextPost` feed item that is the body of the announcement.	Required for creating an announcement if `body` isn't specified Don't specify for updating an announcement.	36.0
`isArchived`	Boolean	Specifies whether the announcement is archived.	Optional	36.0
`parentId`	String	ID of the parent entity for the announcement, that is, a group ID when the announcement appears in a group.	Required for creating an announcement if `feedItemId` isn't specified Don't specify for updating an announcement.	36.0
`sendEmails`	Boolean	Specifies whether the announcement is sent as an email to all group members regardless of their email setting for the group. If Chatter emails aren't enabled for the organization, announcement emails aren't sent. Default value is `false`.	Optional for creating an announcement Don't specify for updating an announcement	36.0

SEE ALSO:

postAnnouncement(communityId, groupId, announcement)

postAnnouncement(communityId, announcement)

ConnectApi.ArticleTopicAssignmentJobInput

An article and topic assignment job.

Property	Type	Description	Required or Optional	Available Version
operation	ConnectApi.ArticleTopicJobType	Type of operation to perform on articles and topics. Values are: • AssignTopicsToArticle—Assign topics to articles in a data category. • UnassignTopicsFromArticle—Unassign topics from articles in a data category.	Required	40.0
topicNames	ConnectApi.TopicNamesInput	List of topic names to assign to or unassign from articles.	Required	40.0

ConnectApi.AssociatedActionsCapabilityInput

A list of action link groups to associate with a feed element. To associate an action link group with a feed element, the call must be made from the Apex namespace that created the action link definition. In addition, the user making the call must have created the definition or have View All Data permission.

An action link is a button on a feed element. Clicking an action link can take a user to a Web page, initiate a file download, or invoke an API call to Salesforce or to an external server. An action link includes a URL and an HTTP method, and can include a request body and header information, such as an OAuth token for authentication. Use action links to integrate Salesforce and third-party services into the feed so that users can drive productivity and accelerate innovation.

Property	Type	Description	Required or Optional	Available Version
actionLinkGroupIds	List<String>	The action link group IDs to associate with the feed element. Associate one Primary and up to 10 total action link groups to a feed item. Action link groups are returned in the order specified in this property. An action link group ID is returned from a call to ConnectApi.ActionLinks.createActionLinkGroupDefinition(communityId, actionLinkGroup).	Required	33.0

SEE ALSO:

ConnectApi.FeedElementCapabilitiesInput

ConnectApi.AssociateRecordsWithRecipientInput

Records associated with the survey invitation.

Property	Type	Description	Required or Optional	Available Version
`associateRecordIds`	List<String>	ID of the associated records.	Required	50.0
`recipientId`	String	Participant ID with whose invitation the record should be associated.	Required	50.0

SEE ALSO:

ConnectApi.SurveyInvitationEmailInput

ConnectApi.AudienceCriteriaInput

Custom recommendation audience criteria type.

This class is abstract and has no public constructor. You can make an instance only of a subclass.

Superclass for:

- ConnectApi.CustomListAudienceCriteriaInput
- ConnectApi.NewUserAudienceCriteriaInput

Property	Type	Description	Required or Optional	Available Version
`type`	`ConnectApi.RecommendationAudienceCriteriaType`	Specifies the custom recommendation audience criteria type. One of these values: • `CustomList`—A custom list of users makes up the audience. • `MaxDaysInCommunity`—New members make up the audience.	Optional If not specified, defaults to `CustomList`.	36.0

SEE ALSO:

ConnectApi.RecommendationAudienceInput

ConnectApi.AudienceCriterionInput

Personalization audience criterion.

Property	Type	Description	Required or Optional	Available Version
`criterion`	List<`ConnectApi.AudienceCriterionValueInput`>	List of mappings of audience criteria fields and values.	Required when creating an audience Optional when updating an audience	48.0

Property	Type	Description	Required or Optional	Available Version
`criterionNumber`	Integer	Number associated with the audience criterion in a formula. For example, (1 AND 2) OR 3. If unspecified, criteria are assigned numbers in the order that they're added.	Optional	48.0
`criterionOperator`	`ConnectApi.AudienceCriteriaOperator`	Operator used in the personalization audience criterion. Values are: • `Contains` • `Equal` • `GreaterThan` • `GreaterThanOrEqual` • `Includes` • `LessThan` • `LessThanOrEqual` • `NotEqual` • `NotIncludes` • `StartsWith`	Required when creating an audience Optional when updating an audience	48.0
`criterionType`	`ConnectApi.AudienceCriteriaType`	Type of personalization audience criterion. Values are: • `Audience`—Criterion based on audience. • `Default`—Audience has no criteria. • `Domain`—Criterion based on domain. • `FieldBased`—Criterion based on object fields. • `GeoLocation`—Criterion based on location. • `Permission`—Criterion based on standard or custom permissions. • `Profile`—Criterion based on profile.	Required when creating an audience Optional when updating an audience	48.0

SEE ALSO:

ConnectApi.AudienceInput

ConnectApi.AudienceCriterionValueInput

Audience criterion value.

Property	Type	Description	Required or Optional	Available Version
`audienceId`	String	ID of an audience.	Required if creating or updating an audience with the `Audience` criterion type.	53.0
`city`	String	City of a user.	Optional if creating or updating an audience with the `GeoLocation` criterion type	48.0
`country`	String	Country of a user.	Required if creating or updating an audience with the `GeoLocation` criterion type	48.0
`domainId`	String	Domain ID of a user.	Required if creating or updating an audience with the `Domain` criterion type	48.0
`entityField`	String	Field of an object.	Required if creating or updating an audience with the `FieldBased` criterion type	48.0
`entityType`	String	Type of object.	Required if creating or updating an audience with the `FieldBased` criterion type	48.0
`fieldValue`	String	Value of a field.	Required if creating or updating an audience with the `FieldBased` criterion type	48.0
`isEnabled`	Boolean	Specifies whether the permission is enabled (`true`) or not (`false`) for a user.	Required if creating or updating an audience with the `Permission` criterion type	48.0
`permission`	String	Valid API name of a standard user or custom permission.	Required if creating or updating an	48.0

Property	Type	Description	Required or Optional	Available Version
			audience with the `Permission` criterion type	
`profileId`	String	Profile ID of a user.	Required if creating or updating an audience with the `Profile` criterion type	48.0
`subdivision`	String	Subdivision of a user.	Required if creating or updating an audience with the `GeoLocation` criterion type and using the `city` property	48.0

SEE ALSO:

ConnectApi.AudienceCriterionInput

ConnectApi.AudienceInput

A personalization audience.

Property	Type	Description	Required or Optional	Available Version
`criteria`	List<`ConnectApi.AudienceCriterionInput`>	List of audience criteria to update or add. An audience can have up to 100 criteria.	Required when creating an audience Optional when updating an audience	48.0
`customFormula`	String	Custom formula for the audience criteria. For example, (1 AND 2) OR 3.	Required when creating an audience with the `formulaFilterType` set to `CustomLogicMatches` Optional, otherwise	48.0

Property	Type	Description	Required or Optional	Available Version
formulaFilterType	ConnectApi.FormulaFilterType	Formula filter type for the personalization audience. Values are: • AllCriteriaMatch—All audience criteria are true (AND operation). • AnyCriterionMatches—Any audience criterion is true (OR operation). • CustomLogicMatches—Audience criteria match the custom formula (for example, (1 AND 2) OR 3).	Required when creating an audience Optional when updating an audience	48.0
name	String	Name of the audience.	Required when creating an audience Optional when updating an audience	48.0

ConnectApi.AuditParamsRequest

Audit Parameters input.

This class is abstract.

Superclass of ConnectApi.BaseRequest.

Property	Type	Description	Required or Optional	Available Version
email	String	Email of the client that made the request.	Optional	50.0
ipAddress	String	IP address of the client that made the request.	Optional	50.0
macAddress	String	Mac address of the client that made the request.	Optional	50.0
phone	String	Phone number of the client that made the request.	Optional	50.0

ConnectApi.AuthApiPaymentMethodRequest

Payment method input representation for payment authorizations.

Subclass of ConnectApi.BaseApiPaymentMethodRequest.

Property	Type	Description	Required or Optional	Available Version
`cardPaymentMethod`	`ConnectApi.CardPaymentMethodRequest`	Card payment method information.	Required	51.0

ConnectApi.AuthorizationReversalRequest

Authorization reversal input consumed by authorization reversal service.

Subclass of ConnectApi.BaseRequest.

Property	Type	Description	Required or Optional	Available Version
`accountId`	String	Account for the payment authorization reversal. Must match the payment authorization's account.		51.0
`amount`	Double	Amount of adjustment applied to the payment authorization.		51.0
`comments`	String	Users can add comments to provide additional details about a record. Maximum of 1,000 characters.		51.0
`effectiveDate`	Datetime	Date that the adjustment takes effect on the authorization.		51.0

ConnectApi.AuthorizationRequest

Payment Authorization input consumed by Payment Authorization service.

Subclass of ConnectApi.BaseRequest.

Property	Type	Description	Required or Optional	Available Version
`accountId`	String	Salesforce account that contains the payment transaction being authorized.	Required	51.0
`amount`	Double	Authorization amount.	Required	51.0
`comments`	String	Optional comments for the payment authorization.	Optional	51.0
`currencyIsoCode`	String	Three-letter ISO 4217 currency code associated with the payment group record.	Required	51.0
`effectiveDate`	Datetime	Date that the authorization will be applied to the transaction.	Required	51.0
`paymentGatewayId`	String	Payment gateway that processes the authorization.	Required	51.0

Property	Type	Description	Required or Optional	Available Version
`paymentGroup`	`ConnectApi.PaymentGroupRequest`	Payment group for the authorization. The payload must reference either a `paymentGroup` or a `paymentGroupId`, but not both.	Optional	51.0
`paymentMethod`	`ConnectApi.AuthApiPaymentMethodRequest`	Payment method used in the payment gateway for the authorization transaction.	Required	51.0

ConnectApi.BannerPhotoInput

A banner photo.

Property	Type	Description	Required or Optional	Available Version
`cropHeight`	Integer	Height of the crop rectangle in pixels.	Optional	36.0
`cropWidth`	Integer	Width of the crop rectangle in pixels.	Optional	36.0
`cropX`	Integer	X position of the crop rectangle from the left edge of the image in pixels. Top left is position (0,0).	Optional	36.0
`cropY`	Integer	Y position of the crop rectangle from the top edge of the image in pixels. Top left is position (0,0).	Optional	36.0
`fileId`	String	18 character ID of an existing file. The key prefix must be 069 and the file must be an image and be smaller than 2 GB. Note: Images uploaded on the Group page and on the User page don't have file IDs and therefore can't be used.	Required	36.0
`versionNumber`	Integer	Version number of an existing file. If not provided, the latest version is used.	Optional	36.0

ConnectApi.BaseApiPaymentMethodRequest

Payment method API input representation.

This class is abstract.

Superclass of:

- ConnectApi.AuthApiPaymentMethodRequest
- ConnectApi.PostAuthApiPaymentMethodRequest

- ConnectApi.SaleApiPaymentMethodRequest

Property	Type	Description	Required or Optional	Available Version
address	ConnectApi.AddressRequest	Payment method address.	Required	51.0
id	String	Payment method record ID. Used in payment transactions.	Required	51.0
saveForFuture	Boolean	Shows whether Salesforce saves the payment method for future use.	Required	51.0

ConnectApi.BasePaymentMethodRequest

Base payment method input representation.

This class is abstract.

Superclass of:

- ConnectApi.AlternativePaymentMethod
- ConnectApi.CardPaymentMethodRequest

No additional properties.

ConnectApi.BaseRequest

Base parameters for making a request to the payment gateway.

This class is abstract.

Subclass of ConnectApi.AuditParamsRequest.

Superclass of:

- ConnectApi.AuthorizationRequest
- ConnectApi.AuthorizationReversalRequest
- ConnectApi.CaptureRequest
- ConnectApi.PaymentMethodTokenizationRequest
- ConnectApi.PostAuthRequest
- ConnectApi.RefundRequest
- ConnectApi.SaleRequest

Property	Type	Description	Required or Optional	Available Version
additionalData	Map<String, String>	An optional map of additional parameters to be sent to the payment gateway.	Optional	50.0
idempotencyKey	String	Idempotency key.	Optional	50.0

ConnectApi.BatchInput

Construct a set of inputs to be passed into a method at the same time.

Use this constructor when there isn't a binary input:

```
ConnectApi.BatchInput(Object input)
```

Use this constructor to pass one binary input:

```
ConnectApi.BatchInput(Object input, ConnectApi.BinaryInput binary)
```

Use this constructor to pass multiple binary inputs:

```
ConnectApi.BatchInput(Object input, List<ConnectApi.BinaryInput> binaries)
```

The constructors takes these parameters:

Argument	Type	Description	Available Version
`input`	Object	An individual input object to be used in the batch operation. For example, for `postFeedElementBatch()`, this should be `ConnectApi.FeedElementInput`.	32.0
`binary`	`ConnectApi.BinaryInput`	A binary file to associate with the input object.	32.0
`binaries`	List<`ConnectApi.BinaryInput`>	A list of binary files to associate with the input object.	32.0

SEE ALSO:

- Post a Batch of Feed Elements
- Post a Batch of Feed Elements with a New (Binary) File

ConnectApi.BinaryInput

Create a `ConnectApi.BinaryInput` object to attach files to feed items and comments, to add repository files, to create managed content, and to replace managed content variants.

The constructor is:

```
ConnectApi.BinaryInput(blob, contentType, filename)
```

The constructor takes these arguments:

Argument	Type	Description	Available Version
`blob`	Blob	Contents of the file to be used for input	28.0
`contentType`	String	MIME type description of the content, such as `image/jpg`	28.0

Argument	Type	Description	Available Version
`filename`	String	File name with the file extension, such as UserPhoto.jpg	28.0

SEE ALSO:

- Post a Feed Element with a New File (Binary) Attachment
- Post a Comment with a New File
- ConnectApi.BatchInput

ConnectApi.BookmarksCapabilityInput

Create or update a bookmark on a feed element.

This class is a subclass of ConnectApi.FeedElementCapabilityInput.

Property	Type	Description	Required or Optional	Available Version
`isBookmarkedByCurrentUser`	Boolean	Specifies if the feed element should be bookmarked for the user (`true`) or not (`false`).	No	32.0

SEE ALSO:

- ConnectApi.FeedElementCapabilitiesInput

ConnectApi.BooleanList

List of Boolean values.

Subclass of ConnectApi.AbstractList.

Property	Type	Description	Required or Optional	Available Version
`values`	List<Boolean>	List of Boolean values to filter on, for example, `[true, false]`.	Optional	63.0

ConnectApi.BotVersionActivationInput

Activation status of the bot version.

Property	Type	Description	Required or Optional	Available Version
`status`	`ConnectApi.BotVersionActivationStatus`	Activation status of the bot version. Values are: • `Active`	Optional	50.0

Property	Type	Description	Required or Optional	Available Version
		• `Inactive` Activation status must be specified in the *`status`* or *`postBody`* parameter.		

ConnectApi.CalculateCartInput

Custom fields for a cart calculation.

Property	Type	Description	Required or Optional	Available Version
`customFields`	List<`SObject`>	Array of sObjects and custom fields for the sObjects. Standard fields are ignored. The custom fields must already be defined for the sObject. Currently, the WebCart, CartItem, and CartDeliveryGroup sObjects are supported. Field-level security rules from the shopper profile are applied to the custom fields. The rules are applied for registered shoppers and for the guest shopper profile. See Create a Cart and Cart Item with Custom Fields in a Commerce Store.	Optional	63.0

ConnectApi.CalculateTaxRequest

Request to sent through the tax adapter to the external tax engine. Inputs with a `TaxTransactionType` of Debit represent a tax calculation request. Inputs with a `TaxTransactionType` of Credit represent a tax cancellation request.

Subclass of ConnectApi.TaxTransactionRequest.

Property	Type	Description	Required or Optional	Available Version
`isCommit`	Boolean	Commits the transaction for tax calculation.	Required	55.0
`taxEngineId`	String	ID of the Salesforce tax engine entity used to represent the external tax engine.	Required	55.0
`taxTransactionType`	`ConnectApi.TaxTransactionType`	Type of tax transaction. Values are: • `Credit`—Transaction is a credit transaction. • `Debit`—Transaction is a debit transaction.	Required	55.0

Property	Type	Description	Required or Optional	Available Version
`taxType`	`ConnectApi.CalculateTaxType`	Type of tax calculation. Values are: • `Actual`—Calculated tax represents the final taxed amount for the transaction. • `Estimated`—Calculated tax represents only an estimated value before the transaction is finalized.	Required	55.0

ConnectApi.CancelAllOrderItemsInputRepresentation

Cancellation of all items in an order.

Property	Type	Description	Required or Optional	Available Version
`changeItemFees`	<redacted>	List of input data for fees, including taxes, associated with the order items being canceled.	Optional	63.0
`excludedItems`	List<String>	List of items excluded from cancellation.	Optional	63.0
`orderSummaryId`	String	ID of the order summary.	Required	63.0
`reason`	String	Reason for the cancellation. The value must match one of the picklist values on the Reason field of the Order Product Summary Change object.	Required	63.0
`reasonText`	String	Reason text used for the return insights. The value has a max of 255 characters.	Optional	63.0

ConnectApi.CanvasCapabilityInput

Create or update a canvas app associated with a feed element.

This class is a subclass of ConnectApi.FeedElementCapabilityInput.

Property	Type	Description	Required or Optional	Available Version
`description`	String	A description of the canvas app. The maximum size is 255 characters.	Optional	32.0
`developerName`	String	The API name (developer name) of the connected app.	Required	32.0
`height`	String	The height of the canvas app in pixels.	Optional	32.0

Property	Type	Description	Required or Optional	Available Version
`namespacePrefix`	String	A unique namespace prefix for the canvas app.	Optional	32.0
`parameters`	String	JSON parameters passed to the canvas app.	Optional	32.0
`thumbnailUrl`	String	A thumbnail URL to a preview image. The maximum thumbnail size is 120 pixels by 120 pixels.	Optional	32.0
`title`	String	A title for the canvas link.	Required	32.0

SEE ALSO:

ConnectApi.FeedElementCapabilitiesInput

ConnectApi.CapacityRequestInputRepresentation

Request related to a location's fulfillment order capacity.

Property	Type	Description	Required or Optional	Available Version
`actionRequestId`	String	Unique string that identifies the request. Can be a UUID. Use the action request IDs in response data to identify which requests succeeded or failed.	Required	55.0
`locationId`	String	ID of the location associated with the request.	Required	55.0

ConnectApi.CaptureRequest

Payment capture input consumed by the payment capture service.

Subclass of ConnectApi.BaseRequest.

Property	Type	Description	Required or Optional	Available Version
`accountId`	String	ID of the account linked to the capture request.	Optional	50.0
`amount`	Double	Amount captured from the previous authorization.	Required	50.0
`clientContext`	String	Context for payment APIs. Used for a payment caller to re-establish context.	Optional	50.0
`comments`	String	Comments for the payment capture.	Optional	50.0

Property	Type	Description	Required or Optional	Available Version
`effectiveDate`	Datetime	Date when the payment becomes effective.	Optional	50.0
`paymentGroup`	`ConnectApi.PaymentGroupRequest`	Details about the payment group record associated with the payment request.	Optional	50.0

ConnectApi.CardPaymentMethodRequest

Card payment method input representation.

Subclass of ConnectApi.BasePaymentMethodRequest.

Property	Type	Description	Required or Optional	Available Version
`accountId`	String	Salesforce Payments account to which this payment method is linked.	Required	51.0
`autoPay`	Boolean	Indicates whether a token for recurring payments is being requested (`true`) or not (`false`). The token lets the payment method be used for recurring payments.	Optional	55.0
`cardCategory`	`ConnectApi.CardCategory`	Card processing type. Valid values are: • `CreditCard` • `DebitCard`	Required	51.0
`cardHolderFirstName`	String	First name of the card holder.	Required	51.0
`cardHolderLastName`	String	Last name of the card holder.	Required	51.0
`cardHolderName`	String	Full name of the card holder.	Required	51.0
`cardNumber`	String	Card number.	Required	51.0
`cardType`	String	Card network type. Valid values are: • `AmericanExpress` • `DinersClub` • `JCB` • `MasterCard` • `Maestro` • `Visa`	Required	51.0
`comments`	String	Optional comments for the card payment method.	Optional	51.0
`cvv`	String	Card Verification Value.	Optional	51.0

Property	Type	Description	Required or Optional	Available Version
email	String	Email address of the card holder.	Required	51.0
expiryMonth	Integer	Card expiration month.	Required	51.0
expiryYear	Integer	Card expiration year.	Required	51.0
nickName	String	Optional nickname for the card.	Optional	51.0
startMonth	Integer	Month the card becomes active.	Optional	51.0
startYear	Integer	Year the card becomes active.	Optional	51.0

ConnectApi.PromotionCartAdjustmentGroupInput

Cart adjustment group for a promotion.

Property	Type	Description	Required or Optional	Available Version
adjustmentBasisReference	String	ID of the associated coupon, if applicable.	Optional	60.0
adjustmentDescription	String	Description of the price adjustment.	Optional	60.0
adjustmentType	ConnectApi.AdjustmentType	Type of price adjustment. Valid values are: • AdjustmentAmount—The adjustment is a fixed amount. • AdjustmentPercentage—The adjustment is a percentage.	Required	60.0
adjustmentValue	String	Price value of the adjustment.	Optional	60.0
baseAmount	String	Total amount of the adjustment.	Optional	60.0
cartId	String	ID of the cart.	Required	60.0
id	String	ID of the cart adjustment group.	Required	60.0
priceAdjustmentCauseId	String	ID of the related promotion.	Optional	60.0
priority	Integer	Where in the sequence of adjustments this adjustment was applied.	Optional	60.0

ConnectApi.cartCouponInput

Cart coupon input.

Property	Type	Description	Required or Optional	Available Version
couponCode	String	The coupon code.	Required	54.0

ConnectApi.PromotionCartDeliveryGroupInput

Cart delivery group input.

Property	Type	Description	Required or Optional	Available Version
cartDeliveryGroupId	String	ID of the cart delivery group.	Required	57.0
cartDeliveryGroupMethods	List<ConnectApi.PromotionCartDeliveryGroupInput on page 1821>	List of cart delivery group methods.	Optional	60.0
deliveryMethodId	String	ID of the order delivery method.	Optional	57.0—59.0

ConnectApi.CartEvaluateShippingInput

Shipping address and custom fields used to calculate shipping costs for a cart.

Property	Type	Description	Required or Optional	Available Version
customFields	List<SObject>	Array of sObjects and custom fields for the sObjects. Standard fields are ignored. The custom fields must already be defined for the sObject. Currently, the WebCart, CartItem, and CartDeliveryGroup sObjects are supported. Field-level security rules from the shopper profile are applied to the custom fields. The rules are applied for registered shoppers and for the guest shopper profile. See Create a Cart and Cart Item with Custom Fields in a Commerce Store.	Optional	63.0
shippingAddress	[illegible]	Shipping address for a cart.	Required	63.0

ConnectApi.CartEvaluateTaxInput

Shipping address and custom fields used to calculate taxes for a cart.

Property	Type	Description	Required or Optional	Available Version
customFields	List<SObject>	Array of sObjects and custom fields for the sObjects. Standard fields are ignored. The custom fields must already be defined for the sObject. Currently, the WebCart, CartItem, and CartDeliveryGroup sObjects are supported. Field-level security rules from the shopper profile are applied to the custom fields. The rules are applied for registered shoppers and for the guest shopper profile. See Create a Cart and Cart Item with Custom Fields in a Commerce Store.	Optional	63.0
shippingAddress	[illegible]	Shipping address for a cart.	Required	63.0

ConnectApi.CartInventoryReservationInputRepresentation (Pilot)

Input representation to create or update a reservation.

Note: This feature is not generally available and is being piloted with certain Customers subject to additional terms and conditions. It is not part of your purchased Services. This feature is subject to change, may be discontinued with no notice at any time in Salesforce's sole discretion, and Salesforce may never make this feature generally available. Make your purchase decisions only on the basis of generally available products and features. This feature is made available on an AS IS basis and use of this feature is at your sole risk.

Property	Type	Description	Required or Optional	Available Version
durationInSeconds	Integer	Reservation duration in seconds.	Required	58.0

ConnectApi.CartItemInput

An item in a cart.

Property	Type	Description	Required or Optional	Available Version
cartDeliveryGroupId	String	ID of the cart delivery group.	Optional	59.0
customFields	List<SObject>	Array of sObjects and custom fields for the sObjects. Standard fields are ignored. The custom fields must already be defined for the sObject. Currently, only the CartItem sObject is supported. Field-level security rules from the shopper profile are applied to the custom fields. The rules are applied	Optional	61.0

Property	Type	Description	Required or Optional	Available Version
		for registered shoppers and for the guest shopper profile. The custom fields can be of type Checkbox, Currency, Date, Email, LongTextArea, Number, Percent, Phone, Text, TextArea, Url, Address, or Location. The `customFields` property isn't supported in stores built on an Aura template. See Create a Cart and Cart Item with Custom Fields in a Commerce Store.		
`productId`	String	ID of the product.	Required when adding an item to a cart Not supported when updating a cart item	49.0
`productSellingModelId`	String	Reserved for future use.	Optional	59.0
`quantity`	String	Quantity of the cart item. Use a value that can be converted to BigDecimal.	Required	49.0
`subscriptionTerm`	Integer on page 3550	Reserved for future use.	Optional	59.0
`type`	`ConnectApi.CartItemType`	Type of item in a cart. Value is `Product`. • `DeliveryCharge` • `Product`	Required when adding an item to a cart Not supported when updating a cart item	49.0

ConnectApi.CartItemPromotionCollectionInputRepresentation

Promotions for a cart item.

Property	Type	Description	Required or Optional	Available Version
`items`	List<`ConnectApi.CartItemPromotionInputRepresentation`>	List of cart items to get the associated promotions.	Required	52.0

ConnectApi.CartItemPromotionInputRepresentation

ID of a cart item associated with a promotion.

Property	Type	Description	Required or Optional	Available Version
`cartItemId`	String	ID of the item associated with the cart.	Optional	52.0

ConnectApi.CartMessagesVisibilityInput

Set the visibility for cart messages.

Property	Type	Description	Required or Optional	Available Version
`visibility`	Boolean	Specifies whether to set cart messages as visible (`true`) or not (`false`).	Required	50.0

ConnectApi.CartInput

A cart.

Property	Type	Description	Required or Optional	Available Version
`currencyIsoCode`	String	Currency ISO code of the cart.	Optional	57.0
`customFields`	List<`SObject`>	Array of sObjects and custom fields for the sObjects. Standard fields are ignored. The custom fields must already be defined for the sObject. Currently, only the WebCart sObject is supported. Field-level security rules from the shopper profile are applied to the custom fields. The rules are applied for registered shoppers and for the guest shopper profile. See Create a Cart and Cart Item with Custom Fields in a Commerce Store.	Optional	61.0
`effectiveAccountId`	String	ID of the buyer account or guest buyer profile for which the request is made. If unspecified, the default value is determined from context.	Optional	49.0
`isSecondary`	Boolean	Specifies whether the cart is secondary (`true`) or not (`false`). If unspecified, defaults to `false`.	Optional	53.0
`name`	String	Name of the cart. The name can have up to 250 Unicode characters. If unspecified, defaults to the generated name.	Optional	49.0
`orderOwnerId`	String	ID of the owner of the order.	Optional	58.0

Property	Type	Description	Required or Optional	Available Version
`type`	`ConnectApi.CartType`	Type of cart. Vaues are: • `Cart`—Cart created by a customer. • `PayNowReadOnly`—Clone of a Template cart that the customer can check out with using the Pay Now feature. • `Template`—Cart created by an internal user. If unspecified, defaults to `Cart`.	Optional	49.0
`typeAsString`	String	Type of the cart provided as a string.	Optional	59.0

ConnectApi.CartShippingAddressInput

A cart shipping address.

Subclass of `ConnectApi.AbstractCheckoutAddressInput`

No additional properties.

ConnectApi.CartToWishlistInput

Copy products from a cart to a wishlist.

Property	Type	Description	Required or Optional	Available Version
`wishlistId`	String	ID of the wishlist to copy cart products to.	Required	50.0

ConnectApi.CdpCalculatedInsightInput

Input representation for a calculated insight.

Property	Type	Description	Required or Optional	Available Version
`apiName`	String	API name of the calculated insight with suffix __cio.	Required for creating a calculated insight Optional for updating a calculated insight	57.0
`createdFromPackage`	Boolean	Specifies whether the calculated insight was created from an installed package (`true`) or not (`false`).	Optional	57.0

Property	Type	Description	Required or Optional	Available Version
`dataSpaceName`	String	Name of the data space.	Optional	57.0
`definitionType`	`ConnectApi.CalculatedInsightDefinitionTypeEnum`	Definition type of the calculated insight. Values are: • `CALCULATED_METRIC` • `CALCULATED_METRIC` • `CALCULATED_METRIC`	Required for creating a calculated insight Optional for updating a calculated insight	57.0
`description`	String	Calculated insight description.	Optional	57.0
`displayName`	String	Calculated insight display name.	Required for creating a calculated insight Optional for updating a calculated insight	57.0
`draft`	Boolean	Specifies whether to save the calculated insight as draft (`true`) or not (`false`).	Optional	57.0
`expression`	String	Calculated insight ANSI SQL expression.	Required for creating a calculated insight Optional for updating a calculated insight	57.0
`packagedCalculatedInsightApiName`	String	API name of the packaged calculated insight.	Optional	57.0

ConnectApi.CdpIdentityResolutionConfigInput

Input representation for creating an identity resolution ruleset.

Property	Type	Description	Required or Optional	Available Version
`configurationType`	`ConnectApi.CdpIdentityResolutionConfigurationType`	Source object for an identity resolution ruleset. Values are: • `Account` • `Individual`	Required	57.0
`description`	String	Description of the identity resolution ruleset.	Optional	57.0
`doesRunAutomatically`	Boolean	Specifies whether automatic job run scheduling is enabled for the ruleset (`true`) or not (`false`). If unspecified, defaults to `false`.	Optional	57.0

Property	Type	Description	Required or Optional	Available Version
label	String	User friendly name of the identity resolution ruleset.	Required	57.0
matchRules	List<ConnectApi.CdpIdentityResolutionMatchRule>	List of match rules for the identity resolution ruleset.	Optional	57.0
reconciliationRules	List<ConnectApi.CdpIdentityResolutionReconciliationRule>	List of reconciliation rules for the identity resolution ruleset.	Required	57.0
rulesetId	String	Extended ID of the ruleset used to differentiate between rulesets created for comparison. The ruleset ID must be unique and can't be longer than 4 characters.	Optional	57.0

ConnectApi.CdpIdentityResolutionConfigPatchInput

Input representation for updating an identity resolution ruleset.

Property	Type	Description	Required or Optional	Available Version
description	String	Description of the identity resolution ruleset.	Optional	57.0
doesRunAutomatically	Boolean	Specifies whether automatic job run scheduling is enabled for the ruleset (`true`) or not (`false`). If unspecified, defaults to `false`.	Optional	57.0
label	String	User friendly name of the identity resolution ruleset.	Required	57.0
matchRules	List<ConnectApi.CdpIdentityResolutionMatchRule>	List of match rules for the identity resolution ruleset.	Optional	57.0
reconciliationRules	List<ConnectApi.CdpIdentityResolutionReconciliationRule>	List of reconciliation rules for the identity resolution ruleset.	Required	57.0

ConnectApi.CdpIdentityResolutionMatchCriterion

Input representation for an identity resolution ruleset's match rule criterion.

Property	Type	Description	Required or Optional	Available Version
caseSensitiveMatch	Boolean	Specifies whether the criterion match is case sensitive (true) or not (false). Available only when matching is based on the party identifier.	Optional	58.0
entityName	String	API name of the Data Model Object the match rule applies to.	Required	57.0
fieldName	String	Name of the field the criterion applies to.	Required	57.0
matchMethodType	ConnectApi.CdpIdentityResolutionMatchMethodType	Match method for a match rule criterion. Values are: • Exact—Exact match. • ExactNormalized—Exact normalized match. • Fuzzy—Fuzzy match with medium precision. • FuzzyHigh—Fuzzy match with high precision. • FuzzyLow—Fuzzy match with low precision.	Required	57.0
partyIdentificationInfo	ConnectApi.CdpIdentityResolutionMatchCriterionPartyIdentificationInfo	Party Identifier information.	Optional	57.0
shouldMatchOnBlank	Boolean	Specifies whether blank fields can be used for matching (true) or not (false).	Required	57.0

SEE ALSO:

ConnectApi.CdpIdentityResolutionMatchRule

ConnectApi.CdpIdentityResolutionMatchCriterionPartyIdentificationInfo

Input representation for information when party identification is used in an identity resolution ruleset's match rule criterion.

Property	Type	Description	Required or Optional	Available Version
partyName	String	Party identification name.	Required if the match rule criterion uses party identification for matching	57.0

Property	Type	Description	Required or Optional	Available Version
`partyType`	String	Party identification type.	Optional	57.0

SEE ALSO:

ConnectApi.CdpIdentityResolutionMatchCriterion

ConnectApi.CdpIdentityResolutionMatchRule

Input representation for an identity resolution ruleset's match rule.

Property	Type	Description	Required or Optional	Available Version
`criteria`	List<`ConnectApi.CdpIdentityResolutionMatchCriterion`>	Object and field the match rule applies to and the match method applied.	Required	57.0
`label`	String	User friendly name for the identity resolution match rule.	Required	57.0

SEE ALSO:

ConnectApi.CdpIdentityResolutionConfigInput

ConnectApi.CdpIdentityResolutionConfigPatchInput

ConnectApi.CdpIdentityResolutionReconciliationFieldRule

Input representation for an identity resolution ruleset's reconciliation rule for a field.

Property	Type	Description	Required or Optional	Available Version
`fieldName`	String	The field that this reconciliation rule applies to.	Required	57.0
`ruleType`	`ConnectApi.CdpIdentityResolutionReconciliationRuleType`	Default reconciliation rule applied to fields in the object the reconciliation rule applies to. Values are: • `LastUpdated` • `MostFrequent` • `SourceSequence`	Required	57.0
`shouldIgnoreEmptyValue`	Boolean	Specifies whether to ignore an empty value (`true`) or not (`false`).	Required	57.0

Property	Type	Description	Required or Optional	Available Version
`sources`	List<`ConnectApi.CdpIdentityResolutionReconciliationSource`>	If `ruleType` is `SourceSequence`, a prioritized list of data sources.	Required if `ruleType` is `SourceSequence`	57.0

SEE ALSO:

ConnectApi.CdpIdentityResolutionReconciliationRule

ConnectApi.CdpIdentityResolutionReconciliationRule

Input representation for an identity resolution ruleset's default reconciliation rule for an object.

Property	Type	Description	Required or Optional	Available Version
`entityName`	String	API name of the Data Model Object the reconciliation rule applies to.	Required	57.0
`fields`	List<`ConnectApi.CdpIdentityResolutionReconciliationFieldRule`>	Field-specific reconciliation rules that override this default rule for the specified field.	Optional	57.0
`ruleType`	`ConnectApi.CdpIdentityResolutionReconciliationRuleType`	Default reconciliation rule applied to fields in the object the reconciliation rule applies to. Values are: • `LastUpdated` • `MostFrequent` • `SourceSequence`	Required	57.0
`shouldIgnoreEmptyValue`	Boolean	Specifies whether to ignore an empty value (`true`) or not (`false`).	Required	57.0
`sources`	List<`ConnectApi.CdpIdentityResolutionReconciliationSource`>	If `ruleType` is `SourceSequence`, a list of data sources in priority order.	Required if `ruleType` is `SourceSequence`	57.0

SEE ALSO:

ConnectApi.CdpIdentityResolutionConfigInput

ConnectApi.CdpIdentityResolutionConfigPatchInput

ConnectApi.CdpIdentityResolutionReconciliationSource

Input representation for an identity resolution default reconciliation rule or field-specific rule using the `SourceSequence` match method.

Property	Type	Description	Required or Optional	Available Version
`name`	String	If the `ruleType` for a reconciliation rule is `SourceSequence`, API name of a source Data Lake Object.	Required if `ruleType` is `SourceSequence`	57.0

SEE ALSO:

- ConnectApi.CdpIdentityResolutionReconciliationRule
- ConnectApi.CdpIdentityResolutionReconciliationFieldRule
- ConnectApi.CdpIdentityResolutionReconciliationFieldRule

ConnectApi.CdpIdentityResolutionRunNowInput

Input representation for running an identity resolution ruleset job on demand.

Property	Type	Description	Required or Optional	Available Version
`callingApp`	String	Calling application.	Optional	57.0
`callingAppInfo`	String	Calling application information.	Optional	57.0

ConnectApi.CdpQueryInput

Data query input.

Property	Type	Description	Required or Optional	Available Version
`sql`	String	ANSI-standard SQL query.	Required	52.0

ConnectApi.CdpSegmentDbtInput

Segment dbt input.

Property	Type	Description	Required or Optional	Available Version
`models`	List<`ConnectApi.CdpSegmentDbtModelInput`>	List of models. The segment data build tool currently supports a single SQL model.	Required	55.0

SEE ALSO:

- ConnectApi.CdpSegmentInput

ConnectApi.CdpSegmentDbtModelInput

Segment dbt model input.

Property	Type	Description	Required or Optional	Available Version
`name`	String	Dbt model name.	Required	55.0
`sql`	String	Dbt SQL. Dbt SQL date strings must be in ISO 8601 format , for example, 2011-02-25T18:24:31.000Z. For details about supported validations, see Supported Validations for Segment Data Build Tool Model SQL.	Required	55.0

SEE ALSO:

ConnectApi.CdpSegmentDbtInput

ConnectApi.CdpSegmentInput

Segment input.

Property	Type	Description	Required or Optional	Available Version
`additionalMetadata`	Map<String, String>	Map of additional metadata.	Optional for creating a segment Not supported for updating a segment	55.0
`dataSpace`	String	Segment dataspace. In API version 59.0 and later, this property is not available. Use the `dataspace` request parameter instead.	Optional	57.0–58.0
`description`	String	Segment description.	Optional	55.0
`developerName`	String	Segment developer name.	Required for creating a segment Not supported for updating a segment	55.0
`displayName`	String	Segment display name.	Optional	57.0
`includeDbt`	`ConnectApi.CdpSegmentDbtInput`	Segment data build tool.	Required	55.0

Property	Type	Description	Required or Optional	Available Version
`publishSchedule`	`ConnectApi.PublishSchedule`	Publish refresh schedule. Values are: • `One`—Refreshes every hour. Used to rapidly publish UI and DBT-based segments. • `Four`—Refreshes every four hours. Used to rapidly publish UI and DBT-based segments. • `Twelve`—Refreshes every twelve hours. • `TwentyFour`—Refreshes every twenty-four hours.	Optional	55.0
`publishScheduleEndDate`	String	Date indicating the end of the publish schedule.	Optional if `publishSchedule` isn't specified	55.0
`publishScheduleStartDateTime`	String	Datetime indicating the start of the publish schedule.	Optional if `publishSchedule` isn't specified	55.0
`segmentOnApiName`	String	API name of the SegmentOn entity.	Optional	57.0
`segmentType`	`ConnectApi.SegmentType`	Type of segment. Value is: • `Dbt`—Data build tool After a segment is created, the segment type can't be changed.	Required for creating a segment Not supported for updating a segment	55.0

ConnectApi.ChangeInputRepresentation

A list of changes to OrderItemSummaries that make up an order change, such as a cancel or return.

Property	Type	Description	Required or Optional	Available Version
`changeItems`	List<`ConnectApi.ChangeItemInputRepresentation`>	List of changes to OrderItemSummaries.	Required	48.0

SEE ALSO:

previewCancel(orderSummaryId, changeInput)

previewReturn(orderSummaryId, changeInput)

submitCancel(orderSummaryId, changeInput)

submitReturn(orderSummaryId, changeInput)

ConnectApi.ChangeItemFeeInputRepresentation

Input representation for Change Item Fee Input

Property	Type	Description	Required or Optional	Available Version
`amount`	Double	Positive value used to calculate the fee amount, as described by the `amountType`.	Required	57.0
`amountType`	String	Describes how the fee amount is calculated. Valid values are: • `AmountWithTax`—Value of `amount` is the fee amount, including tax. • `AmountWithoutTax`—Value of `amount` is the fee amount, not including tax. Tax is calculated on the value and added. • `Percentage`—Value of `amount` is a percentage. To determine the fee amount, `amount` is divided by 100, and then multiplied by the TotalPrice and TotalTaxAmount of the associated OrderItemSummary, prorated for the quantity being returned. • `PercentageGross`—Value of `amount` is a percentage. To determine the fee amount, `amount` is divided by 100, and then multiplied by the TotalLineAmountWithTax of the associated OrderItemSummary, prorated for the quantity being returned.	Required	57.0
`description`	String	Description of the fee.	Required	57.0
`priceBookEntryId`	String	ID of the price book entry associated with the fee product.	Required unless price books are optional in the org	57.0
`product2Id`	String	ID of the product representing the fee.	Required	57.0

Property	Type	Description	Required or Optional	Available Version
`reason`	String	Reason for the fee. The value must match an entry in the OrderProductSummaryChange object's Reason picklist.	Required	57.0

SEE ALSO:

- ConnectApi.ChangeInputRepresentation
- ConnectApi.ChangeItemInputRepresentation
- previewCancel(orderSummaryId, changeInput)
- previewReturn(orderSummaryId, changeInput)
- submitCancel(orderSummaryId, changeInput)
- submitReturn(orderSummaryId, changeInput)

ConnectApi.ChangeItemFeeTaxInputRepresentation

Input representation of taxes associated with a change item fee.

Property	Type	Description	Required or Optional	Available Version
`amount`	Double	Tax amount of the change item fee.	Required	63.0
`description`	String	Description of the change item fee.	Required	63.0
`rate`	Double	Tax rate for the change item fee.	Required	63.0
`taxEffectiveDate`	String	Effective date for the tax.	Required	63.0
`type`	String	Describes how the fee amount is calculated. Valid values are: • `Actual` • `Estimated`	Required	63.0

ConnectApi.ChangeItemFeeWithTaxInputRepresentation

Input representation of a change item fee with taxes.

Property	Type	Description	Required or Optional	Available Version
`amount`	Double	Positive value used to calculate the fee amount.	Required	63.0
`changeItemFees`	[illegible]	List of taxes associated with the change item fees.	Required	63.0

Property	Type	Description	Required or Optional	Available Version
description	String	Description of the fee.	Required	63.0
orderDeliveryGroupSummaryId	String	ID of the order delivery group summary.	Required	63.0
priceBookEntryId	String	ID of the price book entry associated with the fee product.	Required unless price books are optional in the org	63.0
product2Id	String	ID of the product representing the fee.	Required	63.0
reason	String	Reason for the cancellation. The value must match one of the picklist values on the Reason field of the Order Product Summary Change object.	Required	63.0

ConnectApi.ChangeItemInputRepresentation

Change to an order item summary, such as a return or cancel. You specify whether to prorate the associated shipping charge based on the price change. The order item summary can't be a shipping charge.

Property	Type	Description	Required or Optional	Available Version
changeItemFees	List<ChangeItemFee>	List of input data for fees associated with the order item being returned or canceled.	Optional	57.0
orderItemSummaryId	String	ID of the order item summary.	Required	48.0
quantity	Double	Quantity to change. Use a positive value. For example, a value of 2 means "cancel or return 2 units."	Required	48.0
reason	String	Reason for the change. The value must match one of the picklist values on the Reason field of the OrderItemSummaryChange object.	Required	48.0
reasonForChangeText	String	Reason text used for the return insights. The value has a max of 255 characters.	Optional	59.0

Property	Type	Description	Required or Optional	Available Version
shippingReductionFlag	Boolean	Specifies whether to prorate the shipping charge.	Required	48.0

SEE ALSO:

- ConnectApi.ChangeInputRepresentation
- previewCancel(orderSummaryId, changeInput)
- previewReturn(orderSummaryId, changeInput)
- submitCancel(orderSummaryId, changeInput)
- submitReturn(orderSummaryId, changeInput)

ConnectApi.ChatterGroupInput

Chatter group input.

Property	Type	Description	Available
`announcement`	String	The 18-character ID of an announcement. An announcement displays in a designated location in the Salesforce UI until 11:59 p.m. on its expiration date, unless it's deleted or replaced by another announcement.	31.0
`canHaveChatterGuests`	Boolean	`true` if this group allows Chatter customers, `false` otherwise. After this property is set to `true`, it cannot be set to `false`.	29.0
`description`	String	The "Description" section of the group.	29.0
`information`	`ConnectApi.GroupInformationInput`	The "Information" section of a group. If the group is private, this section is visible only to members.	28.0
`isArchived`	Boolean	`true` if the group is archived, `false` otherwise. Defaults to `false`.	29.0
`isAutoArchiveDisabled`	Boolean	`true` if automatic archiving is turned off for the group, `false` otherwise. Defaults to `false`.	29.0
`name`	String	The name of the group.	29.0
`owner`	String	The ID of the group owner. This property is available for PATCH requests only.	29.0

Property	Type	Description	Available
`visibility`	`ConnectApi.GroupVisibilityType`	Group visibility type. • `PrivateAccess`—Only members of the group can see posts to this group. • `PublicAccess`—All users within the Experience Cloud site can see posts to this group. • `Unlisted`—Reserved for future use.	29.0

SEE ALSO:

createGroup(communityId, groupInput)

updateGroup(communityId, groupId, groupInput)

ConnectApi.ChatterStreamInput

A Chatter feed stream.

Property	Type	Description	Required or Optional	Available Version
`description`	String	Description of the stream, up to 1,000 characters.	Optional	39.0
`name`	String	Name of the stream, up to 120 characters.	Required when creating a stream Optional when updating a stream	39.0
`subscriptionsToAdd`	List<`ConnectApi.StreamSubscriptionInput`>	List of up to 25 entities whose feeds are included in the stream. Adding an entity that is already added results in no operation. Including the same entity in `subscriptionsToAdd` and `subscriptionsToRemove` results in no operation.	Optional	39.0
`subscriptionsToRemove`	List<`ConnectApi.StreamSubscriptionInput`>	List of entities whose feeds are removed from the stream. Removing an entity that is already removed results in no operation. Including the same entity in `subscriptionsToAdd` and `subscriptionsToRemove` results in no operation.	Optional when updating a stream Not supported when creating a stream	39.0

ConnectApi.CommentCapabilitiesInput

A container for all capabilities that can be included with a comment.

Property	Type	Description	Required or Optional	Available Version
`content`	ConnectApi.ContentCapabilityInput	Content to attach to the comment.	Optional	32.0
`feedEntityShare`	ConnectApi.FeedEntityShareCapabilityInput	Feed entity to share to the comment.	Optional	42.0
`record`	ConnectApi.RecordCapabilityInput	Existing knowledge article to attach to the comment.	Optional	42.0

SEE ALSO:

ConnectApi.CommentInput

ConnectApi.CommentInput

Comment input used to add rich comments, for example, comments that include mentions or file attachments.

Property	Type	Description	Required or Optional	Available Version
`attachment`	`ConnectApi.FeedItemAttachmentInput`	Specifies an attachment for the comment. Valid values are: • `ContentAttachmentInput` • `NewFileAttachmentInput` `LinkAttachmentInput` is not permitted for comments. **Important:** As of version 32.0, use the `capabilities` property.	Optional	28.0–31.0
`body`	`ConnectApi.MessageBodyInput`	Description of message body. The body can contain up to 10,000 characters and 25 mentions. Because the character limit can change, clients should make a `describeSObjects()` call on the FeedItem or FeedComment object and look at the length of the `Body` or `CommentBody` field to determine the maximum number of allowed characters. To edit this property in a comment, use `updateComment(communityId, commentId, comment)`. Editing comments is supported in version 34.0 and later.	Required	28.0

Property	Type	Description	Required or Optional	Available Version
		Rich text and inline images are supported in comment bodies in version 35.0 and later. Entity links are supported in version 43.0 and later.		
`capabilities`	`ConnectApi.CommentCapabilityInput`	Specifies any capabilities for the comment, such as a file attachment.	Optional	32.0
`threadParentId`	String	ID of the parent comment for a threaded comment.	Optional	44.0

SEE ALSO:

- Post a Comment with a Mention
- Post a Comment with a New File
- Post a Comment with an Existing File
- Post a Rich-Text Comment with Inline Image
- Post a Rich-Text Feed Comment with a Code Block
- Edit a Comment
- postCommentToFeedElement(communityId, feedElementId, comment, feedElementFileUpload)

ConnectApi.CommerceAddressFieldInput

Commerce address field input. This is used to reference custom fields for the address.

Property	Type	Description	Required or Optional	Available Version
`dataName`	String	The name of the custom address field.	Required	54.0
`text`	String	The value of the custom address field.	Optional	54.0

ConnectApi.CommerceAddressInput

Commerce address input.

Property	Type	Description	Required or Optional	Available Version
`addressType`	String	Type of address, for example, `Shipping` or `Billing`.	Optional	54.0
`city`	String	The address city.	Optional	54.0

Property	Type	Description	Required or Optional	Available Version
commerceAddressFieldInputList	List<ConnectApi.CommerceAddressFieldInput>	A list of custom address fields, if any.	Optional	54.0
companyName	String	The address company name.	Optional	57.0
country	String	The address country.	Optional	54.0
countryCode	String	Two-character country code.	Optional	54.0–58.0
firstName	String	The address first name.	Optional	57.0
isDefault	Boolean	Indicates whether a contact's address is the preferred method of communication (`true`) or not (`false`). The default value is `false`.	Optional	54.0
lastName	String	The address last name.	Optional	57.0
middleName	String	The address middle name.	Optional	57.0
name	String	Name of the contact.	Required	54.0
phoneNumber	String	The address phone number.	Optional	57.0
postalCode	String	Zip code or postal code for the address.	Optional	54.0
region	String	The address state.	Optional	54.0
regionCode	String	The address state code.	Optional	54.0–58.0
street	String	The address street.	Optional	54.0

ConnectApi.CompositeCommerceProductInputRepresentation

Composite product input.

Property	Type	Description	Required or Optional	Available Version
attributeSetInfo	ConnectApi.[illegible]	Attribute set information for a variation parent product.	Optional	62.0
categoryIds	List<String>	List of category IDs associated with the product.	Optional	61.0
productFields	Map<String, String>	A map of product field names and their values.	Required	61.0
productMedia	ConnectApi.ProductMedia	Media associated with the product.	Optional	61.0

ConnectApi.ConfirmHeldFOCapacityInputRepresentation

Request to confirm held fulfillment order capacity at one or more locations. Can correspond to one action call.

Property	Type	Description	Required or Optional	Available Version
`confirmHeldFOCapacityRequests`	List<ConnectApi.ConfirmHeldFOCapacityRequestInputRepresentation>	List of requests to confirm held fulfillment order capacity at one or more locations.	Required	55.0

ConnectApi.ConfirmHeldFOCapacityRequestInputRepresentation

Request to confirm held fulfillment order capacity at one or more locations.

Property	Type	Description	Required or Optional	Available Version
`allOrNothing`	Boolean	Controls whether a single failed request cancels all other requests in the list (*`true`*) or whether some requests can succeed if others fail (*`false`*). The default value is *`false`*.	Optional	55.0
`capacityRequests`	List<ConnectApi.CapacityRequestInputRepresentation>	List of requests to confirm held fulfillment order capacity. Each request is for one fulfillment order assigned to one location.	Required	55.0

ConnectApi.ContentCapabilityInput

Attach or update a file on a comment. Use this class to attach a new file or update a file that has already been uploaded to Salesforce.

This class is a subclass of ConnectApi.FeedElementCapabilityInput.

To attach or remove files from a feed post (instead of a comment) in version 36.0 and later, use ConnectApi.FilesCapabilityInput.

Property	Type	Description	Required or Optional	Available Version
`contentDocumentId`	String	ID of the existing content.	Required for existing content	32.0
`description`	String	Description of the file to be uploaded.	Optional	32.0
`sharingOption`	ConnectApi.FileSharingOption	Sharing option of the file. Values are: • `Allowed`—Resharing of the file is allowed. • `Restricted`—Resharing of the file is restricted.	Optional	35.0

Property	Type	Description	Required or Optional	Available Version
`title`	String	Title of the file. This value is used as the file name for new content. For example, if the title is My Title, and the file is a .txt file, the file name is My Title.txt.	Required for new content	32.0

SEE ALSO:

ConnectApi.FeedElementCapabilitiesInput

ConnectApi.ContentHubFieldValueInput

Fields of the item type.

Property	Type	Description	Required or Optional	Available Version
`name`	String	Name of the item field. When updating the metadata of a repository file, only the name field can be updated.	Required	39.0
`value`	String	Value of the item field.	Required	39.0

SEE ALSO:

ConnectApi.ContentHubItemInput

ConnectApi.ContentHubItemInput

Item type ID and fields of the item type.

Property	Type	Description	Required or Optional	Available Version
`fields`	List<`ConnectApi.ContentHubFieldValueInput`>	List of fields for the item.	Required to create a SharePoint file in a repository because the file name is required; otherwise optional	39.0
`itemTypeId`	String	ID of the item type.	Required to create a file in a repository	39.0

ConnectApi.ContractInputRepresentation

Input to create and update contract.

Property	Type	Description	Required or Optional	Available Version
isAutoDocgenRequired	Boolean	Specifies whether automatic document generation is required or not.	Required	56.0
recordTypeName	String	Contract record type name.	Optional	56.0
sourceObjectId	String	Source record ID.	Required	56.0
templateName	String	Document template name for document generation.	Optional	56.0

ConnectApi.CouponCodeRedemptionInput

Input representation for coupon code redemption.

Property	Type	Description	Required or Optional	Available Version
buyer	String	ID of the buyer account or email address for a guest user.	Required	60.0
couponCodes	List<String>	List of coupon codes.	Required	58.0
effectiveAccountId	String	ID of the account.	Required	58.0–59.0
transactionId	String	ID of the transaction, which must be a valid cart ID.	Required	58.0

ConnectApi.CreateCreditMemoInputRepresentation

A list of change orders used to create a credit memo.

Property	Type	Description	Required or Optional	Available Version
changeOrderIds	List<String>	List of IDs of the change orders.	Required	48.0

SEE ALSO:

createCreditMemo(orderSummaryId, creditMemoInput)

ConnectApi.CreateInvoiceFromChangeOrdersInputRepresentation

OrderSummary and associated change orders to create Invoices for.

Property	Type	Description	Required or Optional	Available Version
changeOrderIds	List<String>	List of IDs of change orders to create Invoices for.	Required	56.0
orderSummaryId	String	ID of the associated Order Summary.	Required	56.0

SEE ALSO:

createMultipleInvoices(invoicesInput)

ConnectApi.CreateMultipleInvoicesFromChangeOrdersInputRepresentation

ConnectApi.CreateMultipleInvoicesFromChangeOrdersInputRepresentation

Data about the change orders to create Invoices for.

Property	Type	Description	Required or Optional	Available Version
invoicesFromChangeOrders	List<ConnectApi.CreateInvoiceFromChangeOrdersInputRepresentation>	List of OrderSummary IDs with the IDs of the associated change orders to create Invoices for. Each entry in the list generates one invoice, which combines the change orders in that entry.	Required	56.0

SEE ALSO:

createMultipleInvoices(invoicesInput)

ConnectApi.CreateOrderPaymentSummaryInputRepresentation

An OrderSummary for which to create an OrderPaymentSummary, with the payment authorization or payments to include in it.

Property	Type	Description	Required or Optional	Available Version
orderSummaryId	String	ID of the OrderSummary.	Required	48.0
paymentAuthorizationId	String	ID of the payment authorization.	Either a payment authorization or at least one payment is required.	48.0

Property	Type	Description	Required or Optional	Available Version
`paymentIds`	List<String>	List of IDs of the payments.	Either a payment authorization or at least one payment is required.	48.0

SEE ALSO:

createOrderPaymentSummary(orderPaymentSummaryInput)

ConnectApi.CreateServiceAppointmentInput

Contains information to create a service appointment.

Property	Type	Description	Required or Optional	Available Version
`assignedResources`	List<ConnectApi.AssignedResourceInput>	Represents the service resources to be assigned to a service appointment. Note: When creating an appointment, use `extendedFields` to add values to any of the fields, including custom fields, in `assignedResources` as long as you have edit access to those fields.	Optional	53.0
`lead`	ConnectApi.LeadInput	Represents a prospect or lead. Note: Required to create a service appointment for unauthenticated guest users.	Required if `serviceAppointment` isn't provided.	53.0
`schedulingPolicyId`	String	The ID of the `AppointmentSchedulingPolicy` object. If no scheduling policy is passed in the request body, the default configurations are used. The only scheduling policy configuration that is used in determining time slots is the enforcement of account visiting hours.	Optional	53.0
`serviceAppointment`	ConnectApi.ServiceAppointmentInput	Represents the service appointment details to book an appointment. Note: When creating an appointment, use `extendedFields` to add values	Required if `lead` isn't provided.	53.0

Property	Type	Description	Required or Optional	Available Version
		to any of the fields, including custom fields, in `assignedResources` as long as you have edit access to those fields.		

ConnectApi.CredentialCustomHeaderInput

Credential custom header input.

Property	Type	Description	Required or Optional	Available Version
`headerName`	String	Header name.	Required	58.0
`headerValue`	String	Header value.	Required	58.0
`id`	String	Header ID.	Optional	58.0
`sequenceNumber`	Integer	Sequence number.	Required	58.0

SEE ALSO:

ConnectApi.ExternalCredentialInput

ConnectApi.NamedCredentialInput

ConnectApi.CredentialInput

Credential input.

Property	Type	Description	Required or Optional	Available Version
`authentication Protocol`	`ConnectApi. Credential Authentication Protocol`	Authentication protocol of the external credential. Values are: • `AwsSv4` • `Basic` • `Custom` • `Jwt` • `OAuth`	Required	56.0
`authentication ProtocolVariant`	`ConnectApi. Credential Authentication ProtocolVariant`	Authentication protocol variant of the external credential. Values are: • `AwsSv4_STS`—AWS Signature Version 4 with Security Token Service.	Optional	57.0

Property	Type	Description	Required or Optional	Available Version
		• `ClientCredentialsClientSecret`—OAuth 2.0 Client Credentials client secret. Client secrets are sent in the callout's request body. • `ClientCredentialsClientSecretBasic`—OAuth 2.0 Client Credentials client secret. Client secrets are sent in the callout's authorization header, as with Basic authentication. • `ClientCredentialsJwtAssertion`—OAuth 2.0 Client Credentials JSON Web Token assertion. • `JwtBearer`—OAuth 2.0 JSON Web Token bearer flow. • `NoAuthentication`—No authentication. • `RolesAnywhere`—AWS Signature Version 4 with Identity and Access Management (IAM) Roles Anywhere. If specified, the authentication protocol variant must match the actual protocol variant of the external credential.		
`credentials`	Map<String, `ConnectApi.CredentialValueInput`>	Map of protocol-specific credentials. Authentication protocols have credential allowlists and encryption rules. • `AwsSv4`—`awsAccessKeyId` (not encrypted), `awsSecretAccessKey` (encrypted), `awsRoleArn` (not encrypted) • `Custom`—Any credential name is valid (user sets encryption rules)	Required	56.0
`externalCredential`	String	Fully qualified developer name of the external credential.	Required	56.0
`principalName`	String	Name of the external credential named principal.	Required if `principalType` is `NamedPrincipal`	56.0
`principalType`	`ConnectApi.CredentialPrincipalType`	Type of credential principal. Values are: • `AwsStsPrincipal` • `NamedPrincipal`	Required	56.0

Property	Type	Description	Required or Optional	Available Version
		• `PerUserPrincipal`		

ConnectApi.CredentialValueInput

Credential value input.

Authentication protocols have credential allowlists and encryption rules.

- `AwsSv4`—`awsAccessKeyId` (not encrypted), `awsSecretAccessKey` (encrypted), `awsRoleArn` (not encrypted)
- `Custom`—Any credential name is valid (user sets encryption rules)

Property	Type	Description	Required or Optional	Available Version
`encrypted`	Boolean	Specifies whether the value of the credential is encrypted (`true`) or not (`false`).	Required	56.0
`revision`	Integer	Revision number of a short-lived credential, such as OAuthToken. If the provided revision isn't the latest version, the authentication endpoint refreshes the credential.	Optional	58.0
`value`	String	Value of the credential.	Required	56.0

SEE ALSO:

ConnectApi.CredentialInput

ConnectApi.CustomListAudienceCriteriaInput

Criteria for the custom list type of custom recommendation audience.

Subclass of ConnectApi.AudienceCriteriaInput.

Property	Type	Description	Required or Optional	Available Version
`memberOperationType`	`ConnectApi.RecommendationAudienceMemberOperationType`	The operation to carry out on the audience members. Values are: • `Add`—Adds specified members to the audience. • `Remove`—Removes specified members from the audience.	Required to update a recommendation audience Don't use or specify `null` to create a recommendation audience	36.0

Property	Type	Description	Required or Optional	Available Version
`members`	List<String>	A collection of user IDs. When updating an audience, you can include up to 100 members. An audience can have up to 100,000 members, and each Experience Cloud site can have up to 100 audiences.	Required to update a recommendation audience Don't use or specify `null` to create a recommendation audience	36.0

ConnectApi.DeliveryAddressInputRepresentation

Delivery address.

While each field is optional, at least one combination (latitude and longitude, country and postal code, or city, state, and country) must be included. The fields can't be left empty.

Property	Type	Description	Required or Optional	Available Version
`city`	String	City in the state for the delivery address.	Optional	63.0
`country`	String	Country code for the delivery address.	Optional	63.0
`latitude`	Double	Latitude for the delivery address.	Optional	63.0
`longitude`	Double	Longitude for the delivery address.	Optional	63.0
`postalCode`	String	Postal code of the delivery address.	Optional	63.0
`state`	String	State in the country for the delivery address.	Optional	63.0

ConnectApi.DeliveryEstimationProductInputRepresentation

Delivery estimation product information.

Property	Type	Description	Required or Optional	Available Version
`name`	String	Product name.	Optional	63.0
`productId`	String	ID of the product.	Optional	63.0
`quantity`	Double	Product quantity.	Required	63.0
`stockKeepingUnit`	String	Product's stock keeping unit (SKU).	Required	63.0

ConnectApi.DirectMessageCapabilityInput

Create or update the members of a direct message.

Property	Type	Description	Required or Optional	Available Version
`membersToAdd`	List<String>	List of user IDs for members to include in the direct message.	Required when creating a direct message (POST) Optional when updating a direct message (PATCH)	39.0
`membersToRemove`	List<String>	List of user IDs for members to remove from the direct message.	Optional when updating a direct message (PATCH) Not supported when creating a direct message (POST)	40.0
`subject`	String	Subject of the direct message.	Optional when creating a direct message (POST) Not supported when updating a direct message (PATCH)	39.0

SEE ALSO:

ConnectApi.FeedElementCapabilitiesInput

ConnectApi.DistinctValueRefinementInput

Attribute-based refinement with distinct values for product search.

This class is a subclass of ConnectApi.RefinementInput.

Property	Type	Description	Required or Optional	Available Version
`values`	List<String>	Comma-separated list of attribute values. It considers attribute labels for localization.	Required	52.0

ConnectApi.DoubleList

List of double values.

Subclass of ConnectApi.AbstractList.

Property	Type	Description	Required or Optional	Available Version
`values`	List<Double>	List of Double values to filter on.	Optional	63.0

ConnectApi.EinsteinLlmAdditionalConfigInput

Additional configuration information for the LLM provider.

Property	Type	Description	Required or Optional	Available Version
`additionalParameters`	Map<String, `ConnectApi.WrappedValue`>	Map of parameters and values for the LLM provider.	Optional	60.0
`applicationName`	String	Name of the application.	Required	60.0
`enablePiiMasking`	Boolean	Specifies whether to mask personally identifiable information (PII) (`true`) or not (`false`).	Optional	60.0
`frequencyPenalty`	Double	Use to reduce the repetitiveness of generated tokens. The higher the value, the stronger a penalty is applied to previously present tokens, proportional to how many times they already appeared in the prompt or in prior generations. Minimum value is `0.0`. Maximum value is `1.0`.	Optional	60.0
`maxTokens`	Integer	Maximum number of tokens to generate.	Optional	60.0
`numGenerations`	Integer	Number of generation requests to send to the LLM provider.	Optional	60.0
`presencePenalty`	Double	Use to reduce the repetitiveness of generated tokens. This value is similar to frequency penalty, except that this penalty is applied equally to all tokens that already appeared, regardless of their exact frequencies. Minimum value is `0.0`, and maximum value is `1.0`.	Optional	60.0
`stopSequences`	List<String>	Generated text is cut at the end of the earliest occurrence of a stop sequence.	Optional	60.0
`temperature`	Double	Sampling temperature to use. Higher values mean the model takes more risks. Lower temperatures mean that generations are less random. Minimum value is `0.0`, and maximum value is `1.0`	Optional	60.0

ConnectApi.EinsteinPromptTemplateGenerationsInput

Prompt template input parameters to use for generation.

Property	Type	Description	Required or Optional	Available Version
`additionalConfig`	`ConnectApi.EinsteinLlmAdditionalConfigInput`	Configuration information for the LLM provider.	Required	60.0
`citationMode`	String	Mode of citations for the specified prompt template. Valid values are: • `post_generation`—Citations are generated after the generated response for the specified prompt template. • `off`—Citations aren't generated for the specified prompt template.	Optional	62.0
`inputParams`	Map<String, ConnectApi.WrappedValue>	Parameters and values to resolve the specified prompt template.	Required	60.0
`isPreview`	Boolean	Specifies whether to only resolve the prompt template (`true`) or to resolve the prompt template and generate an LLM response (`false`).	Required	60.0
`outputLanguage`	String	Language code for the language to generate the LLM response in. See Supported Languages in Prompt Template Responses.	Optional	61.0
`tags`	ConnectApi.WrappedValue	Map of wrapped values, such as free-form user feedback, that can be used to resolve a specified prompt template.	Optional	62.0

ConnectApi.EnsureFundsAsyncInputRepresentation

ID of an Invoice to ensure funds for and apply them to.

Property	Type	Description	Required or Optional	Available Version
`invoiceId`	String	ID of the Invoice.	Required	48.0

Property	Type	Description	Required or Optional	Available Version
isConsiderReservedBalanceAmount	Boolean	If true, the reserved balance amount is used for the Order Summary to fund the invoice. If not enough reserved balance amount, any available balance that isn't reserved by another Order Summary is used. If false, any available balance is used.	Optional	59.0

SEE ALSO:

ensureFundsAsync(orderSummaryId, ensureFundsInput)

ConnectApi.EnsureRefundsAsyncInputRepresentation

ID of a credit memo to ensure refunds for, an amount of excess funds to refund, or both. At least one is required. Also includes any invoices for fees that reduce the refund amount, such as return fees. If multiple payment methods are available, you can specify how to distribute the refund.

Property	Type	Description	Required or Optional	Available Version
`creditMemoId`	String	ID of the credit memo that represents a refund amount.	Either `creditMemoId` or `excessFundsAmount` is required	48.0
`excessFundsAmount`	Double	Amount of excess funds to refund.	Either `excessFundsAmount` or `creditMemoId` is required	49.0
`invoicesToPay`	List<`ConnectApi.InvoiceToPayInputRepresentation`>	List of invoices for any fees that reduce the refund, such as return fees.	Optional	56.0
`isAllowPartial`	Boolean	This value controls the behavior when the amounts included in the `sequences` list don't cover the entire refund amount. If this value is false, then the default refund logic is applied to ensure the remaining refund amount. If this value is true, then the unrefunded balance remains on the credit memo. If you don't specify a `sequences` list, this value is ignored and the default refund logic is applied. The default value is false.	Optional	56.0

Property	Type	Description	Required or Optional	Available Version
isReservedBalanceAmountConsidered	Boolean	If true, the refundable amount is used to open the payment balance for the reservedBalanceAmount in the Order Payment Summaries. The remaining refundable amount considers the sequence of order payment summaries, if provided. If false, any reserved balance amount for exchanges is refunded.	Optional	59.0
sequences	List<ConnectApi.SequenceOrderPaymentSummaryInputRepresentation>	Ordered list of refund amounts and OrderPaymentSummaries to apply them to. An OrderPaymentSummary must either belong to the order summary or be a reference to the order summary in the OrderPaymentSummaryReference entity. The process traverses this list in order and stops when it's refunded the full amount.	Optional	56.0

SEE ALSO:

ensureRefundsAsync(orderSummaryId, ensureRefundsInput)

ConnectApi.EntityLinkSegmentInput

An entity link segment.

Subclass of ConnectApi.MessageSegmentInput.

Property	Type	Description	Required or Optional	Available Version
entityId	String	ID of the entity to link to. Only users with access to the entity see it. It's hidden for users without access.	Required	43.0

ConnectApi.EstimateDeliveryDateInputRepresentation

Delivery date estimation information.

Property	Type	Description	Required or Optional	Available Version
deliveryAddress	ConnectApi.DeliveryAddressInputRepresentation on page 1850	Delivery address.	Optional	63.0
locations	String	List of location external references.	Optional	63.0

Property	Type	Description	Required or Optional	Available Version
products	ConnectApi.DeliveryProductInputRepresentation on page 1850	List of products included in delivery estimation.	Required	63.0
shippingCarrier	ConnectApi.ShippingCarrierInputRepresentation on page 1937	Shipping carrier used to deliver the order.	Required	63.0

ConnectApi.ExtendedFieldInput

Contains information about the extended field.

Property	Type	Description	Required or Optional	Available Version
name	String	The name of the field, including custom field.	Optional	53.0
value	String	The value of the field.	Optional	53.0

ConnectApi.ExtensionInput

An extension.

Property	Type	Description	Required or Optional	Available Version
alternativeRepresentation	ConnectApi.AlternativeInput	Alternative representation of the extension.	Required	40.0
extensionId	String	ID of the extension.	Required	40.0
payload	String	Payload associated with the extension.	Required	40.0
payloadVersion	String	Payload version that identifies the structure of the payload associated with the extension.	Optional	40.0

SEE ALSO:

ConnectApi.ExtensionsCapabilityInput

ConnectApi.ExtensionsCapabilityInput

Create or update extensions associated with a feed element.

This class is a subclass of ConnectApi.FeedElementCapabilityInput.

Property	Type	Description	Required or Optional	Available Version
`itemsToAdd`	List<`ConnectApi.ExtensionInput`>	List of extensions to associate with the feed element.	Required for creating an extension Optional for updating an extension	40.0
`itemsToRemove`	List<String>	List of attachment IDs to remove from the feed element.	Optional for updating an extension Don't specify for creating an extension	41.0

SEE ALSO:

ConnectApi.FeedElementCapabilitiesInput

ConnectApi.ExternalAuthIdentityProviderCredentialInput

External auth identity provider credential input.

Property	Type	Description	Required or Optional	Available Version
`credentialName`	String	Name of the external auth identity provider credential.	Required	62.0
`credentialValue`	String	Value of the external auth identity provider credential.	Required	62.0

SEE ALSO:

ConnectApi.ExternalAuthIdentityProviderCredentialsInput

ConnectApi.ExternalAuthIdentityProviderCredentialsInput

External auth identity provider credentials input.

Property	Type	Description	Required or Optional	Available Version
credentials	ConnectApi.ExternalAuthIdentityProviderCredentialInput on page 1857>	List of external auth identity provider credentials to populate.	Required	62.0

SEE ALSO:

createExternalAuthIdentityProviderCredentials(fullName, requestBody)

updateExternalAuthIdentityProviderCredentials(fullName, requestBody)

ConnectApi.ExternalAuthIdentityProviderInput

External auth identity provider input.

Property	Type	Description	Required or Optional	Available Version
authenticationFlow	ConnectApi.IdentityProviderAuthFlow on page 2367	Authentication flow to get tokens to call protected APIs. Values are: • AuthorizationCode	Required	62.0
authenticationProtocol	ConnectApi.IdentityProviderAuthProtocol on page 2367	Authentication protocol required to access the external system. Values are: • OAuth	Required	62.0
authorizeUrl	String	Authorization endpoint URL for the external system.	Required when the authenticationProtocol is OAuth and the authenticationFlow is AuthorizationCode. Otherwise, Optional.	62.0
clientAuthentication	ConnectApi.IdentityProviderClientAuth	Client authentication method that describes how credentials are sent to the authorization server. Values are: • ClientSecretBasic • ClientSecretPost The default value is ClientSecretBasic.	Optional	63.0
description	String	Description of the external auth identity provider.	Optional	62.0

Property	Type	Description	Required or Optional	Available Version
`fullName`	String	Full name of the external auth identity provider. The full name can include a namespace prefix.	Required	62.0
`label`	String	External auth identity provider label.	Required	62.0
`parameters`	List<`ConnectApi.ExternalAuthIdentityProviderParameter`>	List of custom request parameters to customize and extend requests to the identity provider's token endpoint.	Optional	63.0
`standardExternalIdentityProvider`	String	Reference to a standard external auth identity provider.	Optional	63.0
`tokenUrl`	String	Token endpoint URL to retrieve tokens from the external system. Required for all OAuth 2.0 authentication flows.	Required	62.0
`userInfoUrl`	String	User info URL to retrieve user profile information from the external system. Applicable only when the `authenticationProtocol` is `OAuth`.	Optional	62.0

SEE ALSO:

createExternalAuthIdentityProvider(requestBody)

updateExternalAuthIdentityProvider(fullName, requestBody)

ConnectApi.ExternalAuthIdentityProviderParameterInput

External auth identity provider parameter input.

Property	Type	Description	Required or Optional	Available Version
`parameterName`	String	The name of the external auth identity provider parameter.	Required	63.0
`parameterType`	`ConnectApi.ExternalAuthIdentityProviderParameterType`	Parameter type for an external auth identity provider. Values are: • `AuthorizeRequestQueryParameter` • `IdentityProviderOptions` • `RefreshRequestBodyParameter` • `RefreshRequestHttpHeader` • `RefreshRequestQueryParameter`	Required	63.0

Property	Type	Description	Required or Optional	Available Version
		• `TokenRequestBodyParameter` • `TokenRequestHttpHeader` • `TokenRequestQueryParameter`		
`parameterValue`	String	If `parameterType` describes a literal value then the literal value is stored in this property.	Optional	63.0
`sequenceNumber`	Integer	Specifies the order of parameters to apply when an external auth identity provider has more than one parameter. Priority is from lower to higher numbers, for example, `1` is the highest priority.	Optional	63.0

SEE ALSO:

ConnectApi.ExternalAuthIdentityProviderInput

ConnectApi.ExternalCredentialInput

Input used to create or update an external credential.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Property	Type	Description	Required or Optional	Available Version
`authentication Protocol`	`ConnectApi. Credential Authentication Protocol`	Authentication protocol of the external credential. Values are: • `AwsSv4` • `Basic` • `Custom` • `Jwt` • `OAuth`	Required	58.0
`authentication ProtocolVariant`	`ConnectApi. Credential Authentication ProtocolVariant`	Authentication protocol variant of the external credential. Values are: • `AwsSv4_STS`—AWS Signature Version 4 with Security Token Service. • `ClientCredentialsClientSecret`—OAuth 2.0 Client Credentials client secret. Client secrets are sent in the callout's request body.	Optional	58.0

Property	Type	Description	Required or Optional	Available Version
		• `ClientCredentialsClientSecretBasic`—OAuth 2.0 Client Credentials client secret. Client secrets are sent in the callout's authorization header, as with Basic authentication. • `ClientCredentialsJwtAssertion`—OAuth 2.0 Client Credentials JSON Web Token assertion. • `JwtBearer`—OAuth 2.0 JSON Web Token bearer flow. • `NoAuthentication`—No authentication. • `RolesAnywhere`—AWS Signature Version 4 with Identity and Access Management (IAM) Roles Anywhere. If specified, the authentication protocol variant must match the actual protocol variant of the external credential.		
`customHeaders`	List<`ConnectApi.CredentialCustomHeaderInput`>	List of credential custom headers.	Optional	58.0
`developerName`	String	Fully qualified developer name of the external credential.	Required for creating an external credential Optional for updating an external credential	58.0
`masterLabel`	String	External credential label.	Required	58.0
`parameters`	List<`ConnectApi.ExternalCredentialParameterInput`>	List of external credential parameters.	Optional depending on `authenticationProtocol` and `authenticationVariant`	58.0
`principals`	List<`ConnectApi.ExternalCredentialPrincipalInput`>	List of principals the credential has.	Optional	58.0

ConnectApi.ExternalCredentialParameterInput

External credential parameter input.

Property	Type	Description	Required or Optional	Available Version
`id`	String	Parameter ID.	Optional	58.0
`parameterDescription`	String	Parameter description.	Optional	58.0
`parameterName`	String	Parameter name of the external credential.	Required	58.0
`parameterType`	`ConnectApi.ExternalCredentialParameterType`	Parameter type of the external credential. Values are: • `AdditionalRefreshStatusCode` • `AuthParameter` • `AuthProvider` • `AuthProviderUrl` • `AuthProviderUrlQueryParameter` • `JwtBodyClaim` • `JwtHeaderClaim` • `SigningCertificate`	Required	58.0
`parameterValue`	String	Parameter value of the external credential.	Required	58.0

SEE ALSO:

ConnectApi.ExternalCredentialInput

ConnectApi.ExternalCredentialPrincipalInput

ConnectApi.ExternalCredentialPrincipalInput

External credential principal input.

Property	Type	Description	Required or Optional	Available Version
`id`	String	ID of the principal external credential parameter.	Optional	58.0
`parameters`	List<`ConnectApi.ExternalCredentialParameterInput`>	List of external credential parameters.	Optional	58.0
`principalName`	String	Principal name.	Required	58.0
`principalType`	`ConnectApi.CredentialPrincipalType`	Type of credential principal. Values are: • `AwsStsPrincipal` • `NamedPrincipal` • `PerUserPrincipal`	Required	58.0

Property	Type	Description	Required or Optional	Available Version
sequenceNumber	Integer	Sequence number.	Required	58.0

SEE ALSO:

ConnectApi.ExternalCredentialInput

ConnectApi.NamedCredentialInput

ConnectApi.FeedElementCapabilitiesInput

A container for all capabilities that can be included when creating a feed element.

Property	Type	Description	Required or Optional	Available Version
associatedActions	ConnectApi.AssociatedActionsCapabilityInput	Describes actions added to the feed element.	Optional	33.0
bookmarks	ConnectApi.BookmarksCapabilityInput	Describes bookmarks added to the feed element.	Optional	32.0
canvas	ConnectApi.CanvasCapabilityInput	Describes a canvas app added to the feed element.	Optional	32.0
content	ConnectApi.ContentCapabilityInput	Describes content added to the feed element. **Important:** This class isn't available for feed posts in version 36.0 and later. In version 36.0 and later, use ConnectApi.FilesCapabilityInput.	Optional	32.0–35.0
directMessage	ConnectApi.DirectMessageCapabilityInput	Describes the direct message.	Optional	39.0
extensions	ConnectApi.ExtensionsCapabilityInput	Describes the extensions associated with the feed element.	Optional	40.0
feedEntityShare	ConnectApi.FeedEntityShareCapabilityInput	Describes the feed entity shared with the feed element.	Optional	39.0
files	ConnectApi.FilesCapabilityInput	Describes files attached to the feed element.	Optional	36.0

Property	Type	Description	Required or Optional	Available Version
`link`	`ConnectApi.LinkCapabilityInput`	Describes a link added to the feed element.	Optional	32.0
`poll`	`ConnectApi.PollCapabilityInput`	Describes a poll added to the feed element.	Optional	32.0
`questionAndAnswers`	`ConnectApi.QuestionAndAnswersCapabilityInput`	Describes a question and answer capability added to the feed element.	Optional	32.0
`status`	`ConnectApi.StatusCapabilityInput`	Describes the status of the feed element.	Optional	44.0
`topics`	`ConnectApi.TopicsCapabilityInput`	Describes topics assigned to the feed element.	Optional	38.0

SEE ALSO:

ConnectApi.FeedElementInput

ConnectApi.FeedElementCapabilityInput

A feed element capability.

In API version 30.0 and earlier, most feed items can have comments, likes, topics, and so on. In version 31.0 and later, every feed item (and feed element) can have a unique set of *capabilities*. If a capability property exists on a feed element, that capability is available, even if the capability property doesn't have a value. For example, if the `ChatterLikes` capability property exists on a feed element (with or without a value), the context user can like that feed element. If the capability property doesn't exist, it isn't possible to like that feed element. A capability can also contain associated data. For example, the `Moderation` capability contains data about moderation flags.

This class is abstract and has no public constructor. You can make an instance only of a subclass.

This class is a superclass of:

- ConnectApi.AssociatedActionsCapabilityInput
- ConnectApi.BookmarksCapabilityInput
- ConnectApi.CanvasCapabilityInput
- ConnectApi.ContentCapabilityInput
- ConnectApi.DirectMessageCapabilityInput
- ConnectApi.ExtensionsCapabilityInput
- ConnectApi.FeedEntityShareCapabilityInput
- ConnectApi.FilesCapabilityInput
- ConnectApi.LinkCapabilityInput

- ConnectApi.MuteCapabilityInput
- ConnectApi.PollCapabilityInput
- ConnectApi.QuestionAndAnswersCapabilityInput
- ConnectApi.ReadByCapabilityInput
- ConnectApi.RecordCapabilityInput
- ConnectApi.StatusCapabilityInput
- ConnectApi.TopicsCapabilityInput

ConnectApi.FeedElementInput

Feed elements are the top-level items that a feed contains. Feeds are feed element containers.

This class is abstract and has no public constructor. You can make an instance only of a subclass.

Superclass of ConnectApi.FeedItemInput.

Property	Type	Description	Required or Optional	Available Version
`capabilities`	`ConnectApi.FeedElementCapabilitiesInput`	The capabilities that define auxiliary information on this feed element.	Optional	31.0
`feedElementType`	`ConnectApi.FeedElementType`	The type of feed element this input represents.	Required when creating a feed element Optional when updating a feed element	31.0

Property	Type	Description	Required or Optional	Available Version
`subjectId`	String	The ID of the parent this feed element is being posted to. This value can be the ID of a user, group, or record, or the string `me` to indicate the context user. In version 45.0 and later, you can move a feed element from one public group to another by setting this property to the ID of the new public group. You can't include or change any other properties when moving a feed element.	Required	31.0

SEE ALSO:

- Post a Feed Element with a Mention
- Post a Feed Element with Existing Content
- Post a Feed Element with a New File (Binary) Attachment
- Define an Action Link and Post with a Feed Element
- Define an Action Link in a Template and Post with a Feed Element
- Share a Feed Element (in Version 39.0 and Later)
- Edit a Feed Element
- Edit a Question Title and Post
- Post a Rich-Text Feed Element with Inline Image

ConnectApi.FeedEntityShareCapabilityInput

Share a feed entity with a feed post or comment.

This class is a subclass of ConnectApi.FeedElementCapabilityInput.

Property	Type	Description	Required or Optional	Available Version
`feedEntityId`	String	ID of the feed entity to share with the feed post or comment.	Required	39.0

SEE ALSO:

- ConnectApi.FeedElementCapabilitiesInput

ConnectApi.FeedItemInput

Used to create rich feed items, for example, feed items that include @mentions or files.

Subclass of ConnectApi.FeedElementInput as of version 31.0.

Property	Type	Description	Required or Optional	Available Version
`attachment`	`ConnectApi.FeedItemAttachmentInput`	Specifies the attachment for the feed item. The feed item type is inferred based on the provided attachment. **Important:** As of API version 32.0, use the inherited `capabilities` property.	Optional	28.0–31.0
`body`	`ConnectApi.MessageBodyInput`	Message body. The body can contain up to 10,000 characters and 25 mentions. Because the character limit can change, clients should make a `describeSObjects()` call on the FeedItem or FeedComment object and look at the length of the `Body` or `CommentBody` field to determine the maximum number of allowed characters. If you specify `originalFeedElementId` to share a feed item, use the `body` property to add the first comment to the feed item. To edit this property in a feed item, use `updateFeedElement(communityId, feedElementId, feedElement)`. Editing feed posts is supported in version 34.0 and later.	Required unless the feed item has a link capability or a content capability.	28.0
`isBookmarkedByCurrentUser`	Boolean	Specifies if the new feed item should be bookmarked for the user (`true`) or not (`false`). **Important:** As of API version 32.0, use the `capabilities.bookmarks.isBookmarkedByCurrentUser` property.	Optional	28.0–31.0
`originalFeedElementId`	String	To share a feed element, specify its 18-character ID. **Important:** As of API version 39.0, use the `capabilities.feedEntityShare.feedEntityId` property.	Optional	31.0–38.0
`originalFeedItemId`	String	To share a feed item, specify its 18-character ID. **Important:** In API version 32.0–38.0, use the `originalFeedElementId` property. In API version 39.0 and later, use the `capabilities.feedEntityShare.feedEntityId` property.	Optional	28.0–31.0
`visibility`	`ConnectApi.FeedItemVisibilityType` Enum	Type of users who can see a feed item. • `AllUsers`—Visibility is not limited to internal users. • `InternalUsers`—Visibility is limited to internal users.	Optional	28.0

Property	Type	Description	Required or Optional	Available Version
		Default values: • For external users, the default value is `AllUsers`. External users must use this value to see their posts. • For internal users, the default value is `InternalUsers`. Internal users can accept this value or use the value `AllUsers` to allow external users to see their posts. If the parent of the feed item is a user, group, or direct message, the `visibility` of the feed item must be `AllUsers`.		

ConnectApi.FileIdInput

Attach a file that has already been uploaded or remove a file from a feed element.

Property	Type	Description	Required or Optional	Available Version
`id`	String	ID of a file that has already been uploaded.	Required	36.0
`operationType`	`ConnectApi.OperationType`	Operation to carry out on the file. Values are: • `Add`—Adds the file to the feed element. • `Remove`—Removes the file from the feed element. `Remove` operations are processed before `Add` operations. Adding content that is already added and removing content that is already removed result in no operation.	Optional If not specified, defaults to `Add`.	36.0

SEE ALSO:

ConnectApi.FilesCapabilityInput

ConnectApi.FilesCapabilityInput

Attach up to 10 files that have already been uploaded or remove one or more files from a feed element.

This class is a subclass of ConnectApi.FeedElementCapabilityInput.

Property	Type	Description	Required or Optional	Available Version
`items`	List<`ConnectApi.FileIdInput`>	List of file IDs and operations to be carried out on those files.	Required	36.0

SEE ALSO:

ConnectApi.FeedElementCapabilitiesInput

ConnectApi.SearchFilter

Filter input for object search.

Property	Type	Description	Required or Optional	Available Version
`field`	String	Field to use in the filter.	Optional	63.0
`operator`	`ConnectApi.FilterOperator`	Filter operator. Values are: • `EqOp`—Equal • `ExcludesOp`—Excludes • `GtOp`—Greater than • `GteOp`—Greater than or equal • `InOp`—In • `IncludesOp`—Includes • `LikeOp`—Like • `LtOp`—Less than • `LteOp`—Less than or equal • `NeOp`—Not equal • `NinOp`—Not in	Optional	63.0
`values`	List<`ConnectApi.AbstractList`>	Values of the filter, it can be a List of String, Boolean, Long, or Double. Do not mix data types for filter values, for example, `["A", "B", "C"]` is valid, but `["A", -7, false]` isn't.	Optional	63.0

SEE ALSO:

ConnectApi.SearchRequest

ConnectApi.FindRoutesWithFewestSplitsGroupUsingOCIInputRepresentation

Data used to calculate inventory availability and fulfillment routes for one order involving the fewest number of shipment splits.

Property	Type	Description	Required or Optional	Available Version
excludeLocations	List<String>	List of locations to exclude from the routing calculations.	Optional	55.0
maximumNumberOfSplits	Integer	Maximum allowable number of shipment splits. Routing options that involve more than this number of splits are not returned. Note: Each split represents an additional shipment. Specifying a maximum of 0 returns only locations that can fulfill the entire order in a single shipment. A maximum of 1 returns combinations of locations that can fulfill the order in one or two shipments, and so on.	Required	54.0
orderedItems	[illegible]	Each list element represents a quantity of a product to be routed for fulfillment and the assigned location group or location.	At least one element is required	54.0

SEE ALSO:

findRoutesWithFewestSplitsUsingOCI(findRoutesWithFewestSplitsUsingOCIInput)

ConnectApi.FindRoutesWithFewestSplitsUsingOCIInputRepresentation

ConnectApi.FindRoutesWithFewestSplitsInputRepresentation

Data used to calculate order fulfillment routes involving the fewest number of shipment splits.

Property	Type	Description	Required or Optional	Available Version
locationAvailableInventory	List<ConnectApi.LocationAvailabilityInputRepresentation>	Each list element represents the available quantity of a product at an inventory location.	At least one element is required	51.0
maximumNumberOfSplits	Integer	The maximum allowable number of shipment splits. Routing options that involve more than this number of splits are not returned. Note: Each split represents an additional shipment. Specifying a maximum of 0 returns only locations that can fulfill the entire order in a single shipment. A maximum of 1 returns combinations of locations	Required	51.0

Property	Type	Description	Required or Optional	Available Version
		that can fulfill the order in one or two shipments, and so on.		
orderedQuantities	List<ConnectApi.QuantityWithSkuInputRepresentation>	Each list element represents a quantity of a product to be routed for fulfillment.	At least one element is required	51.0

SEE ALSO:

findRoutesWithFewestSplits(findRoutesWithFewestSplitsInputRepresentation)

ConnectApi.FindRoutesWithFewestSplitsUsingOCIInputRepresentation

Data used to calculate order fulfillment routes involving the fewest number of shipment splits, taking into account inventory availability.

Property	Type	Description	Required or Optional	Available Version
[illegible]	[illegible]	Each list element represents a routing request for one order.	At least one element is required	54.0
ociExpandAttributes	Collection	The string value groupEligibilityExclusion excludes specific locations.	Optional	59.0

SEE ALSO:

findRoutesWithFewestSplitsUsingOCI(findRoutesWithFewestSplitsUsingOCIInput)

ConnectApi.FindRoutesWithFewestSplitsUsingOCIItemInputRepresentation

A quantity of a product and a location group or location assigned to fulfill it.

Property	Type	Description	Required or Optional	Available Version
locationGroupIdentifier	String	The External Reference of the location group or location assigned to the order item. If you specify a location group, inventory is considered for all locations belonging to that group.	Required	54.0
quantity	Double	Quantity of the product.	Required	54.0

Property	Type	Description	Required or Optional	Available Version
`stockKeepingUnit`	String	SKU of the product.	Required	54.0

SEE ALSO:

findRoutesWithFewestSplitsUsingOCI(findRoutesWithFewestSplitsUsingOCIInput)

ConnectApi.FindRoutesWithFewestSplitsUsingOCIInputRepresentation

ConnectApi.FindRoutesWithFewestSplitsGroupUsingOCIInputRepresentation

ConnectApi.FormFieldInput

Marketing integration form field.

Property	Type	Description	Required or Optional	Available Version
`name`	String	Name of the marketing integration form field.	Required	53.0
`type`	`ConnectApi.FormFieldType`	Type of marketing integration form field. Values are: • `Boolean` • `Date` • `EmailAddress` • `Number` • `Text`	Required	53.0

SEE ALSO:

ConnectApi.FormInput

ConnectApi.FormInput

Marketing integration form.

Property	Type	Description	Required or Optional	Available Version
`formFieldsList`	List<`ConnectApi.FormFieldInput`>	Fields for the marketing integration form.	Required	53.0
`formName`	String	Name of the marketing integration form.	Required	53.0
`memberIdentificationCode`	String	The member identification code (MID) of the Marketing Cloud Engagement account associated with the form.	Required	53.0

ConnectApi.FormSubmissionFieldInput

Marketing integration form field submission.

Property	Type	Description	Required or Optional	Available Version
`name`	String	Name of the marketing integration form field.	Required	53.0
`value`	String	Value of the marketing integration form field.	Required	53.0

SEE ALSO:

ConnectApi.FormSubmissionInput

ConnectApi.FormSubmissionInput

Marketing integration form submission.

Property	Type	Description	Required or Optional	Available Version
`formFieldsList`	List<`ConnectApi.FormFieldSubmissionInput`>	Fields for the marketing integration form.	Required	53.0

ConnectApi.FulfillmentGroupInputRepresentation

A list of OrderItemSummaries to be fulfilled together, and the fulfillment location to handle them. The fulfillment type is one of the values defined for the Type field on the FulfillmentOrder object, such as “Warehouse” or “Retail Store.” The specified type is assigned to the FulfillmentOrder for this fulfillment group.

Property	Type	Description	Required or Optional	Available Version
`fulfilledFromLocationId`	String	ID of the fulfillment location.	Required	48.0
`fulfillmentType`	String	Fulfillment type. One of the Type field values defined for FulfillmentOrders.	Required	48.0
`orderItemSummaries`	List<`ConnectApi.OrderItemSummaryInputRepresentation`>	List of OrderItemSummaries.	Required	48.0

Property	Type	Description	Required or Optional	Available Version
`referenceId`	String	Reference to this input for use in troubleshooting failures. This value is only used by the APIs for creating fulfillment orders for multiple order delivery group summaries.	Optional	50.0

SEE ALSO:

ConnectApi.FulfillmentOrderInputRepresentation

createFulfillmentOrders(fulfillmentOrderInput)

ConnectApi.FulfillmentOrderInputRepresentation

An OrderDeliveryGroupSummary that defines a delivery method and recipient, and a list of fulfillment groups to assign to FulfillmentOrders. Each fulfillment group is a set of OrderItemSummaries that match the OrderDeliveryGroupSummary and share the same fulfillment location. The method creates a FulfillmentOrder for each fulfillment group and a FulfillmentOrderLineItem for each OrderItemSummary.

Property	Type	Description	Required or Optional	Available Version
`defaultActivationStatus`	String	Default activation status for a new fulfillment order. If you don't specify a value, the default value is `Allocated`, which belongs to the `Activated` status category. This default can be changed, but the replacement status must also have a status category of `Activated`.	Optional	58.0
`fulfillmentGroups`	List<`ConnectApi.FulfillmentGroupInputRepresentation`>	List of fulfillment groups that specify the OrderItemSummaries and fulfillment locations.	Required	48.0
`orderDeliveryGroupSummaryId`	String	ID of the OrderDeliveryGroupSummary.	Required	48.0
`orderSummaryId`	String	ID of the OrderSummary.	Required	48.0

SEE ALSO:

createFulfillmentOrders(fulfillmentOrderInput)

ConnectApi.FulfillmentOrderInvoiceInputRepresentation

Instantiate and include this object with no properties when creating an invoice.

This input class has no properties.

SEE ALSO:

createInvoice(fulfillmentOrderId, invoiceInput)

ConnectApi.FulfillmentOrderLineItemInputRepresentation

A FulfillmentOrderLineItem and quantity to cancel. You can cancel less than the full quantity, in which case you reallocate the canceled quantity to a different FulfillmentOrder.

Property	Type	Description	Required or Optional	Available Version
`fulfillmentOrderLineItemId`	String	ID of the FulfillmentOrderLineItem.	Required	48.0
`quantity`	Double	Quantity to cancel.	Required	48.0

SEE ALSO:

ConnectApi.FulfillmentOrderLineItemsToCancelInputRepresentation

cancelFulfillmentOrderLineItems(fulfillmentOrderId, cancelFulfillmentOrderLineItemsInput)

ConnectApi.FulfillmentOrderLineItemsToCancelInputRepresentation

A list of FulfillmentOrderLineItems and quantities to cancel.

Property	Type	Description	Required or Optional	Available Version
`fulfillmentOrderLineItemsToCancel`	List<`ConnectApi.FulfillmentOrderLineItemInputRepresentation`>	List of FulfillmentOrderLineItems and quantities.	Required	48.0

SEE ALSO:

cancelFulfillmentOrderLineItems(fulfillmentOrderId, cancelFulfillmentOrderLineItemsInput)

ConnectApi.GetFOCapacityValuesRequestInputRepresentation

Locations to get fulfillment order capacity information for.

Property	Type	Description	Required or Optional	Available Version
`locationIds`	List<String>	List of IDs of the locations to get fulfillment order capacity information for.	Required	55.0

ConnectApi.GroupInformationInput

Chatter group information input.

Property	Type	Description	Available Version
`text`	String	The text in the "Information" section of a group.	28.0
`title`	String	The title of the "Information" section of a group.	28.0

SEE ALSO:

ConnectApi.ChatterGroupInput

ConnectApi.HashtagSegmentInput

Include a hashtag in a feed item or comment.

Subclass of ConnectApi.MessageSegmentInput.

Property	Type	Description	Available Version
`tag`	String	Text of the hash tag without the # (hash tag) prefix Note: Closing square brackets ( ]) are not supported in hash tag text. If the text contains a closing square bracket ( ]), the hash tag ends at the bracket.	28.0

SEE ALSO:

ConnectApi.MessageBodyInput

ConnectApi.HoldFOCapacityInputRepresentation

Request to hold fulfillment order capacity at one or more locations. Can correspond to one action call.

Property	Type	Description	Required or Optional	Available Version
`holdFOCapacityRequests`	List<`ConnectApi.HoldFOCapacityRequestInputRepresentation`>	List of requests to hold fulfillment order capacity at one or more locations.	Required	55.0

ConnectApi.HoldFOCapacityRequestInputRepresentation

Request to hold fulfillment order capacity at one or more locations.

Property	Type	Description	Required or Optional	Available Version
`allOrNothing`	Boolean	Controls whether a single failed request cancels all other requests in the list (`true`) or whether some requests can succeed if others fail (`false`). The default value is `false`.	Optional	55.0
`capacityRequests`	List<`ConnectApi.CapacityRequestInputRepresentation`>	List of requests to hold fulfillment order capacity. Each request is for one fulfillment order at one location.	Required	55.0

ConnectApi.InlineImageSegmentInput

An inline image segment.

Subclass of ConnectApi.MessageSegmentInput.

Property	Type	Description	Required or Optional	Available Version
`altText`	String	Alt text for the inline image.	Optional If not specified, the title of the inline image file is used as the alt text.	35.0
`fileId`	String	ID of the inline image file.	Required	35.0

SEE ALSO:

- Post a Rich-Text Feed Element with Inline Image
- ConnectApi.MessageBodyInput

ConnectApi.InnerEnsureFundsAsyncInputRepresentation

ID of an Invoice and ID of the associated OrderSummary.

Property	Type	Description	Required or Optional	Available Version
`invoiceId`	String	ID of the Invoice to ensure funds for.	Required	56.0
`isConsiderReservedBalanceAmt`	Boolean	If true, the reserved balance amount is used for the Order Summary to fund the invoice. If not enough reserved balance amount, any available balance that isn't reserved by another Order Summary is used. If false, any available balance is used.	Optional	59.0

Property	Type	Description	Required or Optional	Available Version
`orderSummaryId`	String	ID of the OrderSummary associated with the Invoice.	Required	56.0

SEE ALSO:

multipleEnsureFundsAsync(multipleEnsureFundsInput)

ConnectApi.MultipleEnsureFundsAsyncInputRepresentation

ConnectApi.InviteInput

An invitation.

Property	Type	Description	Required or Optional	Available Version
`invitees`	List<`String`>	List of email addresses to send the invitation to.	Required	39.0
`message`	String	Message to include in the invitation.	Optional	39.0

ConnectApi.InvoiceToPayInputRepresentation

Invoice for a fee.

Property	Type	Description	Required or Optional	Available Version
`invoiceId`	String	ID of the invoice for a fee.	Required	56.0

SEE ALSO:

ensureRefundsAsync(orderSummaryId, ensureRefundsInput)

ConnectApi.EnsureRefundsAsyncInputRepresentation

ConnectApi.LeadInput

Contains information about a lead or guest user.

Property	Type	Description	Required or Optional	Available Version
`company`	String	The company of the lead.	Optional	53.0
`email`	String	The email address of the lead.	Optional	53.0
`extendedFields`	List<ConnectApi.ExtendedFieldInput>	Use to add values to any of the fields, including custom fields.	Optional	53.0

Property	Type	Description	Required or Optional	Available Version
`firstName`	String	The first name of the lead.	Optional	53.0
`lastName`	String	The last name of the lead.	Optional	53.0
`phone`	String	The phone number of the lead.	Optional	53.0

ConnectApi.LinkCapabilityInput

Create or update a link on a feed element.

This class is a subclass of ConnectApi.FeedElementCapabilityInput.

Property	Type	Description	Required or Optional	Available Version
`url`	String	Link URL. The URL can be to an external site.	Required	32.0
`urlName`	String	Description of the link.	Optional	32.0

SEE ALSO:

ConnectApi.FeedElementCapabilitiesInput

ConnectApi.LinkSegmentInput

Include a link segment in a feed item or comment.

Subclass of ConnectApi.MessageSegmentInput.

Property	Type	Description	Available Version
`url`	String	URL to be used for the link	28.0

SEE ALSO:

ConnectApi.MessageBodyInput

ConnectApi.LocationAvailabilityInputRepresentation

The available quantity of a product at an inventory location.

Property	Type	Description	Required or Optional	Available Version
`externalReferenceId`	String	The external reference ID of the inventory location.	Optional	51.0
`quantity`	Double	The available quantity of the product.	Required	51.0
`stockKeepingUnit`	String	The Stock Keeping Unit of the product.	Required	51.0

ConnectApi.LocationInputRepresentation

Inventory location data used to calculate shipping distance.

Property	Type	Description	Required or Optional	Available Version
`countryCode`	String	The country code of the location.	Required	51.0
`locationIdentifier`	String	The identifier of the location.	Required	51.0
`postalCode`	String	The postal code of the location.	Required	51.0

ConnectApi.LongList

List of long values.

Subclass of ConnectApi.AbstractList.

Property	Type	Description	Required or Optional	Available Version
`values`	List<Long>	List of Long values to filter on.	Optional	63.0

ConnectApi.ManagedContentBodyInput

Input representation for the body of a piece of managed content.

Property	Type	Description	Required or Optional	Available Version
`nodeMap`	`Object`	Body of the managed content version. The format must be Map<String, Object> , where map values are either primitive values like String, Integer, Double, Boolean, or another Map<String, Object>.	Required	60.0

SEE ALSO:

- ConnectApi.ManagedContentDocumentInput
- ConnectApi.ManagedContentVariantUpdateInput

ConnectApi.ManagedContentChannelCreateRepresentation

Input class to create a managed content channel.

Property	Type	Description	Required or Optional	Available Version
`cacheControlMaxAge`	Long	Cache control max age value in seconds.	Optional	62.0

Property	Type	Description	Required or Optional	Available Version
`domain`	String	ID or name of the domain assigned to the public channel.	Optional	62.0
`isDedicatedContentDelivery`	Boolean	Specifies whether the channel has off-core dedicated content delivery enabled (`true`) or not (`false`). Orgs hosted on Hyperforce use off-core dedicated content delivery to deliver content in public channels with high performance and low latency.	Optional	63.0
`isDomainLocked`	Boolean	Specifies whether the domain is locked and can't be changed (`true`) or not (`false`).	Optional	62.0
`isSearchable`	Boolean	Specifies whether the text contents of the channel are searchable (`true`) or not (`false`).	Optional	62.0
`mediaCacheControlMaxAge`	Long	Media cache control max age value in seconds.	Optional	62.0
`name`	String	Name of the managed content channel.	Required	62.0
`targetId`	String	ID of the target associated with the managed content channel.	Required for all channel types except `PublicUnauthenticated`	62.0
`type`	`ConnectApi.ManagedContentChannelType`	Type of managed content channel. Values are: • `CloudToCloud`—Cloud-to-Cloud integrated channel. • `Community`—Experience Cloud site channel. • `ConnectedApp`—Channel served by a connected app. • `PublicUnauthenticated`—Public channel. All published content is publicly available. • `UserPermission`—Channel backed by a system permission. All published content is available only to users with the permission.	Required	62.0

SEE ALSO:

postManagedContentChannel(ManagedContentCreateInputParam)

ConnectApi.ManagedContentChannelUpdateRepresentation

Input class to update a managed content channel.

Property	Type	Description	Required or Optional	Available Version
`cacheControlMaxAge`	Long	Cache control max age value in seconds.	Optional	62.0
`domain`	String	ID or name of the domain assigned to the public channel.	Optional	62.0
`isDedicatedContentDelivery`	Boolean	Specifies whether the channel has off-core dedicated content delivery enabled (`true`) or not (`false`). Orgs hosted on Hyperforce use off-core dedicated content delivery to deliver content in public channels with high performance and low latency.	Optional	63.0
`isDomainLocked`	Boolean	Specifies whether the domain is locked and can't be changed (`true`) or not (`false`).	Optional	62.0
`isSearchable`	Boolean	Specifies whether the text contents of the channel are searchable (`true`) or not (`false`).	Optional	62.0
`mediaCacheControlMaxAge`	Long	Media cache control max age value in seconds.	Optional	62.0
`name`	String	Name of the managed content channel.	Required	62.0
`targetId`	String	ID of the target associated with the managed content channel.	Required for all channel types except `Public Unauthenticated`	62.0

SEE ALSO:

patchManagedContentChannel(channelId, ManagedContentChannelInput)

ConnectApi.ManagedContentDocumentCloneInput

Managed content clone input.

Property	Type	Description	Required or Optional	Available Version
`apiName`	String	API name of the cloned content.	Optional	61.0
`contentSpaceOrFolderId`	String	ID of the target folder for the cloned content. If unspecified, defaults to the folder of the source content.	Optional	61.0

Property	Type	Description	Required or Optional	Available Version
`includeVariants`	Boolean	Specifies whether to include variants (`true`) or not (`false`) when cloning the content. If unspecified, default is `false`.	Optional	61.0
`title`	String	Title of the cloned content. If unspecified, "clone of" is appended to the source content's title.	Optional	61.0

SEE ALSO:

cloneManagedContentDocument(contentKeyOrId, ManagedContentCloneInputParam)

ConnectApi.ManagedContentDocumentInput

Input representation for a piece of managed content.

Property	Type	Description	Required or Optional	Available Version
`apiName`	String	API name of the managed content.	Optional	61.0
`contentBody`	`ConnectApi.ManagedContentBodyInput`	Body of the managed content.	Required	60.0
`contentKey`	String	Content key to assign to the managed content. A content key is a universally unique identifier (UUID) such as MCA4CCV5QS2BAB5H7YRCRPTCWGZQ.	Optional	60.0
`contentSpaceOrFolderId`	String	Content space or folder ID where the content is created.	Required	60.0
`contentType`	String	Fully qualified name of the content type to create. If you're uploading a binary file using a multipart/form-data message, `contentType` must be `sfdc_cms__image` or `sfdc_cms__doc`. You can't create a Form using `sfdc_cms__form`. The `sfdc_cms__form` content type isn't supported.	Required	60.0
`externalId`	String	External ID of the managed content.	Optional	60.0
`title`	String	Title of the managed content.	Required	60.0

Property	Type	Description	Required or Optional	Available Version
`urlName`	String	URL name of the managed content within the org.	Optional	60.0

ConnectApi.ManagedContentPublishInput

Input for publishing content.

Property	Type	Description	Required or Optional	Available Version
`contentIds`	List<String>	IDs of content to publish. All variants of the content are published.	Required if `variantIds` isn't specified	60.0
`contextContentSpaceId`	String	ID of the context workspace. If specified, content from other workspaces is published if it is shared to the specified workspace. If unspecified, the context workspace is derived from the content's origin workspace. All content in the request should belong to the same origin workspace.	Optional	61.0
`description`	String	Description for publish action.	Optional	60.0
`includeContentReferences`	Boolean	Specifies whether to include content references (`true`) or not (`false`).	Optional	60.0
`variantIds`	List<String>	IDs of variants to publish. All variants must be from the same content space.	Required if `contentIds` isn't specified	60.0

ConnectApi.ManagedContentSpaceChannelInputRepresentation

Channel to add or remove from a managed content space.

Property	Type	Description	Required or Optional	Available Version
`channelId`	String	ID of the channel to add or remove from the managed content space.	Required	62.0

Property	Type	Description	Required or Optional	Available Version
`operation`	`ConnectApi.ManagedContentSpaceChannelOperation` on page 2369	Operation to perform on the channel and managed content space. • `Add`—Add a channel to a managed content space. • `Remove`—Remove a channel from a managed content space.	Required	62.0

SEE ALSO:

ConnectApi.ManagedContentSpaceChannelsInputRepresentation

ConnectApi.ManagedContentSpaceChannelsInputRepresentation

Channels to add or remove from the managed content space.

Property	Type	Description	Required or Optional	Available Version
`spaceChannels`	List<`ConnectApi.ManagedContentSpaceChannelInputRepresentation`>	List of channels to add or remove from the managed content space.	Required	62.0

SEE ALSO:

patchManagedContentSpaceChannels(contentSpaceId, spaceChannels)

ConnectApi.ManagedContentUnpublishInput

Input for unpublishing content.

Property	Type	Description	Required or Optional	Available Version
`contentIds`	List<String>	IDs of content to unpublish. All variants of the content are unpublished.	Required if `variantIds` isn't specified	60.0
`contextContentSpaceId`	String	ID of the context workspace. If specified, content from other workspaces is unpublished if it is shared to the specified workspace. If unspecified, the context workspace is derived from the content's origin workspace. All content in the request	Optional	61.0

Property	Type	Description	Required or Optional	Available Version
		should belong to the same origin workspace.		
`description`	String	Description for unpublish action.	Optional	60.0
`variantIds`	List<String>	IDs of variants to unpublish. All variants must be from the same content space.	Required if `contentIds` isn't specified	60.0

ConnectApi.ManagedContentVariantUpdateInput

Input representation for replacing a managed content variant.

Property	Type	Description	Required or Optional	Available Version
`apiName`	String	API name of the managed content variant.	Optional	63.0
`contentBody`	`ConnectApi.ManagedContentBodyInput`	Body of the managed content variant.	Optional	60.0
`title`	String	Title of the managed content variant.	Optional	60.0
`urlName`	String	URL name of the managed content variant within the org.	Optional	60.0

ConnectApi.ManagedTopicPositionCollectionInput

A collection of relative positions of managed topics.

Property	Type	Description	Required or Optional	Available Version
`managedTopicPositions`	List<`ConnectApi.ManagedTopicPositionInput`>	List of relative positions of managed topics. This list can include `Featured` and `Navigational` managed topics and doesn't need to include all managed topics. For more information about reordering managed topics, see the example in reorderManagedTopics(communityId, managedTopicPositionCollection).	Required	32.0

ConnectApi.ManagedTopicPositionInput

Relative position of a managed topic.

Property	Type	Description	Required or Optional	Available Version
`managedTopicId`	String	ID of existing managed topic.	Required	32.0
`position`	Integer	Relative position of the managed topic, indicated by zero-indexed, ascending whole numbers.	Required	32.0

SEE ALSO:

ConnectApi.ManagedTopicPositionCollectionInput

ConnectApi.MarkupBeginSegmentInput

The beginning tag for rich text markup.

Subclass of ConnectApi.MessageSegmentInput.

Property	Type	Description	Required or Optional	Available Version
`altText`	String	Alternative text for the `Hyperlink` segment.	Optional	45.0
`markupType`	`ConnectApi.MarkupType`	Type of rich text markup. • `Bold`—Bold tag. • `Code`—Code tag. • `Hyperlink`—Hyperlink anchor tag. • `Italic`—Italic tag. • `ListItem`—List item tag. • `OrderedList`—Ordered list tag. • `Paragraph`—Paragraph tag. • `Strikethrough`—Strikethrough tag. • `Underline`—Underline tag. • `UnorderedList`—Unordered list tag. Markup segments with a `markupType` of `Code` can include only text segments.	Required	35.0

Property	Type	Description	Required or Optional	Available Version
`url`	String	URL for the `Hyperlink` segment. Supported hyperlink URLs start with `http://` or `https://`.	Required for `Hyperlink`	45.0

SEE ALSO:

Post a Rich-Text Feed Element with Inline Image

ConnectApi.MessageBodyInput

ConnectApi.MarkupEndSegmentInput

The end tag for rich text markup.

Subclass of ConnectApi.MessageSegmentInput

Property	Type	Description	Required or Optional	Available Version
`markupType`	`ConnectApi.MarkupType`	Type of rich text markup. • `Bold`—Bold tag. • `Code`—Code tag. • `Hyperlink`—Hyperlink anchor tag. • `Italic`—Italic tag. • `ListItem`—List item tag. • `OrderedList`—Ordered list tag. • `Paragraph`—Paragraph tag. • `Strikethrough`—Strikethrough tag. • `Underline`—Underline tag. • `UnorderedList`—Unordered list tag.	Required	35.0

SEE ALSO:

Post a Rich-Text Feed Element with Inline Image

ConnectApi.MessageBodyInput

ConnectApi.MCSFolderShareInput

Target to share a managed content space folder with.

Property	Type	Description	Required or Optional	Available Version
`targetId`	String	ID of the target to share the managed content space folder with. Supported target IDs are the root folder IDs of workspaces. To get the root folder ID for a space, use the `getManagedContentSpace(contentSpaceId)` method.	Optional	63.0

SEE ALSO:

ConnectApi.MCSFolderShareCollectionUpdateInput

ConnectApi.MCSFolderShareCollectionUpdateInput

Update the targets that a managed content space folder is shared with.

Property	Type	Description	Required or Optional	Available Version
`shareWith`	List<`ConnectApi.MCSFolderShareInput`>	List of target IDs to share the managed content space folder with. Supported target IDs are the root folder IDs of workspaces. To get the root folder ID for a space, use the `getManagedContentSpace(contentSpaceId)` method.	Required if `unshareWith` isn't specified	63.0
`unshareWith`	List<String>	Comma-separated list of target IDs to unshare the managed content space folder with.	Required if `shareWith` isn't specified	63.0

SEE ALSO:

getMCSFolderShares(folderId)

ConnectApi.MentionSegmentInput

Include an @mention of a user or group in a feed post or comment. When creating a feed post or comment, you can include up to 25 mentions.

Subclass of ConnectApi.MessageSegmentInput.

Property	Type	Description	Available Version
`id`	String	ID of the user or group to mention. To mention a user, use either `id` or `username`. You can't include both. To mention a group, you must use `id`.	28.0 Groups are available in 29.0.
`username`	String	User name of the user to mention. To mention a user, use either `id` or `username`. You can't include both.	38.0

SEE ALSO:

ConnectApi.MessageBodyInput

ConnectApi.MessageBodyInput

Add rich messages to feed items and comments.

Property	Type	Description	Available Version
`messageSegments`	List<`ConnectApi.MessageSegmentInput`>	List of message segments contained in the body	28.0

SEE ALSO:

ConnectApi.FeedItemInput

ConnectApi.CommentInput

ConnectApi.AnnouncementInput

ConnectApi.MessageSegmentInput

Used to add rich message segments to feed items and comments.

This class is abstract and has no public constructor. You can make an instance only of a subclass.

Superclass for:

- ConnectApi.EntityLinkSegmentInput
- ConnectApi.HashtagSegmentInput
- ConnectApi.InlineImageSegmentInput
- ConnectApi.LinkSegmentInput
- ConnectApi.MarkupBeginSegmentInput
- ConnectApi.MarkupEndSegmentInput
- ConnectApi.MentionSegmentInput
- ConnectApi.TextSegmentInput

Use the ConnectApiHelper repository on GitHub to simplify many of the tasks accomplished with ConnectApi.MessageSegmentInput, such as posting with inline images, rich text, and mentions.

Property	Type	Description	Required or Optional	Available Version
`type`	`ConnectApi.MessageSegmentType`	The type of message segment. Values are: • `EntityLink` • `FieldChange` • `FieldChangeName` • `FieldChangeValue` • `Hashtag` • `InlineImage` • `Link` • `MarkupBegin` • `MarkupEnd` • `Mention` • `MoreChanges` • `ResourceLink` • `Text`	Required	23.0

SEE ALSO:

- Edit a Comment
- Edit a Feed Element
- Edit a Question Title and Post
- Post a Rich-Text Feed Element with Inline Image
- ConnectApi.MessageBodyInput

ConnectApi.MultipleEnsureFundsAsyncInputRepresentation

List of Invoices and the associated OrderSummaries.

Property	Type	Description	Required or Optional	Available Version
`asyncInputs`	List<`ConnectApi.InnerEnsureFundsAsyncInputRepresentation`>	List of Invoices to ensure funds for and the associated OrderSummaries.	Required	56.0

SEE ALSO:

- multipleEnsureFundsAsync(multipleEnsureFundsInput)

ConnectApi.MultipleFulfillmentOrderInputRepresentation

List of inputs for creating fulfillment orders.

Property	Type	Description	Required or Optional	Available Version
`fulfillmentOrders`	List<`ConnectApi.FulfillmentOrderInputRepresentation`>	Each element contains the data to create one fulfillment order.	Required	50.0

ConnectApi.MultipleFulfillmentOrderInvoicesInputRepresentation

The FulfillmentOrders to create Invoices for.

Property	Type	Description	Required or Optional	Available Version
`fulfillmentOrderIds`	List<String>	List of IDs of FulfillmentOrders to create Invoices for.	At least one ID is required.	52.0

ConnectApi.MuteCapabilityInput

Mute or unmute a feed element.

This class is a subclass of ConnectApi.FeedElementCapabilityInput.

Property	Type	Description	Required or Optional	Available Version
`isMutedByMe`	Boolean	Indicates whether the feed element is muted for the context user. Default value is `false`.	Required	35.0

SEE ALSO:

setIsMutedByMe(communityId, feedElementId, isMutedByMe)

ConnectApi.NamedCredentialCalloutOptionsInput

Named credential callout options input.

Property	Type	Description	Required or Optional	Available Version
`allowMergeFieldsInBody`	Boolean	Specifies whether to allow merge fields in the HTTP body (`true`) or not (`false`).	Required	58.0
`allowMergeFieldsInHeader`	Boolean	Specifies whether to allow merge fields in the HTTP header (`true`) or not (`false`).	Required	58.0

Property	Type	Description	Required or Optional	Available Version
`generateAuthorizationHeader`	Boolean	Specifies whether to generate an authorization header (`true`) or not (`false`).	Required	58.0

SEE ALSO:

ConnectApi.NamedCredentialInput

ConnectApi.NamedCredentialInput

Input used to create or update a named credential.

Property	Type	Description	Required or Optional	Available Version
`calloutOptions`	`ConnectApi.NamedCredentialCalloutOptionsInput`	Callout options.	Required	58.0
`calloutUrl`	String	URL of the named credential in a callout.	Required	58.0
`customHeaders`	List<`ConnectApi.CredentialCustomHeaderInput`>	Custom HTTP headers.	Optional	58.0
`developerName`	String	Named credential developer name.	Required for creating a named credential Optional for updating a named credential	58.0
`externalCredentials`	List<`ConnectApi.ExternalCredentialInput`>	External credentials used by the named credential. In version 58.0 and later only one external credential is supported.	Required	58.0
`masterLabel`	String	Named credential label.	Required	58.0
`networkConnection`	`ConnectApi.NetworkConnectionInput`	PrivateConnect outbound network connection.	Optional depending on `type`	58.0
`parameters`	List<`ConnectApi.NamedCredentialParameterInput`>	Named credential parameters.	Optional	58.0
`type`	`ConnectApi.NamedCredentialType`	Type of named credential. Values are: • `PrivateEndpoint` • `SecuredEndpoint`	Required	58.0

ConnectApi.NamedCredentialParameterInput

Named credential parameter input.

Property	Type	Description	Required or Optional	Available Version
`id`	String	ID of the parameter.	Optional	58.0
`parameterDescription`	String	Description of the parameter.	Optional	58.0
`parameterName`	String	Name of the parameter.	Required	58.0
`parameterType`	`ConnectApi.NamedCredentialParameterType`	Type of named credential parameter. Values are: • `AllowedManagedPackageNamespaces` • `ClientCertificate`	Required	58.0
`parameterValue`	String	Value of the parameter.	Required	58.0

SEE ALSO:

ConnectApi.NamedCredentialInput

ConnectApi.NBAStrategyInput

A recommendation strategy.

Property	Type	Description	Required or Optional	Available Version
`contextRecordId`	String	ID of the context record. For example, if the next best action is on a case detail page, the ID of the case.	Optional	45.0
`maxResults`	Integer	Maximum number of results. Valid values are from 1 to 25. The default is 3.	Optional	45.0
`strategyContext`	Map<String, String>	Variable and value mappings for the strategy.	Optional	45.0
`debugTrace`	Boolean	Specifies whether to return trace and debug information in the response (`true`) or not (`false`). If unspecified, the default is `false`.	Optional	45.0

ConnectApi.NetworkConnectionInput

Network connection input.

Property	Type	Description	Required or Optional	Available Version
`developerName`	String	Developer name of the network connection.	Required	58.0
`namespace`	String	Namespace of the network connection.	Optional	58.0

SEE ALSO:

ConnectApi.NamedCredentialInput

ConnectApi.NewUserAudienceCriteriaInput

Criteria for the new members type of custom recommendation audience.

Subclass of ConnectApi.AudienceCriteriaInput.

Property	Type	Description	Required or Optional	Available Version
`value`	Double	The maximum number of days since a user became a site member. For example, if you specify *30*, anyone who became a site member in the last 30 days is included in the new members audience.	Required	36.0

ConnectApi.OAuthCredentialAuthUrlInput

OAuth authentication flow.

Property	Type	Description	Required or Optional	Available Version
`externalCredential`	String	Fully qualified developer name of the external credential.	Required	56.0
`principalName`	String	Name of the external credential named principal.	Required if `principalType` is `NamedPrincipal`	56.0
`principalType`	`ConnectApi.CredentialPrincipalType`	Type of credential principal. Values are: • `AwsStsPrincipal` • `NamedPrincipal` • `PerUserPrincipal`	Required	56.0
`returnUrl`	String	Return URL to apply to the authentication URL.	Optional	56.0

ConnectApi.OCICreateReservationInputRepresentation

Data to reserve inventory at one or more Omnichannel Inventory locations or location groups.

Property	Type	Description	Required or Optional	Available Version
`actionRequestId`	String	A UUID that identifies the request. Use the action request IDs in response data to identify which requests succeeded or failed. If the `OmnichannelInventoryService` class's `createReservation` method is called, the actionRequestId is used for the reservationId.	Required	51.0
`allowPartialReservations`	Boolean	When true, if the system can't create the entire reservation, then it attempts to create a partial reservation.	Optional	51.0
`createRecords`	List<`ConnectApi.OCICreateReservationSingleInputRepresentation`>	A list of product quantities and locations or location groups. The list can include up to 100 elements.	At least one element is required	51.0
`expirationSeconds`	Integer	A length of time in seconds. If the reservation isn't fulfilled within this amount of time after the `reservationTime`, then it expires. The maximum value is 14400.	Optional	51.0
`externalRefId`	String	External reference ID.	Optional	51.0
`reservationTime`	String	The time at which to record the reservation. Example: 2020-07-24T21:13:00Z	Optional	51.0

ConnectApi.OCICreateReservationSingleInputRepresentation

A quantity of a product and an Omnichannel Inventory location or location group at which to reserve it.

Property	Type	Description	Required or Optional	Available Version
`locationGroupIdentifier`	String	Identifier of the location group at which to reserve inventory.	Either locationGroupIdentifier or locationIdentifier is required, but not both	51.0
`locationIdentifier`	String	Identifier of the location at which to reserve inventory.	Either locationGroupIdentifier or locationIdentifier	51.0

Property	Type	Description	Required or Optional	Available Version
			is required, but not both	
quantity	Double	The quantity of the product to reserve.	Required	51.0
stockKeepingUnit	String	The SKU of the product to reserve.	Required	51.0

ConnectApi.OCIFulfillReservationInputRepresentation

A list of inventory reservations to fulfill.

Property	Type	Description	Required or Optional	Available Version
fulfillmentRecords	List<ConnectApi.OCIFulfillReservationSingleInputRepresentation>	A list of inventory reservations. The list can include up to 100 elements.	At least one element is required.	51.0
reservationId	String	The ID of the inventory reservation.	Optional	58.0

ConnectApi.OCIFulfillReservationSingleInputRepresentation

An inventory reservation to fulfill.

Property	Type	Description	Required or Optional	Available Version
actionRequestId	String	A UUID that identifies the request. Use the action request IDs in response data to identify which requests succeeded or failed.	Required	51.0
externalRefId	String	The external reference ID of the location that's fulfilling the reservation.	Optional	51.0
locationIdentifier	String	The identifier of the location that's fulfilling the reservation.	Required	51.0
quantity	Double	The quantity being fulfilled.	Required	51.0
reservationId	String	The ID of the inventory reservation.	Optional	58.0
stockKeepingUnit	String	The SKU of the product being fulfilled.	Required	51.0

ConnectApi.OCIGetInventoryAvailabilityInputRepresentation

Details of a request to retrieve inventory availability.

Property	Type	Description	Required or Optional	Available Version
`locationGroupIdentifier`	String	The External Reference of a location group to retrieve inventory availability data for. Specifying this value retrieves inventory data for all locations belonging to this group.	Optional; can't combine with `locationGroupIdentifiers` or `locationIdentifiers`	51.0
`locationGroupIdentifiers`	List<String>	A list of up to 100 External References of location groups to retrieve inventory availability data for.	Optional; can't combine with `locationGroupIdentifier` or `locationIdentifiers`	51.0
`locationIdentifiers`	List<String>	A list of up to 100 External References of locations to retrieve inventory availability data for.	Optional; can't combine with `locationGroupIdentifier` or `locationGroupIdentifiers`	51.0
`stockKeepingUnit`	String	The SKU of a product to retrieve inventory availability data for. Specifying a SKU with no locations or location groups returns availability data for that SKU at all inventory locations that aren't assigned to location groups.	Optional; can't combine with `stockKeepingUnits`	51.0
`stockKeepingUnits`	List<String>	A list of up to 100 SKUs of products to retrieve inventory availability data for.	Optional; can't combine with `stockKeepingUnit`	51.0
`useCache`	Boolean	Whether to fetch the inventory data from the cache. The default value is True..	Optional	51.0

ConnectApi.OCIReleaseReservationInputRepresentation

Details of one or more inventory reservations to release.

Property	Type	Description	Required or Optional	Available Version
`releaseRecords`	List<`ConnectApi.OCIReleaseReservationSingleInputRepresentation`>	List of inventory reservations to release. The list can include up to 100 elements.	At least one element is required.	51.0
`reservationId`	String	The ID of the inventory reservation.	Optional	58.0

ConnectApi.OCIReleaseReservationSingleInputRepresentation

A single inventory reservation to release.

Property	Type	Description	Required or Optional	Available Version
`actionRequestId`	String	A UUID that identifies the request. Use the action request IDs in response data to identify which requests succeeded or failed.	Required	51.0
`externalRefId`	String	The external reference ID of the location or location group that has the reservation.	Optional	51.0
`locationGroupIdentifier`	String	The identifier of the location group that has the reservation.	The identifier for a location or location group, but not both, is required.	51.0
`locationIdentifier`	String	The identifier of the location that has the reservation.	The identifier for a location or location group, but not both, is required.	51.0
`quantity`	Double	The quantity of reserved inventory to release.	Required	51.0
`reservationId`	String	The ID of the inventory reservation.	Optional	58.0
`stockKeepingUnit`	String	The SKU of the product to release.	Required	51.0

ConnectApi.OCITransferReservationInputRepresentation

A list of inventory reservation transfers and specifies whether a single failure cancels the entire list.

Property	Type	Description	Required or Optional	Available Version
`allOrNothing TransferId`	String	Controls whether a single failed transfer cancels all other transfers in the `transferRecords` list. • To allow some transfers in the `transferRecords` list to succeed when others fail, don't include this property. • To cancel all the transfers in the `transferRecords` list when any of them fail, set this property to a UUID. The ID must be unique, but isn't otherwise used in this version.	Optional	51.0
`reservationId`	String	The ID of the inventory reservation.	Optional	58.0

Property	Type	Description	Required or Optional	Available Version
transferRecords	List<ConnectApi.OCITransferReservationSingleInputRepresentation>	A list of inventory reservation transfers. The list can include up to 100 elements.	At least one element is required.	51.0

ConnectApi.OCITransferReservationSingleInputRepresentation

An inventory reservation transfer.

Property	Type	Description	Required or Optional	Available Version
actionRequestId	String	A UUID that identifies the request. Use the action request IDs in response data to identify which requests succeeded or failed.	Required	51.0
externalRefId	String	The external reference ID of the location receiving the transfer.	Optional	51.0
fromLocationGroupIdentifier	String	The identifier of the location group sending the reservation.	The identifier for a sending location or location group, but not both, is required	51.0
fromLocationIdentifier	String	The identifier of the location sending the reservation.	The identifier for a sending location or location group, but not both, is required	51.0
ignoreAvailabilityCheck	Boolean	If true, force the transfer even if the receiving location doesn't have sufficient available inventory. The default value is false.	Optional	52.0
quantity	Double	The quantity of inventory being transferred.	Required	51.0
reservationId	String	The ID of the inventory reservation.	Optional	58.0
stockKeepingUnit	String	The SKU of the product being transferred.	Required	51.0
toLocationGroupIdentifier	String	The identifier of the location group receiving the reservation.	The identifier for a receiving location or location group, but not both, is required	51.0
toLocationIdentifier	String	The identifier of the location receiving the reservation.	The identifier for a receiving location or location group, but not both, is required	51.0

ConnectApi.OCIUpdateReservationInputRepresentation

Data to update one or more Omnichannel Inventory item reservations.

Property	Type	Description	Required or Optional	Available Version
actionRequestId	String	Unique and idempotent action request ID. Use in response data to identify which requests succeeded or failed.	Required	61.0
allowPartialReservations	Boolean	When true, if the system can't update the entire reservation, then it attempts to update a partial reservation.	Optional	61.0
externalRefId	String	External reference ID.	Optional	61.0
reservationId	String	The ID of the inventory reservation.	Optional	61.0
reservationTime	String	The time the reservation was updated. Example: 2020-07-24T21:13:00Z.	Optional	61.0
updateAllOrNothingRecords	List	Controls whether a single failed request updates all other requests in the list (true) or whether some requests can succeed if others fail (false). The default value is false.	Optional	61.0
updateRecords	List<ConnectApi.OCIUpdateReservationSingleInputRepresentation> on page 1901 []	A list of product quantities and locations or location groups. The list can have up to 100 elements.	At least one element is required	61.0

ConnectApi.OCIUpdateReservationSingleInputRepresentation

Data to update one Omnichannel Inventory reservation item.

Property	Type	Description	Required or Optional	Available Version
locationGroupIdentifier	String	Identifier of the location group where the inventory is reserved.	Either locationGroupIdentifier or locationIdentifier are required, but not both	61.0
locationIdentifier	String	Identifier of the location where the inventory is reserved.	Either locationGroupIdentifier or locationIdentifier are required, but not both	61.0
quantity	Double	The quantity of the product to update.	Required	61.0
stockKeepingUnit	String	The SKU of the product to update.	Required	61.0

ConnectApi.SearchOrderBy

Order by parameter for object search.

Property	Type	Description	Required or Optional	Available Version
`field`	String	Field to sort the results by.	Optional	63.0
`order`	`ConnectApi.OrderDirection`	Order direction. Values are: • `Ascending` • `Descending`	Optional	63.0
`orderNulls`	`ConnectApi.OrderNulls`	Null value order. Values are: • `Firsts`—Null values are sorted first. • `Lasts`—Null values are sorted last.	Optional	63.0

SEE ALSO:

ConnectApi.SearchRequest

ConnectApi.OrderItemSummaryInputRepresentation

An OrderItemSummary and quantity.

Property	Type	Description	Required or Optional	Available Version
`orderItemSummaryId`	String	ID of the OrderItemSummary.	Required	48.0
`quantity`	Double	Quantity to include. Can't contain a fraction or a decimal.	Required	48.0

SEE ALSO:

ConnectApi.FulfillmentGroupInputRepresentation

ConnectApi.FulfillmentOrderInputRepresentation

createFulfillmentOrders(fulfillmentOrderInput)

ConnectApi.OrderItemSummaryAdjustmentCollectionInput

Collection of order item summaries.

Property	Type	Description	Required or Optional	Available Version
`orderItemSummaries`	List<`ConnectApi.OrderItemSummaryAdjustmentInput`>	List of order item summaries.	Required	53.0

ConnectApi.OrderItemSummaryAdjustmentInput

Order item summary.

Property	Type	Description	Required or Optional	Available Version
`orderItemSummaryId`	String	ID of the order item summary.	Required	53.0

SEE ALSO:

ConnectApi.OrderItemSummaryAdjustmentCollectionInput

ConnectApi.OrderSummaryAdjustmentAggregatesAsyncInput

Order summary IDs for calculating adjustment aggregates.

Property	Type	Description	Required or Optional	Available Version
`orderSummaryIds`	List< String>	List of order summary IDs.	Required	55.0

ConnectApi.OrderSummaryInputRepresentation

An order from which to create an OrderSummary. Optionally, you can specify OrderSummary-specific information such as its Status and whether it is managed in Salesforce Order Management.

Property	Type	Description	Required or Optional	Available Version
`businessModel`	String	The order's business model. It can have one of these values: • B2B • B2C	Optional	53.0
`externalReferenceIdentifier`	String	Used internally to prevent duplicate records. This value is case-sensitive.	Optional	56.0
`name`	String	Specifies an OrderNumber to assign to the order summary.	Optional	50.0

Property	Type	Description	Required or Optional	Available Version
`orderId`	String	ID of the original order.	Required	48.0
`orderLifeCycleType`	String	Specifies whether the order is managed in Salesforce Order Management or by an external system. It can have one of these values: • `MANAGED`—Managed in Salesforce Order Management. • `UNMANAGED`—Managed by an external system. If no value is specified, the default is `MANAGED`.	Optional	49.0
`sourceProcess`	String	Describes the order process creating the OrderSummary. It can have one of these values: • `Exchange`—An Exchange process. • `OrderOnBehalf`—An Order on Behalf Of process. • `Standard`—Any process other than Exchange or Order on Behalf Of. If no value is specified, the default is `Standard`.	Optional	57.0
`status`	String	Specifies a status to assign to the order summary. The value must match one of the picklist values on the Status field of the OrderSummary object.	Optional	50.0

SEE ALSO:

createOrderSummary(orderSummaryInput)

ConnectApi.OrderSummaryLookupInput

Order summary lookup input.

Property	Type	Description	Required or Optional	Available Version
`orderSummaryIdOrRefNumber`	String	Either the order summary ID or reference number value.	Required	58.0

Property	Type	Description	Required or Optional	Available Version
`verification`	`ConnectApi.OrderSummaryVerificationInput`	Verification attributes for guest shoppers.	Optional	58.0

ConnectApi.OrderSummaryVerificationInput

Order summary verification input.

Property	Type	Description	Required or Optional	Available Version
`email`	String	Guest shopper or registered buyer's email address.	Optional	58.0
`lastName`	String	Guest shopper or registered buyer's last name.	Optional	58.0
`phoneNumber`	String	Guest shopper or registered buyer's phone number.	Optional	58.0

ConnectApi.OrderToCartInput

Input for action adding an order to a cart.

Property	Type	Description	Required or Optional	Available Version
`cartStateOrId`	String	Cart state (`active` or `current`) or the ID of the cart to which the products from an order are to be copied.	Required	57.0

ConnectApi.PaymentGroupRequest

Payment group input consumed by a payment group service.

Property	Type	Description	Required or Optional	Available Version
`createPaymentGroup`	Boolean	Specifies whether Salesforce needs to create a payment group (`true`) or not (`false`).	Optional	50.0
`currencyIsoCode`	String	Three-letter ISO 4217 currency code associated with the payment group record.	Optional	50.0
`id`	String	ID of the payment group record.	Optional	50.0
`sourceObjectId`	String	Source object ID of the payment group record. Supports only OrderId.	Optional	50.0

ConnectApi.PaymentInfoInputRepresentation

Payment information about additional funds required for an order.

Property	Type	Description	Required or Optional	Available Version
lastPaymentGatewayLogId	String	Last payment gateway log ID for the new order payment summary.	Optional	60.0
name	String	Overrides the default name of the order payment summary created.	Optional	60.0
paymentAuthorizationId	String	Payment authorization ID to be used if needed to fund the exchange order.	Optional	60.0
paymentIds	List<String>	Payment IDs for the new order payment summary.	Optional	60.0
paymentMethodId	String	Payment method ID for the new order payment summary.	Optional	60.0

ConnectApi.PaymentMethodTokenizationRequest

Payment method tokenization input consumed by the payment tokenization service.

Subclass of ConnectApi.BaseRequest.

Property	Type	Description	Required or Optional	Available Version
address	ConnectApi.AddressRequest	Address of the payment method.	Required	52.0
cardPaymentMethod	ConnectApi.CardPaymentMethodRequest	Object representation of the card payment method.	Required	52.0
paymentGatewayId	String	ID of the card payment method's payment gateway.	Required	52.0

ConnectApi.PhotoInput

Specify how to crop a photo that has already been uploaded.

Property	Type	Description	Available version
cropSize	Integer	The length, in pixels, of any edge of the crop square.	29.0
cropX	Integer	The position X, in pixels, from the left edge of the image to the start of the crop square. Top left is position (0,0).	29.0
cropY	Integer	The position Y, in pixels, from the top edge of the image to the start of the crop square. Top left is position (0,0).	29.0

Property	Type	Description	Available version
`fileId`	String	18 character ID of an existing file. The key prefix must be 069 and the file must be an image and be smaller than 2 GB. Note: Images uploaded on the Group page and on the User page don't have file IDs and therefore can't be used.	25.0
`versionNumber`	Integer	Version number of the existing content. If not provided, the latest version is used.	25.0

SEE ALSO:

- setPhotoWithAttributes(communityId, groupId, photo)
- setPhotoWithAttributes(communityId, groupId, photo, fileUpload)
- updateRecommendationDefinitionPhotoWithAttributes(communityId, recommendationDefinitionId, photo)
- updateRecommendationDefinitionPhotoWithAttributes(communityId, recommendationDefinitionId, photo, fileUpload)
- setPhotoWithAttributes(communityId, userId, photo)
- setPhotoWithAttributes(communityId, userId, photo, fileUpload)

ConnectApi.PinCapabilityInput

Pin or unpin a feed element to a feed.

Property	Type	Description	Required or Optional	Available Version
`entityId`	String	ID of the entity to pin or unpin. In version 41.0 and later, `entityId` must be a feed item ID. In version 41.0–42.0, only one feed item can be pinned per feed. In version 43.0 and later, three feed items can be pinned per feed.	Required	41.0
`isPinned`	Boolean	Specifies whether to pin (`true`) or unpin (`false`) the entity.	Required	41.0

ConnectApi.PollCapabilityInput

Create, update, or vote on a poll on a feed element.

This class is a subclass of ConnectApi.FeedElementCapabilityInput.

Property	Type	Description	Required or Optional	Available Version
`choices`	List<String>	The choices used to create a new poll. You must specify 2–10 poll choices for each poll.	Required for creating a poll	32.0

Property	Type	Description	Required or Optional	Available Version
myChoiceId	String	ID of an existing choice on the feed poll. Used to vote on an existing poll.	Required for voting on a poll	32.0

SEE ALSO:

ConnectApi.FeedElementCapabilitiesInput

ConnectApi.PostAuthApiPaymentMethodRequest

Payment method input for post authorization.

Subclass of ConnectApi.BaseApiPaymentMethodRequest.

Property	Type	Description	Required or Optional	Available Version
alternativePaymentMethod	ConnectApi.AlternativePaymentMethod	Alternative payment method.	Required	54.0
cardPaymentMethod	ConnectApi.CardPaymentMethodRequest	Card payment method.	Required	54.0

ConnectApi.PostAuthRequest

Payment post authorization input consumed by the payment post authorization service.

Subclass of ConnectApi.BaseRequest.

Property	Type	Description	Required or Optional	Available Version
accountId	String	ID of the account of the customer for the authorized payment.	Required	54.0
amount	Double	Amount of the post authorization.	Required	54.0
comments	String	Comments for payment post authorization. Maximum of 1000 characters.	Optional	54.0
currencyIsoCode	String	Three-letter ISO 4217 currency code associated with the payment group record.	Optional	54.0
effectiveDate	Datetime	Date that the payment post authorization occurs.	Required	54.0
paymentGatewayId	String	Payment gateway that evaluates the post authorization.	Required	54.0

Property	Type	Description	Required or Optional	Available Version
paymentGroup	ConnectApi.PaymentGroupRequest	Payment group associated with or to be created for the request. Request must contain either a paymentGroupId or paymentGroup, but not both.	Optional	54.0
paymentMethod	ConnectApi.PostAuthApiPaymentMethodRequest	Payment method sent for the post authorization.	Required.	54.0

ConnectApi.PreviewCartToExchangeOrderInputRepresentation

Information required to preview a cart to exchange order.

Property	Type	Description	Required or Optional	Available Version
exchangeCartId	String	ID of the cart used for adding items to the exchange order.	Required	60.0
orderSummaryId	String	Order summary ID.	Required	60.0
referenceId	String	Return order ID.	Required	60.0
reservationType	String	The type of the reservation.	Optional	61.0

ConnectApi.PricingInput

Pricing for multiple products.

Property	Type	Description	Required or Optional	Available Version
pricingLineItems	List<ConnectApi.PricingLineItemInput>	Up to 500 line items for pricing.	Required	49.0

ConnectApi.PricingLineItemInput

Pricing line item.

Property	Type	Description	Required or Optional	Available Version
productId	String	ID of the product to price.	Required	49.0

SEE ALSO:

ConnectApi.PricingInput

ConnectApi.ProductSearchGroupingInput

Grouping information for product search results.

Property	Type	Description	Required or Optional	Available Version
`groupingOption`	`ConnectApi.CommerceSearchGroupingOption`	Grouping option for search results. Values are: • `BestMatch`—Search results are grouped by the best-match product of the variation group. • `NoGrouping`—Search results aren't grouped. • `VariationParent`—Search results are grouped by the variation parent.	Required	52.0
`topProductType`	`ConnectApi.CommerceSearchTopProductType`	Type of the top product to return for each product group in search results. Value is: • `VariationParent` If `NoGrouping` is specified for `groupingOption`, `topProductType` is ignored.	Optional	52.0—62.0

ConnectApi.ProductSearchInput

Product search.

Property	Type	Description	Required or Optional	Available Version
`categoryId`	String	Category ID returns results for products in this category or its subcategories. If you omit `categoryId` from the request, all categories are searched. If you specify `categoryId` and `searchTerm`, only products in the specified category are searched.	Required if `searchTerm` isn't specified	52.0
`fields`	List<String>	Product fields to return in search results. Search results include fields you have access to. If unspecified, returns the `Name`, `Description`, `StockKeepingUnit`, `ProductCode`, and `Family` fields.	Optional	52.0
`grouping`	`ConnectApi.ProductSearchGroupingInput`	Specifies whether to group products in search results and how to group them. If unspecified, the default is the value	Optional	52.0

Property	Type	Description	Required or Optional	Available Version
		specified in **Search** > **Results Display Settings** > **Results Grouping**.		
`includePrices`	Boolean	Specifies whether to include prices for products in search results (`true`) or not (`false`). If unspecified, defaults to `false`.	Optional	52.0
`includeQuantityRule`	Boolean	Specifies whether to include purchase quantity rule information for products in search results (`true`) or not (`false`). If unspecified, defaults to `false`.	Optional	52.0
`page`	Integer	Number of the page you want returned. Starts at 0. If you pass in `null` or 0, the first page is returned.	Optional	52.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are from 1 through 200. If unspecified, the default is the value specified in Results per Page in **Search** > **Results Display Settings**.	Optional	52.0
`refinements`	List<`ConnectApi.RefinementInput`>	List up to nine refinements (facets) for search results. Buyers or shoppers can select up to 20 values for each refinement.	Optional	52.0
`searchTerm`	String	List of up to 32 space-separated search terms.	Required if `categoryId` isn't specified	52.0
`sortRuleId`	String	ID of the sort rule that specifies the order of products in the search results.	Optional	52.0

ConnectApi.ProductVariationInputRepresentation

Variation product input.

Property	Type	Description	Required or Optional	Available Version
`variationAttributes`	Map<String, String>	Mapping of variation attributes (API name and value) associated with the product.	Required	62.0

ConnectApi.PromotionCartDeliveryGroupInput

IDs of the cart delivery group and its delivery method.

Property	Type	Description	Required or Optional	Available Version
`cartDeliveryGroupId`	String	ID of the cart delivery group.	Required	57.0
`deliveryMethodId`	String	ID of the order delivery method.	Required	57.0

SEE ALSO:

ConnectApi.PromotionEvaluateInput

evaluate(salesTransaction)

ConnectApi.PromotionCartInput

Cart during promotion evaluation.

Property	Type	Description	Required or Optional	Available Version
`cartAdjustmentGroups`	ConnectApi.PromotionCartAdjustmentGroupInput[]	Collection of cart adjustment groups associated with the items in the cart.	Optional	60.0
`cartDeliveryGroups`	ConnectApi.PromotionCartDeliveryGroupInput[]	Collection of cart delivery groups associated with the items in the cart. Available if shipping promotions are enabled.	Required when evaluating shipping promotions	60.0
`cartItems`	List<`ConnectApi.PromotionCartItemInput`>	A collection of items in the cart.	Required	57.0
`currencyIsoCode`	String	Three-letter ISO 4217 currency code associated with the cart.	Required for multi-currency orgs	57.0
`id`	String	ID of the cart.	Optional	57.0

SEE ALSO:

ConnectApi.PromotionEvaluateInput

evaluate(salesTransaction)

ConnectApi.PromotionCartItemInput

Item in a cart during promotion evaluation.

Property	Type	Description	Required or Optional	Available Version
`cartDeliveryGroupId`	String	ID of the cart delivery group.	Required when evaluating shipping promotions	57.0
`cartId`	String	ID of the cart.	Optional	57.0
`id`	String	ID of the cart item. Must be unique across all items in the cart.	Required	57.0
`itemDescription`	String	Description of the cart item.	Optional	57.0
`itemName`	String	Name of the cart item.	Optional	57.0
`listPrice`	String	List price of the cart item.	Optional	57.0
`product2Id`	String	Product ID of the cart item.	Required if `sku` isn't specified	57.0
`quantity`	String	Number of items in the cart.	Required	57.0
`salesPrice`	String	Sales price of the cart item. This is the price per quantity and the value used to compute the discount. If `salesPrice` and `totalLineBaseAmount` are specified, `totalLineBaseAmount` is used.	Required if `totalLineBaseAmount` isn't specified	57.0
`sku`	String	Stock keeping unit (SKU) of the cart item.	Required if `product2Id` isn't specified	57.0
`totalLineBaseAmount`	String	Total amount for the cart item, equal to sales price multiplied by quantity. This value is used to compute the discount. If `salesPrice` and `totalLineBaseAmount` are specified, `totalLineBaseAmount` is used.	Required if `salesPrice` isn't specified	57.0
`totalListBaseAmount`	String	Total amount for the cart item based on list price and quantity.	Optional	57.0
`type`	`ConnectApi.CartItemType`	Type of item in a cart. Values are: • `DeliveryCharge` • `Product`	Required when evaluating shipping promotions	57.0

SEE ALSO:

ConnectApi.PromotionCartInput
ConnectApi.PromotionEvaluateInput
evaluate(salesTransaction)

ConnectApi.PromotionEvaluateInput

Find promotions that the customer is eligible for and compute their discounts.

Property	Type	Description	Required or Optional	Available Version
`cart`	ConnectApi.PromotionCartInput	Cart and its items.	Required	57.0
`cartDeliveryGroups`	[illegible]	List of cart delivery groups associated with the items in the cart. Available if shipping promotions are enabled.	Required when evaluating shipping promotions	57.0
`couponCodes`	List<String>	Collection of coupon codes to enable promotions. A customer can apply a maximum of two coupons per cart.	Optional	57.0
`effectiveAccountId`	String	ID of the account for which the request is made.	Required if `segments` isn't specified	57.0
`isItemizeHeaderAdjustments`	Boolean	Specifies whether order-level adjustments are itemized (`true`) or not (`false`). If unspecified, the default value is `false`.	Optional	57.0
`parentProducts`	List<ConnectApi.PromotionParentProductsInput>	Collection of parent product IDs mapped to their variation product IDs.	Optional	57.0
`productCategories`	List<ConnectApi.PromotionProductCategoriesInput>	Collection of product IDs mapped to their associated category IDs.	Optional	57.0
`segments`	List<String>	All promotions associated with promotion segments specified in this collection are active and can be evaluated against the cart. Additionally, any segments associated with a store or buyer group are also still evaluated against the cart. If this field is not present, only the promotions associated with a store or buyer group are evaluated.	Optional	57.0
`webStoreId`	String	ID of the store for which the request is made. If unspecified, defined segments must be used instead.	Optional	57.0

SEE ALSO:

evaluate(salesTransaction)

ConnectApi.PromotionParentProductsInput

IDs of a parent product and variation product.

Property	Type	Description	Required or Optional	Available Version
`childProductId`	String	ID of the variation product.	Required	57.0
`parentProductId`	String	ID of the parent product.	Required	57.0

SEE ALSO:

ConnectApi.PromotionEvaluateInput

evaluate(salesTransaction)

ConnectApi.PromotionProductCategoriesInput

IDs of a product and associated category.

Property	Type	Description	Required or Optional	Available Version
`categoryId`	String	ID of the category.	Required	57.0
`productId`	String	ID of the product.	Required	57.0

SEE ALSO:

ConnectApi.PromotionEvaluateInput

evaluate(salesTransaction)

ConnectApi.QuantityWithSkuInputRepresentation

A quantity of a product.

Property	Type	Description	Required or Optional	Available Version
`quantity`	Double	Quantity of the product.	Required	51.0
`stockKeepingUnit`	String	SKU of the product.	Required	51.0

ConnectApi.QuestionAndAnswersCapabilityInput

Create or edit a question feed element or set the best answer of the existing question feed element.

This class is a subclass of ConnectApi.FeedElementCapabilityInput.

Property	Type	Description	Required or Optional	Available Version
`bestAnswerId`	String	A comment ID to use as a best answer for a question feed element. The best answer comment must already exist on the question feed element.	Required to update a feed element. Not supported when posting a feed element.	32.0
`questionTitle`	String	Title for a question feed element. To edit the title of a question, use `updateFeedElement(communityId, feedElementId, feedElement)`. Editing question titles is supported in version 34.0 and later.	Required to post a feed element. Not supported when updating a feed element.	32.0

SEE ALSO:

- Edit a Question Title and Post
- ConnectApi.FeedElementCapabilitiesInput

ConnectApi.RankAverageDistanceInputRepresentation

An order recipient's geographic location and information about sets of inventory locations that can fulfill the order.

Property	Type	Description	Required or Optional	Available Version
`deliveryCountryCode`	String	The country code of the order recipient.	Required	51.0
`deliveryPostalCode`	String	The postal code of the order recipient.	Required	51.0
`distanceUnit`	String	Specify *`mi`* or *`km`* to return average distances in miles or kilometers, respectively.		51.0
`sortResult`	String	Specify *`ASC`* or *`DESC`* to rank the results by average shipping distance in ascending or descending order, respectively.		51.0
`targetLocations`	List<ConnectApi.TargetLocationInputRepresentation>	Each element is a set of inventory locations that can combine to fulfill the order.	At least one element is required	51.0

ConnectApi.ReadByCapabilityInput

Mark feed elements as read by the context user.

This class is a subclass of ConnectApi.FeedElementCapabilityInput.

Property	Type	Description	Required or Optional	Available Version
`isReadByMe`	Boolean	Specifies to mark the feed element as read (`true`) for the context user.	Required	40.0
`lastReadDateByMe`	Datetime	Specifies the last date, in ISO 8601 format, when the feed element is marked as read for the context user. If you don't specify a date or you specify a future date, the current system date is used.	Optional	40.0

SEE ALSO:

ConnectApi.FeedElementCapabilitiesInput

ConnectApi.SequenceOrderPaymentSummaryInputRepresentation

Amount to apply to specified OrderPaymentSummary as part of a payment or refund.

Property	Type	Description	Required or Optional	Available Version
`amount`	Double	Amount to apply to the OrderPaymentSummary.	Optional	56.0
`orderPaymentSummaryId`	String	ID of the OrderPaymentSummary to apply the Amount to.	Required	56.0

SEE ALSO:

ensureRefundsAsync(orderSummaryId, ensureRefundsInput)

ConnectApi.EnsureRefundsAsyncInputRepresentation

ConnectApi.sharedOrderPaymentSummarySequenceInputRepresentation

Shared order payment summary sequence.

Property	Type	Description	Required or Optional	Available Version
`orderPaymentSummaryId`	String	Shared order payment summary ID.	Required	60.0
`reservedBalanceAmount`	Double	Balance amount to be reserved.	Required	60.0

ConnectApi.SubmitCartToExchangeOrderInputRepresentation

Information required for a submit cart to exchange order action.

Property	Type	Description	Required or Optional	Available Version
exchangeCartId	String	ID of the cart used for adding items to the exchange order.	Required	60.0
orderNumber	String	Order number.	Optional	60.0
orderSummaryId	String	Order summary ID.	Required	60.0
paymentInfoList	[illegible] on page 1906>	List of payment information when additional funds are needed for the newly created exchange order.	Optional	60.0
referenceId	String	Return order ID.	Optional	60.0
reservationType	String	The reservation that's created by the submit API. The possible values are Full, which means there's a reservation against the exchange cart, or None if there's no reservation.	Optional	61.0
sequences	[illegible] on page 1917>	Ordered list of order payment summaries and reserved balance amounts to apply them to.	Optional	60.0

ConnectApi.RecipientEngagementContextInput

Context based on which the survey invitation is sent to a participant.

Property	Type	Description	Required or Optional	Available Version
recipientEngagementContext	Map<String, String>	Map each recipient with the context based on which the survey invitation is emailed.	Required	50.0
recipientId	String	Participant ID with whom the engagement context should be associated.	Required	50.0

SEE ALSO:

ConnectApi.SurveyInvitationEmailInput

ConnectApi.RecommendationAudienceInput

A custom recommendation audience.

Property	Type	Description	Required or Optional	Available Version
`criteria`	ConnectApi.AudienceCriteriaInput	The criteria for the custom recommendation audience type.	Optional If not specified when creating a recommendation audience, the audience criteria type defaults to custom list.	36.0
`memberOperationType`	`ConnectApi.RecommendationAudienceMemberOperationType`	**Important:** This property is available only in version 35.0. In version 36.0 and later, use ConnectApi.CustomListAudienceCriteriaInput. The operation to carry out on the audience members. • `Add`—Adds specified members to the audience. • `Remove`—Removes specified members from the audience.	Required to update a recommendation audience Don't use or specify `null` to create a recommendation audience	35.0 only
`members`	List<String>	**Important:** This property is available only in version 35.0. In version 36.0 and later, use ConnectApi.CustomListAudienceCriteriaInput. A collection of user IDs. When updating an audience, you can include up to 100 members. An audience can have up to 100,000 members, and each Experience Cloud site can have up to 100 audiences.	Required to update a recommendation audience Don't use or specify `null` to create a recommendation audience	35.0 only
`name`	String	The unique name of the custom recommendation audience.	Optional to update a recommendation audience Required to create a recommendation audience	35.0

SEE ALSO:

createRecommendationAudience(communityId, recommendationAudience)

ConnectApi.RecommendationDefinitionInput

A custom recommendation definition.

Property	Type	Description	Required or Optional	Available Version
`actionUrl`	String	URL for acting on the custom recommendation, for example, the URL to join a group.	Required to create a recommendation definition Optional to update a recommendation definition	35.0
`actionUrlName`	String	Text label for the action URL in the user interface, for example, "Launch."	Required to create a recommendation definition Optional to update a recommendation definition	35.0
`explanation`	String	Explanation, or body, of the custom recommendation.	Required to create a recommendation definition Optional to update a recommendation definition	35.0
`name`	String	Name of the custom recommendation definition. The name is displayed in Setup.	Required to create a recommendation definition Optional to update a recommendation definition	35.0
`title`	String	Title of the custom recommendation definition.	Optional	35.0

SEE ALSO:

createRecommendationDefinition(communityId, recommendationDefinition)

ConnectApi.RecommendationReactionInput

A reaction to a recommendation produced by a recommendation strategy.

Property	Type	Description	Required or Optional	Available Version
`aiModel`	String	Reserved for future use.	Optional	47.0
`contextRecordId`	String	ID of the context record. For example, if the next best action is on a case detail page, the ID of the case.	Optional	45.0
`executionId`	String	ID of the original recommendation strategy execution.	Optional	45.0
`externalId`	String	External ID of the recommendation. This ID doesn't need to be a Salesforce 18-character ID. For example, it can be a product number from an external system.	Optional	46.0
`onBehalfOfId`	String	ID of the user or entity for which the reaction took place.	Optional	45.0
`reactionType`	`ConnectApi.RecommendationReactionType`	Type of reaction to a recommendation. Values are: • `Accepted` • `Rejected`	Required	45.0
`recommendationMode`	String	Reserved for future use.	Optional	46.0
`recommendationScore`	Double	Reserved for future use.	Optional	46.0
`strategyName`	String	Name of the recommendation strategy.	Required	45.0
`targetActionId`	String	ID of the target action.	Optional	45.0
`targetActionName`	String	Name of the target action.	Required	45.0
`targetId`	String	ID of the recommendation that is being reacted to.	Required	45.0

ConnectApi.RecordCapabilityInput

Attach an existing knowledge article to a comment.

This class is a subclass of ConnectApi.FeedElementCapabilityInput.

Property	Type	Description	Required or Optional	Available Version
`recordId`	String	ID of the existing knowledge article to attach.	Required	42.0

ConnectApi.RecordsetFilterCriteriaInput

A set of recordset filter criteria applied to records, such as service appointment records.

Property	Type	Description	Required or Optional	Available Version
`criteriaIds`	List<String>	Recordset filter criteria IDs.	Required	53.0
`enforceSharing`	Boolean	Determines whether record sharing checks are enforced (`true`) or not (`false`) during the execution of this call.	Optional	53.0
`filteredObjectName`	String	Object that the filter is applied to.	Required	53.0
`recordIds`	List<String>	List of record IDs of the filtered object.	Required	53.0

ConnectApi.ReferencedRefundRequest

Referenced refund input.

Subclass of ConnectApi.RefundRequest.

Property	Type	Description	Required or Optional	Available Version
`accountId`	String	ID of the account linked to the referenced refund request.	Optional	50.0
`amount`	Double	Amount refunded.	Required	50.0
`clientContext`	String	Context for payment APIs. Used for a payment caller to re-establish context.	Optional	50.0
`comments`	String	Optional comments for the refund.	Optional	50.0
`effectiveDate`	Datetime	Date when the refund becomes effective.	Optional	50.0
`paymentGroup`	`ConnectApi.PaymentGroupRequest`	Payment group details associated with the refund request.	Optional	50.0

ConnectApi.RefinementInput

Attribute-based refinement input for product search.

This class is abstract and is a superclass of ConnectApi.DistinctValueRefinementInput.

Property	Type	Description	Required or Optional	Available Version
`attributeType`	`ConnectApi.CommerceSearchAttributeType`	Search attribute type. • `Custom` • `ProductAttribute`	Required	52.0

Property	Type	Description	Required or Optional	Available Version
		• `Standard`		
`nameOrId`	String	Developer name of the attribute. In version 52.0 and later, the ID of the attribute isn't supported.	Required	52.0
`type`	`ConnectApi.CommerceSearchFacetType`	Search facet type. Value is: • `DistinctValue`	Required	52.0

ConnectApi.RefundRequest

Refund input.

This class is abstract.

Subclass of ConnectApi.BaseRequest.

No additional properties.

Superclass of ConnectApi.ReferencedRefundRequest.

ConnectApi.ReleaseHeldFOCapacityInputRepresentation

Request to release held fulfillment order capacity at one or more locations. Can correspond to one action call.

Property	Type	Description	Required or Optional	Available Version
`releaseHeldFOCapacityRequests`	List<`ConnectApi.ReleaseHeldFOCapacityRequestInputRepresentation`>	List of requests to release held fulfillment order capacity at one or more locations.	Required	55.0

ConnectApi.ReleaseHeldFOCapacityRequestInputRepresentation

Request to release held fulfillment order capacity at one or more locations.

Property	Type	Description	Required or Optional	Available Version
`allOrNothing`	Boolean	Controls whether a single failed request cancels all other requests in the list (*`true`*) or whether some requests can succeed if others fail (*`false`*). The default value is *`false`*.	Optional	55.0

Property	Type	Description	Required or Optional	Available Version
capacityRequests	List<ConnectApi.CapacityRequestInputRepresentation>	List of requests to release held fulfillment order capacity. Each request is for capacity for one fulfillment order held at one location.	Required	55.0

ConnectApi.RequestHeaderInput

An HTTP request header name and value pair.

Property	Type	Description	Required or Optional	Available Version
name	String	The name of the request header.	Required	33.0
value	String	The value of the request header.	Required	33.0

SEE ALSO:

Define an Action Link and Post with a Feed Element

ConnectApi.ReturnItemsInputRepresentation

Data about products and delivery charges to return, as well as associated return fees.

Property	Type	Description	Required or Optional	Available Version
returnOrderItemDeliveryCharges	List<ConnectApi.ReturnOrderItemDeliveryChargeInputRepresentation>	List of ReturnOrderLineItems to return that represent delivery charges.	Optional	52.0
returnOrderItemFees	List<ConnectApi.ReturnOrderItemFeeInputRepresentation>	List of ReturnOrderLineItems to process that represent return fees.	Optional	56.0
returnOrderItems	List<ConnectApi.ReturnOrderItemInputRepresentation>	List of ReturnOrderLineItems to process that represent products, along with data about how to process them.	Required	52.0

SEE ALSO:

returnItems(returnOrderId, returnItemsInput)

ConnectApi.ReturnOrderInputRepresentation

Data for creating a ReturnOrder and ReturnOrderLineItems.

Property	Type	Description	Required or Optional	Available Version
`orderSummaryId`	String	ID of the OrderSummary containing the items to be returned. The OrderSummary's OrderLifeCycleType must be Managed.	Required	50.0
`returnOrder LifeCycleType`	String	The LifeCycleType of the ReturnOrder. Possible values are: • Managed—Process the ReturnOrder using the APIs and actions. It can generate change orders and affects financial fields and rollup calculations. • Unmanaged—The ReturnOrder is for tracking purposes only. It isn't involved in any financial calculations and doesn't generate any change orders. The system doesn't prevent the creation of duplicate ReturnOrderLineItems in an unmanaged ReturnOrder for the same OrderItem.	Required	51.0
`returnOrderLineItems`	List<`ConnectApi.ReturnOrderLineItemInputRepresentation`>	List of data for creating ReturnOrderLineItems.	At least one element is required	50.0
`status`	String	Status to assign the ReturnOrder. This value must match an entry in the ReturnOrder object's Status picklist.	Required	51.0

SEE ALSO:

createReturnOrder(returnOrderInput)

ConnectApi.ReturnOrderItemDeliveryChargeInputRepresentation

ID of a ReturnOrderLineItem representing a delivery charge.

Property	Type	Description	Required or Optional	Available Version
`returnOrder LineItemId`	String	ID of a ReturnOrderLineItem to return.	Required	52.0

SEE ALSO:

ConnectApi.ReturnItemsInputRepresentation

returnItems(returnOrderId, returnItemsInput)

ConnectApi.ReturnOrderItemFeeInputRepresentation

ID of a ReturnOrderLineItem representing a return fee, and instructions for updating it. After the update, the ReturnOrderLineItem is read-only. Any remaining quantity of the fee to be processed is added to a new ReturnOrderLineItem.

Property	Type	Description	Required or Optional	Available Version
`quantityReturned`	Double	Quantity of the ReturnOrderLineItem to process. When the fee is a fixed amount, the charge is determined by multiplying the total fee amount by this value divided by the expected quantity. For example, if the fee amount is $10 and the expected quantity is 2, then if the `quantityReturned` is 1, $5 is charged. This value normally equals the quantity returned of the ReturnOrderLineItem for the returned item that the fee applies to. The value must be greater than zero. If this value plus `quantityToCancel` is less than the expected quantity, then the remaining quantity to be returned is added to a new ReturnOrderLineItem.	Required	56.0
`quantityToCancel`	Double	Quantity of the ReturnOrderLineItem to remove. This value normally equals the quantity canceled of the ReturnOrderLineItem for the returned item that the fee applies to. This value can also be used to cancel a portion of the fee. The value must be zero or greater. If this value plus `quantityReturned` is less than the expected quantity, then the remaining quantity to be returned is added to a new ReturnOrderLineItem.	Required	56.0
`returnOrderLineItemId`	String	ID of the ReturnOrderLineItem representing the return fee.	Required	56.0

SEE ALSO:

ConnectApi.ReturnItemsInputRepresentation

returnItems(returnOrderId, returnItemsInput)

ConnectApi.ReturnOrderItemInputRepresentation

ID of a ReturnOrderLineItem and instructions for updating it. After the update, the ReturnOrderLineItem is read-only. Any remaining quantity to be returned is added to a new ReturnOrderLineItem.

Property	Type	Description	Required or Optional	Available Version
quantityReceived	Double	The quantity of the ReturnOrderLineItem that has been received. The value must be zero or greater. This value isn't used by any standard features, but is provided for use in customizations.	Required	52.0
quantityRejected	Double	The quantity of the ReturnOrderLineItem that has been rejected for return. The value must be zero or greater. This value isn't used by any standard features, but is provided for use in customizations.	Required	52.0
quantityReturned	Double	The quantity of the ReturnOrderLineItem that has been returned. The value must be greater than zero. If this value plus quantityToCancel is less than the expected return quantity, then the remaining quantity to be returned is added to a new ReturnOrderLineItem.	Required	52.0
quantityToCancel	Double	The quantity of the ReturnOrderLineItem to remove because it's not being returned. The value must be zero or greater. If this value plus quantityReturned is less than the expected return quantity, then the remaining quantity to be returned is added to a new ReturnOrderLineItem.	Required	52.0
reasonForRejection	String	The reason why the rejected quantity, if any, was rejected. This value isn't used by any standard features, but is provided for use in customizations.	Optional	52.0
returnOrderLineItemId	String	The ID of the ReturnOrderLineItem.	Required	52.0

SEE ALSO:

ConnectApi.ReturnItemsInputRepresentation

returnItems(returnOrderId, returnItemsInput)

ConnectApi.ReturnOrderLineItemInputRepresentation

Data for creating a ReturnOrderLineItem for an order item being returned, including data to create ReturnOrderLineItems representing any return fees associated with it.

Property	Type	Description	Required or Optional	Available Version
canReduceShipping	Boolean	Whether to refund any associated shipping charge.	Required	50.0
orderItemSummaryId	String	ID of the associated OrderItemSummary. If the OrderItemSummary already has an associated ReturnOrderLineItem, then you must specify a different reasonForReturn. Duplicating the reason breaks the financial calculations.	Required	50.0
quantityExpected	Double	Quantity expected to be returned. This value also applies to any fees specified in returnOrderLineItemFees.	Required	50.0
quantityReceived	Double	Quantity already physically returned.	Optional	50.0
reasonForReturn	String	Reason for the return. The value must match an entry in both the OrderSummaryChange Reason field and the ReturnOrderLineItem object's ReasonForReturn picklist.	Required if the returnOrder LifeCycleType is MANAGED.	50.0
returnOrder LineItemFees	List<ConnectApi. ReturnOrder LineItemFee InputRepresentation>	List of input data for return fees associated with the order item being returned. A ReturnOrderLineItem of Type Fee is created to represent each fee.	Optional	56.0

SEE ALSO:

ConnectApi.ReturnOrderInputRepresentation

createReturnOrder(returnOrderInput)

ConnectApi.ReturnOrderLineItemFeeInputRepresentation

Data for creating a ReturnOrderLineItem that represents a return fee.

Property	Type	Description	Required or Optional	Available Version
amount	Double	Value used to calculate the fee amount, as described by the amountType. It must be a positive value.	Required	56.0
amountType	String	Describes how the fee amount is calculated. It can have one of these values: • AmountWithTax—Value of amount is the fee amount, including tax.	Required	56.0

Property	Type	Description	Required or Optional	Available Version
		• `AmountWithoutTax`—Value of `amount` is the fee amount, not including tax. Tax is calculated on the value and added. • `Percentage`—Value of `amount` is a percentage. To determine the fee amount, `amount` is divided by 100, and then multiplied by the TotalPrice and TotalTaxAmount of the associated OrderItemSummary, prorated for the quantity being returned. • `PercentageGross`—Value of `amount` is a percentage. To determine the fee amount, `amount` is divided by 100, and then multiplied by the TotalLineAmountWithTax of the associated OrderItemSummary, prorated for the quantity being returned.		
`description`	String	Description of the fee.	Required	56.0
`product2Id`	String	ID of the product representing the fee.	Required	56.0
`reason`	String	Reason for the fee. The value must match an entry in the ReturnOrderLineItem object's ReasonForReturn picklist.	Required	56.0

SEE ALSO:

ConnectApi.ReturnOrderInputRepresentation

createReturnOrder(returnOrderInput)

ConnectApi.ReturnOrderLineItemInputRepresentation

ConnectApi.SaleApiPaymentMethodRequest

Payment method request for sale.

Subclass of ConnectApi.BaseApiPaymentMethodRequest.

Property	Type	Description	Required or Optional	Available Version
`cardPaymentMethod`	ConnectApi.CardPaymentMethodRequest	Payment method used in a sale request.	Required	54.0

ConnectApi.SaleRequest

Payment sale input consumed by the payment sale service.

Subclass of ConnectApi.BaseRequest.

Property	Type	Description	Required or Optional	Available Version
`accountId`	String	Reference to account.	Required	54.0
`amount`	Double	The amount of the sale request.	Required	54.0
`comments`	String	Optional comment for the sale request.	Optional	54.0
`currencyIsoCode`	String	Three-letter ISO 4217 currency code associated with the payment output.	Required	54.0
`effectiveDate`	Datetime	Date that the sale request takes effect.	Required	54.0
`paymentGatewayId`	String	The payment gateway that receives the sale request.	Required	54.0
`paymentGroup`	ConnectApi.PaymentGroupRequest	Payment group information for the sale request.	Optional	54.0
`paymentMethod`	ConnectApi.SalePaymentMethodRequest	Payment method used within the sale request.	Reqiured	54.0

ConnectApi.ScheduledRecommendationInput

A scheduled custom recommendation.

Property	Type	Description	Required or Optional	Available Version
`channel`	`ConnectApi.RecommendationChannel`	A way to tie custom recommendations together. For example, display recommendations in specific places in the UI or show recommendations based on time of day or geographic locations. Values are: • `CustomChannel1`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels. For example, community managers can use Experience Builder to determine where recommendations appear. • `CustomChannel2`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels.	Optional for creating a scheduled recommendation If not specified, defaults to `DefaultChannel`. Don't use when updating a scheduled recommendation	36.0

Property	Type	Description	Required or Optional	Available Version
		• `CustomChannel3`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels. • `CustomChannel4`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels. • `CustomChannel5`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels. • `DefaultChannel`—Default recommendation channel. Recommendations appear by default on the Home and Question Detail pages of Customer Service and Partner Central Experience Builder templates. They also appear in the feed in the Salesforce mobile web and anywhere community managers add recommendations using Experience Builder. Use these channel values; you can't rename or create other channels.		
`enabled`	Boolean	Indicates whether scheduling is enabled. If `true`, the custom recommendation is enabled and appears in Experience Cloud sites. If `false`, custom recommendations in feeds in Salesforce mobile web aren't removed, but no new custom recommendations appear. In Customer Service and Partner Central sites, disabled custom recommendations no longer appear.	Optional	35.0
`rank`	Integer	Relative rank of the scheduled custom recommendation indicated by ascending whole numbers starting with 1. Setting the rank is comparable to an insertion into an ordered list. The scheduled custom recommendation is inserted into the position specified by the `rank`. The `rank` of all the scheduled custom recommendations after it is pushed down.	Optional	35.0

Property	Type	Description	Required or Optional	Available Version
		See Ranking scheduled custom recommendations example. If the specified `rank` is larger than the size of the list, the scheduled custom recommendation is put at the end of the list. The `rank` of the scheduled custom recommendation is the size of the list, instead of the one specified. If a `rank` is not specified, the scheduled custom recommendation is put at the end of the list.		
`recommendationAudienceId`	String	ID of the audience for this scheduled custom recommendation.When updating a scheduled custom recommendation, specify `ALL` to remove the association between a custom recommendation audience and a scheduled custom recommendation.	Optional	35.0
`recommendationDefinitionId`	String	ID of the custom recommendation definition that this scheduled recommendation schedules.	Required to create a scheduled recommendation You can't specify a `recommendationDefinitionId` when updating a scheduled recommendation.	35.0

Ranking scheduled custom recommendations example

If you have these scheduled custom recommendations:

Scheduled Recommendations	Rank
ScheduledRecommendationA	1
ScheduledRecommendationB	2
ScheduledRecommendationC	3

And you include this information in the Scheduled Custom Recommendation Input:

Scheduled Recommendation	Rank
ScheduledRecommendationD	2

The result is:

Scheduled Recommendation	Rank
ScheduledRecommendationA	1
ScheduledRecommendationD	2
ScheduledRecommendationB	3
ScheduledRecommendationC	4

SEE ALSO:

createScheduledRecommendation(communityId, scheduledRecommendation)

ConnectApi.SearchDataCategory

Data category input for object search.

Property	Type	Description	Required or Optional	Available Version
`categories`	List<String>	List of data category names to filter.	Optional	63.0
`groupName`	String	Name of the data category group to filter.	Optional	63.0
`operator`	`ConnectApi.DataCategoryOperator`	Data category operator. • `Above`—Queries the data category and all of its parent categories. • `AboveOrBelow`—Queries the data category, all of its parent categories, and all of its subcategories. • `At`—Queries the data category. • `Below`—Queries the data category and all of its subcategories.	Optional	63.0

SEE ALSO:

ConnectApi.SearchRequest

ConnectApi.SearchRequest

Search request input for searching an object.

Property	Type	Description	Required or Optional	Available Version
`q`	String	Query term to search on. Query term must be two or more characters.	Required	63.0

Property	Type	Description	Required or Optional	Available Version
configurationName	String	Name of the search configuration to apply. Look up search configuration names from Search Manager.	Optional	63.0
dataCategories	List<ConnectApi.SearchDataCategory>	List of data categories to filter.	Optional	63.0
displayFields	List<String>	List of fields to display and return in the search results. By default, the fields displayed are defined by the search layout.	Optional	63.0
filters	List<ConnectApi.SearchFilter>	List of filters to apply.	Optional	63.0
highlights	Boolean	Specifies whether search generates a text highlight (`true`) or not (`false`). The default value is `false`.	Optional	63.0
offset	Integer	Search page offset position. Default value is `0`, which indicates displaying results from the beginning without skipping any entries.	Optional	63.0
orderBy	List<ConnectApi.SearchOrderBy>	Order by input for object search.	Optional	63.0
pageSize	Integer	Number of results in a page. Valid values are from 1 through 1999. If unspecified the default value is `20`.	Optional	63.0
spellcheck	Boolean	Specifies whether search should apply spellcheck (`true`) or not (`false`). The default value is `true`.	Optional	63.0

SEE ALSO:

find(objectApiName, request)

ConnectApi.SellerDetailsRequest

Seller details for the tax calculation

Property	Type	Description	Required or Optional	Available Version
code	String	Code used to identify the seller of the taxed items.		55.0

ConnectApi.ServiceAppointmentInput

Contains information about the service appointment.

Property	Type	Description	Required or Optional	Available Version
additionalInformation	String	Additional details about the service appointment.	Optional	53.0
appointmentMode	ConnectApiSvcApptModeEnum	Mode of the service appointment. • `Group`— Service appointment mode is Group. • `Regular`— Default mode of service appointment.	Optional	60.0
appointmentType	String	Type of the appointment.	Optional	53.0
attendeeLimit	Integer	Maximum number of customers that's allowed to attend the service appointment.	Required if the appointment mode is Group.	60.0
city	String	Name of the city.	Optional	53.0
comments	String	Comments about the appointment.	Optional	53.0
contactId	String	ID of the contact associated with the parent record.	Optional	53.0
country	String	Name of the country.	Optional	53.0
description	String	Description of the appointment.	Optional	53.0
engagementChannelTypeId	String	ID of the engagement channel type to associate with the appointment. You can use engagement channel type only if: • **Schedule Appointments Using Engagement Channels** is enabled in Salesforce Scheduler Settings in your Salesforce org. • Shifts are defined in the scheduling policy. For more information on setting up shifts in the scheduling policy, see Define Shift Rules in Scheduling Policy. Note: Engagement channel types are not supported with operating hours rules in the scheduling policy.	Optional	56.0

Property	Type	Description	Required or Optional	Available Version
extendedFields	List<ConnectApi.ExtendedFieldInput>	Values to add to any of the fields, including custom fields.	Optional	53.0
parentRecordId	String	ID of the parent record associated with the account.	Required if `lead` isn't provided.	53.0
postalCode	String	Postal code of the city.	Optional	53.0
schedEndTime	Datetime	Time at which the appointment is scheduled to end.	Optional	53.0
schedStartTime	Datetime	Time at which the appointment is scheduled to start.	Optional	53.0
serviceTerritoryId	String	ID of the service territory associated with the service appointment.	Optional	53.0
state	String	Name of the state.	Optional	53.0
street	String	Name of the street.	Optional	53.0
subject	String	Short phrase describing the appointment.	Optional	53.0
workTypeId	String	ID of the work type associated with the service appointment. If specified, it is added to the service appointment record.	Optional	53.0

ConnectApi.ShiftsFromPatternInput

Shifts from a pattern.

Property	Type	Description	Required or Optional	Available Version
schedulingEndDate	String	Scheduling end date in YYYY-MM-DD format. Provide `schedulingEndDate` or `schedulingOccurrences`. Don't provide both.	Required if `schedulingOccurrences` isn't provided	51.0
schedulingOccurrences	Integer	Number of scheduling occurrences. Provide `schedulingEndDate` or `schedulingOccurrences`. Don't provide both.	Required if `schedulingEndDate` isn't provided	51.0
schedulingStartDate	String	Scheduling start date in YYYY-MM-DD format.	Required	51.0

Property	Type	Description	Required or Optional	Available Version
`serviceResourceId`	String	ID of the service resource to assign shifts to. In version 53.0 and later, use `serviceResourceIdList`.	Optional	51.0–52.0
`serviceResourceIdList`	List<String>	List of service resource IDs to assign shifts to.	Optional	53.0
`serviceTerritoryId`	String	ID of the service territory to assign shifts to.	Optional	51.0
`shiftStatus`	String	Status of the shifts. Default values are: • `Confirmed` • `Published` • `Tentative` Additional status values can be created.	Optional	52.0

ConnectApi.ShippingCarrierInputRepresentation

Shipping carrier.

Property	Type	Description	Required or Optional	Available Version
`externalReference`	String	Unique code, reference, or identifier for the shipping carrier used by external systems.	Optional	63.0
`shippingCarrierMethods`	[illegible] on page 1937	List of shipping carrier methods.	Required	63.0

ConnectApi.ShippingCarrierMethodInputRepresentation

Shipping carrier method external references.

Property	Type	Description	Required or Optional	Available Version
`externalReference`	String	Unique code, reference, or identifier for the shipping carrier method used by external systems.	Required	63.0

ConnectApi.SocialPostMassApprovalInput

List of social post ids and the action to approve or reject publishing them.

Property	Type	Description	Required or Optional	Available Version
`isApproved`	Boolean	Specifies whether to approve (`true`) or reject (`false`) publishing the social posts. If unspecified, defaults to `false`.	Optional	46.0
`socialPostIdList`	List<String>	A list of up to 200 social post IDs.	Required	46.0

ConnectApi.StatusCapabilityInput

Change the status of a feed post or comment.

This class is a subclass of ConnectApi.FeedElementCapabilityInput.

Property	Type	Description	Required or Optional	Available Version
`feedEntityStatus`	`ConnectApi.FeedEntityStatus`	Status of the feed post or comment. Values are: • `Draft`—The feed post isn't published but is visible to the author and users with Modify All Data or View All Data permission. Comments can't be drafts. • `Isolated`—The feed post or comment is isolated, and only admins can see it. • `PendingReview`—The feed post or comment isn't approved yet and therefore isn't published or visible. • `Published`—The feed post or comment is approved and visible. Posts that have a status of `PendingReview` or `Published` can't be changed to a status of `Draft` and vice versa. Only admins can change the status of a post or comment to or from `Isolated` status.	Required	37.0

SEE ALSO:

ConnectApi.FeedElementCapabilitiesInput

ConnectApi.StreamSubscriptionInput

An entity to subscribe to for a Chatter feed stream.

Property	Type	Description	Required or Optional	Available Version
`entityId`	String	The ID of any feed-enabled entity, such as a group, record, or user, that the context user can access. When subscribed, the entity's feed is included in the feed stream.	Required	39.0

SEE ALSO:

ConnectApi.ChatterStreamInput

ConnectApi.StringList

List of string values.

Subclass of ConnectApi.AbstractList.

Property	Type	Description	Required or Optional	Available Version
`values`	List<String>	List of string values to filter on, for example, `["A", "B", "C"]`.	Optional	63.0

ConnectApi.SurveyInvitationEmailInput

Survey invitation email.

Property	Type	Description	Required or Optional	Available Version
`allowGuestUserResponse`	Boolean	Specifies whether participants who don't have a Salesforce account can respond (`true`) or not (`false`).	Required	50.0
`allowParticipantsAccessTheirResponse`	Boolean	Specifies whether participants can see their responses (`true`) or not (`false`).	Required	50.0
`associateRecordsWithRecipients`	List<`ConnectApi.AssociateRecordsWithRecipientInput`>	Maps each recipient with another record that must be associated with the recipient's survey invitation.	Optional	50.0
`body`	String	Content of the email. Specify the email body in case you don't specify an email template. The email body must contain one of the following merge fields: • To embed a link to launch the survey: [[SURVEY_INVITATION_URL]]	Optional	50.0

Property	Type	Description	Required or Optional	Available Version
		• To embed a survey question: {{{SurveyQuestion.QuestionName}}} and {{{SurveyQuestion.QuestionHtmlContent}}}		
`collectAnonymousResponse`	Boolean	Specifies whether participants can respond anonymously (`true`) or not (`false`).	Required	50.0
`communityId`	String	ID of the site that's used to open the survey for users outside your org.	Optional	50.0
`emailTemplateId`	String	ID of the Lightning email template that's used to send the survey invitation. The template must contain the required merge fields that embed either the survey link or a question in the email. Only Lightning email templates are used to send survey invitations.	Optional	50.0
`fromEmailAddress`	String	Email ID of the user or the org-wide email address associated with the user's profile.	Required	50.0
`invitationExpirationDate`	Datetime	Date on which the survey invitation expires.	Optional	50.0
`invitationOwner`	String	ID of the owner of the survey invitation records.	Optional	50.0
`isPersonalInvitation`	Boolean	Specifies whether an unique invitation is created for each participant (`true`) or not (`false`). When a participant responds using a personal invitation, the response record is associated with the participant's Salesforce record.	Required	50.0
`recipientEngagementContexts`	List<`ConnectApi.RecipientEngagementContextInput`>	Maps each recipient with the context based on which the survey invitation is emailed.	Optional	50.0
`recipients`	List<String>	List of up to 300 IDs of leads, contacts, or users to whom the survey invitation is emailed.	Required	50.0
`shareInvitationsWith`	List<String>	IDs of the users with whom the survey invitation records must be shared. The invitation records are shared with Read access.	Optional	50.0
`subject`	String	Subject of the email. Specify the subject in case you don't specify an email template.	Optional	50.0

Property	Type	Description	Required or Optional	Available Version
`surveyQuestionIds`	List<String>	IDs of the questions that are embedded in the email. You can send an email invitation for questions of the following types: Net Promoter Score (NPS), rating, and score.	Optional	50.0

ConnectApi.TargetCollectionInput

Collection of targets to create.

Property	Type	Description	Required or Optional	Available Version
`targets`	List<`ConnectApi.TargetInput`>	List of targets to create.	Required	48.0

ConnectApi.TargetCollectionUpdateInput

Collection of targets to update.

Property	Type	Description	Required or Optional	Available Version
`targets`	List<`ConnectApi.TargetUpdateInput`>	List of targets to update.	Required	48.0

ConnectApi.TargetInput

Target to create.

Property	Type	Description	Required or Optional	Available Version
`audienceId`	String	ID of the audience to assign to the target.	Required	48.0
`groupName`	String	Group name of the target. Groups bundle related target and audience pairs. You can have up to 2,000 groups and 500 targets per group. To determine the group name for targets of type `ExperienceVariation`, see Personalization Target Developer and Group Names in the *Experience Cloud Developer Guide*.	Required	48.0

Property	Type	Description	Required or Optional	Available Version
`priority`	Integer	Priority of the target. Within a group, priority determines which target is returned if the user matches more than one audience.	Optional	48.0
`publishStatus`	`ConnectApi.PublishStatus`	The publish status of the target. Values are: • `Draft` • `Live` We recommend setting the publish status to `Draft`. If you specify `Live`, your changes revert after you publish the site.	Optional	48.0
`targetType`	String	Type of target, indicating the nature of the data being targeted. Supported values include: • `ExperienceVariation` (version 48.0 and later) • Custom object API names, such as ***`CustomObjectName`*`__c`** (version 48.0 and later) • `NavigationLinkSet` (version 49.0 and later) • `Topic` (version 49.0 and later) • `CollaborationGroup` (version 49.0 and later) • `KnowledgeArticle` (version 49.0 and later) • `ContentDocument` (version 49.0 and later) • `ManagedContent` (version 49.0 and later) • `Report` (version 49.0 and later) • `Dashboard` (version 49.0 and later) You can have up to 2,500 `ExperienceVariation` targets and 25,000 record targets.	Required	48.0

Property	Type	Description	Required or Optional	Available Version
`targetValue`	String	Value of the target. If `targetType` is `ExperienceVariation`, `targetValue` is the developer name of the experience variation. If `targetType` is ***`CustomObjectName`*****`__c`**, `targetValue` is the ID of the custom object. To determine the developer name for targets of type `ExperienceVariation`, see Personalization Target Developer and Group Names in the *Experience Cloud Developer Guide*.	Required	48.0

SEE ALSO:

ConnectApi.TargetCollectionInput

ConnectApi.TargetLocationInputRepresentation

A set of inventory locations that together can fulfill an order.

Property	Type	Description	Required or Optional	Available Version
`locations`	List<`ConnectApi.LocationInputRepresentation`>	A list of locations with information about their country and postal codes.	Required	51.0

ConnectApi.TargetUpdateInput

Target to update.

Property	Type	Description	Required or Optional	Available Version
`audienceId`	String	ID of the audience to assign to the target.	Required if `priority` isn't specified. Otherwise, Optional	48.0
`priority`	Integer	Priority of the target. Within a group, priority determines which target is returned if the user matches more than one audience.	Required if `audienceId` isn't specified. Otherwise, Optional	48.0

Property	Type	Description	Required or Optional	Available Version
`targetId`	String	ID of the target to update.	Required	48.0

SEE ALSO:

ConnectApi.TargetCollectionUpdateInput

ConnectApi.TaxAddressRequest

Address input representation for tax calculation.

Name	Type	Description	Required or Optional	Available Version
`city`	String	City.	Optional	55.0
`country`	String	Country.	Optional	55.0
`latitude`	Double	Latitude.	Optional	55.0
`locationCode`	String	Location code.	Optional	55.0
`longitude`	Double	Longitude.	Optional	55.0
`postalCode`	String	Postal code.	Optional	55.0
`state`	String	State.	Optional	55.0
`street`	String	Street.	Optional	55.0

ConnectApi.TaxAddressesRequest

Addresses, including the Bill To address, Ship From address, Ship to address, and Sold To address.

Name	Type	Description	Required or Optional	Available Version
`billTo`	`ConnectApi.TaxAddressRequest`	Bill To address.	Optional	55.0
`shipFrom`	`ConnectApi.TaxAddressRequest`	Ship From address.	Optional	55.0
`shipTo`	`ConnectApi.TaxAddressRequest`	Ship To address.	Optional	55.0

Name	Type	Description	Required or Optional	Available Version
soldTo	ConnectApi.TaxAddressRequest	Sold To address.	Optional	55.0

ConnectApi.TaxCustomerDetailsRequest

Customer details for the tax calculation.

Property	Type	Description	Required or Optional	Available Version
accountId	String	ID of the customer's account.	Optional	55.0
code	String	Customer code.	Optional	55.0
exemptionNo	String	Tax exemption number.	Optional	55.0
exemptionReason	String	Tax exemption reason.	Optional	55.0

ConnectApi.TaxLineItemRequest

A list of line items passed to the tax engine for tax calculation.

Property	Type	Description	Required or Optional	Available Version
addresses	ConnectApi.TaxAddressesRequest	Addresses, including the Bill To address, Ship From address, Ship To address, and Sold To address.	Optional	55.0
amount	Double	Amount of the line item.	Optional	55.0
description	String	Description of the line item.	Optional	55.0
effectiveDate	Datetime	Date to apply the tax calculation to the line item.	Optional	55.0
lineNumber	String	Line number of the line item.	Optional	55.0
productCode	String	Product code of the line item.	Optional	55.0
quantity	Double	Quantity of the line item.	Optional	55.0
taxCode	String	Tax code for the line item.	Optional	55.0

ConnectApi.TaxTransactionRequest

Information about the tax transaction sent to the tax adapter as part of a tax calculation request.

This class is abstract.

Superclass of ConnectApi.CalculateTaxRequest.

Property	Type	Description	Required or Optional	Available Version
addresses	ConnectApi.TaxAddressesRequest	Addresses, including the Bill To address, Ship From address, Ship to address, and Sold To address.	Optional	55.0
currencyIsoCode	String	Three-letter ISO 4217 currency code associated with the payment group record.	Optional	55.0
customerDetails	ConnectApi.TaxCustomerDetailsRequest	Customer details for the tax calculation.	Optional	55.0
description	String	Information about whether the tax transaction failed or was successful.	Optional	55.0
documentCode	String	Document code.	Optional	55.0
effectiveDate	Datetime	The date that tax is applied to the taxed entity.	Required	55.0
lineItems	List<ConnectApi.TaxLineItemRequest>	The line items on which tax was calculated.	Required	55.0
referenceDocumentCode	String	The original document code. Used in case of subsequent transactions such as credit tax.	Optional	55.0
referenceEntityId	String	ID of the reference entity used during tax calculation.	Optional	55.0
transactionDate	Datetime	The date that the tax transaction occurred.	Optional	53.0

ConnectApi.TextClassificationsInputRepresentation

Text classification information associating classifiers and text to be classified.

Property	Type	Description	Required or Optional	Available Version
classifiers	List<String>	List of classifiers according to which text has to be classified.	Required	59.0
textList	List<String>	List of text to be classified.	Required	59.0

ConnectApi.TextSegmentInput

Include a text segment in a feed item or comment.

Subclass of ConnectApi.MessageSegmentInput.

Property	Type	Description	Available Version
`text`	String	Plain text for this segment. If hashtags or links are detected in *`text`*, they're included in the comment as hashtag and link segments. Mentions aren't detected in *`text`* and aren't separated out of the text. Mentions require `ConnectApi.MentionSegmentInput`.	28.0

SEE ALSO:

- Edit a Comment
- Edit a Feed Element
- Edit a Question Title and Post
- Post a Rich-Text Feed Element with Inline Image
- ConnectApi.MessageBodyInput

ConnectApi.TopicInput

Update a topic's name or description or merge topics.

Property	Type	Description	Available Version
`description`	String	Description of the topic	29.0
`idsToMerge`	List<String>	List of up to five secondary topic IDs to merge with the primary topic If any of the secondary topics are navigational or featured topics, they lose their topic type, topic images, and children topics. Their feed items are reassigned to the primary topic. If you merge a topic with a content topic, the content associations are preserved. If you merge a topic with an inactive endorsee, the endorsement isn't mapped to the primary topic.	33.0
`name`	String	Name of the topic Use this property to change only the capitalization and spacing of the topic name.	29.0

SEE ALSO:

- updateTopic(communityId, topicId, topic)

ConnectApi.TopicNamesInput

A list of topic names to replace currently assigned topics. Also a list of suggested topics to assign.

Property	Type	Description	Required or Optional	Available Version
`topicNames`	List<String>	A list of up to 10 topic names for a feed item or 100 topic names for a record.	Required	35.0

Property	Type	Description	Required or Optional	Available Version
topicSuggestions	List<String>	A list of suggested topics to assign to a record or feed item to improve future topic suggestions.	Optional	37.0

SEE ALSO:

reassignTopicsByName(communityId, recordId, topicNames)

ConnectApi.ArticleTopicAssignmentJobInput

ConnectApi.TopicsCapabilityInput

Assign topics to a feed element.

Property	Type	Description	Required or Optional	Available Version
contextTopicName	String	Name of the parent topic in the site to which the feed element belongs.	Optional	38.0
topics	List<String>	List of topics to assign to the feed element.	Required	38.0

SEE ALSO:

ConnectApi.FeedElementCapabilitiesInput

ConnectApi.UpdateServiceAppointmentInput

Contains information to update a service appointment.

Property	Type	Description	Required or Optional	Available Version
serviceAppointmentId	String	The ID of the service appointment to be modified.	Required	53.0
assignedResources	List<ConnectApi.AssignedResourceInput>	Represents the service resources who are assigned to a service appointment. When updating an appointment, pass the complete list of required resources. If you don't pass a resource who is already assigned to the appointment, the API deletes that assigned resource. For example, suppose that an existing service appointment has assigned resources: A and B and you pass B and C in assigned resources in the PATCH request. The API checks the resource availability of B and C	Optional	53.0

Property	Type	Description	Required or Optional	Available Version
		for existing work type and service territory, and if both are available, the service appointment gets updated with: • Resource A—Deleted • ResourceB—Updated • ResourceC—Created However, if you don't pass any of the assigned resources, the API assumes there's no change. Note: When creating an appointment, use `extendedFields` to add values to any of the fields, including custom fields, in `assignedResources` as long as you have edit access to those fields.		
`lead`	ConnectApi.LeadInput	Represents a prospect or lead. Note: Required to create a service appointment for unauthenticated guest users.	Required if `serviceAppointment` isn't provided.	53.0
`schedulingPolicyId`	String	The ID of the `AppointmentSchedulingPolicy` object. If no scheduling policy is passed in the request body, the default configurations are used. The only scheduling policy configuration that is used in determining time slots is the enforcement of account visiting hours.	Optional	53.0
`serviceAppointment`	ConnectApi.ServiceAppointmentInput	Represents the service appointment details to book an appointment. When updating an appointment, pass only the fields that must be updated. Note: When creating an appointment, use `extendedFields` to add values to any of the fields, including custom fields, in `assignedResources` as long as you have edit access to those fields.	Required if `lead` isn't provided.	53.0

ConnectApi.UpDownVoteCapabilityInput

Upvote or downvote a feed element or a comment.

Property	Type	Description	Required or Optional	Available Version
`vote`	`ConnectApi.UpDownVoteValue`	Type of vote for a feed element or comment. Values are: • `Down` • `None` • `Up`	Required	41.0

ConnectApi.UserInput

Update a user's About Me information.

Property	Type	Description	Available Version
`aboutMe`	String	The `aboutMe` property of a `ConnectApi.UserDetail` output object. This property populates the About Me section of the user profile, which is visible to all members of an Experience Cloud site or org.	29.0

SEE ALSO:

updateUser(communityId, userId, userInput)

ConnectApi.WishlistInput

Create a wishlist.

Property	Type	Description	Required or Optional	Available Version
`name`	String	Name of the wishlist.	Required	49.0
`products`	List<`ConnectApi.WishlistItemInput`>	List of products to add to the wishlist.	Optional	49.0

ConnectApi.WishlistItemInput

Item to update or add to a wishlist.

Property	Type	Description	Required or Optional	Available Version
`productId`	String	ID of the product to update or add to the wishlist.	Required	49.0

SEE ALSO:

ConnectApi.WishlistInput

ConnectApi.WishlistUpdateInput

Update a wishlist name.

Property	Type	Description	Required or Optional	Available Version
`name`	String	Wishlist name to update.	Required	50.0

ConnectApi.WrappedValue

Value wrapped for use as an object.

Property	Type	Description	Required or Optional	Available Version
`value`	`Object`	Value to wrap.	Required	60.0

Retired `ConnectApi` Input Classes

These `ConnectApi` input classes are retired.

IN THIS SECTION:

ConnectApi.CanvasAttachmentInput

Used to attach a canvas app to a feed item.

ConnectApi.ContentAttachmentInput

Used to attach existing content to a comment or feed item.

ConnectApi.DatacloudOrderInput

Input representation for a Datacloud order to purchase contacts or companies and retrieve purchase information.

ConnectApi.FeedItemAttachmentInput

Used to attach a file to a feed item.

ConnectApi.LinkAttachmentInput

Add links to a feed item.

ConnectApi.NewFileAttachmentInput

Attach a new file to a feed item.

ConnectApi.PollAttachmentInput

Attach a poll to a feed item.

ConnectApi.CanvasAttachmentInput

Used to attach a canvas app to a feed item.

Important: This class isn't available in version 32.0 and later. In version 32.0 and later, use ConnectApi.CanvasCapabilityInput.

Subclass of Connectapi.FeedItemAttachmentInput.

Property	Type	Description	Available Version
`description`	String	Optional. The description of the canvas app.	29.0–31.0
`developerName`	String	The developer name (API name) of the canvas app	29.0–31.0
`height`	String	Optional. The height of the canvas app in pixels. Default height is 200 pixels.	29.0–31.0
`namespacePrefix`	String	Optional. The namespace prefix of the Developer Edition organization in which the canvas app was created.	29.0–31.0
`parameters`	String	Optional. Parameters passed to the canvas app in JSON format. Example: `{'isUpdated'='true'}`	29.0–31.0
`thumbnailUrl`	String	Optional. A URL to a thumbnail image for the canvas app. Maximum dimensions are 120x120 pixels.	29.0–31.0
`title`	String	The title of the link used to call the canvas app.	29.0–31.0

ConnectApi.ContentAttachmentInput

Used to attach existing content to a comment or feed item.

Important: This class isn't available in version 32.0 and later. In version 32.0 and later, use ConnectApi.ContentCapabilityInput.

Subclass of ConnectApi.FeedItemAttachmentInput.

Property	Type	Description	Available Version
`contentDocumentId`	String	ID of the existing content.	28.0–31.0

ConnectApi.DataCloudOrderInput

Input representation for a Datacloud order to purchase contacts or companies and retrieve purchase information.

Property	Type	Description	Required or Optional	Available Version
`companyIds`	String	A comma-separated list of identification numbers for the companies to be purchased. You can't include any contact IDs or your purchase fails.	Required to purchase companies	32.0
`contactIds`	String	A comma-separated list of identification numbers for the contacts to be purchased. You can't include any company IDs or your purchase fails.	Required to purchase contacts	32.0
`userType`	ConnectDatacloudUserTypeEnum	Indicates the Data.com user type to be used. There are two user types. • `Monthly` (default) • `Listpool`	Optional	32.0

SEE ALSO:

postOrder(orderInput)

ConnectApi.FeedItemAttachmentInput

Used to attach a file to a feed item.

Important: This class isn't available in version 32.0 and later. In version 32.0 and later, use ConnectApi.FeedElementCapabilityInput.

This class is abstract and has no public constructor. You can make an instance only of a subclass.

Superclass for:

- ConnectApi.CanvasAttachmentInput
- ConnectApi.ContentAttachmentInput
- ConnectApi.LinkAttachmentInput
- ConnectApi.NewFileAttachmentInput
- ConnectApi.PollAttachmentInput

ConnectApi.LinkAttachmentInput

Add links to a feed item.

Important: This class isn't available in version 32.0 and later. In version 32.0 and later, use ConnectApi.LinkCapabilityInput.

Subclass of ConnectApi.FeedItemAttachmentInput.

Property	Type	Description	Available Version
`url`	String	URL to be used for the link	28.0–31.0
`urlName`	String	Title of the link	28.0–31.0

ConnectApi.NewFileAttachmentInput

Attach a new file to a feed item.

Important: This class isn't available in version 32.0 and later. In version 32.0 and later, use ConnectApi.ContentCapabilityInput.

The actual binary file, that is the attachment, is provided as part of the BinaryInput in the method that takes this attachment input, such as `postFeedItem` or `postComment`.

Subclass of ConnectApi.FeedItemAttachmentInput.

Property	Type	Description	Available Version
`description`	String	Description of the file to be uploaded.	28.0–31.0
`title`	String	The title of the file. This value is required and is also used as the file name. For example, if the title is My Title, and the file is a .txt file, the file name is My Title.txt.	28.0–31.0

ConnectApi.PollAttachmentInput

Attach a poll to a feed item.

Important: This class isn't available in version 32.0 and later. In version 32.0 and later, use ConnectApi.PollCapabilityInput.

Subclass of ConnectApi.FeedItemAttachmentInput.

Property	Type	Description	Available Version
`pollChoices`	List<String>	The text labels for the poll items. Polls must contain between 2 to 10 poll choices.	28.0–31.0

`ConnectApi` Output Classes

Most `ConnectApi` methods return instances of `ConnectApi` output classes.

All properties are read-only, except for instances of output classes created within test code.

All output classes are concrete unless marked abstract in this documentation.

All concrete output classes have no-argument constructors that you can invoke only from test code. See Testing ConnectApi Code.

ConnectApi.AbstractCartItem

A cart item.

This class is abstract.

Superclass of:

- ConnectApi.CartItem
- ConnectApi.CartItemWithoutPrice

Property Name	Type	Description	Available Version
billingFrequency	ConnectApi.BillingFrequency	Reserved for future use.	59.0
cartDeliveryGroupId	String	ID of the cart delivery group.	60.0
cartId	String	ID of the cart.	49.0
cartItemId	String	ID of the item.	49.0
childProductCount	Integer on page 3550	Number of child products in the cart that are associated with the item. A cart item can have child products if the `productClass` of the item is `Bundle`. For nested bundles, which include a child product that's also a bundle, `childProductCount` includes all child products.	62.0
customFields	List<SObject>	Array of sObjects and viewable custom fields for the sObjects. Standard fields are ignored. Currently, only the CartItem sObject is supported. Field-level security rules from the shopper profile are applied to the custom fields. The rules are applied for registered shoppers and for the guest shopper profile.	61.0
messagesSummary	ConnectApi.CartMessagesSummary	Messages summary for the item.	49.0
name	String	Name of the item.	49.0
parentCartItemId	String	ID of the item's parent cart item. The value is empty if the item is a top-level cart item.	62.0
productDetails	ConnectApi.CartItemProduct	Summary of the product details.	49.0
productId	String	ID of the product.	49.0
productSellingModelId	String	Reserved for future use.	59.0
quantity	String	Quantity of the item.	49.0
sellingModelType	ConnectApi.SellingModelType	Reserved for future use.	60.0
subscriptionTerm	Integer on page 3550	Reserved for future use.	59.0

Property Name	Type	Description	Available Version
`type`	`ConnectApi.CartItemType`	Type of item in a cart. Values are: • `DeliveryCharge` • `Product`	49.0

SEE ALSO:

ConnectApi.CartItemResult

ConnectApi.AbstractContentHubItemType

An item type associated with a repository folder.

This class is abstract.

Superclass of:

- ConnectApi.ContentHubItemTypeDetail
- ConnectApi.ContentHubItemTypeSummary

Property Name	Type	Description	Available Version
`contentStreamSupport`	`ConnectApi.ContentHubStreamSupport`	Support for content streaming. Values are: • `ContentStreamAllowed` • `ContentStreamNotAllowed` • `ContentStreamRequired`	39.0
`description`	String	Description of the item type.	39.0
`displayName`	String	Display name of the item type.	39.0
`id`	String	ID of the item type.	39.0
`isVersionable`	Boolean	Indicates whether the item type can have versions.	39.0
`url`	String	URL to the detailed information of the item type.	39.0

ConnectApi.AbstractDirectoryEntrySummary

A directory entry with summary information.

This class is abstract.

Superclass of:

- ConnectApi.RepositoryGroupSummary
- ConnectApi.RepositoryUserSummary

Property Name	Type	Description	Available Version
`domain`	String	Domain of the directory entry.	39.0

Property Name	Type	Description	Available Version
email	String	Email of the directory entry.	39.0
id	String	ID of the directory entry.	39.0
type	ConnectApi.ContentHubDirectoryEntryType	Type of directory entry. Values are: • GroupEntry • UserEntry	39.0

ConnectApi.AbstractExtensionInformation

Extension information.

This class is abstract.

Superclass of ConnectApi.LightningExtensionInformation.

Property Name	Type	Description	Available Version
extensionInformationType	ConnectApi.ExtensionInformationType	Information type of the extension. Values are: • Lightning	40.0

ConnectApi.AbstractGatewayCommonResponse

Payment gateway response fields commonly used in payment services.

This class is abstract.

Superclass of ConnectApi.AbstractGatewayResponse.

Property Name	Type	Description	Available Version
gatewayAvsCode	String	Used to verify the address mapped to a payment method when the payments platform requests tokenization from the payment gateway.	50.0
gatewayDate	Datetime	Date when the notification occurred. Some gateways don't send this value.	50.0
gatewayMessage	String	Error messages that the gateway returned for the notification request. Maximum length of 255 characters.	50.0
gatewayResultCode	String	Gateway-specific result code. You can map the result code to a Salesforce-specific result code. Maximum length of 64 characters.	50.0
gatewayResultCodeDescription	String	A description of the gateway-specific result code that a payment gateway returned. Maximum length of 1,000 characters.	50.0

Property Name	Type	Description	Available Version
`salesforceResultCode`	String	The Salesforce result code for the gateway result code.	50.0

ConnectApi.AbstractGatewayResponse

Payment gateway response fields used in sale, authorization, and capture services.

This class is abstract.

Subclass of ConnectApi.AbstractGatewayCommonResponse.

Super class of:

- ConnectApi.AuthReversalGatewayResponse
- ConnectApi.AuthorizationGatewayResponse
- ConnectApi.AuthorizationReversalResponse
- ConnectApi.CaptureGatewayResponse
- ConnectApi.PaymentMethodTokenizationGatewayResponse
- ConnectApi.PostAuthGatewayResponse
- ConnectApi.RefundGatewayResponse
- ConnectApi.SaleGatewayResponse

Property Name	Type	Description	Available Version
`gatewayReferenceDetails`	String	Provides information about the gateway communication.	50.0
`gatewayReferenceNumber`	String	Unique transaction ID created by the payment gateway.	50.0

ConnectApi.AbstractManagedContentChannelRepresentation

Managed content channel.

This class is abstract.

Super class of:

- ConnectApi.ManagedContentChannel
- ConnectApi.ManagedContentChannelSummary

No additional properties.

SEE ALSO:

ConnectApi.ManagedContentChannelsRepresentation

ConnectApi.AbstractManagedContentDeliveryDocument

Managed content delivery document.

This class is abstract.

Superclass of:

- ConnectApi.ManagedContentDeliveryDocument
- ConnectApi.ManagedContentDeliveryDocumentSummary

Property Name	Type	Description	Available Version
`contentKey`	String	Globally unique identifier (GUID) for the managed content.	55.0
`contentType`	`ConnectApi.ManagedContentTypeSummary`	Type of managed content.	55.0
`language`	String	Language locale of the managed content.	55.0
`managedContentId`	String	ID of the managed content.	55.0
`publishedDate`	Datetime	Most recent publish date of the managed content.	55.0
`resourceUrl`	String	URL to the single content delivery resource.	55.0
`title`	String	Title of the managed content.	55.0
`unauthenticatedUrl`	String	Public URL for the managed content.	55.0
`urlName`	String	URL name of the managed content.	55.0

SEE ALSO:

ConnectApi.ManagedContentDeliveryDocumentCollection

ConnectApi.AbstractManagedContentReference

Managed content reference.

This class is abstract.

Superclass of:

- ConnectApi.ManagedContentReference
- ConnectApi.ManagedContentReferenceSummary

Property Name	Type	Description	Available Version
`contentKey`	String	Unique identifier for the managed content reference.	54.0
`managedContentId`	String	ID of the managed content reference.	54.0
`resourceUrl`	String	URL to the single content delivery resource.	55.0

SEE ALSO:

ConnectApi.ManagedContentDeliveryDocumentCollection

ConnectApi.AbstractMessageBody

Abstract message body.

This class is abstract.

Superclass of:

- ConnectApi.FeedBody
- ConnectApi.MessageBody

Name	Type	Description	Available Version
`isRichText`	Boolean	Indicates whether the body is rich text.	35.0
`messageSegments`	List<`ConnectApi.MessageSegment`>	List of message segments	28.0
`text`	String	Display-ready text. Use this text if you don't want to process the message segments.	28.0

ConnectApi.AbstractNBAAction

A recommended action of recommendation strategy.

This class is abstract.

Superclass of ConnectApi.NBAFlowAction.

Property Name	Type	Description	Available Version
`parameters`	List<`ConnectApi.NBAActionParameter`>	List of parameters to pass to the action.	45.0
`type`	`ConnectApi.NBAActionType`	Type of action. Values are: • `Flow`—Automated process tool with multiple subtypes.	45.0

SEE ALSO:

ConnectApi.NBARecommendation

ConnectApi.AbstractNBATarget

A recommendation target of a recommendation strategy.

This class is abstract.

Superclass of ConnectApi.NBANativeRecommendation.

Property Name	Type	Description	Available Version
type	ConnectApi.NBATargetType	Type of target. Values are: • Recommendation	45.0

SEE ALSO:

ConnectApi.NBARecommendation

ConnectApi.AbstractRecommendation

A Chatter, custom, or static recommendation.

This class is abstract.

Superclass of:

- ConnectApi.EntityRecommendation
- ConnectApi.NonEntityRecommendation

  ConnectApi.NonEntityRecommendation isn't used in version 34.0 and later. In version 34.0 and later, ConnectApi.EntityRecommendation is used for all recommendations.

Property Name	Type	Description	Available Version
explanation	ConnectApi.RecommendationExplanation	The Chatter, custom, or static recommendation explanation.	32.0
platformActionGroup	ConnectApi.PlatformActionGroup	A platform action group instance with state appropriate for the context user.	34.0
recommendationType	ConnectApi.RecommendationType	Specifies the type of record being recommended.	32.0
url	String	URL for the Chatter, custom, or static recommendation.	34.0

SEE ALSO:

ConnectApi.RecommendationsCapability

ConnectApi.RecommendationCollection

ConnectApi.AbstractRecommendationExplanation

Explanation for a Chatter recommendation.

This class is abstract.

Superclass of ConnectApi.RecommendationExplanation.

Property Name	Type	Description	Available Version
`summary`	String	Summary explanation for the Chatter recommendation.	32.0
`type`	`ConnectApi.RecommendationExplanationType`	Indicates the reason for the Chatter recommendation. • `ArticleHasRelatedContent`—Articles with related content to a context article. • `ArticleViewedTogether`—Articles often viewed together with the article that the context user just viewed. • `ArticleViewedTogetherWithViewers`—Articles often viewed together with other records that the context user views. • `Custom`—Custom recommendations. • `FilePopular`—Files with many followers or views. • `FileViewedTogether`—Files often viewed at the same time as other files that the context user views. • `FollowedTogetherWithFollowees`—Users often followed together with other records that the context user follows. • `GroupMembersFollowed`—Groups with members that the context user follows. • `GroupNew`—Recently created groups. • `GroupPopular`—Groups with many active members. • `ItemViewedTogether`—Records often viewed at the same time as other records that the context user views. • `PopularApp`—Applications that are popular. • `RecordOwned`—Records that the context user owns. • `RecordParentOfFollowed`—Parent records of records that the context user follows. • `RecordViewed`—Records that the context user recently viewed. • `TopicFollowedTogether`—Topics often followed together with the record that the context user just followed. • `TopicFollowedTogetherWithFollowees`—Topics often followed together with other records that the context user follows.	32.0

Property Name	Type	Description	Available Version
		• `TopicPopularFollowed`—Topics with many followers. • `TopicPopularLiked`—Topics on posts that have many likes. • `UserDirectReport`—Users who report to the context user. • `UserFollowedTogether`—Users often followed together with the record that the context user followed . • `UserFollowsSameUsers`—Users who follow the same users as the context user. • `UserManager`—The context user's manager. • `UserNew`—Recently created users. • `UserPeer`—Users who report to the same manager as the context user. • `UserPopular`—Users with many followers. • `UserViewingSameRecords`—Users who view the same records as the context user.	

ConnectApi.AbstractRecordField

A field on a record.

This class is abstract.

Superclass of:

- ConnectApi.BlankRecordField
- ConnectApi.LabeledRecordField

Message segments in a feed item are typed as `ConnectApi.MessageSegment`. Feed item capabilities are typed as `ConnectApi.FeedItemCapability`. Record fields are typed as `ConnectApi.AbstractRecordField`. These classes are all abstract and have several concrete subclasses. At runtime you can use `instanceof` to check the concrete types of these objects and then safely proceed with the corresponding downcast. When you downcast, you must have a default case that handles unknown subclasses.

Important: The composition of a feed can change between releases. Write your code to handle instances of unknown subclasses.

Name	Type	Description	Available Version
`type`	String	Type of the field. One of these values: • `Address` • `Blank` • `Boolean` • `Compound` • `CreatedBy` • `Date` • `DateTime` • `Email` • `LastModifiedBy` • `Location` • `Name` • `Number` • `Percent` • `Phone` • `Picklist` • `Reference` • `Text` • `Time`	29.0

SEE ALSO:

ConnectApi.RecordViewSection

ConnectApi.AbstractRecordView

A view of any record in the org, including a custom object record. This object is used if a specialized object, such as User or ChatterGroup, isn't available for the record type.

This class is abstract.

Subclass of ConnectApi.ActorWithId.

Superclass of:

- ConnectApi.RecordSummary
- ConnectApi.RecordView

Name	Type	Description	Available Version
`name`	String	The localized name of the record.	29.0

ConnectApi.AbstractRepositoryFile

A repository file.

This class is abstract.

Subclass of ConnectApi.AbstractRepositoryItem.

Superclass of:

- ConnectApi.RepositoryFileDetail
- ConnectApi.RepositoryFileSummary

Property Name	Type	Description	Available Version
`checkinComment`	String	Checkin comment of the file.	39.0
`contentBody`	String	Text of the file's content if available, otherwise `null`.	43.0
`contentSize`	Integer	Length in bytes of the content of the file.	39.0
`downloadUrl`	String	URL to the repository file content.	39.0
`external ContentUrl`	String	URL of this file's content in the external system.	39.0
`external DocumentUrl`	String	URL of this file in the external system.	39.0
`external FilePermission Information`	`ConnectApi. ExternalFile PermissionInformation`	External file permission information, such as available groups, available permission types, and current sharing status, or `null` if `includeExternalFilePermissionsInfo` is `false`.	39.0
`mimeType`	String	Mime type of the file.	39.0
`previewUrl Thumbnail`	String	URL to the thumbnail preview (240 x 180 PNG).	39.0
`previewUrl ThumbnailBig`	String	URL to the big thumbnail preview (720 x 480 PNG).	39.0
`previewUrl ThumbnailTiny`	String	URL to the tiny thumbnail preview (120 x 90 PNG).	39.0
`previewsUrl`	String	URL to the previews.	39.0
`title`	String	Title of the file.	39.0
`versionId`	String	ID of the file version in the external system.	39.0

ConnectApi.AbstractRepositoryFolder

A repository folder.

This class is abstract.

Subclass of ConnectApi.AbstractRepositoryItem.

Superclass of:

- ConnectApi.RepositoryFolderDetail
- ConnectApi.RepositoryFolderSummary

Property Name	Type	Description	Available Version
`externalFolderUrl`	String	URL of this folder in the external system.	39.0
`folderItemsUrl`	String	URL that lists the files and folders in this folder.	39.0
`path`	String	Absolute path of the folder in the external system.	39.0

ConnectApi.AbstractRepositoryItem

A repository item.

This class is abstract.

Superclass of:

- ConnectApi.AbstractRepositoryFile
- ConnectApi.AbstractRepositoryFolder

Property Name	Type	Description	Available Version
`createdBy`	String	Name of the user who created the item.	39.0
`createdDate`	Datetime	Date the item was created.	39.0
`description`	String	Description of the Item.	39.0
`id`	String	ID of the item.	39.0
`itemTypeUrl`	String	URL to the item type information.	39.0
`modifiedBy`	String	Name of the user who last modified the item.	39.0
`modifiedDate`	Datetime	Date the item was last modified.	39.0
`motif`	`ConnectApi.Motif`	Motif of the item.	39.0
`name`	String	Name of the item.	39.0
`repository`	`ConnectApi.Reference`	Item external repository.	39.0
`type`	String	Item type, `file` or `folder`.	39.0
`url`	String	The URL to the item.	39.0

ConnectApi.AbstractUserMissionActivity

User activity associated with missions.

This class is abstract.

Superclass of:

- ConnectApi.UserMission
- ConnectApi.UserMissionActivity

Property Name	Type	Description	Available Version
`activityCount`	Integer	Number of mission activities of the specified type for the user.	45.0
`activityType`	String	Type of mission activity for a user. Values are: • `FeedItemAnswerAQuestion`—User answered a question. • `FeedItemLikeSomething`—User liked a post or comment. • `FeedItemMarkAnswerAsBest`—User marked an answer as the best answer. • `FeedItemPostQuestion`—User posted a question. • `FeedItemReceiveAComment`—User received a comment on a post. • `FeedItemReceiveALike`—User received a like on a post or comment. • `FeedItemReceiveAnAnswer`—User received an answer to a question. • `FeedItemWriteAComment`—User commented on a post. • `FeedItemWriteAPost`—User made a post. • `FeedItemYourAnswerMarkedBest`—User's answer was marked as the best answer.	45.0

SEE ALSO:

ConnectApi.UserMissionActivityCollection

ConnectApi.ActionLinkDefinition

The definition of an action link. Action link definition can be sensitive to a third party (for example, OAuth bearer token headers). For this reason, only calls made from the Apex namespace that created the action link definition can read, modify, or delete the definition. In addition, the user making the call must have created the definition or have View All Data permission.

Property Name	Type	Description	Available Version
`actionUrl`	String	The action link URL. For example, a `Ui` action link URL is a Web page. A `Download` action link URL is a link to the file to download. `Ui` and `Download` action link URLs are provided to clients. An `Api` or	33.0

Property Name	Type	Description	Available Version
		`ApiAsync` action link URL is a REST resource. `Api` and `ApiAsync` action link URLs aren't provided to clients. Links to Salesforce can be relative. All other links must be absolute and start with `https://`.	
`createdDate`	Datetime	ISO 8601 format date string, for example, 2011-02-25T18:24:31.000Z.	33.0
`excludedUserId`	String	ID of a single user to exclude from performing the action. If you specify an `excludedUserId`, you can't specify a `userId`.	33.0
`groupDefault`	Boolean	`true` if this action is the default action link in the action link group; `false` otherwise. There can be only one default action link per action link group. The default action link gets distinct styling in the Salesforce UI.	33.0
`headers`	List<`ConnectApi.RequestHeader`>	The request headers for the `Api` and `ApiAsync` action link types.	33.0
`id`	String	The 18-character ID for the action link definition.	33.0
`label`	String	A custom label to display on the action link button. A `label` value can be set only in an action link template. Action links have four statuses: NewStatus, PendingStatus, SuccessStatus, and FailedStatus. These strings are appended to the label for each status: • *label* • *label* Pending • *label* Success • *label* Failed For example, if the value of `label` is "See Example," the values of the four action link states are: See Example, See Example Pending, See Example Success, and See Example Failed. An action link can use either `label` or `labelKey` to generate label names, it can't use both. If `label` has a value, the value of `labelKey` is `None`. If `labelKey` has a value other than `None`, the value of `label` is `null`.	34.0
`labelKey`	String	Key for the set of labels to show in the user interface. A set includes labels for these states: NewStatus, PendingStatus, SuccessStatus, FailedStatus. For example, if you use the `Approve` key, you get these labels: Approve, Pending, Approved, Failed.	33.0

Property Name	Type	Description	Available Version
		For a complete list of label keys, see Action Links Labels in the *Connect REST API Developer Guide*.	
`method`	`ConnectApi.HttpRequestMethod`	The HTTP method. One of these values: • `HttpDelete`—Returns HTTP 204 on success. Response body or output class is empty. • `HttpGet`—Returns HTTP 200 on success. • `HttpHead`—Returns HTTP 200 on success. Response body or output class is empty. • `HttpPatch`—Returns HTTP 200 on success or HTTP 204 if the response body or output class is empty. • `HttpPost`—Returns HTTP 201 on success or HTTP 204 if the response body or output class is empty. Exceptions are the batch posting resources and methods, which return HTTP 200 on success. • `HttpPut`—Return HTTP 200 on success or HTTP 204 if the response body or output class is empty.	33.0
`modifiedDate`	Datetime	ISO 8601 format date string, for example, 2011-02-25T18:24:31.000Z.	33.0
`requestBody`	String	The request body for `Api` and `ApiAsync` action link types. Note: Escape quotation mark characters in the `requestBody` value.	33.0
`requires Confirmation`	Boolean	`true` to require the user to confirm the action; `false` otherwise.	33.0
`templateId`	String	The ID of the action link template from which to instantiate this action link. If the action link isn't associated with a template, the value is `null`.	33.0
`type`	`ConnectApi.ActionLinkType`	Defines the type of action link. Values are: • `Api`—The action link calls a synchronous API at the action URL. Salesforce sets the status to `SuccessfulStatus` or `FailedStatus` based on the HTTP status code returned by your server. • `ApiAsync`—The action link calls an asynchronous API at the action URL. The action remains in a `PendingStatus` state until a third party makes a request to	33.0

Property Name	Type	Description	Available Version
		`/connect/action-links/`***`actionLinkId`*** to set the status to `SuccessfulStatus` or `FailedStatus` when the asynchronous operation is complete. • `Download`—The action link downloads a file from the action URL. • `Ui`—The action link takes the user to a web page at the action URL. Note: Invoking `ApiAsync` action links from an app requires a call to set the status. However, there isn't currently a way to set the status of an action link using Apex. To set the status, use Connect REST API. See the Action Link resource in the Connect REST API Developer Guidefor more information.	
`userId`	String	The ID of the user who can execute the action. If not specified or `null`, any user can execute the action. If you specify a `userId`, you can't specify an `excludedUserId`.	33.0

SEE ALSO:

ConnectApi.ActionLinkGroupDefinition

ConnectApi.ActionLinkDiagnosticInfo

Any diagnostic information that may exist for an executed action link. Diagnostic info is provided only for users who can access the action link.

Property Name	Type	Description	Available Version
`diagnosticInfo`	String	Any diagnostic information returned when an action link is executed. Diagnostic information is provided only for users who can access the action link.	33.0
`url`	String	The URL for this action link diagnostic information.	33.0

ConnectApi.ActionLinkGroupDefinition

The definition of an action link group. Information in the action link group definition can be sensitive to a third party (for example, OAuth bearer token headers). For this reason, only calls made from the Apex namespace that created the action link group definition can read, modify, or delete the definition. In addition, the user making the call must have created the definition or have View All Data permission.

Property Name	Type	Description	Available Version
actionLinks	List<ConnectApi.ActionLinkDefinition>	A collection of action link definitions that make up the action link group. Within an action link group, action links are displayed in the order listed in the actionLinks property of the ConnectApi.ActionLinkGroupDefinitionInput class. Within a feed item, action link groups are displayed in the order specified in the actionLinkGroupIds property of the ConnectApi.AssociatedActionsCapabilityInput class.	33.0
category	ConnectApi.PlatformActionGroupCategory	Indicates the priority and location of the action links. Values are: • Primary—The action link group is displayed in the body of the feed element. • Overflow—The action link group is displayed in the overflow menu of the feed element.	33.0
createdDate	Datetime	ISO 8601 date string, for example, 2011-02-25T18:24:31.000Z.	33.0
executionsAllowed	ConnectApi.ActionLinkExecutionsAllowed	Defines the number of times an action link can be executed. Values are: • Once—An action link can be executed only one time across all users. • OncePerUser—An action link can be executed only one time for each user. • Unlimited—An action link can be executed an unlimited number of times by each user. If the action link's actionType is Api or ApiAsync, you can't use this value.	33.0
expirationDate	Datetime	ISO 8601 date string, for example, 2011-02-25T18:24:31.000Z, that represents the date and time this action group expires and can no longer be executed. If the value is null, there isn't an expiration date.	33.0
id	String	18-character ID of the action link group definition.	33.0
modifiedDate	Datetime	ISO 8601 date string, for example, 2011-02-25T18:24:31.000Z.	33.0
templateId	String	The ID of the action link group template from which to instantiate this action link group, or null if this group isn't associated with a template.	33.0
url	String	The URL for this action link group definition.	33.0

ConnectApi.ActivitySharingResult

The results of sharing a captured email or event.

Property Name	Type	Description	Available Version
`success`	Boolean	Whether the share operation succeeded or not.	39.0

ConnectApi.Actor

Actor.

This class is abstract.

Superclass of:

- ConnectApi.ActorWithId
- ConnectApi.RecommendedObject
- ConnectApi.UnauthenticatedUser

Name	Type	Description	Available Version
`name`	String	Name of the actor, such as the group name.	28.0
`type`	String	One of the following: • `file` • `group` • `recommendedObject` (version 34.0 and later) • `unauthenticateduser` • `user` • *`record type name`*—the name of the record type, such as `myCustomObject__c` or Account	28.0

SEE ALSO:

ConnectApi.CaseCommentCapability
ConnectApi.EntityRecommendation
ConnectApi.EditCapability
ConnectApi.FeedEntitySummary
ConnectApi.FeedItem
ConnectApi.FeedItemSummary
ConnectApi.Subscription

ConnectApi.ActorWithId

Actor with ID.

This class is abstract.

Subclass of ConnectApi.Actor.

Superclass of:

- ConnectApi.AbstractRecordView
- ConnectApi.ArticleSummary
- ConnectApi.ChatterGroup
- ConnectApi.ContentHubRepository
- ConnectApi.File
- ConnectApi.RelatedFeedPost
- ConnectApi.User

Name	Type	Description	Available Version
`id`	String	Actor's 18-character ID	28.0
`motif`	`ConnectApi.Motif`	An icon that identifies the actor as a user, group, file, or custom object. The icon isn't the user or group photo, and it isn't a preview of the file. The motif can also contain the object's base color.	28.0
`mySubscription`	`ConnectApi.Reference`	If the context user is following the item, this contains information about the subscription, else returns `null`.	28.0
`url`	String	Connect REST API URL for the resource	28.0

SEE ALSO:

ConnectApi.FeedElement
ConnectApi.FeedEntitySummary
ConnectApi.GroupRecord
ConnectApi.MentionSegment
ConnectApi.RecordSummaryList

ConnectApi.Address

Address.

Name	Type	Description	Available Version
`city`	String	Name of the city	28.0
`country`	String	Name of the country	28.0
`formattedAddress`	String	Formatted address per the locale of the context user	28.0
`state`	String	Name of the state, province, or so on	28.0
`street`	String	Street number	28.0

Name	Type	Description	Available Version
`zip`	String	Zip or postal code	28.0

SEE ALSO:

- ConnectApi.DatacloudCompany
- ConnectApi.DatacloudContact
- ConnectApi.UserDetail

ConnectApi.AdjustOrderSummaryOutputRepresentation

Output representation of the financial changes for an adjust items action. For a preview action, these values are the expected output. For a submit action, these values are the actual output.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
`changeBalances`	`ConnectApi.ChangeItemOutputRepresentation`	Expected (for preview) or actual (for submit) financial values for the price adjustment action. Most of the values match the change order values. If two change orders are returned, then these values combine them. The sign of a value in this output is the opposite of the corresponding value on a change order record. For example, a discount is a positive value in `changeBalances` and a negative value on a change order record.	49.0
`inFulfillmentChangeOrderId`	String	ID of the change Order that holds the financial changes applicable to OrderItemSummary quantities that are in the process of being fulfilled. This change Order is only created for a request that specified an `allocatedItemsChangeOrderType` of InFulfillment. For an adjustPreview call, this value is always null.	55.0
`orderSummaryId`	String	ID of the OrderSummary.	49.0
`postFulfillmentChangeOrderId`	String	ID of the change Order that holds the financial changes applicable to OrderItemSummary quantities that have been fulfilled. For an adjustPreview call, this value is always null.	49.0
`preFulfillmentChangeOrderId`	String	ID of the change Order that holds the financial changes applicable to OrderItemSummary quantities that have not been fulfilled. If the request specified an `allocatedItemsChangeOrderType` of PreFulfillment, this change Order also includes the changes applicable to OrderItemSummary quantities	49.0

Property Name	Type	Description	Available Version
		that are in the process of being fulfilled. For an adjustPreview call, this value is always null.	

ConnectApi.Alternative

Alternative representation for an extension on a feed element.

Property Name	Type	Description	Available Version
`textRepresentation`	String	Text representation of the extension.	40.0
`thumbnailUrl`	String	Thumbnail URL to the extension.	40.0
`title`	String	Title of the extension.	40.0

ConnectApi.AlternativePaymentMethodOutput

Alternative payment method details output.

Property Name	Type	Description	Available Version
`accountId`	String	Salesforce Payments account to which this payment method is linked.	56.0
`comments`	String	Details about a record added by a user. Maximum of 1,000 characters.	56.0
`email`	String	Email address of the card holder.	56.0
`gatewayToken`	String	A unique, alphanumeric ID, called a token, that a payment gateway generates when it first processes a payment. The token replaces the actual payment data so that the data is kept secure. This token is stored as encrypted text, and can be used for recurring payments.	56.0
`gatewayTokenDetails`	String	Detailed information about the gateway token.	56.0
`name`	String	Name that you assign to the payment method object.	56.0

ConnectApi.Announcement

An announcement displays in a designated location in the Salesforce UI until 11:59 p.m. on its expiration date, unless it's deleted or replaced by another announcement.

Name	Type	Description	Available Version
`expirationDate`	Datetime	The Salesforce UI displays an announcement until 11:59 p.m. on this date unless another announcement is posted first. The Salesforce UI ignores the time value in the `expirationDate`. However, you can use the time value to create your own display logic in your own UI.	31.0
`feedElement`	`ConnectApi.FeedElement`	The feed element that contains the body of the announcement and its associated comments, likes, and so on.	31.0
`id`	String	18-character ID of the announcement.	31.0
`isArchived`	Boolean	Specifies whether the announcement is archived.	36.0
`sendEmails`	Boolean	Specifies whether the announcement is sent as an email to all group members.	36.0
`url`	String	The URL to the announcement.	33.0

SEE ALSO:

ConnectApi.AnnouncementPage

ConnectApi.ChatterGroup

ConnectApi.AnnouncementPage

A collection of announcements.

Name	Type	Description	Available Version
`announcements`	`List<ConnectApi.Announcement>`	A collection of `ConnectApi.Announcement` objects.	31.0
`currentPageUrl`	String	Connect REST API URL identifying the current page.	31.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	31.0
`previousPageUrl`	String	Connect REST API URL identifying the previous page, or `null` if there isn't a previous page.	31.0

ConnectApi.SearchAppliedOrderBy

The applied order for object search.

Property Name	Type	Description	Available Version
`field`	String	Field used to sort the results.	63.0

Property Name	Type	Description	Available Version
`order`	`ConnectApi.OrderDirection`	Order direction. Values are: • `Ascending` • `Descending`	63.0
`orderNulls`	`ConnectApi.OrderNulls`	Null value order. Values are: • `Firsts`—Null values are sorted first. • `Lasts`—Null values are sorted last.	63.0

SEE ALSO:

ConnectApi.SearchObject

ConnectApi.ApprovalCapability

If a feed element has this capability, it includes information about an approval.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
`id`	String	The work item ID. The work item ID is `null` if there isn't a pending work item associated with the approval record.	32.0
`postTemplateFields`	List<`ConnectApi.ApprovalPostTemplateField`>	The details of the approval post template field.	32.0
`processInstanceStepId`	String	The process instance step ID. The associated record represents one step in an approval process.	32.0
`status`	`ConnectApi.WorkflowProcessStatus`	The status of the approval.	32.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.ApprovalIntent

Approval intent for a social post.

Property Name	Type	Description	Available Version
`isRecallable`	Boolean	Specifies whether the social post can be recalled (`true`) or not (`false`).	45.0

SEE ALSO:

ConnectApi.SocialPostIntents

ConnectApi.ApprovalPostTemplateField

Approval post template field.

Name	Type	Description	Available Version
`displayName`	String	The field name.	28.0
`displayValue`	String	The field value or `null` if the field is set to `null`.	28.0
`record`	`ConnectApi.Reference`	A record ID. If no record exists or if the reference is `null`, this value is `null`.	28.0

SEE ALSO:

ConnectApi.ApprovalCapability

ConnectApi.ArticleItem

Article item in question and answers suggestions.

Property Name	Type	Description	Available Version
`id`	String	Id of the article.	32.0
`rating`	Double	The rating of the article.	32.0
`title`	String	Title of the article.	32.0
`urlLink`	String	Link URL of the article.	32.0
`viewCount`	Integer	Number of votes given to the article.	32.0

SEE ALSO:

ConnectApi.QuestionAndAnswersSuggestions

ConnectApi.ArticleSummary

A knowledge article summary.

Subclass of ConnectApi.ActorWithId.

Property Name	Type	Description	Available Version
`articleType`	String	Type of the knowledge article.	37.0
`knowledgeArticleVersionId`	String	ID of the knowledge article version.	39.0
`lastPublishedDate`	Datetime	Last published date of the knowledge article.	37.0
`rating`	Double	The rating of the article.	37.0
`summary`	String	Summary of the knowledge article contents.	37.0
`title`	String	Title of the knowledge article.	37.0
`urlName`	String	URL name of the knowledge article.	37.0
`viewCount`	Integer	Number of times the knowledge article has been viewed.	38.0

ConnectApi.AssociatedActionsCapability

If a feed element has this capability, it has platform actions associated with it.

Property Name	Type	Description	Available Version
`platformActionGroups`	List<`ConnectApi.PlatformActionGroup`>	The platform action groups associated with a feed element. Platform action groups are returned in the order specified in the `ConnectApi.AssociatedActionsCapabilityInput` class.	33.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.AsyncOutputRepresentation

Output representation of the async operation.

Subclass of ConnectApi.BaseAsyncOutputRepresentation.

No additional properties.

SEE ALSO:

multipleEnsureFundsAsync(multipleEnsureFundsInput)

ConnectApi.MultipleAsyncOutputRepresentation

ConnectApi.Audience

A personalization audience.

Property Name	Type	Description	Available Version
`criteria`	List<`ConnectApi.AudienceCriteriaDetail`>	Criteria details for the audience.	48.0
`customFormula`	String	Custom formula for the audience criteria. For example, (1 AND 2) OR 3.	48.0
`formulaFilterType`	`ConnectApi.FormulaFilterType`	Formula filter type for the personalization audience. Values are: • `AllCriteriaMatch`—All audience criteria are true (AND operation). • `AnyCriterionMatches`—Any audience criterion is true (OR operation). • `CustomLogicMatches`—Audience criteria match the custom formula (for example, (1 AND 2) OR 3).	48.0
`id`	String	ID of the audience.	48.0
`name`	String	Name of the audience.	48.0
`targets`	List<`ConnectApi.AudienceTargetAssignment`>	Target assignments for the audience.	48.0
`url`	String	URL to this audience.	48.0

SEE ALSO:

ConnectApi.AudienceCollection

ConnectApi.AudienceCollection

Collection of personalization audiences.

Property Name	Type	Description	Available Version
`audiences`	List<`ConnectApi.Audience`>	Collection of audiences.	48.0

ConnectApi.AudienceCriteria

Custom recommendation audience criteria.

This class is abstract.

This class is a superclass of:

- ConnectApi.CustomListAudienceCriteria
- ConnectApi.NewUserAudienceCriteria

Property Name	Type	Description	Available Version
type	ConnectApi.RecommendationAudienceCriteriaType	Specifies the custom recommendation audience criteria type. One of these values: • CustomList—A custom list of users makes up the audience. • MaxDaysInCommunity—New members make up the audience.	36.0

SEE ALSO:

ConnectApi.RecommendationAudience

ConnectApi.AudienceCriteriaDetail

Personalization audience criteria.

Property Name	Type	Description	Available Version
criterion	List<ConnectApi.AudienceCriterionDetail>	List of mappings of audience criteria fields and values.	48.0
criterionNumber	Integer	Number associated with the audience criterion in a formula. For example, (1 AND 2) OR 3. If unspecified, criteria are assigned numbers in the order that they're added.	48.0
criterionOperator	ConnectApi.AudienceCriteriaOperator	Operator used in the personalization audience criterion. Values are: • Contains • Equal • GreaterThan • GreaterThanOrEqual • Includes • LessThan • LessThanOrEqual • NotEqual • NotIncludes • StartsWith	48.0

Property Name	Type	Description	Available Version
`criterionType`	`ConnectApi.AudienceCriteriaType`	Type of personalization audience criterion. Values are: • `Audience`—Criterion based on audience. • `Default`—Audience has no criteria. • `Domain`—Criterion based on domain. • `FieldBased`—Criterion based on object fields. • `GeoLocation`—Criterion based on location. • `Permission`—Criterion based on standard or custom permissions. • `Profile`—Criterion based on profile.	48.0

SEE ALSO:

ConnectApi.Audience

ConnectApi.AudienceCriterionDetail

Audience criterion information.

Property Name	Type	Description	Available Version
`value`	Map<String, String>	Mapping of an audience criterion value and field.	48.0

SEE ALSO:

ConnectApi.AudienceCriteriaDetail

ConnectApi.AudienceTarget

Personalization audience assigned to a target.

Property Name	Type	Description	Available Version
`audienceName`	String	Name of the audience assigned to the target.	48.0
`id`	String	ID of the audience assigned to the target.	48.0
`url`	String	URL to the audience assigned to the target.	48.0

SEE ALSO:

ConnectApi.Target

ConnectApi.AudienceTargetAssignment

Target assignments for a personalization audience.

Property Name	Type	Description	Available Version
`formulaScope`	`ConnectApi.FormulaScope`	Formula scope of the target.	51.0
`groupName`	String	Group name of the target. Groups bundle related target and audience pairs.	48.0
`id`	String	ID of the target.	48.0
`isMatch`	Boolean	Specifies whether the target matches the current context (`true`) or doesn't (`false`).	48.0
`priority`	Integer	Priority of the target. Within a group, priority determines which target is returned if the user matches more than one audience.	48.0
`publishStatus`	`ConnectApi.PublishStatus`	Publish status of the target. Values are: • `Draft` • `Live`	48.0
`targetType`	String	Type of target, indicating the nature of the data being targeted.	48.0
`targetValue`	String	Value of the target.	48.0
`url`	String	URL to the target.	48.0

SEE ALSO:

ConnectApi.Audience

ConnectApi.AuthReversalGatewayResponse

Authorization Reversal Gateway Response Representation.

Subclass of ConnectApi.AbstractGatewayResponse.

No additional properties.

ConnectApi.AuthorizationGatewayResponse

Payment gateway authorization response representation.

Subclass of ConnectApi.AbstractGatewayResponse.

Property Name	Type	Description	Available Version
`gatewayAuthorizationCode`	String	Gateway authorization code.	51.0

ConnectApi.AuthorizationResponse

Payment Authorization output representation.

Property Name	Type	Description	Available Version
error	ConnectApi.ErrorResponse	Error representation for the payment authorization.	51.0
gatewayResponse	ConnectApi.AuthorizationGatewayResponse	Gateway response representation for the payment authorization.	51.0
paymentAuthorization	ConnectApi.PaymentAuthorizationResponse	Payment authorization representation.	51.0
paymentGatewayLogs	List<ConnectApi.GatewayLogResponse>	Gateway log list representation for the payment authorization.	51.0
paymentGroup	ConnectApi.PaymentGroupResponse	Payment group representation for the payment authorization.	51.0
paymentMethod	ConnectApi.PaymentMethodResponse	Payment method representation for the payment authorization.	51.0

ConnectApi.AuthorizationReversalResponse

Authorization Reversal output representation.

Property Name	Type	Description	Available Version
error	ConnectApi.ErrorResponse	Error response representation for the authorization reversal.	51.0
gatewayResponse	ConnectApi.AuthReversalGatewayResponse	Gateway response representation for authorization reversal.	51.0
paymentAuthAdjustment	ConnectApi.PaymentAuthAdjustmentResponse	Payment authorization adjustment response representation for the authorization reversal.	51.0
paymentGatewayLogs	List<ConnectApi.GatewayLogResponse>	Gateway log collection representation for the authorization reversal.	51.0

ConnectApi.AvailableLocationOutputRepresentation

A set of inventory locations that can combine to fulfill an order.

Property Name	Type	Description	Available Version
locations	List<String>	A list of inventory locations.	51.0

SEE ALSO:

findRoutesWithFewestSplits(findRoutesWithFewestSplitsInputRepresentation)

findRoutesWithFewestSplitsUsingOCI(findRoutesWithFewestSplitsUsingOCIInput)

ConnectApi.FindRoutesWithFewestSplitsOutputRepresentation

ConnectApi.FindRoutesWithFewestSplitsUsingOCIOutputRepresentation

ConnectApi.AverageDistanceResultOutputRepresentation

Wraps inventory location shipping distance calculation results.

Property Name	Type	Description	Available Version
distanceCalculation	ConnectApi.DistanceCalculationOutputRepresentation	Results of the shipping distance calculations.	51.0

ConnectApi.BalanceStatePreviewOutputRepresentation

The generated preview of all balances for an order or a cart, including totals, adjustments, and taxes.

Property Name	Type	Description	Available Version
grandTotalAmount	Double	The grand total of the exchanges, including adjustments, fees, delivery cost, and taxes.	61.0
totalAdjustmentAmount	Double	The total amount the order was adjusted by, not including tax.	61.0
totalAdjustmentAmountWithTax	Double	The total amount the order was adjusted by, including tax.	61.0
totalAmount	Double	The total amount being return, not including tax.	61.0
totalAmountWithTax	Double	The total amount being return, including tax.	61.0
totalDeliveryAmount	Double	The total cost for delivery, not including tax.	61.0
totalDeliveryAmountWithTax	Double	The total cost for delivery, including tax.	61.0
totalFeeAmount	Double	The combined total of all fees charged, not including tax.	61.0
totalFeeAmountWithTax	Double	The combined total of all fees charged, including tax.	61.0
totalTaxAmount	Double	The combined total of all taxes.	61.0

ConnectApi.BannerCapability

If a feed element has this capability, it has a banner motif and style.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
`motif`	`ConnectApi.Motif`	A banner motif.	31.0
`style`	`ConnectApi.BannerStyle`	Decorates a feed item with a color and set of icons. Possible value: • `Announcement`—An announcement displays in a designated location in the Salesforce UI until 11:59 p.m. on its expiration date, unless it's deleted or replaced by another announcement.	31.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.BannerPhoto

A banner photo.

Property Name	Type	Description	Available Version
`bannerPhotoUrl`	String	URL to the banner photo in a large format. This URL is available only to authenticated users.	36.0
`bannerPhoto VersionId`	String	18-character version ID of the banner photo.	36.0
`url`	String	URL to the banner photo.	36.0

SEE ALSO:

ConnectApi.ChatterGroup

ConnectApi.UserDetail

ConnectApi.BaseAsyncOutputRepresentation

Base Order Management async output class.

This class is abstract.

Subclass of ConnectApi.BaseOutputRepresentation.

Superclass of:

- ConnectApi.AsyncOutputRepresentation
- ConnectApi.EnsureFundsAsyncOutputRepresentation
- ConnectApi.EnsureRefundsAsyncOutputRepresentation

Property Name	Type	Description	Available Version
`backgroundOperationId`	String	ID of the background operation.	48.0

ConnectApi.BaseInvoiceOutputRepresentation

Base Order Management Invoice output class.

This class is abstract.

Subclass of ConnectApi.BaseOutputRepresentation.

Superclass of ConnectApi.ChangeOrdersInvoiceOutputRepresentation.

Property Name	Type	Description	Available Version
`invoiceId`	String	ID of the created invoice.	56.0

ConnectApi.BaseManagedSocialAccount

Base information describing a managed social account or fan page of a social network.

This class is abstract.

Superclass of ConnectApi.ManagedSocialAccount.

Property Name	Type	Description	Available Version
`defaultResponseAccountId`	String	Default response account to use when replying to posts sent to this account.	44.0
`displayName`	String	Real name (or user name if real name not available) for this account on the social network.	44.0
`externalPictureUrl`	String	URL to the account's avatar image.	44.0
`id`	String	Internal SFDC ID for this managed social account.	44.0
`label`	String	Label for the social account.	44.0
`profileUrl`	String	URL to the account's profile.	44.0
`socialNetwork`	`ConnectApi.SocialNetworkProvider`	Social network that this account belongs to. Values are: • `Facebook` • `GooglePlus` • `Instagram` • `InstagramBusiness` • `KakaoTalk` • `Kik` • `Line`	44.0

Property Name	Type	Description	Available Version
		• LinkedIn • Messenger • Other • Pinterest • QQ • Rypple • SinaWeibo • SMS • Snapchat • Telegram • Twitter • VKontakte • WeChat • WhatsApp • YouTube	
uniqueName	String	Unique name used for distinguishing same name fan pages; acts like a user name for a fan page.	44.0
username	String	Unique user name or handle for this account on the social network.	44.0

ConnectApi.BaseOutputRepresentation

Base Order Management output class.

This class is abstract.

Superclass of:

- ConnectApi.AdjustOrderSummaryOutputRepresentation
- ConnectApi.BaseAsyncOutputRepresentation
- ConnectApi.BaseInvoiceOutputRepresentation
- ConnectApi.ConfirmHeldFOCapacityOutputRepresentation
- ConnectApi.CreateCreditMemoOutputRepresentation
- ConnectApi.CreateMultipleInvoicesFromChangeOrdersOutputRepresentation
- ConnectApi.CreateOrderPaymentSummaryOutputRepresentation
- ConnectApi.EnsureFundsAsyncOutputRepresentation
- ConnectApi.EnsureRefundsAsyncOutputRepresentation
- ConnectApi.FindRoutesWithFewestSplitsOutputRepresentation
- ConnectApi.FindRoutesWithFewestSplitsUsingOCIOutputRepresentation
- ConnectApi.FindRoutesWithFewestSplitsWithInventoryOutputRepresentation
- ConnectApi.FulfillmentGroupOutputRepresentation

- ConnectApi.FulfillmentOrderCancelLineItemsOutputRepresentation
- ConnectApi.FulfillmentOrderInvoiceOutputRepresentation
- ConnectApi.FulfillmentOrderOutputRepresentation
- ConnectApi.GetFOCapacityValuesOutputRepresentation
- ConnectApi.HoldFOCapacityOutputRepresentation
- ConnectApi.MultipleAsyncOutputRepresentation
- ConnectApi.MultipleFulfillmentOrderInvoicesOutputRepresentation
- ConnectApi.MultipleFulfillmentOrderOutputRepresentation
- ConnectApi.OrderSummaryOutputRepresentation
- ConnectApi.PreviewCancelOutputRepresentation
- ConnectApi.PreviewReturnOutputRepresentation
- ConnectApi.ProductDetailsOutputRepresentation
- ConnectApi.RankAverageDistanceOutputRepresentation
- ConnectApi.RegisterGuestBuyerOutputRepresentation
- ConnectApi.ReleaseHeldFOCapacityOutputRepresentation
- ConnectApi.ReturnItemsOutputRepresentation
- ConnectApi.ReturnOrderItemSplitLineOutputRepresentation
- ConnectApi.ReturnOrderOutputRepresentation
- ConnectApi.SubmitCancelOutputRepresentation
- ConnectApi.SubmitReturnOutputRepresentation

Property Name	Type	Description	Available Version
`errors`	List<`ConnectApi.ErrorResponse`>	Any errors that were returned.	48.0
`success`	Boolean	Indicates whether the transaction was successful.	48.0

ConnectApi.BatchResult

The result of an operation returned by a batch method.

Usage

Calls to batch methods return a list of `BatchResult` objects. Each element in the `BatchResult` list corresponds to the strings in the list parameter passed to the batch method. The first element in the `BatchResult` list matches the first string passed in the list parameter, the second element corresponds with the second string, and so on. If only one string is passed, the `BatchResult` list contains a single element.

Example

The following example shows how to obtain and iterate through the returned `ConnectApi.BatchResult` objects. The code adds two group IDs to a list. One of group IDs is incorrect, which causes a failure when the code calls the batch method. After it calls the batch method, it iterates through the results to determine whether the operation was successful or not for each group ID in the list. The

code writes the ID of every group that was processed successfully to the debug log. The code writes an error message for every failed group.

This example generates one successful operation and one failure.

```
List<String> myList = new List<String>();
// Add one correct group ID.
myList.add('0F9D00000000oOT');
// Add one incorrect group ID.
myList.add('0F9D00000000izf');

ConnectApi.BatchResult[] batchResults = ConnectApi.ChatterGroups.getGroupBatch(null,
myList);

// Iterate through each returned result.
for (ConnectApi.BatchResult batchResult : batchResults) {
    if (batchResult.isSuccess()) {
        // Operation was successful.
        // Print the group ID.
        ConnectApi.ChatterGroupSummary groupSummary;
        if(batchResult.getResult() instanceof ConnectApi.ChatterGroupSummary) {
           groupSummary = (ConnectApi.ChatterGroupSummary) batchResult.getResult();
        }
        System.debug('SUCCESS');
        System.debug(groupSummary.id);
    }
    else {
        // Operation failed. Print errors.
        System.debug('FAILURE');
        System.debug(batchResult.getErrorMessage());
    }
}
```

IN THIS SECTION:

BatchResult Methods

These are instance methods for `BatchResult`.

BatchResult Methods

These are instance methods for `BatchResult`.

IN THIS SECTION:

getError()

If an error occurred, returns a `ConnectApi.ConnectApiException` object providing the error code and description.

getErrorMessage()

Returns a String that contains an error message.

getErrorTypeName()

Returns a String that contains the name of the error type.

getResult()

Returns an object that contains the results of the batch operation. The object is typed according to the batch method. For example, if you call `getMembershipBatch()`, a successful call to BatchResult `getResult()` returns a `ConnectApi.GroupMembership` object.

isSuccess()

Returns a Boolean that is set to `true` if the batch operation was successful for this object, `false` otherwise.

getError()

If an error occurred, returns a `ConnectApi.ConnectApiException` object providing the error code and description.

Signature

```
public ConnectApi.ConnectApiException getError()
```

Return Value

Type: `ConnectApi.ConnectApiException`

getErrorMessage()

Returns a String that contains an error message.

Signature

```
public String getErrorMessage()
```

Return Value

Type: String

Usage

The error message doesn't make a round trip through a Visualforce view state, because exceptions can't be serialized.

getErrorTypeName()

Returns a String that contains the name of the error type.

Signature

```
public String getErrorTypeName()
```

Return Value

Type: String

getResult()

Returns an object that contains the results of the batch operation. The object is typed according to the batch method. For example, if you call `getMembershipBatch()`, a successful call to BatchResult `getResult()` returns a `ConnectApi.GroupMembership` object.

Signature

```
public Object getResult()
```

Return Value

Type: Object

isSuccess()

Returns a Boolean that is set to `true` if the batch operation was successful for this object, `false` otherwise.

Signature

```
public Boolean isSuccess()
```

Return Value

Type: Boolean

ConnectApi.BlankRecordField

Record field displayed as a place holder in a grid of fields.

Subclass of ConnectApi.AbstractRecordField.

ConnectApi.PromotionBonusProduct

Bonus product for a promotion.

Property Name	Type	Description	Available Version
`adjustmentBasisReference`	String	ID of the associated coupon, if applicable.	58.0
`bonusProductId`	String	ID of the bonus product.	58.0
`causeId`	String	ID of the related promotion.	58.0
`qualifyingItems`	List<`ConnectApi.PromotionCartItemKey` on page 2004>	List of qualifying cart items and their related quantity.	58.0

ConnectApi.BookmarksCapability

If a feed element has this capability, the context user can bookmark it.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
`isBookmarkedByCurrentUser`	Boolean	Indicates whether the feed element has been bookmarked by the context user (`true`) or not (`false`).	32.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.BookmarkSummary

Summary of a bookmark.

Subclass of ConnectApi.UserFeedEntityActivitySummary.

No additional properties.

ConnectApi.BotInfoRepresentation

Information about the bot associated with the conversation application.

Property Name	Type	Description	Available Version
`botId`	String	ID of the bot.	54.0
`botName`	String	Name of the bot.	54.0
`lastModifiedDate`	Datetime	Last modified date of the bot definition.	54.0

ConnectApi.BotVersionActivationInfo

Success or failure information of the bot version activation.

Property Name	Type	Description	Available Version
`isActivated`	Boolean	Indicates whether the bot is active or not.	51.0
`messages`	List<String>	Failure messages.	50.0
`success`	Boolean	Indicates whether the activation was successful or not.	50.0

ConnectApi.BundleCapability

If a feed element has this capability, it has a container of feed elements called a *bundle*.

This class is abstract.

Subclass of ConnectApi.FeedElementCapability.

Superclass of:

- ConnectApi.GenericBundleCapability
- ConnectApi.TrackedChangeBundleCapability

Property Name	Type	Description	Available Version
`bundleType`	`ConnectApi.BundleType`	Defines this feed element's bundle type. The bundle type determines what additional information appears in the bundle.	31.0
`page`	`ConnectApi.FeedElementPage`	A collection of feed elements.	31.0
`totalElements`	Integer	The total number of feed elements that this bundle aggregates.	31.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.CalculateCartResult

Result of a cart calculate request. Includes a cart summary with calculated cart values.

Subclass of ConnectApi.CommerceResultRepresentationBase

Property Name	Type	Description	Available Version
`data`	ConnectApi.CartSummary	A cart summary with calculated cart values.	62.0

ConnectApi.CalculateTaxResponse

Shows the results of a tax calculation request.

Subclass of ConnectApi.TaxTransactionResponse.

Property Name	Type	Description	Available Version
`adapterError`	`ConnectApi.ErrorResponse`	Adapter error.	55.0
`status`	`ConnectApi.TaxTransactionStatus`	Status of a tax transaction. Values are: • `Committed`—Tax has been committed to the transaction. • `Uncommitted`—Tax hasn't been committed to the transaction.	55.0

Property Name	Type	Description	Available Version
taxEngineLogs	List<ConnectApi.TaxEngineLogResponse>	Tax engine logs.	55.0
taxTransactionType	ConnectApi.TaxTransactionType	Type of tax transaction. Values are: • Credit—Transaction is a credit transaction. • Debit—Transaction is a debit transaction.	55.0
taxType	ConnectApi.CalculateTaxType	Type of tax calculation. Values are: • Actual—Calculated tax represents the final taxed amount for the transaction. • Estimated—Calculated tax represents only an estimated value before the transaction is finalized.	55.0

ConnectApi.CallCollaborationCapability

If a feed element has this capability, it has a recording comment.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
commentEndTime	Integer	End time of the comment on the media player, in seconds.	51.0
commentStartTime	Integer	Start time of the comment on the media player, in seconds.	51.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.CancelAllOrderItemsAsyncOutputRepresentation

ID of the asynchronous background operation.

Subclass of ConnectApi.BaseAsyncOutputRepresentation on page 1986.

Property Name	Type	Description	Available Version
asyncOperationLogId	String	ID of the background operation.	63.0

ConnectApi.CandidateAnswersStatus

The status of candidate answers on a feed element.

Property Name	Type	Description	Available Version
`hasCandidateAnswers`	Boolean	Indicates whether candidate answers are available for a question.	41.0
`hasCandidateAnswersPublished`	Boolean	Indicates whether any candidate answers are published.	41.0
`hasCandidateAnswersRated`	Boolean	Indicates whether any candidate answers are rated.	41.0

SEE ALSO:

ConnectApi.QuestionAndAnswersCapability

ConnectApi.CanvasCapability

If a feed element has this capability, it renders a canvas app.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
`description`	String	A description of the canvas app. The maximum size is 255 characters.	32.0
`developerName`	String	The API name (developer name) of the connected app.	32.0
`height`	String	The height of the canvas app in pixels.	32.0
`icon`	`ConnectApi.Icon`	The icon for the canvas app.	32.0
`namespacePrefix`	String	A unique namespace prefix for the canvas app.	32.0
`parameters`	String	JSON parameters passed to the canvas app.	32.0
`thumbnailUrl`	String	A thumbnail URL to a preview image. The maximum thumbnail size is 120 pixels by 120 pixels.	32.0
`title`	String	A title for the canvas link.	32.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.CapacityResponseOutputRepresentation

Response to a request related to a location's fulfillment capacity.

Property Name	Type	Description	Available Version
`actionRequestId`	String	Unique string that identifies the original capacity request.	55.0

Property Name	Type	Description	Available Version
`error`	`ConnectApi.ErrorResponse`	Error returned by the request, if any.	55.0
`success`	Boolean	Indicates whether the request was successful.	55.0

ConnectApi.CaptureGatewayResponse

Gateway capture response.

Subclass of ConnectApi.AbstractGatewayResponse.

No additional properties.

ConnectApi.CaptureResponse

Capture output.

Property Name	Type	Description	Available Version
`error`	`ConnectApi.ErrorResponse`	Error response representation for an authorization capture.	50.0
`gatewayResponse`	`ConnectApi.CaptureGatewayResponse`	Gateway log response containing details about gateway logs created during the process of the capture request.	50.0
`payment`	`ConnectApi.PaymentResponse`	Payment response object for the capture request. Contains the information related to a payment object created during request processing.	50.0
`paymentGatewayLogs`	List<`ConnectApi.GatewayLogResponse`>	Gateway log collection for an authorization capture.	50.0
`paymentGroup`	`ConnectApi.PaymentGroupResponse`	Payment group associated with the capture request.	50.0

ConnectApi.CardPaymentMethodOutput

Card payment method details output.

Property Name	Type	Description	Available Version
`accountId`	String	Salesforce Payments account to which this payment method is linked.	56.0
`autoPay`	Boolean	Indicates whether a token for recurring payments is being requested (`true`) or not (`false`). The token lets the payment method be used for recurring payments.	56.0

Property Name	Type	Description	Available Version
`cardBin`	String	Bank Identification Number (BIN). The BIN is the first 4-6 numbers on a payment card that identifies the card issuer.	56.0
`cardCategory`	ConnectApi.CardCategory	• `CreditCard` • `DebitCard`	56.0
`cardHolderFirstName`	String	First name of the card holder	56.0
`cardHolderLastName`	String	Last name of the card holder	56.0
`cardHolderName`	String	Full name of card holder	56.0
`cardLastFour`	String	Last four digits on a card.	56.0
`cardType`	ConnectApi.CardType	Credit card issuer. • `AmericanExpress` • `DinersClub` • `JCB` • `Maestro` • `MasterCard` • `Visa`	56.0
`comments`	String	Details about a record added by a user. Maximum of 1,000 characters.	56.0
`displayCardNumber`	String	Card displayed number	56.0
`email`	String	Email address of the card holder.	56.0
`expiryMonth`	Integer	Card expiration month	56.0
`expiryYear`	Integer	Card expiration year	56.0
`nickName`	String	Optional card nickname	56.0
`startMonth`	Integer	Month when card becomes active	56.0
`startYear`	Integer	Year when card becomes active	56.0

ConnectApi.CartCoupon

Cart Coupon representation.

Property Name	Type	Description	Available Version
`cartCouponId`	String	ID of the cart coupon code.	54.0
`couponCode`	String	The coupon code a buyer can use to manually apply a promotion to the cart.	54.0

ConnectApi.CartCouponCollection

Collection of coupons related to a cart.

Property Name	Type	Description	Available Version
`cartCoupons`	ConnectApi.CartCouponList	Collection of coupons.	54.0
`cartId`	String	ID of the cart.	54.0
`cartStatus`	`ConnectApi.CartStatus`	Status of the cart. Values are: • `Active`—Cart is created and available for modifications, like adding or removing products or promotions. • `Checkout`—Cart is in checkout. If the customer modifies the cart, the current checkout session is canceled. • `Closed`—Checkout is complete and an order was created. The cart cannot be modified. • `PendingClosed`—Cart is marked to be closed, but the request isn't completed yet. The cart can't be modified. This value is available in API version 57.0 and later. • `PendingDelete`—Cart is marked for delete, but the request isn't completed yet. The cart can't be modified. • `Processing`—Cart is processing. For example, taxes are being calculated. The cart can't be modified.	54.0
`ownerId`	String	ID of the user who owns the cart.	54.0

ConnectApi.CartCouponList

List of coupons for a cart.

Property Name	Type	Description	Available Version
`coupons`	List<ConnectApi.CartCoupon>	List of coupons associated with a cart.	54.0

ConnectApi.PromotionCartDeliveryGroupMethod

Cart delivery group method for a promotion.

Property Name	Type	Description	Available Version
`cartDeliveryMethodAdjustments`	List<`ConnectApi.PromotionCart`	List of cart delivery method adjustments.	60.0

Property Name	Type	Description	Available Version
	DeliveryMethodAdjustment on page 2259>		
cartDeliveryMethodId	String	ID of the cart delivery method.	60.0
price	String	Price	60.0
shippingRateId	String	ID of the shipping rate.	62.0
totalAdjustmentBaseAmount	String	Total amount of the adjustment.	60.0
totalNetAmount	String	Final price of the cart item after all adjustments.	60.0

ConnectApi.PromotionCartDeliveryGroup

Cart delivery group for a promotion.

Property Name	Type	Description	Available Version
cartDeliveryGroupId	String	ID of the cart delivery group.	60.0
cartDeliveryGroupMethods	List<ConnectApi.PromotionCartDeliveryGroupMethod on page 1999>	List of methods for the delivery group.	60.0
id	String	ID of the cart.	60.0

ConnectApi.CartInventoryItemReservationOutputRepresentation (Pilot)

Inventory item reservation.

Note: This feature is not generally available and is being piloted with certain Customers subject to additional terms and conditions. It is not part of your purchased Services. This feature is subject to change, may be discontinued with no notice at any time in Salesforce's sole discretion, and Salesforce may never make this feature generally available. Make your purchase decisions only on the basis of generally available products and features. This feature is made available on an AS IS basis and use of this feature is at your sole risk.

Property Name	Type	Description	Available Version
errorCode	String	Error code for this reservation item.	58.0
errorMessage	String	Error message for this reservation item.	58.0
id	String	ID for this reservation item.	58.0
itemReservationSourceId	String	Item reservation source ID for this reservation item.	58.0

Property Name	Type	Description	Available Version
`productId`	String	Product ID for this reservation item.	58.0
`quantity`	Double	Quantity for this reservation item.	58.0
`reservedAtLocationId`	String	Reserved at location or group ID for this reservation item.	58.0

ConnectApi.CartInventoryReservationOutputRepresentation (Pilot)

Inventory Reservation

Note: This feature is not generally available and is being piloted with certain Customers subject to additional terms and conditions. It is not part of your purchased Services. This feature is subject to change, may be discontinued with no notice at any time in Salesforce's sole discretion, and Salesforce may never make this feature generally available. Make your purchase decisions only on the basis of generally available products and features. This feature is made available on an AS IS basis and use of this feature is at your sole risk.

Property Name	Type	Description	Available Version
`errorCode`	String	Error code for this reservation.	58.0
`errorMessage`	String	Error message for this reservation.	58.0
`inventoryItemReservations`	List<`ConnectApi.CartInventoryItemReservationOutputRepresentation`>	Collection of inventory item reservations.	58.0
`reservationIdentifier`	String	Reservation identifier.	58.0
`success`	Boolean	Indicates whether the transaction was successful.	58.0

ConnectApi.CartItem

An item in a cart.

Subclass of ConnectApi.AbstractCartItem.

Property Name	Type	Description	Available Version
`currencyIsoCode`	String	Currency ISO code of the cart.	57.0
`firstPymtTotalAmount`	String	For subscription items, the first payment amount after adjustments and taxes. For non-subscription products, the value is the same as `totalAmount`.	60.0
`firstPymtTotalListPrice`	String	For subscription items, the total list price of the first payment. For non-subscription items, the value is the same as `totalListPrice`.	63.0

Property Name	Type	Description	Available Version
`firstPymtTotalPrice`	String	For subscription items, the first term price, including adjustments but excluding taxes. For non-subscription items, the total price, including adjustments but excluding taxes.	60.0
`firstPymtTotalTax`	String	For subscription items, the tax amount on the first payment. For non-subscription products, the value is the same as `totalTax`.	60.0
`itemizedAdjustmentAmount`	String	Total itemized adjustment amount for the item, including promotions and excluding taxes.	52.0
`listPrice`	String	List price for the item.	49.0
`productClass`	`ConnectApi.ProductClass`	Class of product. Values are: • `Bundle` • `Set` • `Simple` • `Variation` • `VariationParent`	62.0
`salesPrice`	String	Sales price for the item.	49.0
`totalAdjustmentAmount`	String	Total adjustment amount for the item.	50.0
`totalAmount`	String	Total amount for the item.	49.0
`totalListPrice`	String	Total list price for the item.	49.0
`totalPrice`	String	Total price for the item including adjustments but excluding taxes.	49.0
`totalTax`	String	Total tax for the item.	49.0
`unitAdjustedPrice`	String	Unit price, including tier level discounts, for the item. This value is informational only and isn't used in pricing calculations.	50.0
`unitAdjustedPriceWithItemAdj`	String	Unit price, including both tier and item level discounts, for the item. This value is informational only and isn't used in pricing calculations.	61.0
`unitAdjustmentAmount`	String	Tier level adjustments made to the unit price for the item. This value is informational only and isn't used in pricing calculations.	50.0
`unitItemAdjustmentAmount`	String	Item level adjustments made to the unit price for the item. This value is informational only and isn't used in pricing calculations.	61.0

ConnectApi.CartItemBasic

Represents limited details about a cart item.

Property Name	Type	Description	Available Version
`cartItemId`	String	ID of the cart item.	60.0
`name`	String	Name of the cart item.	60.0
`productId`	String	ID of the product associated with the cart item.	60.0
`quantity`	String	Quantity of the cart item.	60.0
`type`	`ConnectApi.CartItemType`	Type of item in a cart. Value is Product.	60.0

ConnectApi.CartItemBasicResult

Represents the result of a cart request.

Property Name	Type	Description	Available Version
`cartItem`	ConnectApi.CartItemBasic on page 2003	Item in a cart.	60.0
`message`	String	Error message when the request is not successful.	60.0
`status`	String	Status of the request.	60.0

ConnectApi.CartItemCollection

A collection of items in a cart.

Property Name	Type	Description	Available Version
`cartCoupons`	`ConnectApi.CartCouponCollection`	Collection of coupons in the cart.	59.0
`cartItems`	List<`ConnectApi.CartItemResult`>	Collection of cart item results.	49.0
`cartPromotions`	`ConnectApi.CartPromotionCollection`	Collection of promotions in the cart.	59.0
`cartSummary`	`ConnectApi.CartSummary`	Summary of the cart.	49.0
`currentPage`	Integer	Current page of cart items. The value matches the requested page number, unless the requested page number exceeds the total number of pages. In this scenario, the current page is the highest available page number.	60.0

Property Name	Type	Description	Available Version
`currentPageToken`	String	Token identifying the current page.	49.0
`currentPageUrl`	String	Connect REST API URL identifying the current page.	49.0
`hasErrors`	Boolean	Specifies whether at least one of the results contains an error.	49.0
`nextPageToken`	String	Token identifying the next page, or `null` if there isn't a next page.	49.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	49.0
`previousPageToken`	String	Token identifying the previous page, or `null` if there isn't a previous page.	49.0
`previousPageUrl`	String	Connect REST API URL identifying the previous page, or `null` if there isn't a previous page.	49.0
`totalItemCount`	Integer	Total number of unique products in the cart.	60.0
`totalNumberOfPages`	Integer	Total number of pages for the given page size.	60.0

ConnectApi.PromotionCartItemKey

Cart item key for a bonus product in a promotion.

Property Name	Type	Description	Available Version
`cartItemId`	String	ID of the cart item.	58.0
`cartItem QuantityIdentifier`	String	Quantity ID of the cart item.	58.0

ConnectApi.CartItemProduct

Product summary for a cart item.

Property Name	Type	Description	Available Version
`fields`	Map<String, String>	Map of product fields and values.	49.0
`name`	String	Name of the product.	49.0
`productId`	String	ID of the product.	49.0
`productSubscriptionInformation`		Reserved for future use.	59.0
`purchaseQuantityRule`	`ConnectApi.PurchaseQuantityRule`	If one exists, purchase quantity rule for the product.	52.0

Property Name	Type	Description	Available Version
`sku`	String	SKU of the product.	49.0
`thumbnailImage`	`ConnectApi.ProductMedia`	Thumbnail image of the product.	49.0
`variationAttributes`	Map<String, `ConnectApi.CartProductAttribute`>	Variation attributes associated with the product.	50.0

SEE ALSO:

ConnectApi.AbstractCartItem

ConnectApi.WishlistItem

ConnectApi.CartItemPromotionCollectionOutputRepresentation

Promotions for the items in a cart.

Property Name	Type	Description	Available Version
`currencyIsoCode`	String	Currency code associated with the cart.	53.0
`items`	Map<String, `ConnectApi.CartPromotionListRepresentation`>	Collection of promotions.	52.0

ConnectApi.CartItemResult

Result after requesting a cart item.

Property Name	Type	Description	Available Version
`cartItem`	`ConnectApi.AbstractCartItem`	Cart item.	49.0
`message`	String	Message when the request isn't successful.	49.0
`status`	String	Status for the request.	49.0

SEE ALSO:

ConnectApi.CartItemCollection

ConnectApi.WishlistToCartResult

ConnectApi.CartItemWithoutPrice

An item without price information in a cart.

Subclass of ConnectApi.AbstractCartItem.

No additional properties.

ConnectApi.CartMessage

Cart message.

Property Name	Type	Description	Available Version
`message`	String	Cart message.	49.0
`messageId`	String	ID of the object supplying the message.	49.0
`relatedEntityId`	String	ID of the entity, for example, cart, cart item, or cart tax, associated with the message.	49.0
`severity`	`ConnectApi.CartMessageSeverity`	Severity of cart message. Values are: • `Error` • `Info` • `Warning`	49.0
`type`	String	Type of message. Standard values include `Inventory`, `Taxes`, `Pricing`, `Shipping`, `Entitlement`, `SystemError`, and `Other`.	49.0
`visible`	Boolean	Specifies whether the message is visible (`true`) or dismissed (`false`).	49.0

SEE ALSO:

ConnectApi.CartMessagesSummary

ConnectApi.CartMessagesSummary

Cart messages summary.

Property Name	Type	Description	Available Version
`errorCount`	Integer	In `ConnectApi.CartItemResult`, the count of messages with the `Error` severity level.	49.0
`hasErrors`	Boolean	Specifies whether there are messages related to the entity (`true`) or not (`false`).	49.0
`limitedMessages`	List<`ConnectApi.CartMessage`>	In `ConnectApi.CartItemResult`, a limited list of messages related to the cart item. In `ConnectApi.CartSummary`, each message	49.0

Property Name	Type	Description	Available Version
		can be related to the cart or to another cart-related entity.	
`relatedEntityId`	String	In `ConnectApi.CartItemResult`, the ID of the related cart item. In `ConnectApi.CartSummary`, each message can be related to the cart or to another cart-related entity.	49.0
`totalLineItemsWithErrors`	Integer	In `ConnectApi.CartItemResult`, either `null` if the cart item has no errors or `1` if the cart item has errors. In `ConnectApi.CartSummary`, the total number of product line items that contain errors.	50.0

SEE ALSO:

ConnectApi.AbstractCartItem

ConnectApi.CartMessagesVisibilityResult

Result of setting the visibility for cart messages.

Property Name	Type	Description	Available Version
`visibility`	Boolean	Specifies whether cart messages are set to visible (`true`) or not (`false`).	50.0

ConnectApi.CartProductAttribute

Product attribute for a cart item.

Property Name	Type	Description	Available Version
`label`	String	Label or display name of the attribute.	50.0
`sequence`	Integer	Sequence of the attribute within the attribute set.	50.0
`value`	String	Display value of the attribute.	50.0

SEE ALSO:

ConnectApi.CartItemProduct

ConnectApi.CartPromotionCollection

All the promotions associated with the cart.

Property Name	Type	Description	Available Version
`cartId`	String	ID of the cart.	53.0
`cartStatus`	`ConnectApi.CartStatus`	Status of the cart. Values are: • `Active`—Cart is created and available for modifications, like adding or removing products or promotions. • `Checkout`—Cart is in checkout. If the customer modifies the cart, the current checkout session is canceled. • `Closed`—Checkout is complete and an order was created. The cart cannot be modified. • `PendingClosed`—Cart is marked to be closed, but the request isn't completed yet. The cart can't be modified. This value is available in API version 57.0 and later. • `PendingDelete`—Cart is marked for delete, but the request isn't completed yet. The cart can't be modified. • `Processing`—Cart is processing. For example, taxes are being calculated. The cart can't be modified.	53.0
`currencyIsoCode`	String	Currency ISO code of the cart.	53.0
`promotions`	List<`ConnectApi.CartPromotionList`>	Collection of promotions.	53.0

ConnectApi.CartPromotionList

A list of promotions for a cart.

Property Name	Type	Description	Available Version
`promotions`	List<`ConnectApi.CartPromotionOutputRepresentation`>	Promotions associated with a cart.	54.0

ConnectApi.CartPromotionOutputRepresentation

A promotion associated with a cart.

Property Name	Type	Description	Available Version
`adjustmentAmount`	String	Adjustment amount out of the promotion.	53.0

Property Name	Type	Description	Available Version
couponCode	String	Coupon code for a promotion. A coupon code is available only for manual promotions, not for automatic promotions.	54.0
currencyIsoCode	String	Currency ISO code associated with the cart.	57.0
displayName	String	Localized display name of the promotion.	52.0
promotionId	String	ID of the promotion.	53.0
targetType	ConnectApi.CartPromotionType	Promotion target type. Values are: • Cart—Promotion targets a cart. • Item—Promotion targets an item in a cart.	53.0
termsAndConditions	String	Localized terms and conditions for the promotion.	53.0

ConnectApi.CartSummary

A cart summary.

Property Name	Type	Description	Available Version
accountId	String	ID of the account for the cart.	49.0
asyncOperationStatus	String	Asynchronous processing status of the cart, if asynchronous processing is enabled for the store. This property returns Completed in Apex, because Apex operations always run synchronously.	59.0
cartId	String	ID of the cart.	49.0
currencyIsoCode	String	Three-letter ISO 4217 currency code associated with the cart.	49.0
customFields	List<SObject>	Array of sObjects and viewable custom fields for the sObjects. Field-level security rules from the shopper profile are applied to the custom fields. The rules are applied for registered shoppers and for the guest shopper profile. If no custom fields were specified, returns an empty collection.	61.0
firstPymtGrandTotalAmount	String	First payment amount for subscription products, plus the total payment amount for non-subscription products. Includes taxes.	60.0
firstPymtTotalAmount	String	The total amount on the first payment of the cart.	63.0
firstPymtTotalListPrice	String	The total list price on the first payment of the cart.	63.0

Property Name	Type	Description	Available Version
`firstPymtTotalTaxAmount`	String	Tax amount on the first payment for any subscription products, plus the total tax amount on non-subscription products.	60.0
`grandTotalAmount`	String	Grand total amount including shipping and tax for items in the cart, in the currency of the cart.	49.0
`isSecondary`	Boolean	Specifies whether the cart is secondary (`true`) or not (`false`).	53.0
`name`	String	Name of the cart.	49.0
`ownerId`	String	ID of the owner of the cart.	49.0
`ownerOrderId`	String	ID of the owner of the order.	58.0
`purchaseOrderNumber`	String	Purchase order for the cart.	50.0
`status`	`ConnectApi.CartStatus`	Status of the cart. Values are: • `Active`—Cart is created and available for modifications, like adding or removing products or promotions. • `Checkout`—Cart is in checkout. If the customer modifies the cart, the current checkout session is canceled. • `Closed`—Checkout is complete and an order was created. The cart cannot be modified. • `PendingClosed`—Cart is marked to be closed, but the request isn't completed yet. The cart can't be modified. This value is available in API version 57.0 and later. • `PendingDelete`—Cart is marked for delete, but the request isn't completed yet. The cart can't be modified. • `Processing`—Cart is processing. For example, taxes are being calculated. The cart can't be modified.	49.0
`taxType`	String	Tax type of the cart. • `Automatic`—Automatic taxation policy. • `Gross`—Gross taxation policy. • `Net`—Net taxation policy.	55.0
`totalAmountWithItemAdjustment`	String	Total amount, including both tier and item level discounts but excluding cart level discounts, for all items in the cart.	61.0

Property Name	Type	Description	Available Version
`totalCartLevelAdjustmentAmount`	String	Total cart level discount amount for the cart.	61.0
`totalChargeAmount`	String	Total amount for shipping and other charges in the currency of the cart. Includes adjustments from shipping promotions.	49.0
`totalListPrice`	String	Total list price for all cart items plus shipping adjustments.	49.0
`totalProductAmount`	String	Total amount including discounts, but excluding shipping and tax, for product items in the cart.	49.0
`totalProductAmountAfterAdjustments`	String	Total product amount, including promotions.	52.0
`totalProductCount`	String	Total count of items in the cart. This field may not be accurate when faster add-to-cart is turned on and quantity rules are enabled for products in the cart.	49.0
`totalProductLineItemCount`	Integer	Total count of line items, of the type Product, in the cart.	60.0
`totalProductListAmount`	String	Total list price for all products in the cart without any adjustments.	59.0
`totalPromotionalAdjustmentAmount`	String	Total promotional adjustment amount for items in the cart.	52.0
`totalSubProductCount`	String	Total quantity of all cart items with the selling model type Evergreen or Term-Defined.	60.0
`totalTaxAmount`	String	Total tax amount for the cart, including tax on shipping, if applicable.	49.0
`type`	`ConnectApi.CartType`	Type of cart. Values are: • `Cart`—Cart created by a customer. • `PayNowReadOnly`—Clone of a Template cart that the customer can check out with using the Pay Now feature. • `Template`—Cart created by an internal user.	49.0
`uniqueProductCount`	Integer	Total count of unique items, or SKUs, in the cart. This field is supported when faster add-to-cart is turned off.	49.0
`webstoreId`	String	ID of the webstore of the cart.	49.0

SEE ALSO:

ConnectApi.CartItemCollection

ConnectApi.CartToWishlistResult

Result of copying products from a cart to a wishlist.

Property Name	Type	Description	Available Version
`productsAddedCount`	Integer	Number of products copied from the cart to the wishlist.	50.0
`wishlistId`	String	ID of the wishlist that cart products were copied to.	50.0

ConnectApi.CaseCommentCapability

If a feed element has this capability, it has a case comment on the case feed.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
`actorType`	`ConnectApi.CaseActorType`	Specifies the type of user who made the comment.	32.0
`createdBy`	`ConnectApi.Actor`	Information about the user who created the comment.	32.0
`createdDate`	Datetime	ISO 8601 date string, for example, 2011-02-25T18:24:31.000Z.	32.0
`eventType`	`ConnectApi.CaseCommentEventType`	Specifies an event type for a comment in the case feed.	32.0
`id`	String	18-character ID of case comment.	32.0
`published`	Boolean	Specifies whether the comment has been published.	32.0
`text`	String	Text of the case comment.	32.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.CdpActionResponse

Customer Data Platform action response.

This class is abstract.

Superclass of:

- ConnectApi.CdpCalculatedInsightStandardActionResponseRepresentation
- ConnectApi.CdpSegmentActionOutput in API version 59.0 and later.

Property Name	Type	Description	Available Version
errors	List<ConnectApi.CdpErrorResponse>	List of errors that resulted from the action.	57.0
success	Boolean	Indicates whether the call was successful (true) or not (false).	57.0

ConnectApi.CdpCalculatedInsightDataSource

Calculated insight data source.

Property Name	Type	Description	Available Version
sourceApiName	String	Data source API name.	57.0
type	String	Data source type.	57.0

SEE ALSO:

- ConnectApi.CdpCalculatedInsightDimension
- ConnectApi.CdpCalculatedInsightMeasure

ConnectApi.CdpCalculatedInsightDimension

Calculated insight dimension.

Property Name	Type	Description	Available Version
apiName	String	API name of the dimension.	57.0
creationType	String	Creation type of the dimension.	57.0
dataSource	ConnectApi.CdpCalculatedInsightDataSource	Data source of the dimension.	57.0
dataType	String	Data type of the dimension.	57.0
dateGranularity	String	Date granularity of the dimension.	57.0
displayName	String	Display name of the dimension.	57.0
fieldRole	String	Field role of the dimension.	57.0
formula	String	Formula of the dimension.	57.0

SEE ALSO:

- ConnectApi.CdpCalculatedInsightOutput

ConnectApi.CdpCalculatedInsightMeasure

Calculated insight measure.

Property Name	Type	Description	Available Version
`apiName`	String	API name of the measure.	57.0
`creationType`	String	Creation type of the measure.	57.0
`dataSource`	`ConnectApi.CdpCalculatedInsightDataSource`	Data source of the measure.	57.0
`dataType`	String	Data type of the measure.	57.0
`displayName`	String	Display name of the measure.	57.0
`fieldAggregationType`	String	Field aggregation type of the measure.	57.0
`fieldRole`	String	Field role of the measure.	57.0
`formula`	String	Formula of the measure.	57.0

SEE ALSO:

ConnectApi.CdpCalculatedInsightOutput

ConnectApi.CdpCalculatedInsightOutput

Calculated insight.

Property Name	Type	Description	Available Version
`apiName`	String	API name of the calculated insight.	57.0
`calculatedInsightStatus`	String	Status of the calculated insight.	57.0
`creationType`	String	Creation type of the calculated insight.	57.0
`dataSpace`	String	Data space of the calculated insight.	57.0
`definitionStatus`	String	Definition status of the calculated insight.	57.0
`definitionType`	String	Definition type of the calculated insight.	57.0
`description`	String	Description of the calculated insight.	57.0
`dimensions`	List<`ConnectApi.CdpCalculatedInsightDimension`>	Dimensions of the calculated insight.	57.0
`displayName`	String	Display name of the calculated insight.	57.0
`expression`	String	Expression of the calculated insight.	57.0

Property Name	Type	Description	Available Version
`isEnabled`	Boolean	Specifies whether the calculated insight is enabled (`true`) or not (`false`).	57.0
`lastCalcInsightStatusDateTime`	String	Last calculated insight status date and time.	57.0
`lastCalcInsightStatusErrorCode`	String	Last calculated insight status error code.	57.0
`lastRunDateTime`	String	Last run date and time of the calculated insight.	57.0
`lastRunStatus`	String	Last run status of the calculated insight.	57.0
`lastRunStatusDateTime`	String	Last run status date and time of the calculated insight.	57.0
`lastRunStatusErrorCode`	String	Last run status error code of the calculated insight.	57.0
`measures`	List<`ConnectApi.CdpCalculatedInsightMeasure`>	Measures of the calculated insight.	57.0

SEE ALSO:

ConnectApi.CdpCalculatedInsightPageData

ConnectApi.CdpCalculatedInsightPage

Collection of calculated insights.

Property Name	Type	Description	Available Version
`collection`	`ConnectApi.CdpCalculatedInsightPageData`	Collection of calculated insights.	57.0

ConnectApi.CdpCalculatedInsightPageData

Calculated insight collection data.

Property Name	Type	Description	Available Version
`count`	Integer	Number of results returned in the page.	57.0
`currentPageToken`	String	Token identifying the current page.	57.0
`currentPageUrl`	String	Connect REST API URL identifying the current page.	57.0

Property Name	Type	Description	Available Version
`items`	List<`ConnectApi.CdpCalculatedInsightOutput`>	List of calculated insights.	57.0
`nextPageToken`	String	Token identifying the next page, or `null` if there isn't a next page.	57.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	57.0
`previousPageToken`	String	Token identifying the previous page, or `null` if there isn't a previous page.	57.0
`previousPageUrl`	String	Connect REST API URL identifying the previous page, or `null` if there isn't a previous page.	57.0
`total`	Integer	Total row count of calculated insights.	57.0

SEE ALSO:

ConnectApi.CdpCalculatedInsightPage

ConnectApi.CdpCalculatedInsightStandardActionResponseRepresentation

Response of the calculated insight run action.

Subclass of ConnectApi.CdpActionResponse.

No additional properties.

ConnectApi.CdpErrorResponse

Error response.

Property Name	Type	Description	Available Version
`errorCode`	String	Error code.	57.0
`message`	String	Message stating the reason for the error, if any.	57.0

SEE ALSO:

ConnectApi.CdpActionResponse

ConnectApi.CdpIdentityResolutionMatchCriterionOutput

Identity resolution ruleset's match rule criterion.

Property Name	Type	Description	Available Version
caseSensitiveMatch	Boolean	Specifies whether the criterion match is case sensitive (true) or not (false). Available only when matching is based on the party identifier.	58.0
entityName	String	API name of the Data Model Object the match rule applies to.	57.0
fieldName	String	Name of the field the criterion applies to.	57.0
matchMethodType	ConnectApi.CdpIdentityResolutionMatchMethodType	Match method for a match rule criterion. Values are: • Exact—Exact match. • ExactNormalized—Exact normalized match. • Fuzzy—Fuzzy match with medium precision. • FuzzyHigh—Fuzzy match with high precision. • FuzzyLow—Fuzzy match with low precision.	57.0
partyIdentificationInfo	ConnectApi.CdpIdentityResolutionMatchCriterionPartyIdentificationInfoOutput	Party Identifier information.	57.0
shouldMatchOnBlank	Boolean	Specifies whether blank fields can be used for matching (true) or not (false).	57.0

SEE ALSO:

ConnectApi.CdpIdentityResolutionMatchRuleOutput

ConnectApi.CdpIdentityResolutionMatchCriterionPartyIdentificationInfoOutput

Information when party identification is used in an identity resolution ruleset's match rule criterion.

Property Name	Type	Description	Available Version
partyName	String	Party identification name.	57.0
partyType	String	Party identification type.	57.0

SEE ALSO:

ConnectApi.CdpIdentityResolutionMatchCriterionOutput

ConnectApi.CdpIdentityResolutionMatchRuleOutput

Identity resolution ruleset's match rule.

Property Name	Type	Description	Available Version
criteria	List<ConnectApi.CdpIdentityResolutionMatchCriterionOutput>	Object and field the match rule applies to and the match method applied.	57.0
label	String	User friendly name for the identity resolution match rule.	57.0

SEE ALSO:

ConnectApi.CdpIdentityResolutionOutput

ConnectApi.CdpIdentityResolutionOutput

Identity resolution ruleset.

Property Name	Type	Description	Available Version
anonymousUnifiedProfiles	Long	Count of anonymous unified profiles created by running the identity resolution ruleset.	57.0
configurationType	ConnectApi.CdpIdentityResolutionConfigurationType	Source object for an identity resolution ruleset. Values are: • Account • Individual	57.0
consolidationRate	Double	Consolidation rate resulting from the run of an identity resolution ruleset.	57.0
dataSpaceName	String	Data space used as source data for an identity resolution ruleset.	57.0
description	String	Description of the identity resolution ruleset.	57.0
doesRunAutomatically	Boolean	Specifies whether automatic job run scheduling is enabled for the ruleset (true) or not (false). If unspecified, defaults to false.	57.0
id	String	Identity resolution ruleset's ID. This is not the identity resolution's extended ruleset ID (rulesetId).	57.0
knownUnifiedProfiles	Long	Count of known unified profiles created by running the identity resolution ruleset.	57.0
label	String	User friendly name of the identity resolution ruleset.	57.0
lastJobCompleted	Datetime	Date and time the last job completed.	57.0
lastJobStatus	String	Last job's status. Possible values are: • SUCCESS • IN_PROGRESS	57.0

Property Name	Type	Description	Available Version
		• FAIL • SCHEDULED • SKIPPED	
matchRules	List<ConnectApi.CdpIdentityResolutionMatchRuleOutput>	List of match rules.	57.0
matchedSourceProfiles	Long	Count of matched source profiles identified by running the identity resolution ruleset.	57.0
objectApiName	String	Object name of the identity resolution ruleset.	57.0
reconciliationRules	List<ConnectApi.CdpIdentityResolutionReconciliationRuleOutput>	List of reconciliation rules.	57.0
rulesetId	String	Extension ID of a ruleset. The ruleset ID must be unique and no longer than 4 characters. This ID is not the identifying ID for the ruleset.	57.0
rulesetStatus	String	Status of a ruleset job. Possible values are: • NEW • PUBLISHING • PUBLISHED • ERROR • DELETING • DELETE_FAILED	57.0
sourceProfiles	Long	Count of source profiles that were processed by a ruleset job.	57.0
totalUnifiedProfiles	Long	Count of unified profiles created by running the identity resolution ruleset.	57.0

SEE ALSO:

ConnectApi.CdpIdentityResolutionsOutput

ConnectApi.CdpIdentityResolutionReconciliationFieldRuleOutput

Identity resolution ruleset's reconciliation rule for a field.

Property Name	Type	Description	Available Version
fieldName	String	The field that this reconciliation rule applies to.	57.0

Property Name	Type	Description	Available Version
ruleType	ConnectApi.CdpIdentityResolutionReconciliationRuleType	Default reconciliation rule applied to fields in the object the reconciliation rule applies to. Values are: • LastUpdated • MostFrequent • SourceSequence	57.0
shouldIgnoreEmptyValue	Boolean	Specifies whether to ignore an empty value (true) or not (false).	57.0
sources	List<ConnectApi.CdpIdentityResolutionReconciliationSourceOutput>	If ruleType is SourceSequence, a prioritized list of data sources.	57.0

SEE ALSO:

ConnectApi.CdpIdentityResolutionReconciliationRuleOutput

ConnectApi.CdpIdentityResolutionReconciliationRuleOutput

Identity resolution ruleset's reconciliation rule for an object.

Property Name	Type	Description	Available Version
entityName	String	API name of the Data Model Object the reconciliation rule applies to.	57.0
fields	List<ConnectApi.CdpIdentityResolutionReconciliationFieldRuleOutput>	Field-specific reconciliation rules that override this default rule for the specified field.	57.0
linkDmoName	String	API name of the unified link object created by the identity resolution process.	57.0
ruleType	ConnectApi.CdpIdentityResolutionReconciliationRuleType	Default reconciliation rule applied to fields in the object the reconciliation rule applies to. Values are: • LastUpdated • MostFrequent • SourceSequence	57.0
shouldIgnoreEmptyValue	Boolean	Specifies whether to ignore an empty value (true) or not (false).	57.0
sources	List<ConnectApi.CdpIdentityResolutionReconciliationSourceOutput>	If ruleType is SourceSequence, a list of data sources in priority order.	57.0

Property Name	Type	Description	Available Version
`unifiedDmoName`	String	API name of the unified data model object created by the identity resolution process.	57.0

SEE ALSO:

ConnectApi.CdpIdentityResolutionOutput

ConnectApi.CdpIdentityResolutionReconciliationSourceOutput

Source for an identity resolution default reconciliation rule or field-specific rule using the `SourceSequence` match method.

Property Name	Type	Description	Available Version
`name`	String	If the `ruleType` for a reconciliation rule is `SourceSequence`, API name of a source Data Lake Object.	57.0

SEE ALSO:

ConnectApi.CdpIdentityResolutionReconciliationRuleOutput

ConnectApi.CdpIdentityResolutionReconciliationFieldRuleOutput

ConnectApi.CdpIdentityResolutionRunNowOutput

Identity resolution ruleset run now output.

Property Name	Type	Description	Available Version
`resultCode`	`ConnectApi.CdpIdentityResolutionRunNowResultCode`	Result of an identity resolution ruleset job run. Values are: • `ExceededMaximumNumberOfSuccessfulRunsAllowedIn24Hours` • `IdentityResolutionJobIsAlreadyRunning` • `NoPendingChangesJobRunSkipped` • `SuccessfullySubmittedIdentityResolutionJobRunRequest`	57.0

ConnectApi.CdpIdentityResolutionsOutput

Identity resolution rulesets.

Property Name	Type	Description	Available Version
identityResolutions	List<ConnectApi.CdpIdentityResolutionOutput>	List of identity resolution rulesets.	57.0

ConnectApi.CdpQueryDataOutput

Query data output.

Property Name	Type	Description	Available Version
data	List<Object>	Result data set.	54.0

ConnectApi.CdpQueryMetadataItem

Metadata item.

Property Name	Type	Description	Available Version
placeInOrder	Integer	Attribute place order in the result.	55.0
type	String	Metadata type for column.	55.0
typeCode	Integer	Metadata type code.	55.0

SEE ALSO:

ConnectApi.CdpQueryOutputV2

ConnectApi.CdpQueryMetadataOutput

Query metadata result.

Property Name	Type	Description	Available Version
metadata	List<Object>	Metadata set.	52.0

ConnectApi.CdpQueryOutput

Query result.

Property Name	Type	Description	Available Version
data	List<Object>	Result data set.	52.0
done	Boolean	Specifies whether the query is done (true) or not (false).	52.0
endTime	String	Query end time.	52.0

Property Name	Type	Description	Available Version
`metadata`	Map<String, Object>	Result metadata set.	52.0
`queryId`	String	Query ID.	52.0
`rowCount`	Integer	Number of rows in the result data set.	52.0
`startTime`	String	Query start time.	52.0

ConnectApi.CdpQueryOutputV2

Query output for the V2 API.

Property Name	Type	Description	Available Version
`data`	List<`ConnectApi.CdpQueryV2Row`> (in version 55.0 and later) List<Object> (version 54.0 only)	Result data set.	54.0
`done`	Boolean	Specifies whether the query is done (`true`) or not (`false`).	54.0
`endTime`	String	Query end time.	54.0
`metadata`	Map<String, `ConnectApi.CdpQueryMetadataItem`> (version 55.0 and later) Map<String, Object> (version 54.0 only)	Result metadata set.	54.0
`nextBatchId`	String	Next batch ID. Use this property as the *`nextBatchId`* parameter in the `nextBatchAnsiSqlV2(nextBatchId)` method to get the next batch of data.	54.0
`queryId`	String	Query ID.	54.0
`rowCount`	Integer	Number of rows in the result data set.	54.0
`startTime`	String	Query start time.	54.0

ConnectApi.CdpQueryV2Row

Row in the query output for the V2 API.

Property Name	Type	Description	Available Version
`rowData`	List<Object>	Row values.	55.0

SEE ALSO:

ConnectApi.CdpQueryOutputV2

ConnectApi.CdpSegmentActionOutput

Segment action.

Subclass of ConnectApi.CdpActionResponse in API version 59.0 and later.

Property Name	Type	Description	Available Version
`errorCode`	String	Error code associated with the action, if any.	57.0
`errorMessage`	String	Error message associated with the action, if any.	57.0
`jobId`	String	Job ID for the publish job.	56.0
`partitionId`	String	ID of the partition.	56.0
`publishStatus`	String	Publish status of the segment.	57.0
`segmentApiName`	String	API name of the segment.	59.0
`segmentId`	String	ID of the segment.	56.0

ConnectApi.CdpSegmentContainerOutput

Segment container.

Property Name	Type	Description	Available Version
`batchSize`	Integer	Number of items returned.	56.0
`offset`	Integer	Number of rows skipped before returning results.	56.0
`orderByExpression`	String	Expression indicating how results are ordered.	56.0
`segments`	List<`ConnectApi.CdpSegmentOutput`>	List of segments.	55.0

ConnectApi.CdpSegmentDbtModel

Segment dbt model.

Property Name	Type	Description	Available Version
`name`	String	Dbt model name.	55.0

Property Name	Type	Description	Available Version
sql	String	Dbt model SQL.	55.0

SEE ALSO:

ConnectApi.CdpSegmentDbtPipeline

ConnectApi.CdpSegmentDbtPipeline

Segment dbt pipeline.

Property Name	Type	Description	Available Version
models	List<ConnectApi.CdpSegmentDbtModel>	Dbt models.	55.0

SEE ALSO:

ConnectApi.CdpSegmentOutput

ConnectApi.CdpSegmentMemberOutput

Data Cloud segment member output.

Property Name	Type	Description	Available Version
data	List<ConnectApi.CdpSegmentMemberRowOutput>	Result data set.	58.0
endTime	Datetime	Query end time.	58.0
filter	String	Filter information for the query.	58.0
limit	Integer	Batch size information.	58.0
nextPageUrl	String	URL for the next page.	58.0
offSet	Integer	Offset information.	58.0
orderBy	String	Order by information.	58.0
rowCount	Integer	Total row count.	58.0
startTime	Datetime	Query start time.	58.0
totalCount	Integer	Total count of records.	58.0

ConnectApi.CdpSegmentMemberRowOutput

Data Cloud segment member row output.

Property Name	Type	Description	Available Version
`deltaType`	String	Delta type, for example, `new`, `existing`, or `removed`.	58.0
`id`	String	Segment member ID.	58.0
`kqId`	String	Fully qualified key ID.	58.0
`snapshotType`	String	Type of snapshot, for example, `full` or `incremental`.	58.0
`timestamp`	String	Timestamp.	58.0
`versionStamp`	String	Version timestamp.	58.0

SEE ALSO:

ConnectApi.CdpSegmentMemberOutput

ConnectApi.CdpSegmentMembershipTableOutput

Data Cloud segment membership table.

Property Name	Type	Description	Available Version
`historyTable`	String	Segment membership history table.	58.0
`latestTable`	String	Segment membership latest table.	58.0
`profileTable`	String	Segment membership profile table.	58.0

SEE ALSO:

ConnectApi.CdpSegmentOutput

ConnectApi.CdpSegmentOutput

Segment.

Property Name	Type	Description	Available Version
`apiName`	String	API name of the segment.	56.0
`dataSpace`	String	Data space of the segment.	57.0
`description`	String	Segment description.	55.0
`developerName`	String	Segment developer name.	55.0

Property Name	Type	Description	Available Version
displayName	String	Segment display name.	57.0
excludeCriteria	String	Segment exclude criteria.	57.0
includeCriteria	String	Segment include criteria.	57.0
includeDbt	ConnectApi.CdpSegmentDbtPipeline	Segment dbt pipeline.	55.0
lookalikeCriteria		Reserved for future use.	56.0
marketSegmentDefinitionId	String	ID of the market segment definition.	55.0
marketSegmentId	String	ID of the market segment.	56.0
nextPublishDateTime	String	Date and time of the next segment publish.	57.0
publishInterval	String	Segment publish interval.	55.0
publishScheduleEndDate	String	Publish schedule end date.	55.0
publishScheduleStartDateTime	String	Publish schedule start date time.	55.0
publishStatus	String	Segment publish status.	55.0
segmentMembershipDmo	ConnectApi.CdpSegmentMembershipTableOutput	Segment membership tables.	58.0
segmentMembershipTable	String	Name of the segment membership table.	56.0
segmentOnApiName	String	API name of the SegmentOn entity.	56.0
segmentOnId	String	ID of the DMO segment.	55.0
segmentStatus	String	Segment status.	55.0
segmentType	String	Type of segment.	56.0

SEE ALSO:

ConnectApi.CdpSegmentContainerOutput

ConnectApi.ChangeItemOutputRepresentation

The financial changes resulting from a change to one or more OrderItemSummaries. Most of the values represent the deltas of the values on the associated OrderSummary. The sign of each value is the opposite of the corresponding value on a change order record. For example, a discount is a positive value here and a negative value on a change order record.

Property Name	Type	Description	Available Version
grandTotalAmount	Double	Change to the GrandTotalAmount field.	48.0
totalAdjDeliveryAmtWithTax	Double	Change to the TotalAdjDeliveryAmtWithTax field.	49.0
totalAdjDistAmountWithTax	Double	Change to the TotalAdjDistAmountWithTax field.	49.0
totalAdjProductAmtWithTax	Double	Change to the TotalAdjProductAmtWithTax field.	49.0
totalAdjustedDeliveryAmount	Double	Change to the TotalAdjustedDeliveryAmount field.	48.0
totalAdjustedDeliveryTaxAmount	Double	Change to the TotalAdjustedDeliveryTaxAmount field.	48.0
totalAdjustedProductAmount	Double	Change to the TotalAdjustedProductAmount field.	48.0
totalAdjustedProductTaxAmount	Double	Change to the TotalAdjustedProductTaxAmount field.	48.0
totalAdjustmentDistributedAmount	Double	Change to the TotalAdjustmentDistributedAmount field.	48.0
totalAdjustmentDistributedTaxAmount	Double	Change to the TotalAdjustmentDistributedTaxAmount field.	48.0
totalAmount	Double	Change to the TotalAmount field.	48.0
totalExcessFundsAmount	Double	Amount of excess funds available on the OrderPaymentSummaries related to the OrderSummary. It is equal to the captured amount that is owed as a refund but is not associated with an invoice or credit memo. Excess funds normally occur when order items are canceled before fulfillment but after payment has been captured. This situation is not common in the US, where funds are normally authorized but not captured until the fulfillment process begins. This value includes all current excess funds related to the OrderSummary, not only the funds related to the current change.	48.0
totalFeeAmount	Double	Total amount of the fees charged for the change.	57.0
totalFeeTaxAmount	Double	Total amount of tax on the fees charged for the change.	57.0
totalRefundableAmount	Double	Total amount available to be refunded. It is the sum of the excess funds and any outstanding change order grand total amounts that apply to post-fulfillment changes. This value includes all	48.0

Property Name	Type	Description	Available Version
		current refundable amounts related to the OrderSummary, not only the amount related to the current change.	
`totalRequiredFundsAmount`	Double	The required funds associated with added order items.	54.0
`totalTaxAmount`	Double	Change to the TotalTaxAmount field.	48.0

SEE ALSO:

- ConnectApi.PreviewCancelOutputRepresentation
- ConnectApi.PreviewReturnOutputRepresentation
- ConnectApi.SubmitCancelOutputRepresentation
- ConnectApi.SubmitReturnOutputRepresentation

ConnectApi.ChangeOrdersInvoiceOutputRepresentation

List of IDs of invoices created for change orders.

Subclass of ConnectApi.BaseInvoiceOutputRepresentation.

No additional properties.

SEE ALSO:

- createMultipleInvoices(invoicesInput)
- ConnectApi.CreateMultipleInvoicesFromChangeOrdersOutputRepresentation

ConnectApi.ChatterActivity

Chatter activity.

Name	Type	Description	Available Version
`commentCount`	Integer	Total number of comments in the org or site made by the user.	28.0
`commentReceivedCount`	Integer	Total number of comments in the org or site received by the user.	28.0
`likeReceivedCount`	Integer	Total number of likes and upvotes (in version 45.0 and later) on posts and comments in the org or site received by the user.	28.0
`postCount`	Integer	Total number of posts in the org or site made by the user.	28.0

SEE ALSO:

- ConnectApi.UserDetail

ConnectApi.ChatterActivitySummary

Summary of Chatter activity.

Subclass of ConnectApi.UserFeedEntityActivitySummary.

Property Name	Type	Description	Available Version
`commentCount`	Integer	Total number of comments in the org or site made by the user.	42.0
`commentReceivedCount`	Integer	Total number of comments in the org or site received by the user.	42.0
`likeReceivedCount`	Integer	Total number of likes and upvotes (in version 45.0 and later) on posts and comments in the org or site received by the user.	42.0
`postCount`	Integer	Total number of posts in the org or site made by the user.	42.0

ConnectApi.ChatterConversation

Chatter conversation.

Name	Type	Description	Available Version
`conversationId`	String	ID for the conversation.	29.0
`conversationUrl`	String	Connect REST API URL identifying the conversation.	29.0
`members`	`List<ConnectApi.UserSummary>`	List of users in the conversation.	29.0
`messages`	`ConnectApi.ChatterMessagePage`	Content of the conversation.	29.0
`read`	Boolean	Specifies if the conversation is read (`true`) or not read (`false`).	29.0

ConnectApi.ChatterConversationPage

Chatter conversation page.

Name	Type	Description	Available Version
`conversations`	`List<ConnectApi.ChatterConversationSummary>`	List of conversations on the page.	29.0
`currentPageToken`	String	Token identifying the current page.	29.0
`currentPageUrl`	String	Connect REST API URL identifying the current page.	29.0

Name	Type	Description	Available Version
`nextPageToken`	String	Token identifying the next page, or `null` if there isn't a next page.	29.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	29.0

ConnectApi.ChatterConversationSummary

Chatter conversation summary.

Name	Type	Description	Available Version
`id`	String	ID for the conversation summary.	29.0
`latestMessage`	`ConnectApi.ChatterMessage`	Contents of the latest message.	29.0
`members`	`List<ConnectApi.UserSummary>`	List of members in the conversation.	29.0
`read`	Boolean	Specifies if the conversation is read (`true`) or not read (`false`).	29.0
`url`	String	Connect REST API URL to the conversation summary.	29.0

SEE ALSO:

ConnectApi.ChatterConversationPage

ConnectApi.ChatterGroup

Chatter group.

This class is abstract.

Subclass of ConnectApi.ActorWithId.

Superclass of:

- ConnectApi.ChatterGroupDetail
- ConnectApi.ChatterGroupSummary

Name	Type	Description	Available Version
`additionalLabel`	String	An extra label for the group, for example, "Archived," "Private," or "Private With Customers." If there isn't an extra label, the value is `null`.	30.0
`announcement`	`ConnectApi.Announcement`	The current announcement for this group. An announcement displays in a designated location in the Salesforce UI until 11:59 p.m. on its expiration date, unless it's deleted or replaced by another announcement.	31.0

Name	Type	Description	Available Version
bannerPhoto	ConnectApi.BannerPhoto	The banner photo for the group.	36.0
canHaveChatterGuests	Boolean	true if this group allows Chatter guests.	28.0
community	ConnectApi.Reference	Information about the Experience Cloud site the group is in.	28.0
description	String	Group's description.	28.0
emailToChatterAddress	String	Group's email address for posting to this group by email. Returns null if Chatter emails and posting to Chatter by email aren't both enabled in your organization.	30.0
isArchived	Boolean	Specifies whether the group is archived (true) or not (false).	29.0
isAutoArchiveDisabled	Boolean	Specifies whether automatic archiving is disabled for the group (true) or not (false).	29.0
isBroadcast	Boolean	Specifies whether the group is a broadcast group (true) or not (false). In a broadcast group, only group owners and managers can post to the group.	36.0
lastFeedElementPostDate	Datetime	ISO 8601 date string, for example, 2011-02-25T18:24:31.000Z, of the most recent feed element posted to the group.	31.0
lastFeedItemPostDate	Datetime	ISO 8601 date string, for example, 2011-02-25T18:24:31.000Z, of the most recent feed item posted to the group. Use lastFeedElementPosted.	28.0–30.0
memberCount	Integer	Total number of group members.	28.0
myRole	ConnectApi.GroupMembershipType	Type of membership the user has with the group. • GroupOwner • GroupManager • NotAMember • NotAMemberPrivateRequested • StandardMember	28.0
mySubscription	ConnectApi.Reference	If the context user is a member of this group, contains information about that subscription; otherwise, returns null.	28.0
name	String	Name of the group.	28.0
owner	ConnectApi.UserSummary	Information about the owner of the group.	28.0
photo	ConnectApi.Photo	Information about the group photo.	28.0

Name	Type	Description	Available Version
`visibility`	`Connectapi.GroupVisibilityType`	Group visibility type. Valid values are: • `PrivateAccess`—Only members of the group can see posts to this group. • `PublicAccess`—All users within the Experience Cloud site can see posts to this group. • `Unlisted`—Reserved for future use.	28.0

ConnectApi.ChatterGroupDetail

Chatter group details.

Subclass of ConnectApi.ChatterGroup.

Name	Type	Description	Available Version
`fileCount`	Integer	The number of files posted to the group.	28.0
`information`	`ConnectApi.GroupInformation`	Describes the Information section of the group. If the group is private, this section is visible only to members. If the context user is not a member of the group or does not have Modify All Data or View All Data permission, this value is `null`.	28.0
`pendingRequests`	Integer	The number of requests to join a group that are in a pending state.	29.0

SEE ALSO:

ConnectApi.ChatterGroupPage

ConnectApi.UserGroupDetailPage

ConnectApi.ChatterGroupPage

Page of groups.

Name	Type	Description	Available Version
`currentPageUrl`	String	Connect REST API URL identifying the current page.	28.0
`groups`	`List<ConnectApi.ChatterGroupDetail>`	List of group details.	28.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	28.0

Name	Type	Description	Available Version
previousPageUrl	String	Connect REST API URL identifying the previous page, or null if there isn't a previous page.	28.0

ConnectApi.ChatterGroupSummary

Chatter group summary.

Subclass of ConnectApi.ChatterGroup.

Name	Type	Description	Available Version
fileCount	Integer	The number of files posted to the group.	28.0

SEE ALSO:

- ConnectApi.ChatterGroupSummaryPage
- ConnectApi.UserGroupPage

ConnectApi.ChatterGroupSummaryPage

Page of group summaries.

Name	Type	Description	Available Version
currentPageUrl	String	Connect REST API URL identifying the current page.	29.0
groups	List<ConnectApi.ChatterGroupSummary>	List of group summary objects.	29.0
nextPageUrl	String	Connect REST API URL identifying the next page, or null if there isn't a next page.	29.0
previousPageUrl	String	Connect REST API URL identifying the previous page, or null if there isn't a previous page.	29.0

ConnectApi.ChatterLike

Chatter like information.

Name	Type	Description	Available Version
id	String	Like's 18-character ID	28.0

Name	Type	Description	Available Version
`likedItem`	`ConnectApi.Reference`	A reference to the liked comment or feed element.	28.0
`url`	String	Like's Connect REST API URL	28.0
`user`	`ConnectApi.UserSummary`	Like's creator	28.0

SEE ALSO:

ConnectApi.ChatterLikePage

ConnectApi.ChatterLikePage

Page of Chatter likes.

Name	Type	Description	Available Version
`currentPageToken`	Integer	Token identifying the current page.	28.0
`currentPageUrl`	String	Connect REST API URL identifying the current page.	28.0
`items`	`List<ConnectApi.ChatterLike>`	List of likes.	32.0
`likes`	`List<ConnectApi.ChatterLike>`	List of likes. **Important:** As of API version 32.0, use the `items` property.	28.0–31.0
`nextPageToken`	Integer	Token identifying the next page, or `null` if there isn't a next page.	28.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	28.0
`previousPageToken`	Integer	Token identifying the previous page, or `null` if there isn't a previous page.	28.0
`previousPageUrl`	String	Connect REST API URL identifying the previous page, or `null` if there isn't a previous page.	28.0
`total`	Integer	Total number of likes across all pages.	28.0

SEE ALSO:

ConnectApi.ChatterLikesCapability

ConnectApi.Comment

ConnectApi.ChatterLikesCapability

If a feed element has this capability, the context user can like it. Exposes information about existing likes.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
`isLikedBy CurrentUser`	Boolean	Indicates whether the feed element is liked by the context user (`true`) or not (`false`).	32.0
`page`	`ConnectApi. ChatterLikePage`	Likes information for this feed element.	32.0
`likesMessage`	`ConnectApi. MessageBody`	A message body that describes who likes the feed element.	32.0
`myLike`	`ConnectApi. Reference`	If the context user has liked the feed element, this property is a reference to the specific like, `null` otherwise.	32.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.ChatterMessage

Chatter message.

Name	Type	Description	Available Version
`body`	`ConnectApi.MessageBody`	Contents of the message.	29.0
`conversationId`	String	ID for the conversation.	29.0
`conversationUrl`	String	Connect REST API URL identifying the conversation.	29.0
`id`	String	ID of the message.	29.0
`recipients`	`List<ConnectApi.UserSummary>`	List of the recipients of the message.	29.0
`sender`	`ConnectApi.UserSummary`	Sender of the message.	29.0
`sendingCommunity`	`ConnectApi.Reference`	Information about the Experience Cloud site from which the message was sent. Returns `null` for the default site or if digital experiences isn't enabled.	32.0
`sentDate`	Datetime	The date and time the message was sent.	29.0

Name	Type	Description	Available Version
`url`	String	Connect REST API URL identifying the current page of the conversation.	29.0

SEE ALSO:

ConnectApi.ChatterConversationSummary

ConnectApi.ChatterMessagePage

ConnectApi.ChatterMessagePage

Chatter message page.

Name	Type	Description	Available Version
`currentPageToken`	String	Token identifying the current page.	29.0
`currentPageUrl`	String	Connect REST API URL identifying the current page.	29.0
`messages`	`List<ConnectApi.ChatterMessage>`	Messages on the current page.	29.0
`nextPageToken`	String	Token identifying the next page, or `null` if there isn't a next page.	29.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	29.0

SEE ALSO:

ConnectApi.ChatterConversation

ConnectApi.ChatterStream

A Chatter feed stream.

Property Name	Type	Description	Available Version
`community`	ConnectApi.CommunitySummary	Experience Cloud site where the stream is.	41.0
`createdDate`	Datetime	Date the stream was created.	39.0
`description`	String	Description of the stream.	39.0
`id`	String	18-character ID of the stream.	39.0
`name`	String	Name of the stream.	39.0
`subscriptions`	List<ConnectApi.FeedEntity>	List of entities whose feeds are included in the stream.	39.0

Property Name	Type	Description	Available Version
`url`	String	URL to the stream.	39.0

SEE ALSO:

ConnectApi.ChatterStreamPage

ConnectApi.ChatterStreamPage

A collection of Chatter feed streams.

Property Name	Type	Description	Available Version
`currentPageUrl`	String	URL to the current page of streams.	39.0
`items`	List<`ConnectApi.ChatterStream`>	List of streams.	39.0
`nextPageUrl`	String	URL to the next page of streams. In version 39.0, all streams are included in `currentPageUrl` and `nextPageUrl` is `null`.	39.0
`total`	Integer	Total number of streams in the collection.	39.0

ConnectApi.ClientInfo

Client information.

Name	Type	Description	Available Version
`applicationName`	String	Name of the connected app used for authentication.	28.0
`applicationUrl`	String	Value from the `Info URL` field of the connected app used for authentication.	28.0

SEE ALSO:

ConnectApi.Comment

ConnectApi.FeedItem

ConnectApi.CloseCapability

If a feed element has this capability, users with permission can close it.

Users can't edit (specifically the feed item body or title), comment on, or delete a closed feed element. If the closed feed element is a poll, users can't vote on it. Users can't edit (specifically the comment body) or delete a comment on a closed feed element or select or remove it as best answer.

Admins and moderators can edit and delete closed feed elements and comments on closed feed elements. Admins and moderators can select or remove the best answer status on comments on closed feed elements.

Property Name	Type	Description	Available Version
`canContextUserUpdateIsClosed`	Boolean	Specifies whether the context user has permission to set the feed element to closed (`true`) or not (`false`).	43.0
`isClosed`	Boolean	Specifies whether the feed element is closed (`true`) or not (`false`).	43.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.Comment

A comment.

Name	Type	Description	Available Version
`attachment`	`ConnectApi.FeedItemAttachment`	If the comment contains an attachment, property value is `ContentAttachment`. If the comment does not contain an attachment, it is `null`. **Important:** As of version 32.0, use the `capabilities` property.	28.0–31.0
`body`	`ConnectApi.FeedBody`	Body of the comment.	28.0
`capabilities`	`ConnectApi.CommentCapabilities`	Capabilities associated with the comment, such as any file attachments.	32.0
`clientInfo`	`ConnectApi.ClientInfo`	Information about the connected app used to authenticate the connection.	28.0
`createdDate`	Datetime	ISO 8601 date string, for example, 2011-02-25T18:24:31.000Z.	28.0
`feedElement`	`ConnectApi.Reference`	Feed element on which the comment is posted.	
`feedItem`	`ConnectApi.Reference`	Feed item on which the comment is posted. **Important:** As of version 32.0, use the `feedElement` property.	28.0–31.0
`id`	String	Comment's 18–character ID.	28.0
`isDeleteRestricted`	Boolean	If this property is `true`, the context user can't delete the comment. If this property is `false`, the context user might be able to delete the comment.	28.0

Name	Type	Description	Available Version
`likes`	`ConnectApi.ChatterLikePage`	The first page of likes for the comment. This property has no information for comments on direct messages.	28.0
`likesMessage`	`ConnectApi.MessageBody`	A message body that describes who likes the comment. This property is `null` for comments on direct messages.	28.0
`moderationFlags`	`ConnectApi.ModerationFlags`	Information about the moderation flags on a comment. If `ConnectApi.Features.communityModeration` is `false`, this property is `null`.	29.0
`myLike`	`ConnectApi.Reference`	If the context user liked the comment, this property is a reference to the specific like, `null` otherwise. This property is `null` for comments on direct messages.	28.0
`parent`	`ConnectApi.Reference`	Information about the parent feed-item for this comment.	28.0
`relativeCreatedDate`	String	The created date formatted as a relative, localized string, for example, "17m ago" or "Yesterday."	28.0
`threadLevel`	Integer	Level of nesting for a comment. 0 indicates a standard comment with a parent post. 1 indicates a threaded comment with a parent comment and a parent post. 2 indicates a threaded comment with two parent comments and a parent post. The UI is limited to these three levels.	44.0
`threadParentId`	String	ID of the parent comment for a threaded comment.	44.0
`type`	`ConnectApi.CommentType`	Type of comment. • `ContentComment`—Comment holds a content capability. • `TextComment`—Comment contains only text.	28.0
`url`	String	Connect REST API URL to this comment.	28.0
`user`	`ConnectApi.UserSummary`	Information about the comment author.	28.0

SEE ALSO:

ConnectApi.CommentPage

ConnectApi.QuestionAndAnswersCapability

ConnectApi.CommentCapabilities

A set of capabilities on a comment.

Property Name	Type	Description	Available Version
comments	ConnectApi.CommentsCapability	If a comment has this capability, it has threaded comments.	44.0
content	ConnectApi.ContentCapability	If a comment has this capability, it has a file attachment. Most ConnectApi.ContentCapability properties are null if the content has been deleted from the feed element or if the access has changed to private.	32.0
edit	ConnectApi.EditCapability	If a comment has this capability, users who have permission can edit it.	34.0
feedEntityShare	ConnectApi.FeedEntityShareCapability	If a comment has this capability, a feed entity is shared with it.	42.0
record	ConnectApi.RecordCapability	If a comment has this capability, it has a record attachment.	42.0
status	ConnectApi.StatusCapability	If a comment has this capability, it has a status that determines its visibility.	38.0
upDownVote	ConnectApi.UpDownVoteCapability	If a comment has this capability, users can upvote or downvote it.	41.0
verified	ConnectApi.VerifiedCapability	If a comment has this capability, users with permission can mark it as verified or unverified.	41.0

SEE ALSO:

ConnectApi.Comment

ConnectApi.CommentPage

A page of comments.

Name	Type	Description	Available Version
comments	List<ConnectApi.Comment>	Collection of comments. **Important:** As of version 32.0, use the items property.	28.0–31.0
currentPageToken	String	Token identifying the current page.	28.0
currentPageUrl	String	Connect REST API URL identifying the current page.	28.0
items	List<ConnectApi.Comment>	Collection of comments for this feed element.	32.0

Name	Type	Description	Available Version
`nextPageToken`	String	Token identifying the next page, or `null` if there isn't a next page. If you want to read more of the comments in search results, all the comments in the thread are refreshed, not just the ones that match the search term. Avoid using `nextPageToken` until the comments are refreshed.	28.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page. If you want to read more of the comments in search results, all the comments in the thread are refreshed, not just the ones that match the search term. Avoid using `nextPageUrl` until the comments are refreshed.	28.0
`previousPageToken`	String	Token identifying the previous page, or `null` if there isn't a previous page.	44.0
`previousPageUrl`	String	Connect REST API URL identifying the previous page, or `null` if there isn't a previous page.	44.0
`total`	Integer	Total number of published comments for the parent feed element.	28.0

SEE ALSO:

ConnectApi.CommentsCapability

ConnectApi.CommentSummary

Summary of the comment.

Subclass of ConnectApi.UserActivitySummary.

Property Name	Type	Description	Available Version
`commentId`	String	ID of the comment.	42.0

ConnectApi.CommentsCapability

If a feed element or comment has this capability, the context user can add a comment to it.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
page	ConnectApi.CommentPage	The comments information for this feed element or comment. Threaded comments are supported in version 44.0 and later.	32.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.CommerceActionResult

Result of executing a commerce action.

Property Name	Type	Description	Available Version
isSuccess	Boolean	Specifies whether the action is a success (`true`) or not (`false`).	53.0
message	String	Action result message.	53.0

ConnectApi.CommerceAddressCollection

A collection of Commerce addresses.

Property Name	Type	Description	Available Version
count	Integer	Count of addresses.	54.0
currentPageToken	String	Token to the current page of addresses.	54.0
currentPageUrl	String	URL to the current page of addresses.	54.0
items	List<ConnectApi.CommerceAddressOutput>	Address Details	54.0
nextPageToken	String	Token to the next page of addresses.	54.0
nextPageUrl	String	URL to the next page of addresses.	54.0
pageSize	Integer	Page size for addresses.	54.0
previousPageToken	String	Token to previous page of addresses.	54.0
previousPageUrl	String	URL to the previous page of addresses.	54.0
sortOrder	ConnectApi.CommerceAddressSort	Sort order for Commerce addresses. • `CreatedDateAsc`—Sort in ascending order of created date. • `CreatedDateDesc`—Sort in descending order of created date. • `NameAsc`—Sort in ascending order of name.	54.0

Property Name	Type	Description	Available Version
		• `NameDesc`—Sort in descending order of name.	

ConnectApi.CommerceAddressOutput

Address for a Commerce account.

Property Name	Type	Description	Available Version
`addressId`	String	ID of the address.	54.0
`addressType`	String	Type of address (for example, "Shipping" or "Billing").	54.0
`city`	String	The address city.	54.0
`companyName`	String	The address company name.	57.0
`country`	String	The address country.	54.0
`countryCode`	String	Two-character country code.	54.0–58.0
`fields`	Map<String, Record Field>	A list of custom address fields, if any.	54.0
`firstName`	String	The address first name.	57.0
`isDefault`	Boolean	Indicates whether a contact's address is the preferred method of communication (`true`) or not (`false`). The default value is `false`.	54.0
`lastName`	String	The address last name.	57.0
`middleName`	String	The address middle name.	57.0
`name`	String	Name of the contact.	54.0
`phoneNumber`	String	The address phone number.	57.0
`postalCode`	String	Zip code or postal code for the address.	54.0
`region`	String	The address state.	54.0
`regionCode`	String	The address state code.	54.0–58.0
`street`	String	The address street.	54.0

ConnectApi.CommerceProductSearchResults

Product search results information.

Property Name	Type	Description	Available Version
`categories`	`ConnectApi.SearchCategory`	Categories from the search results.	52.0

Property Name	Type	Description	Available Version
correlationId	String	Reserved for future use.	55.0
facets	List<ConnectApi.SearchFacet>	Facets from the search results.	52.0
locale	String	Locale of the search results.	52.0
productsPage	ConnectApi.ProductSummaryPage	Page of products from the search results.	52.0

ConnectApi.CommerceProductSellingModel

Product selling model information.

Property Name	Type	Description	Available Version
isSubscriptionProduct	Boolean	Indicates whether the product selling model is a subscription product or not.	59.0

ConnectApi.CommerceProductSummary

Summary of a product in product search results.

Property Name	Type	Description	Available Version
defaultImage	ConnectApi.ProductMedia	Default image of the product.	55.0
fields	Map<String, ConnectApi.FieldValue>	Map of fields belonging to the product.	55.0
id	String	ID of the product.	55.0
name	String	Name of the product.	55.0
prices	ConnectApi.PricingResultLineItem	Prices of the product.	55.0
productClass	ConnectApi.ProductClass	Class of product. Values are: • Bundle • Set • Simple • Variation • VariationParent	55.0

Property Name	Type	Description	Available Version
`productSellingModelInformation`	`ConnectApi.CommerceProductSellingModel`	Product selling model information.	59.0
`productVariationInfo`	ConnectApiProductVariationInfo on page 2257	Product variation attributes, metadata, and mappings of attribute combinations to variation product IDs. This field isn't available in stores with displayable fields enabled.	63.0
`purchaseQuantityRule`	`ConnectApi.PurchaseQuantityRule`	If one exists, purchase quantity rule for the product.	58.0
`urlName`	String	SEO-friendly URL name for the product.	59.0
`variationAttributeSet`	`ConnectApi.ProductAttributeSetSummary`	Variation attribute set that's associated with the product.	55.0

ConnectApi.CommerceProductSummaryPage

Collection of product summary representations in product search results.

Property Name	Type	Description	Available Version
`pageSize`	Integer	Number of products per page in search results.	55.0
`products`	List<ConnectApi.CommerceProductSummary>	Collection of product summaries.	55.0
`total`	Long	Number of products in search results across all pages.	55.0

ConnectApi.CommerceResultRepresentationBase

Base cart calculate output class.

This class is abstract.

Superclass of:

- ConnectApi.CalculateCartResult

Property Name	Type	Description	Available Version
`message`	String	Message related to the request.	62.0
`status`	String	Asynchronous processing status of the cart, if asynchronous processing is enabled for the store. This property returns `Completed` in Apex, because Apex operations always run synchronously.	62.0

ConnectApi.CommerceSearchIndex

Index information.

Property Name	Type	Description	Available Version
`completionDate`	Datetime	Completion date and time of the index.	52.0
`createdDate`	Datetime	Creation date of the index.	52.0
`creationType`	`ConnectApi.CommerceSearchIndexCreationType`	Creation type of the index. Values are: • `Manual` • `Scheduled`	52.0
`id`	String	ID of the index.	52.0
`indexBuildType`	`ConnectApi.CommerceSearchIndexBuildType`	Build type of the index. Values are: • `Full` • `Incremental`	57.0
`indexStatus`	`ConnectApi.CommerceSearchIndexStatus`	Status of the index. Values are: • `Completed` • `Failed` • `InProgress`	52.0
`indexUsage`	`ConnectApi.CommerceSearchIndexUsage`	Usage of the index. Values are: • `Live` • `OutOfUse`	52.0
`isIncrementable`	Boolean	Specifies whether the index allows incremental indexing (`true`) or not (`false`).	57.0
`lastCatalogSnapshotTime`	Datetime	Catalog snapshot time of the index.	57.0
`message`	String	Detailed message for the index status.	52.0

SEE ALSO:

ConnectApi.CommerceSearchIndexCollection

ConnectApi.CommerceSearchIndexCollection

Collection of indexes.

Property Name	Type	Description	Available Version
`indexes`	List<`ConnectApi.CommerceSearchIndex`>	List of up to two indexes. Returns the completed, live index and either the in-progress, out-of-use index or the most-recently-failed, out-of-use index.	52.0

ConnectApi.CommerceSearchIndexLog

Search index log information.

Property Name	Type	Description	Available Version
`catalogSnapshotTime`	Datetime	Catalog snapshot time of the index build.	57.0
`completionDate`	Datetime	Completion date of the index build.	57.0
`createdById`	String	ID of the user who initiated the index build.	57.0
`indexBuildStatus`	`ConnectApi.CommerceSearchIndexStatus`	Status of the index. Values are: • `Completed` • `Failed` • `InProgress`	57.0
`indexBuildType`	`ConnectApi.CommerceSearchIndexBuildType`	Build type of the index. Values are: • `Full` • `Incremental`	57.0
`indexId`	String	ID of the index build.	57.0
`message`	String	Detailed message for the index build status.	57.0
`numberOfProducts`	Integer	Number of new or changed products in the index build.	57.0

ConnectApi.CommerceSearchIndexLogCollection

Collection of search index logs for a webstore.

Property Name	Type	Description	Available Version
`indexLogs`	List<`ConnectApi.CommerceSearchIndexLog`>	List of up to 100 index logs sorted by most recent catalog snapshot time of the index.	57.0

ConnectApi.Community

Experience Cloud site.

Name	Type	Description	Available Version
`allowChatterAccessWithoutLogin`	Boolean	Specifies if guest users can access public groups without logging in.	31.0

Name	Type	Description	Available Version
`allowMembersToFlag`	Boolean	Specifies if members can flag content.	30.0
`builderBasedSnaEnabled`	Boolean	Specifies whether the Service Not Available page is an auto-generated Experience Builder-based page (`true`) or a static resource page that's set in **Workspaces** > **Administration** > **Pages** (`false`).	52.0
`builderUrl`	String	Experience Builder URL for the site.	56.0
`contentSpaceId`	String	ID of the managed content space associated with the enhanced site.	62.0
`description`	String	Site description.	28.0
`guestMemberVisibilityEnabled`	Boolean	Specifies whether guest members can see other members (`true`) or not (`false`).	47.0
`id`	String	Site ID.	28.0
`imageOptimizationCDNEnabled`	Boolean	Specifies whether images are optimized for guest users on all devices for sites using Salesforce's CDN for Digital Experiences (`true`) or not (`false`).	56.0
`invitationsEnabled`	Boolean	Specifies whether users can invite other external users.	28.0
`knowledgeableEnabled`	Boolean	Specifies whether knowledgeable people and endorsements are available for topics (`true`), or not (`false`).	30.0
`loginUrl`	String	Login URL for the site.	36.0
`memberVisibilityEnabled`	Boolean	Specifies whether members can see other members (`true`) or not (`false`).	45.0
`name`	String	Site name.	28.0
`nicknameDisplayEnabled`	Boolean	Specifies whether nicknames are displayed.	32.0
`privateMessagesEnabled`	Boolean	Specifies whether members can send and receive private messages to and from other members (`true`) or not (`false`).	30.0
`reputationEnabled`	Boolean	Specifies whether reputation is calculated and displayed for members.	31.0
`sendWelcomeEmail`	Boolean	Specifies whether emails are sent to all new users when they join.	28.0
`siteAsContainerEnabled`	Boolean	Specifies whether the site is an Experience Builder site (`true`) or a Salesforce Tabs + Visualforce site (`false`).	41.0
`siteUrl`	String	URL for the site, which is your Experience Cloud sites domain plus a URL prefix. For example, *`MyDomainName`*`.my.site.com/customers`.	30.0
`status`	`ConnectApi.CommunityStatus` Enum	Status of the Experience Cloud site. • `Live` • `Inactive`	28.0

Name	Type	Description	Available Version
		• `UnderConstruction`	
`templateName`	String	Name of the Experience Builder template.	46.0
`url`	String	Connect REST API URL to the site.	28.0
`urlPathPrefix`	String	Site-specific URL prefix. For example, in the site URL *`MyDomainName`*`.my.site.com/customers`, `customers` is the `UrlPathPrefix`.	28.0

SEE ALSO:

ConnectApi.CommunityPage

ConnectApi.CommunityPage

Page of Experience Cloud sites.

Name	Type	Description	Available Version
`communities`	`List<ConnectApi.Community>`	List of Experience Cloud sites the context user has access to.	28.0
`total`	Integer	Total number of Experience Cloud sites.	28.0

ConnectApi.CommunitySummary

Summary of an Experience Cloud site.

Property Name	Type	Description	Available Version
`id`	String	18-character ID of the site.	41.0
`name`	String	Localized name of the site.	41.0

SEE ALSO:

ConnectApi.UserActivitySummary

ConnectApi.CompanyVerifySummary

Company verify summary.

Subclass of ConnectApi.UserFeedEntityActivitySummary.

No additional properties.

ConnectApi.ComplexSegment

Complex segments of field changes.

This class is abstract.

Subclass of ConnectApi.MessageSegment.

Superclass of ConnectApi.FieldChangeSegment.

Name	Type	Description	Available Version
`segments`	`List<ConnectApi.MessageSegment>`	List of message segments.	28.0

ConnectApi.CompositeCommerceProductOutputRepresentation

Details of a composite product.

Property Name	Type	Description	Available Version
`errors`	List<`ConnectApi.ErrorResponse`>	List of any errors that were returned, including the error code and error message.	61.0
`productId`	String	ID of the product record created.	61.0
`success`	Boolean	Indicates whether the product was successfully created (`true`) or not (`false`).	61.0

ConnectApi.CompositeCommerceVariationOutputRepresentation

Details of composite product variations.

Property Name	Type	Description	Available Version
`errors`	List<`ConnectApi.ErrorResponse`>	List of any errors that were returned, including the error code and error message.	62.0
`productIds`	List<String>	IDs of the created product variations.	62.0
`success`	Boolean	Indicates whether the product variations were successfully created (`true`) or not (`false`).	62.0

ConnectApi.CompoundRecordField

Record field that is a composite of subfields.

Subclass of ConnectApi.LabeledRecordField.

Name	Type	Description	Available Version
fields	List<ConnectApi.AbstractRecordField>	Collection of subfields that make up the compound field.	29.0

ConnectApi.ConfirmHeldFOCapacityOutputRepresentation

Response to a request to confirm held fulfillment order capacity at one or more locations. Can correspond to one action call.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
confirmHeldFOCapacityResponses	List<ConnectApi.ConfirmHeldFOCapacityResponseOutputRepresentation>	List of responses to the requests to confirm held fulfillment order capacity at one or more locations.	55.0

ConnectApi.ConfirmHeldFOCapacityResponseOutputRepresentation

Response to a request to confirm held fulfillment order capacity at one or more locations.

Property Name	Type	Description	Available Version
capacityResponses	List<ConnectApi.CapacityResponseOutputRepresentation>	List of responses to the requests to confirm held fulfillment order capacity at individual locations.	55.0

ConnectApi.Content

A file attached to a feed item.

Property Name	Type	Description	Available Version
checksum	String	MD5 checksum for the file.	36.0
contentUrl	String	URL of the content for links.	36.0
description	String	Description of the attachment.	36.0
downloadUrl	String	URL to the content.	36.0
fileExtension	String	Extension of the file.	36.0
fileSize	String	Size of the file in bytes. If size can't be determined, returns `unknown`.	36.0
fileType	String	Type of file, such as PDF.	36.0
hasPdfPreview	Boolean	`true` if the file has a PDF preview available; `false` otherwise.	36.0

Property Name	Type	Description	Available Version
`id`	String	18-character ID of the content.	36.0
`imageDetails`	ConnectApi.ContentImageFileDetails	Image details, or `null` if the file isn't an image.	40.0
`isInMyFileSync`	Boolean	`true` if the file is synced with Salesforce Files Sync. Note: Salesforce Files Sync was retired on May 25, 2018.	36.0
`mimeType`	String	MIME type of the file.	36.0
`renditionUrl`	String	URL to the rendition resource for the file. For shared files, renditions process asynchronously after upload. For private files, renditions process when the first file preview is requested, and aren't available immediately after the file is uploaded.	36.0
`renditionUrl 240By180`	String	URL to the 240 x 180 pixel rendition resource for the file. For shared files, renditions process asynchronously after upload. For private files, renditions process when the first file preview is requested, and aren't available immediately after the file is uploaded.	36.0
`renditionUrl 720By480`	String	URL to the 720 x 480 pixel rendition resource for the file. For shared files, renditions process asynchronously after upload. For private files, renditions process when the first file preview is requested, and aren't available immediately after the file is uploaded.	36.0
`sharingOption`	`ConnectApi. FileSharingOption`	Sharing option of the file. Values are: • `Allowed`—Resharing of the file is allowed. • `Restricted`—Resharing of the file is restricted.	36.0
`textPreview`	String	Text preview of the file if available; `null` otherwise.	36.0
`thumb120By90 RenditionStatus`	String	Specifies the rendering status of the 120 x 90 preview image of the file. One of these values: • `Processing`—Image is being rendered. • `Failed`—Rendering process failed. • `Success`—Rendering process was successful. • `Na`—Rendering is not available for this image.	36.0
`thumb240By180 RenditionStatus`	String	Specifies the rendering status of the 240 x 180 preview image of the file. One of these values: • `Processing`—Image is being rendered. • `Failed`—Rendering process failed.	36.0

Property Name	Type	Description	Available Version
		• `Success`—Rendering process was successful. • `Na`—Rendering is not available for this image.	
`thumb720By480RenditionStatus`	String	Specifies the rendering status of the 720 x 480 preview image of the file. One of these values: • `Processing`—Image is being rendered. • `Failed`—Rendering process failed. • `Success`—Rendering process was successful. • `Na`—Rendering is not available for this image.	36.0
`title`	String	Title of the file.	36.0
`versionId`	String	Version ID of the file.	36.0

SEE ALSO:

ConnectApi.FilesCapability

ConnectApi.ContentCapability

If a comment has this capability, it has a file attachment.

Subclass of ConnectApi.FeedElementCapability.

For files attached to a feed post (instead of a comment) in version 36.0 and later, use `ConnectApi.FilesCapability`.

If content is deleted from a feed element after it's posted or if the access to the content is changed to private, the `ConnectApi.ContentCapability` exists, however most of its properties are null.

Property Name	Type	Description	Available Version
`checksum`	String	MD5 checksum for the file.	32.0
`contentUrl`	String	URL of the content for links and Google docs.	32.0
`description`	String	Description of the attachment.	32.0
`downloadUrl`	String	URL to the content.	32.0
`fileExtension`	String	Extension of the file.	32.0
`fileSize`	String	Size of the file in bytes. If size cannot be determined, returns `Unknown`.	32.0
`fileType`	String	Type of file.	32.0
`hasPdfPreview`	Boolean	`true` if the file has a PDF preview available, `false` otherwise.	32.0
`id`	String	18-character ID of the content.	32.0

Property Name	Type	Description	Available Version
`isInMyFileSync`	Boolean	`true` if the file is synced with Salesforce Files Sync; `false` otherwise. Note: Salesforce Files Sync was retired on May 25, 2018.	32.0
`mimeType`	String	MIME type of the file.	32.0
`renditionUrl`	String	URL to the rendition resource for the file. Renditions are processed asynchronously and may not be available immediately after the file has been uploaded.	32.0
`renditionUrl240By180`	String	URL to the 240x180 size rendition resource for the file. Renditions are processed asynchronously and may not be available immediately after the file has been uploaded.	32.0
`renditionUrl720By480`	String	URL to the 720x480 size rendition resource for the file. Renditions are processed asynchronously and may not be available immediately after the file has been uploaded.	32.0
`sharingOption`	`ConnectApi.FileSharingOption`	Sharing option of the file. Values are: • `Allowed`—Resharing of the file is allowed. • `Restricted`—Resharing of the file is restricted.	35.0
`textPreview`	String	Text preview of the file if available, `null` otherwise. The maximum number of characters is 200.	32.0
`thumb120By90RenditionStatus`	String	The status of the rendering of the 120x90 pixel sized preview image of the file. Should be either Processing, Failed, Success, or Na if unavailable.	32.0
`thumb240By180RenditionStatus`	String	The status of the rendering of the 240x180 pixel sized preview image of the file. Should be either Processing, Failed, Success, or Na if unavailable.	32.0
`thumb720By480RenditionStatus`	String	The status of the rendering of the 720x480 pixel sized preview image of the file. Should be either Processing, Failed, Success, or Na if unavailable.	32.0
`title`	String	Title of the file.	32.0
`versionId`	String	Version ID of the file.	32.0

SEE ALSO:

ConnectApi.CommentCapabilities

ConnectApi.ContentHubAllowedItemTypeCollection

The item types that the context user is allowed to create in a repository folder.

Property Name	Type	Description	Available Version
`allowedItemTypes`	List<`ConnectApi.ContentHubItemTypeSummary`>	A collection of item types that the context user is allowed to create in a repository folder.	39.0

ConnectApi.ContentHubFieldDefinition

A field definition.

Property Name	Type	Description	Available Version
`displayName`	String	Label or caption of this field.	39.0
`isMandatory`	Boolean	Specifies whether this field is mandatory for the item type.	39.0
`maxLength`	Integer	Maximum length of the value of this field.	39.0
`name`	String	Name of the field.	39.0
`type`	`ConnectApi.ContentHubVariableType`	Data type of the value of the field. Values are: • `BooleanType` • `DateTimeType` • `DecimalType` • `HtmlType` • `IdType` • `IntegerType` • `StringType` • `UriType` • `XmlType`	39.0

SEE ALSO:

ConnectApi.ContentHubItemTypeDetail

ConnectApi.ContentHubItemTypeDetail

The details of an item type associated with a repository folder.

Subclass of ConnectApi.AbstractContentHubItemType.

Property Name	Type	Description	Available Version
`fields`	List<`ConnectApi.ContentHubFieldDefinition`>	A list of fields that the context user is allowed to set in the metadata of this item type.	39.0

ConnectApi.ContentHubItemTypeSummary

The summary of an item type associated with a repository folder.

Subclass of ConnectApi.AbstractContentHubItemType.

No additional properties.

SEE ALSO:

ConnectApi.ContentHubAllowedItemTypeCollection

ConnectApi.ContentHubPermissionType

A permission type.

Property Name	Type	Description	Available Version
`id`	String	Internal ID of the permission type in the repository.	39.0
`label`	String	Label as returned by the repository.	39.0

SEE ALSO:

ConnectApi.ExternalFilePermissionInformation

ConnectApi.ContentHubProviderType

The type of repository.

Property Name	Type	Description	Available Version
`label`	String	Localized label of the provider type.	39.0

Property Name	Type	Description	Available Version
`type`	String	Provider type. One of these values: • `ContentHubBox` • `ContentHubGDrive` • `ContentHubSharepoint` • `ContentHubSharepointOffice365` • `ContentHubSharepointOneDrive` • `SimpleUrl`	39.0

SEE ALSO:

ConnectApi.ContentHubRepository

ConnectApi.ContentHubRepository

A repository.

Subclass of ConnectApi.ActorWithId.

Property Name	Type	Description	Available Version
`authentication`	ConnectApi.ContentHubRepositoryAuthentication	Repository authentication information.	40.0
`features`	`ConnectApi.ContentHubRepositoryFeatures`	Repository features.	39.0
`label`	String	Repository label.	39.0
`name`	String	Repository name.	39.0
`providerType`	`ConnectApi.ContentHubProviderType`	Repository provider type.	39.0
`rootFolderItemsUrl`	String	URL to the list of items in the repository root folder.	39.0

SEE ALSO:

ConnectApi.ContentHubRepositoryCollection

ConnectApi.ContentHubRepositoryAuthentication

Authentication information for a repository.

Property Name	Type	Description	Available Version
`authFlowUrl`	String	Depends on the `authProtocol`. • `NoAuthentication`—`null`. • `Oauth`—URL to start the OAuth flow. • `Password`—URL to the authentication settings for external systems.	40.0
`authProtocol`	ConnectApi.ContentHubAuthenticationProtocol	Authentication protocol used for the repository. Values are: • `NoAuthentication`—Repository doesn't require authentication. • `Oauth`—Repository uses OAuth authentication protocol. • `Password`—Repository uses user name and password authentication protocol.	40.0
`userHasAuthSettings`	Boolean	Specifies whether the user has credentials or the administrator configured the external data source to use the same set of credentials for every user (`true`). Otherwise, `false`.	40.0

SEE ALSO:

ConnectApi.ContentHubRepository

ConnectApi.ContentHubRepositoryCollection

A collection of repositories.

Property Name	Type	Description	Available Version
`currentPageUrl`	String	URL to the current page of repositories.	39.0
`nextPageUrl`	String	URL to the next page of repositories, or `null` if there isn't a next page.	39.0
`previousPageUrl`	String	URL to the previous page of repositories, or `null` if there isn't a previous page.	39.0
`repositories`	List<`ConnectApi.ContentHubRepository`>	Collection of repositories.	39.0

ConnectApi.ContentHubRepositoryFeatures

The features of a repository.

Property Name	Type	Description	Available Version
`canBrowse`	Boolean	Specifies whether the repository's folder hierarchy can be browsed (`true`) or not (`false`).	39.0
`canSearch`	Boolean	Specifies whether the repository can be searched (`true`) or not (`false`).	39.0

SEE ALSO:

ConnectApi.ContentHubRepository

ConnectApi.ContentImageFileDetails

Image file details.

Property Name	Type	Description	Available Version
`height`	Integer	Image's height in pixels.	40.0
`imageFormat`	String	Image's format.	40.0
`orientation`	String	Image's EXIF orientation value, if present.	40.0
`width`	Integer	Image's width in pixels.	40.0

SEE ALSO:

ConnectApi.InlineImageSegment

ConnectApi.ContractOutputRepresentation

Contract list.

Property Name	Type	Description	Available Version
`data`	List<`String`>	Record IDs of the contacts.	56.0

ConnectApi.ConversationApplicationDefinitionDetailRespresentation

Information about the conversation application definition.

Property Name	Type	Description	Available Version
`botInfo`	`ConnectApi.BotInfoRepresentation`	Basic information of the bot associated with this conversation application.	54.0
`errorMessage`	String	Error message for the failed get operation.	54.0

Property Name	Type	Description	Available Version
integrationApplication	ConnectApi.ConversationApplicationIntegrationType	Conversation application integration types. Values are: • Api • Slack	54.0
integrationName	String	Name of the conversation application.	54.0
isSuccess	Boolean	Success indicator of the get operation.	54.0
runtimeUrl	String	Base URL of the bot runtime API.	54.0

ConnectApi.CouponCodeRedemptionCollection

Collection of coupon code redemption results.

Property Name	Type	Description	Available Version
couponCodeRedemptionResults	List<ConnectApi.CouponCodeRedemptionResult>	List of coupon code redemption results.	58.0

ConnectApi.CouponCodeRedemptionResult

Coupon code redemption result.

Property Name	Type	Description	Available Version
availableRedemptions	Integer	Number of coupon code redemptions available.	58.0
couponCode	String	Coupon code.	58.0
errorMsg	String	Error message when coupon code redemption isn't successful.	58.0
isSuccess	Boolean	Specifies whether increasing or decreasing the coupon code redemption is successful (true) or not (false).	58.0
redemptionLimit	Integer	Number of coupon code redemptions allowed.	58.0

ConnectApi.CreateCreditMemoOutputRepresentation

ID of a created Credit Memo.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
creditMemoId	String	ID of the created Credit Memo.	48.0

ConnectApi.CreateMultipleInvoicesFromChangeOrdersOutputRepresentation

List of lists of invoices created from change orders for fees.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
`invoices`	List<`ConnectApi.ChangeOrdersInvoiceOutputRepresentation`>	List of IDs of invoices created from change orders for fees. Include these invoice IDs when calling Ensure Refunds for the return that the fees applied to.	56.0

SEE ALSO:

createMultipleInvoices(invoicesInput)

ConnectApi.CreateOrderPaymentSummaryOutputRepresentation

ID of the created Order Payment Summary.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
`orderPaymentSummaryId`	String	ID of the Order Payment Summary.	48.0

ConnectApi.Credential

Credential.

Property Name	Type	Description	Available Version
`authenticationProtocol`	`ConnectApi.CredentialAuthenticationProtocol`	Authentication protocol of the external credential. Values are: • `AwsSv4` • `Basic` • `Custom` • `Jwt` • `OAuth`	56.0
`authenticationProtocolVariant`	`ConnectApi.CredentialAuthenticationProtocolVariant`	Authentication protocol variant of the external credential. Values are: • `AwsSv4_STS`—AWS Signature Version 4 with Security Token Service. • `ClientCredentialsClientSecret`—OAuth 2.0 Client Credentials client secret. Client secrets are sent in the callout's request body.	57.0

Property Name	Type	Description	Available Version
		• `ClientCredentialsClientSecretBasic`—OAuth 2.0 Client Credentials client secret. Client secrets are sent in the callout's authorization header, as with Basic authentication. • `ClientCredentialsJwtAssertion`—OAuth 2.0 Client Credentials JSON Web Token assertion. • `JwtBearer`—OAuth 2.0 JSON Web Token bearer flow. • `NoAuthentication`—No authentication. • `RolesAnywhere`—AWS Signature Version 4 with Identity and Access Management (IAM) Roles Anywhere.	
`authenticationStatus`	`ConnectApi.CredentialAuthenticationStatus`	Status of the credential authentication. Values are: • `Configured`—Credential has all required credentials for at least one principal. • `NotConfigured`—Credential isn't configured. • `Unknown`—Credential status can't be determined because the authentication protocol is custom.	56.0
`credentials`	Map<String, `ConnectApi.CredentialValue`>	Map of protocol-specific credentials.	56.0
`externalCredential`	String	Fully qualified developer name of the external credential.	56.0
`principalName`	String	Name of the external credential named principal.	56.0
`principalType`	`ConnectApi.CredentialPrincipalType`	Type of credential principal. Values are: • `AwsStsPrincipal` • `NamedPrincipal` • `PerUserPrincipal`	56.0

ConnectApi.CredentialCustomHeader

Credential custom header.

Property Name	Type	Description	Available Version
`headerName`	String	Header name.	57.0
`headerValue`	String	Header value that can contain formulas.	57.0

Property Name	Type	Description	Available Version
`id`	String	ID of the customer header parameter.	58.0
`sequenceNumber`	Integer	Sequence number of the header. The sequence number determines the order of the header.	57.0

SEE ALSO:

ConnectApi.ExternalCredential

ConnectApi.NamedCredential

ConnectApi.CredentialValue

Credential value.

Property Name	Type	Description	Available Version
`encrypted`	Boolean	Specifies whether the credential value is encrypted (`true`) or not (`false`).	56.0
`revision`	Integer	Revision number of a short-lived credential.	58.0
`value`	String	Value of the credential.	56.0

SEE ALSO:

ConnectApi.Credential

ConnectApi.CurrencyRecordField

Record field containing a currency value.

Subclass of ConnectApi.LabeledRecordField.

ConnectApi.CustomListAudienceCriteria

Criteria for the custom list type of custom recommendation audience.

Subclass of ConnectApi.AudienceCriteria.

Property Name	Type	Description	Available Version
`memberCount`	Integer	Total number of members in the custom recommendation audience.	36.0
`members`	ConnectApi.UserReferencePage	Members of the custom recommendation audience.	36.0

ConnectApi.DashboardComponentSnapshot

Represents both dashboard component snapshots and alerts you receive when a dashboard component value crosses a threshold.

Property Name	Type	Description	Available Version
componentId	String	18-character ID of the dashboard component.	32.0
componentName	String	The dashboard component name.	32.0
dashboardBodyText	String	Display this text next to the actor in the feed element.Use this text in place of the default body text.	32.0
dashboardId	String	18-character ID of the dashboard.	32.0
dashboardName	String	The name of the dashboard.	32.0
fullSizeImageUrl	String	The source URL to retrieve the full-size image of a snapshot. Access this URL with OAuth credentials.	32.0
lastRefreshDate	Datetime	ISO 8601 date specifying when this dashboard component was last refreshed.	32.0
lastRefreshDateDisplayText	String	Display text for the last refresh date, for example, "Last Refreshed on October 31, 2013."	32.0
runningUser	ConnectApi.UserSummary	The running user of the dashboard at the time the snapshot was posted. This value may be null. Each dashboard has a running user, whose security settings determine which data to display in a dashboard.	32.0
thumbnailUrl	String	The source URL to retrieve the thumbnail image of a snapshot. Access this URL with OAuth credentials.	32.0

SEE ALSO:

ConnectApi.DashboardComponentSnapshotCapability

ConnectApi.DatacloudCompanies

ConnectApi.DashboardComponentSnapshotCapability

If a feed element has this capability, it has a dashboard component snapshot. A snapshot is a static image of a dashboard component at a specific point in time.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
dashboardComponentSnapshot	ConnectApi.DashboardComponentSnapshot	The dashboard component snapshot.	32.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.DataCategoryMetadata

Data category metadata for the object.

Property Name	Type	Description	Available Version
`groupName`	String	Group name of the data category.	63.0
`label`	String	Label of the data category.	63.0
`values`	Map<String, `ConnectApi.DataCategoryValueMetadata`>	Map of values for the current data category.	63.0

SEE ALSO:

ConnectApi.ObjectMetadata

ConnectApi.DataCategoryValueMetadata

Data category values for the object's data category.

Property Name	Type	Description	Available Version
`label`	String	Label of the data category group.	63.0
`valueName`	String	Name of the data category group.	63.0

SEE ALSO:

ConnectApi.ObjectMetadata

ConnectApi.DateEstimationOutputRepresentation

Date estimation for product delivery.

Property Name	Type	Description	Available Version
`max`	String	Maximum estimated date for delivery.	63.0
`min`	String	Minimum estimated date for delivery.	63.0
`type`	String	Estimation type.	63.0

ConnectApi.DateRecordField

Record field containing a date.

Subclass of ConnectApi.LabeledRecordField.

Name	Type	Description	Available Version
`dateValue`	Datetime	Date that a machine can read. Ignore the trailing `00:00:00.000Z` characters.	29.0

ConnectApi.DeleteIntent

Delete intent for a social post.

Property Name	Type	Description	Available Version
`managedSocialAccount`	`ConnectApi.ManagedSocialAccount`	Managed social account that deletes the social post.	45.0

SEE ALSO:

ConnectApi.DeleteIntents

ConnectApi.DeleteIntents

List of delete intents for a social post.

Property Name	Type	Description	Available Version
`deletes`	List<`ConnectApi.DeleteIntent`>	List of delete intents for the social post.	45.0

SEE ALSO:

ConnectApi.SocialPostIntents

ConnectApi.DeleteSocialPostIntent

Delete intent for the social post.

Property Name	Type	Description	Available Version
`socialAccountId`	String	ID of the social account that deletes the social post.	46.0
`socialPostId`	String	ID of the social post to delete.	46.0

ConnectApi.DeliveryEstimateOutputRepresentation

Delivery estimation information for products.

Property Name	Type	Description	Available Version
error	ConnectApi.DeliveryEstimationErrorOutputRepresentation on page 2068	Any error that was returned, including the error code and error message.	63.0
location	String	Location external reference.	63.0
productDeliveryEstimations	List<ConnectApi.ProductDeliveryEstimationOutputRepresentation> on page 2244	List of product delivery estimations.	63.0
shippingCarrierMethodExternalReference	String	Shipping carrier method external reference.	63.0

ConnectApi.DeliveryEstimationErrorOutputRepresentation

Delivery estimation error.

Property Name	Type	Description	Available Version
code	String	Error code.	63.0
message	String	Error message, if any.	63.0

ConnectApi.DigestJob

Represents a successfully enqueued API digest job request.

Property Name	Type	Description	Available Version
period	ConnectApi.DigestPeriod	Time period that's included in a Chatter email digest. Values are: • DailyDigest—The email includes up to the 50 latest posts from the previous day. • WeeklyDigest—The email includes up to the 50 latest posts from the previous week.	37.0

ConnectApi.DirectMessageCapability

If a feed element has this capability, it's a direct message.

Property Name	Type	Description	Available Version
memberChanges	ConnectApi.DirectMessageMemberActivityPage	Member activities of the direct message, with the most recent activity first.	40.0
members	ConnectApi.DirectMessageMemberPage	Members included in the direct message.	39.0

Property Name	Type	Description	Available Version
originalMembers	ConnectApi.DirectMessageMemberPage	Original members of the direct message.	40.0
subject	String	Subject of the direct message.	39.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.DirectMessageMemberActivity

Direct message member activity.

Property Name	Type	Description	Available Version
activityDate	Datetime	Direct message member activity date.	40.0
actor	ConnectApi.UserSummary	User who changed the direct message membership.	40.0
membersAdded	ConnectApi.DirectMessageMemberPage	Members added to the direct message as part of the activity.	40.0
membersRemoved	ConnectApi.DirectMessageMemberPage	Members removed from the direct message as part of the activity.	40.0

SEE ALSO:

ConnectApi.DirectMessageMemberActivityPage

ConnectApi.DirectMessageMemberActivityPage

A page of direct message member activities.

Property Name	Type	Description	Available Version
activities	List<ConnectApi.DirectMessageMemberActivity>	Collection of direct message member activities.	40.0
currentPageToken	String	Token identifying the current page.	40.0
currentPageUrl	String	Connect REST API URL identifying the current page.	40.0
nextPageToken	String	Token identifying the next page, or `null` if there isn't a next page.	40.0

Property Name	Type	Description	Available Version
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	40.0

SEE ALSO:

ConnectApi.DirectMessageCapability

ConnectApi.DirectMessageMemberPage

A collection of direct message members.

Property Name	Type	Description	Available Version
`currentPageToken`	String	Page token to access the current page of direct message members.	39.0
`currentPageUrl`	String	URL to the current page of direct message members.	39.0
`nextPageToken`	String	Page token to access the next page of direct message members.	39.0
`nextPageUrl`	String	URL to the next page of direct message members.	39.0
`users`	List<`ConnectApi.UserSummary`>	Collection of direct message members.	39.0

SEE ALSO:

ConnectApi.DirectMessageCapability

ConnectApi.DirectMessageCapability

ConnectApi.DirectMessageMemberActivity

ConnectApi.DistanceCalculationOutputRepresentation

Shipping distance data for a set of inventory locations.

Property Name	Type	Description	Available Version
`averageDistance`	Double	The average distance from the locations to the order recipient.	51.0
`locations`	List<`ConnectApi.LocationOutputRepresentation`>	The list of locations and their distances to the order recipient.	51.0
`rank`	Integer	This result's rank among all results by average distance to the order recipient.	51.0

ConnectApi.DistinctFacetValue

Distinct facet value.

This class is a subclass of ConnectApi.FacetValue.

Property Name	Type	Description	Available Version
`displayName`	String	Display name of the facet value.	52.0
`nameOrId`	String	Developer name of the attribute.	52.0
`productCount`	Long	Number of products in the search result that match the facet value.	52.0

SEE ALSO:

ConnectApi.DistinctValueSearchFacet

ConnectApi.DistinctValueSearchFacet

Facet with distinct values in product search results.

This class is a subclass of ConnectApi.SearchFacet.

Property Name	Type	Description	Available Version
`values`	List<`ConnectApi.DistinctFacetValue`>	Values of the facet found in the search result. Sorted by display name in alphabetical order.	52.0

ConnectApi.DownVoteSummary

Summary of a downvote.

Subclass of ConnectApi.UserFeedEntityActivitySummary.

No additional properties.

ConnectApi.EditCapability

If a feed element or comment has this capability, users who have permission can edit it.

Property Name	Type	Description	Available Version
`isEditRestricted`	Boolean	Specifies whether editing this feed element or comment is restricted. If `true`, the context user can't edit this feed element or comment. If `false`, the context user may or may not have permission to edit this feed element or comment. To determine if the context user can edit a feed element or comment, use the `isFeedElementEditableByMe(communityId, feedElementId)` or	34.0

Property Name	Type	Description	Available Version
		`isCommentEditableByMe(communityId, commentId)` method.	
`isEditableByMeUrl`	String	The URL to check if the context user is able to edit this feed element or comment.	34.0
`lastEditedBy`	`ConnectApi.Actor`	Who last edited this feed element or comment.	34.0
`lastEditedDate`	Datetime	The most recent edit date of this feed element or comment.	34.0
`latestRevision`	Integer	The most recent revision of this feed element or comment.	34.0
`relativeLastEditedDate`	String	Relative last edited date, for example, "2h ago."	34.0

SEE ALSO:

- ConnectApi.CommentCapabilities
- ConnectApi.FeedElementCapabilities

ConnectApi.EinsteinLlmGenAiSourceReference

Source from a data provider.

Property Name	Type	Description	Available Version
`contents`	List<`ConnectApi.EinsteinLlmGenerationGenAiSourceContentInfo`>	List of values from the source that are used for grounding a generated response.	62.0
`metadata`	List<`ConnectApi.EinsteinLlmGenerationGenAiSourceReferenceInfo`>	List of metadata for the source, such as URLs or record IDs.	62.0

ConnectApi.EinsteinLlmGenerationCitationOutput

Source information associated with a generated response.

Property Name	Type	Description	Available Version
`citedReferences`	List<`ConnectApi.EinsteinLlmGenerationGenAiCitedReference`>	List of metadata for the sources that are cited in the generated response.	62.0

Property Name	Type	Description	Available Version
sourceReferences	List<ConnectApi.EinsteinLlmGenAiSourceReference>	List of sources from one or more data providers.	62.0

ConnectApi.EinsteinLlmGenerationContentQualityOutput

Quality information about the generated response.

Property Name	Type	Description	Available Version
isToxicityDetected	Boolean	Specifies whether the generated response contains toxic language (true) or not (false).	61.0

ConnectApi.EinsteinLlmGenerationGenAiCitedReference

Metadata from an input source that is cited in a generated response.

Property Name	Type	Description	Available Version
link	String	URL for the source that is cited in a generated response.	62.0
sourceObjectApiName	String	API name of the source record that is cited in a generated response.	62.0
sourceObjectRecordId	String	ID of the source record that is cited in a generated response.	62.0

ConnectApi.EinsteinLlmGenerationGenAiSourceContentInfo

Values from a source that is cited in a generated response.

Property Name	Type	Description	Available Version
content	String	Text from the source that is cited in the generated response.	62.0
fieldName	String	API name of the field that is cited in the generated response, such as Opportunity.Amount.	62.0
objectName	String	API name of the object that is cited in the generated response, such as Opportunity.	62.0

ConnectApi.EinsteinLlmGenerationGenAiSourceReferenceInfo

Metadata from a source from a data provider.

Property Name	Type	Description	Available Version
`link`	String	URL for the source from the data provider.	62.0
`sourceObjectApiName`	String	API name of the source record.	62.0
`sourceObjectRecordId`	String	ID of the source record.	62.0

ConnectApi.EinsteinLLMGenerationItemOutput

Generated response from the LLM provider.

Property Name	Type	Description	Available Version
`contentQualityRepresentation`	`ConnectApi.EinsteinLlmGenerationsContentQuality`	Specifies whether the generated response contains toxic language `(true)` or not `(false)`.	61.0
`parameters`	String	Parameter values for the LLM provider.	60.0
`responseId`	String	ID of the generated response.	60.0
`safetyScoreRepresentation`	`ConnectApi.EinsteinLlmGenerationSafetyScoreOutput`	Safety score information related to the generated response.	60.0
`text`	String	Text of generated response.	60.0

ConnectApi.EinsteinLlmGenerationSafetyScoreOutput

Safety score information related to the LLM response.

Property Name	Type	Description	Available Version
`hateScore`	Double	A higher value means the generated response is more likely to contain text that expresses, incites, or promotes hatred, violence, or severe harm towards the targeted group. Minimum value of `0.0`. Maximum value of `1.0`.	60.0
`physicalScore`	Double	A higher value means the generated response is more likely to contain text with unsafe advice that may harm the user or others physically, or text that promotes, encourages, or depicts acts of self-harm. Minimum value of `0.0`. Maximum value of `1.0`.	60.0

Property Name	Type	Description	Available Version
`profanityScore`	Double	A higher value means the generated response is more likely to contain swear words, curse words, or obscene or profane language. Minimum value of `0.0`. Maximum value of `1.0`.	60.0
`safetyScore`	Double	Overall safety score based on the `hateScore`, `physicalScore`, `profanityScore`, `sexualScore`, `toxicityScore`, and `violenceScore`. A higher value means the generated response is more likely to be safe. Minimum value of `0.0`. Maximum value of `1.0`.	60.0
`sexualScore`	Double	A higher value means the generated response is more likely to contain text meant to arouse sexual excitement or promote sexual services. Minimum value of `0.0`. Maximum value of `1.0`.	60.0
`toxicityScore`	Double	A higher value means the generated response is more likely to contain text that is rude, disrespectful, or unreasonable. Minimum value of `0.0`. Maximum value of `1.0`.	60.0
`violenceScore`	Double	A higher value means the generated response is more likely to contain text that promotes or glorifies violence or celebrates the suffering or humiliation of others. Minimum value of `0.0`. Maximum value of `1.0`.	60.0

ConnectApi.EinsteinPromptRecordCollectionOutputRepresentation

List of prompt template records.

Property Name	Type	Description	Available Version
`hasMoreRecords`	Boolean	Specifies whether the query returned more prompt template records `(true)` or not `(false)`.	62.0
`promptRecords`	List<`ConnectApi.EinsteinPromptRecordRepresentation`>	List of prompt template records returned.	62.0
`totalPromptRecords`	Integer	Number of prompt template records returned.	62.0

ConnectApi.EinsteinPromptRecordFieldRepresentation

Field values for a prompt template record field.

Property Name	Type	Description	Available Version
`displayValue`	String	Visible value of a prompt template record field.	62.0
`value`	Object	Raw data value of a prompt template record field.	62.0

ConnectApi.EinsteinPromptRecordRepresentation

Prompt template record.

Property Name	Type	Description	Available Version
`apiName`	String	API name of the prompt template record.	62.0
`childRelationships`	Map<String, `ConnectApi.EinsteinPromptRecordRepresentation`>	Map of prompt template name and prompt template records that are versions of the prompt template.	62.0
`fields`	Map<String, `ConnectApi.EinsteinPromptRecordFieldRepresentation`>	Map of field name and prompt template record fields.	62.0
`id`	String	ID of the prompt template record.	62.0
`isStandard`	Boolean	Specifies whether the prompt template record is a standard prompt template `(true)` or user-created prompt template `(false)`.	62.0

ConnectApi.EinsteinPromptTemplateAttachment

Data for file attachments to prompts.

Property Name	Type	Description	Available Version
`fileExtension`	String	Extension of the file attachment.	63.0
`id`	String	Content document ID of the file attachment.	63.0
`latestPublishedVersion`	String	Content version ID of the file attachment.	63.0
`title`	String	Title of the file attachment.	63.0

ConnectApi.EinsteinPromptTemplateGenerationsError

Error response to a prompt template generation request.

Property Name	Type	Description	Available Version
errorMessage	String	Message stating the reason for the error, if any.	60.0
httpErrorCode	String	HTTP status code, if any.	60.0
localized ErrorMessage	String	Translated error message, if available.	60.0
messageCode	String	Message code associated with the error message, if any.	60.0

ConnectApi.EinsteinPromptTemplateGenerationsRepresentation

Generated response from the LLM provider and resolved prompt template text.

Property Name	Type	Description	Available Version
citations	ConnectApi. EinsteinLlmGeneration CitationOutput	Source information associated with the generated responses.	62.0
fileData	List<ConnectApi. EinsteinPrompt TemplateAttachment>	File data of the input files used in generation.	63.0
generationErrors	List<ConnectApi. EinsteinPrompt Template GenerationsError>	List of errors associated with the generated responses, if any.	60.0
generations	List< ConnectApi. EinsteinLlm GenerationItem Output>	List of generated responses from the LLM provider.	60.0
isSummarized	Boolean	Specifies whether the generated response contains summarized text (true) or not (false).	61.0
parameters	ConnectApi. WrappedMapObject	Map of parameters and values for the LLM provider parameters.	60.0
prompt	String	Prompt template text with resolved inputs.	60.0
promptTemplate DevName	String	Developer name or ID of the prompt template record.	60.0
requestId	String	ID of the generation request sent to the LLM provider.	60.0
requestMessages	List<ConnectApi. EinsteinPrompt TemplateMask Content>	List of resolved prompt templates with masked data and masking information.	61.0

Property Name	Type	Description	Available Version
responseMessages	List<ConnectApi.EinsteinPromptTemplateMaskContent>	List of generated responses with masked data and masking information for the specified prompt template.	61.0
slotsMaskingInformation	List<ConnectApi.EinsteinPromptTemplateMaskData>	List of original and placeholder values of the masked data.	61.0

ConnectApi.EinsteinPromptTemplateMaskContentRepresentation

Generated response with masked data and masking information for a prompt template.

Property Name	Type	Description	Available Version
content	String	Text of generated response or resolved prompt template with masked data.	61.0
moderationSettings	ConnectApi.EinsteinPromptTemplateMaskSettingsRepresentation	Data masking settings for the specified prompt template.	61.0
role	String	Role in the Salesforce role hierarchy of the user executing the prompt template.	61.0

ConnectApi.EinsteinPromptTemplateMaskDataRepresentation

Information about masked data for a prompt template.

Property Name	Type	Description	Available Version
originalValue	String	Original value of the masked data.	61.0
placeHolder	String	Placeholder value of the masked data.	61.0
recognizers	List<String>	Reserved for internal use.	61.0

ConnectApi.EinsteinPromptTemplateMaskSettingsRepresentation

Data masking settings for a prompt template.

Property Name	Type	Description	Available Version
enableModeration	Boolean	Specifies whether data masking is enabled (true) or not (false).	61.0

ConnectApi.EmailAddress

Email address.

Name	Type	Description	Available Version
`displayName`	String	The display name for the email address.	29.0
`emailAddress`	String	The email address.	29.0
`relatedRecord`	ConnectApi.RecordSummary	The summary of a related record, for example, a contact or user summary.	36.0

SEE ALSO:

ConnectApi.EmailMessageCapability

ConnectApi.EmailAttachment

An email attachment in an email message.

Property Name	Type	Description	Available Version
`attachment`	`ConnectApi.RecordSummary`	Record summary of the attachment.	36.0
`contentType`	String	Type of attachment.	36.0
`fileName`	String	Name of the attachment.	36.0

SEE ALSO:

ConnectApi.EmailMessageCapability

ConnectApi.EmailMergeFieldCollectionInfo

The merge fields for an object.

Property Name	Type	Description	Available Version
`mergeFields`	List<String>	List of merge fields for a single object.	39.0

SEE ALSO:

ConnectApi.EmailMergeFieldInfo

ConnectApi.EmailMergeFieldInfo

The map for objects and their merge fields.

Property Name	Type	Description	Available Version
entityToMergeFieldsMap	Map<String, ConnectApi.EmailMergeFieldCollectionInfo>	Map for multiple objects and their merge field collections.	39.0

ConnectApi.EmailMessageCapability

If a feed element has this capability, it has an email message from a case.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
attachments	List<ConnectApi.EmailAttachment>	Attachments in the email message.	36.0
automationType	String	Automation type of the email message. • aiAssisted—The email message was created with the assistance of AI. • aiAutomated—The email message was created automatically by AI.	63.0
bccAddresses	List<ConnectApi.EmailAddress>	BCC addresses for the email message.	36.0
body	String	Body of the email message.	36.0
ccAddresses	List<ConnectApi.EmailAddress>	CC addresses for the email message.	36.0
direction	ConnectApi.EmailMessageDirection	Direction of the email message. Values are: • Inbound—An inbound message (sent by a customer). • Outbound—An outbound message (sent to a customer by a support agent).	32.0
emailMessageId	String	ID of the email message.	32.0
fromAddress	ConnectApi.EmailAddress	From address for the email message.	36.0
htmlExpandEmailThread	Integer	Start location of previous email thread.	47.0
isRichText	Boolean	Indicates whether the body of the email message is in rich text format.	36.0
status	ConnectApi.EmailMessageStatus	Status of an email message on a case. Values are: • DraftStatus • ForwardedStatus	47.0

Property Name	Type	Description	Available Version
		• `NewStatus` • `ReadStatus` • `RepliedStatus` • `SentStatus`	
`subject`	String	Subject of the email message.	32.0
`textBody`	String	Body of the email message. **Important:** In version 36.0 and later, use the `body` property.	32.0–35.0
`toAddresses`	List<`ConnectApi.EmailAddress`>	To addresses of the email message.	32.0
`totalAttachments`	Integer	Total number of attachments in the email message.	38.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.Emoji

An emoji.

Property Name	Type	Description	Available Version
`category`	String	Emoji category.	39.0
`shortcut`	String	Emoji shortcut.	39.0
`unicodeCharacter`	String	Emoji's unicode character.	39.0

SEE ALSO:

ConnectApi.EmojiCollection

ConnectApi.EmojiCollection

A collection of emoji.

Property Name	Type	Description	Available Version
`emojis`	List<`ConnectApi.Emoji`>	A collection of emoji.	39.0

SEE ALSO:

ConnectApi.SupportedEmojis

ConnectApi.EnhancedLinkCapability

If a feed element has this capability, it has a link that may contain supplemental information like an icon, a title, and a description.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
`description`	String	A description with a 500 character limit.	32.0
`icon`	`ConnectApi.Icon`	A icon.	32.0
`linkRecordId`	String	A ID associated with the link if the link URL refers to a Salesforce record.	32.0
`linkUrl`	String	A link URL to a detail page if available content can't display inline.	32.0
`title`	String	A title to a detail page.	32.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.EnsureFundsAsyncOutputRepresentation

ID of the asynchronous background operation. This output only includes the operation ID, regardless of whether a call is made to an external payment gateway. It doesn't include any errors from the operation.

Subclass of ConnectApi.BaseAsyncOutputRepresentation.

No additional properties.

SEE ALSO:

ensureFundsAsync(orderSummaryId, ensureFundsInput)

ConnectApi.EnsureRefundsAsyncOutputRepresentation

ID of the asynchronous background operation. This output only includes the operation ID, regardless of whether a call is made to an external payment gateway. It doesn't include any errors from the operation.

Subclass of ConnectApi.BaseAsyncOutputRepresentation.

No additional properties.

SEE ALSO:

ensureRefundsAsync(orderSummaryId, ensureRefundsInput)

ConnectApi.EntityLabel

An entity's label.

Property Name	Type	Description	Available Version
`label`	String	Localized singular label of the entity.	40.0
`labelPlural`	String	Localized plural label of the entity.	40.0

SEE ALSO:

ConnectApi.RecordSummary

ConnectApi.EntityLinkSegment

Entity link segment.

Subclass of ConnectApi.MessageSegment.

Name	Type	Description	Available Version
`motif`	`ConnectApi.MotifClass`	A set of small, medium, and large icons that indicate whether the entity is a file, group, record, or user. The motif can also contain the object's base color.	28.0
`reference`	`ConnectApi.Reference`	A reference to the link object if applicable, otherwise, `null`.	28.0

ConnectApi.EntityRecommendation

A Chatter, custom, or static recommendation.

Subclass of ConnectApi.AbstractRecommendation.

Property Name	Type	Description	Available Version
`actOnUrl`	String	For user, file, group, topic, and record `entity` types, use this Connect REST URL with a POST request to take action on the recommendation. For `ConnectApi.RecommendedObject` `entity` types, such as custom recommendations, use the `actionUrl` property of the ConnectApi.PlatformAction to take action on the recommendation.	32.0
`action`	`ConnectApi.RecommendationActionType`	Specifies the action to take on a recommendation. • `follow`—Follow a file, record, topic, or user. • `join`—Join a group. • `view`—View a file, group, article, record, user, custom, or static recommendation.	32.0

Property Name	Type	Description	Available Version
entity	ConnectApi.Actor	The entity with which the receiver is recommended to take action.	32.0

ConnectApi.ErrorResponse

Base error response.

Property Name	Type	Description	Available Version
errorCode	String	Error code.	48.0
message	String	More error detail, if available.	48.0

SEE ALSO:

ConnectApi.BaseOutputRepresentation

ConnectApi.EstimateDeliveryDateOutputRepresentation

Estimated delivery dates.

Property Name	Type	Description	Available Version
deliveryEstimates	List<ConnectApi.DeliveryEstimateOutputRepresentation> on page 2067	List of delivery estimations.	63.0
estimatedDeliveryReference	String	Unique code, reference, or identifier for the estimated delivery used by external systems.	63.0

ConnectApi.Extension

An extension.

Property Name	Type	Description	Available Version
alternativeRepresentation	ConnectApi.Alternative	Alternative representation of the extension.	40.0
attachmentId	String	Attachment ID of the extension.	41.0
extensionId	String	ID of the extension.	40.0
payload	String	Payload associated with the extension.	40.0

Property Name	Type	Description	Available Version
payloadVersion	String	Payload version that identifies the structure of the payload associated with the extension.	40.0

SEE ALSO:

ConnectApi.ExtensionsCapability

ConnectApi.ExtensionDefinition

An extension's definition.

Property Name	Type	Description	Available Version
canAccess	Boolean	Indicates whether users can access the extension when it's associated with a feed element.	40.0
canCreate	Boolean	Indicates whether users can create a feed element with the extension in the org.	40.0
createdDate	Datetime	Date when the extension was created.	40.0
description	String	Description of the extension.	40.0
iconUrl	String	URL to the icon for the extension.	40.0
id	String	ID of the extension.	40.0
informationCollection	List<ConnectApi.AbstractExtensionInformation>	Collection of extension information.	40.0
isEnabledInCommunity	Boolean	Indicates whether the extension is enabled in the site.	40.0 only
isEnabledInLightningPublisher	Boolean	Indicates whether the extension is enabled in the Lightning publisher.	40.0 only
name	String	Name of the extension.	40.0
position	Integer	Position in which the extension is displayed in the publisher.	41.0

SEE ALSO:

ConnectApi.ExtensionDefinitions

ConnectApi.ExtensionDefinitions

A collection of extension definitions.

Property Name	Type	Description	Available Version
currentPageToken	String	Token identifying the current page.	40.0
currentPageUrl	String	Connect REST API URL identifying the current page.	40.0
extensionDefinitions	List<ConnectApi.ExtensionDefinition>	Collection of extension definitions.	40.0
nextPageToken	String	Token identifying the next page, or null if there isn't a next page.	40.0
nextPageUrl	String	Connect REST API URL identifying the next page, or null if there isn't a next page.	40.0
total	Integer	Total number of extensions returned.	40.0

ConnectApi.ExtensionsCapability

If a feed element has this capability, it has one or more extension attachments.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
items	List<ConnectApi.Extension>	List of extensions associated with the feed element.	40.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.ExternalAuthIdentityProvider

External auth identity provider.

Property Name	Type	Description	Available Version
authenticationFlow	ConnectApi.IdentityProviderAuthFlow	Authentication flow to get tokens to call protected APIs. Values are: • AuthorizationCode	62.0
authenticationProtocol	ConnectApi.IdentityProviderAuthProtocol	Authentication protocol required to access the external system. Values are: • OAuth	62.0
authorizeUrl	String	Authorization endpoint URL for the external system.	62.0
callbackUrl	String	For the Authorization Code authentication flow, the callback URL that's used by the external system after authorization.	62.0

Property Name	Type	Description	Available Version
clientAuthentication	ConnectApi.IdentityProviderClientAuth	Client authentication method that describes how credentials are sent to the authorization server. Values are: • ClientSecretBasic • ClientSecretPost	63.0
createdByNamespace	String	Namespace of the package that created the external auth identity provider.	62.0
credentials	List<ConnectApi.ExternalAuthIdentityProviderCredential>	List of the external auth identity provider credentials.	62.0
description	String	Description of the external auth identity provider.	62.0
fullName	String	Full name of the external auth identity provider. The full name can include a namespace prefix.	62.0
id	String	External auth identity provider ID.	62.0
label	String	External auth identity provider label.	62.0
parameters	List<ConnectApi.ExternalAuthIdentityProviderParameter>	List of custom request parameters.	63.0
standardExternalIdentityProvider	String	Reference to a standard external auth identity provider.	63.0
tokenUrl	String	Token endpoint URL to retrieve tokens from the external system. Required for all OAuth 2.0 authentication flows.	62.0
url	String	Connect REST API URL for the external auth identity provider.	62.0
userInfoUrl	String	User info URL to retrieve user profile information from the external system.	62.0

SEE ALSO:

ConnectApi.ExternalAuthIdentityProviderList

createExternalAuthIdentityProvider(requestBody)

getExternalAuthIdentityProvider(fullName)

updateExternalAuthIdentityProvider(fullName, requestBody)

ConnectApi.ExternalAuthIdentityProviderCredential

External auth identity provider credential.

Property Name	Type	Description	Available Version
`credentialName`	String	Name of the external auth identity provider credential.	62.0
`credentialValue`	String	Value of the external auth identity provider credential.	62.0
`encrypted`	Boolean	Indicates whether the external auth identity provider credential is encrypted (`true`) or not (`false`).	62.0

SEE ALSO:

ConnectApi.ExternalAuthIdentityProviderCredentials

ConnectApi.ExternalAuthIdentityProviderCredentials

List of an external auth identity provider's credentials.

Property Name	Type	Description	Available Version
`credentials`	[ConnectApi.ExternalAuthIdentityProviderCredential on page 2088]>	List of external auth identity provider credentials.	62.0

SEE ALSO:

getExternalAuthIdentityProviderCredentials(fullName)

createExternalAuthIdentityProviderCredentials(fullName, requestBody)

updateExternalAuthIdentityProviderCredentials(fullName, requestBody)

ConnectApi.ExternalAuthIdentityProviderList

List of external auth identity providers in the org.

Property Name	Type	Description	Available Version
`externalAuthIdentityProviders`	[ConnectApi.ExternalAuthIdentityProvider on page 2086]>	List of external auth identity providers.	62.0

SEE ALSO:

getExternalAuthIdentityProviders()

ConnectApi.ExternalAuthIdentityProviderParameter

External auth identity provider parameter.

Property Name	Type	Description	Available Version
parameterName	String	The name of the external auth identity provider parameter.	63.0
parameterType	ConnectApi.ExternalAuthIdentityProviderParameterType	Parameter type for an external auth identity provider. Values are: • AuthorizeRequestQueryParameter • IdentityProviderOptions • RefreshRequestBodyParameter • RefreshRequestHttpHeader • RefreshRequestQueryParameter • TokenRequestBodyParameter • TokenRequestHttpHeader • TokenRequestQueryParameter	63.0
parameterValue	String	If parameterType describes a literal value then the literal value is stored in this property.	63.0
sequenceNumber	Integer	Specifies the order of parameters to apply when an external auth identity provider has more than one parameter. Priority is from lower to higher numbers, for example, 1 is the highest priority.	63.0

SEE ALSO:

ConnectApi.ExternalAuthIdentityProvider

ConnectApi.ExternalCredential

External credential, including the named credentials and principals associated with it and the type and status of each principal.

If you don't have the View Setup and Configuration permission, some properties are empty or show limited information.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Property Name	Type	Description	Available Version
authenticationProtocol	ConnectApi.CredentialAuthenticationProtocol	Authentication protocol of the external credential. Values are: • AwsSv4 • Basic • Custom • Jwt • OAuth	56.0

Property Name	Type	Description	Available Version
authenticationProtocolVariant	ConnectApi.CredentialAuthenticationProtocolVariant	Authentication protocol variant of the external credential. Values are: • AwsSv4_STS—AWS Signature Version 4 with Security Token Service. • ClientCredentialsClientSecret—OAuth 2.0 Client Credentials client secret. Client secrets are sent in the callout's request body. • ClientCredentialsClientSecretBasic—OAuth 2.0 Client Credentials client secret. Client secrets are sent in the callout's authorization header, as with Basic authentication. • ClientCredentialsJwtAssertion—OAuth 2.0 Client Credentials JSON Web Token assertion. • JwtBearer—OAuth 2.0 JSON Web Token bearer flow. • NoAuthentication—No authentication. • RolesAnywhere—AWS Signature Version 4 with Identity and Access Management (IAM) Roles Anywhere.	57.0
authenticationStatus	ConnectApi.CredentialAuthenticationStatus	Status of the credential authentication. Values are: • Configured—Credential has all required credentials for at least one principal. • NotConfigured—Credential isn't configured. • Unknown—Credential status can't be determined because the authentication protocol is custom.	56.0
createdByNamespace	String	Namespace of the package that created the external credential.	59.0
customHeaders	List<ConnectApi.CredentialCustomHeader>	List of custom headers.	57.0
developerName	String	Fully qualified developer name of the external credential.	56.0
id	String	External credential ID.	58.0
masterLabel	String	External credential label.	56.0
parameters	List<ConnectApi.ExternalCredentialParameter>	List of parameters of the external credential.	57.0

Property Name	Type	Description	Available Version
`principals`	List<`ConnectApi.ExternalCredentialPrincipal`>	List of principals the credential has.	56.0
`relatedNamedCredentials`	List<`ConnectApi.NamedCredential`>	List of named credentials associated to the external credential.	56.0
`url`	String	Connect REST API URL for the external credential.	58.0

SEE ALSO:

ConnectApi.ExternalCredentialList

ConnectApi.NamedCredential

ConnectApi.ExternalCredentialList

List of external credentials.

Property Name	Type	Description	Available Version
`externalCredentials`	List<`ConnectApi.ExternalCredential`>	List of external credentials.	56.0

ConnectApi.ExternalCredentialParameter

External credential parameter.

Property Name	Type	Description	Available Version
`id`	String	Parameter ID.	58.0
`parameterDescription`	String	Parameter description.	58.0
`parameterName`	String	Parameter name. If the `parameterType` is `AuthParameter`, valid values are: • `AwsAccountId`—Valid for AwsSv4. • `AwsProfileArn`—Valid for AwsSv4 with RolesAnywhere. • `AwsRegion`—Valid for AwsSv4. • `AwsService`—Valid for AwsSv4. • `AwsStsDuration`—Valid for AwsSv4 with STS or RolesAnywhere. • `AwsStsExternalId`—Valid for AwsSv4 with STS. • `AwsTrustAnchorArn`—Valid for AwsSv4 with RolesAnywhere.	57.0

Property Name	Type	Description	Available Version
		• `Scope`—Valid for OAuth. Other parameter types can be any value.	
`parameterType`	`ConnectApi.ExternalCredentialParameterType`	Parameter type of the external credential. Values are: • `AdditionalRefreshStatusCode` • `AuthParameter` • `AuthProvider` • `AuthProviderUrl` • `AuthProviderUrlQueryParameter` • `JwtBodyClaim` • `JwtHeaderClaim` • `SigningCertificate`	57.0
`parameterValue`	String	Parameter value. If the `parameterType` is `JwtBodyClaim` or `JwtHeaderClaim`, the parameter value can contain formulas. If the `parameterType` is `AuthProvider` or `SigningCertificate`, the parameter value is the fully qualified entity name of the corresponding entity.	57.0

SEE ALSO:

ConnectApi.ExternalCredential

ConnectApi.ExternalCredentialPrincipal

External credential principal.

If you don't have the View Setup and Configuration permission, some properties are empty or show limited information.

Property Name	Type	Description	Available Version
`authenticationStatus`	`ConnectApi.CredentialAuthenticationStatus`	Status of the credential authentication. Values are: • `Configured`—Credential has all required credentials for at least one principal. • `NotConfigured`—Credential isn't configured. • `Unknown`—Credential status can't be determined because the authentication protocol is custom.	56.0
`id`	String	ID of the external credential principal.	58.0

Property Name	Type	Description	Available Version
`parameters`	List<`ConnectApi.ExternalCredentialParameter`>	List of external credential parameters.	58.0
`principalAccess`	List<`ConnectApi.ExternalCredentialPrincipalAccess`>	List of access entities associated with the external credential principal.	58.0
`principalName`	String	Name of the external credential named principal.	56.0
`principalType`	`ConnectApi.CredentialPrincipalType`	Type of credential principal. Values are: • `AwsStsPrincipal` • `NamedPrincipal` • `PerUserPrincipal`	56.0
`sequenceNumber`	Integer	Sequence number of the external credential principal.	58.0

SEE ALSO:

ConnectApi.ExternalCredential

ConnectApi.ExternalCredentialPrincipalAccess

External credential principal access.

Property Name	Type	Description	Available Version
`developerName`	String	Developer name of the associated access entity.	58.0
`id`	String	ID of the associated access entity.	58.0
`type`	`ConnectApi.ExternalCredentialPrincipalAccessType`	Access type of the external credential principal. Values are: • `PermissionSet` • `PermissionSetGroup` • `Profile`	58.0

SEE ALSO:

ConnectApi.ExternalCredentialPrincipal

ConnectApi.ExternalFilePermissionInformation

External file permission information.

Property Name	Type	Description	Available Version
externalFilePermissionTypes	List<ConnectApi.ContentHubPermissionType>	Available permission types for the parent folder of the external file, or null for non-external files or when includeExternalFilePermissionsInfo is false.	39.0
externalFilePermissionsFailure	Boolean	true if the retrieval of external file information failed or if includeExternalFilePermissionsInfo is false; false otherwise.	39.0
externalFilePermissionsInfoFailureReason	String	Explanation of the failure if a failure occurred and includeExternalFilePermissionsInfo is true; null otherwise.	39.0
externalFileSharingStatus	ConnectApi.ContentHubExternalItemSharingType	Sharing status for the external file. Values are: • DomainSharing—File is shared with the domain. • PrivateSharing—File is private or shared only with individuals. • PublicSharing—File is publicly shared. Value is null for non-external files or when includeExternalFilePermissionsInfo is false.	39.0
repositoryPublicGroups	List<ConnectApi.RepositoryGroupSummary>	Available public groups in the external repository or null for non-external files or when includeExternalFilePermissionsInfo is false.	39.0

SEE ALSO:

ConnectApi.AbstractRepositoryFile

ConnectApi.ExternalManagedAccountAddressOutput

Default shipping address for an externally managed account.

Property Name	Type	Description	Available Version
city	String	City of the external managed account record.	53.0
country	String	Country of the external managed account record.	53.0
geolocationAccuracy	String	Geolocation accuracy of the external managed account record.	53.0
latitude	String	Latitude of the external managed account record.	53.0

Property Name	Type	Description	Available Version
`longitude`	String	Longitude of the external managed account record.	53.0
`state`	String	State of the external managed account record.	53.0
`street`	String	Street of the external managed account record.	53.0
`zip`	String	Postal code of the external managed account record.	53.0

SEE ALSO:

ConnectApi.ExternalManagedAccountOutput

ConnectApi.ExternalManagedAccountCollectionOutput

Collection of externally managed accounts.

Property Name	Type	Description	Available Version
`externalManagedAccounts`	List<`ConnectApi.ExternalManagedAccountOutput`>	Collection of externally managed accounts.	49.0
`totalExternalManagedAccounts`	Integer	Total number of externally managed accounts.	49.0

ConnectApi.ExternalManagedAccountOutput

Externally managed account.

Property Name	Type	Description	Available Version
`accountId`	String	ID of the account managed by another account.	49.0
`accountName`	String	Name of the external managed account record.	53.0
`address`	`ConnectApi.ExternalManagedAccountAddressOutput`	Default shipping address of the external managed account.	53.0
`externalManagedAccountId`	String	ID of the external managed account record.	49.0
`isMyAccount`	Boolean	Specifies whether the account is the context user's account (`true`) or not (`false`).	53.0

SEE ALSO:

ConnectApi.ExternalManagedAccountCollectionOutput

ConnectApi.FacetValue

Facet value.

This class is abstract and is a superclass of ConnectApi.DistinctFacetValue.

Property Name	Type	Description	Available Version
`type`	`ConnectApi.CommerceSearchFacetType`	Search facet type. Value is: • `DistinctValue`	52.0

ConnectApi.Features

Features available to the context user in the org.

Property	Type	Description	Available Version
`activityReminderNotificationsEnabled`	Boolean	Reserved for future use.	37.0
`chatter`	Boolean	Specifies whether Chatter is enabled.	28.0
`chatterActivity`	Boolean	Specifies whether user details include information about Chatter activity.	28.0
`chatterAnswers`	Boolean	Specifies whether Chatter Answers is enabled. Note: With the Spring '18 release, Salesforce no longer supports Chatter Answers. Users of Chatter Answers can post, answer, comment, or view existing Chatter Answers data, but support and updates are scheduled to end. We recommend transitioning to Chatter Questions. For more information, see End of Support for Chatter Answers in Spring '18.	29.0
`chatterGlobalInfluence`	Boolean	Specifies whether user details include global Chatter activity.	28.0
`chatterGroupRecords`	Boolean	Specifies whether Chatter groups can have records associated with them.	30.0
`chatterGroupRecordSharing`	Boolean	Specifies whether Chatter records are implicitly shared among group members when records are added to groups.	30.0
`chatterMessages`	Boolean	Specifies whether Chatter messages are enabled.	28.0
`chatterTopics`	Boolean	Specifies whether topics are enabled.	28.0
`communitiesEnabled`	Boolean	Specifies whether digital experiences is enabled.	31.0

Property	Type	Description	Available Version
`communityModeration`	Boolean	Specifies whether moderation is enabled.	29.0
`communityReputation`	Boolean	Specifies whether reputation is enabled.	32.0
`customFiscalCalendarEnabled`	Boolean	Specifies whether custom fiscal calendar is enabled.	63.0
`dashboardComponentSnapshots`	Boolean	Specifies whether the user can post dashboard component snapshots.	28.0
`defaultCurrencyIsoCode`	String	ISO code of the default currency. Applicable only when `multiCurrency` is `false`.	28.0
`defaultLocale`	String	Specifies the Default locale.	63.0
`einsteinVoiceEnabled`	Boolean	Reserved for future use.	46.0
`einsteinVoiceInPilotEnabled`	Boolean	Reserved for future use.	46.0
`einsteinVoiceLoggingEnabled`	Boolean	Reserved for future use.	46.0
`einsteinVoiceProviderId`	Integer	Reserved for future use.	46.0
`favoritesEnabled`	Boolean	Specifies whether favorites in Lightning are enabled.	41.0
`feedPolling`	Boolean	Reserved for future use.	28.0
`feedStreamEnabled`	Boolean	Specifies whether Chatter feed streams are enabled.	39.0
`files`	Boolean	Specifies whether files can act as resources for Connect REST API.	28.0
`filesOnComments`	Boolean	Specifies whether files can be attached to comments.	28.0
`fiscalYearStartMonth`	Integer	Specifies the number that corresponds to the month that starts the fiscal year for the org. For example, if the org's fiscal year starts in February, the value is 2.	63.0
`forecasting3AggregatedEnabled`	Boolean	Specifies whether aggregated forecasting is enabled for mobile clients.	38.0

Property	Type	Description	Available Version
`forecastingEnabled`	Boolean	Specifies whether forecasting is enabled.	38.0
`forecastingPeriodRange`	Integer	Range of the forecasting period.	38.0
`forecastingPeriodStart`	Integer	Start index for the forecasting period.	38.0
`forecastingPeriodType`	`ConnectApi.PeriodType`	Time period used for forecasting. Values are: • `Month` • `Quarter` • `Week` • `Year`	38.0
`groupsCanFollow`	Boolean	Reserved for future use.	28.0–29.0
`ideas`	Boolean	Specifies whether Ideas is enabled.	29.0
`liveAgentHostName`	String	Live Agent host name configured for the org.	41.0
`managedTopicsEnabled`	Boolean	Specifies whether managed topics are enabled.	32.0
`maxEntitySubscriptionsPerStream`	Integer	Specifies the maximum number of feed-enabled entities that can be subscribed to in a Chatter stream.	39.0
`maxFilesPerFeedItem`	Integer	Specifies the maximum number of files that can be added to a feed item.	36.0
`maxStreamsPerPerson`	Integer	Specifies the maximum number of Chatter streams that a user can have.	39.0
`mobileNotificationsEnabled`	Boolean	Reserved for future use.	29.0
`multiCurrency`	Boolean	Specifies whether the org uses multiple currencies (`true`) or not (`false`). When `false`, the `defaultCurrencyIsoCode` indicates the ISO code of the default currency.	28.0
`offlineEditEnabled`	Boolean	Specifies whether the offline object permissions are enabled for Salesforce for Android and Salesforce for iOS mobile clients.	37.0
`publisherActions`	Boolean	Specifies whether actions in the publisher are enabled.	28.0
`storeDataOnDevicesEnabled`	Boolean	Specifies whether the Salesforce for Android and Salesforce for iOS can use secure, persistent storage on mobile devices to cache data.	30.0

Property	Type	Description	Available Version
`thanksAllowed`	Boolean	Reserved for future use.	28.0
`trendingTopics`	Boolean	Specifies whether trending topics are enabled.	28.0
`userNavItemsEnabled`	Boolean	Specifies whether users can customize the navigation bar in Lightning.	41.0
`usesStartDateAsFiscalYearName`	Boolean	Specifies whether the calendar year when the fiscal year begins is referred to as the year of the company's fiscal year. For example, if the fiscal year begins in November 2025, but is referred to as fiscal year 2026, the value is `false`.	63.0
`viralInvitesAllowed`	Boolean	Specifies whether existing Chatter users can invite people in their company to use Chatter.	28.0
`wave`	Boolean	Specifies whether CRM Analytics is enabled.	36.0

SEE ALSO:

getSettings()

ConnectApi.OrganizationSettings

ConnectApi.Feed

Chatter feed.

Name	Type	Description	Available Version
`feedElementPostUrl`	String	Connect REST API URL to post feed elements to this subject.	31.0
`feedElements`	ConnectApi.FeedElementPage	Page of feed elements for the feed specified in `redirectedFeedType`. Otherwise, `null`.	40.0
`feedElementsUrl`	String	Connect REST API URL to feed elements.	31.0
`feedItemsUrl`	String	Connect REST API URL to feed items.	28.0–31.0
`isModifedUrl`	String	Connect REST API URL with a *`since`* request parameter that contains an opaque token that describes when the feed was last modified. Returns `null` if the feed isn't a news feed. Use this URL to poll a news feed for updates. **Important:** This feature is available through a Feed Polling pilot program. This pilot program is closed and not accepting new participants.	28.0
`pinnedFeedElementsUrl`	String	URL to pinned feed items.	41.0

Name	Type	Description	Available Version
`redirectedFeedFilter`	`ConnectApi.FeedFilter`	Filter for the feed specified in `redirectedFeedType`. Otherwise, `null`.	42.0
`redirectedFeedSort`	`ConnectApi.FeedSortOrder`	Sort order for the feed specified in `redirectedFeedType`. Otherwise, `null`.	42.0
`redirectedFeedType`	`ConnectApi.FeedType`	Specifies which feed is returned if `pageSize` is specified. Otherwise, `null`.	40.0
`respectsMute`	Boolean	Indicates whether the feed respects the mute feature. If `true`, the feed shows the ability to mute or unmute each element, depending on the value of `isMutedByMe`; `null` if the mute feature is disabled for the organization.	35.0

ConnectApi.FeedBody

Feed body.

Subclass of ConnectApi.AbstractMessageBody.

No additional properties.

SEE ALSO:

- ConnectApi.Comment
- ConnectApi.FeedElement
- ConnectApi.FeedEntitySummary

ConnectApi.FeedDirectory

Directory of feeds and favorites.

Name	Type	Description	Available Version
`favorites`	`List<ConnectApi.FeedFavorite>`	A list of feed favorites	30.0
`feeds`	`List<ConnectApi.FeedDirectoryItem>`	A list of feeds	30.0

ConnectApi.FeedDirectoryItem

Definition of a feed.

Name	Type	Description	Available Version
`feedElementsUrl`	String	Connect REST API resource URL for the feed elements.	
`feedItemsUrl`	String	Connect REST API resource URL for the feed items of a specific feed.	30.0–31.0
`feedType`	`ConnectApi.FeedType` Enum	The feed type. One of these values: • `Bookmarks`—Contains all feed items saved as bookmarks by the context user. • `Company`—Contains all feed items except feed items of type `TrackedChange`. To see the feed item, the user must have sharing access to its parent. • `DirectMessageModeration`—Contains all direct messages that are flagged for moderation. The Direct Message Moderation feed is available only to users with Moderate Experiences Chatter Messages permissions. • `DirectMessages`—Contains all feed items of the context user's direct messages. • `Draft`—Contains all the feed items that the context user drafted. • `Files`—Contains all feed items that contain files posted by people or groups that the context user follows. • `Filter`—Contains the news feed filtered to contain feed items whose parent is a specified object type. • `Groups`—Contains all feed items from all groups the context user either owns or is a member of. • `Home`—Contains all feed items associated with any managed topic in an Experience Cloud site. • `Isolated`—Contains all the feed items and comments that are isolated. • `Landing`—Contains all feed items that best drive user engagement when the feed is requested. Allows clients to avoid an empty feed when there aren't many personalized feed items. • `Moderation`—Contains all feed items that are flagged for moderation, except direct messages. The moderation feed is available only to users with Moderate Experiences Feeds permissions. • `Mute`—Contains all feed items that the context user muted. • `News`—Contains all updates for people the context user follows, groups the user is a member of, and files and records the user is following. Contains all updates for records whose parent is the context user. • `PendingReview`—Contains all feed items and comments that are pending review. • `People`—Contains all feed items posted by all people the context user follows.	30.0

Name	Type	Description	Available Version
		• `Record`—Contains all feed items whose parent is a specified record, which could be a group, user, object, file, or any other standard or custom object. When the record is a group, the feed also contains feed items that mention the group. When the record is a user, the feed contains only feed items on that user. You can get another user's record feed. • `Streams`—Contains all feed items for any combination of up to 25 feed-enabled entities that the context user subscribes to in a stream. Examples of feed-enabled entities include people, groups, and records, • `To`—Contains all feed items with mentions of the context user. Contains feed items the context user commented on and feed items created by the context user that are commented on. • `Topics`—Contains all feed items that include the specified topic. • `UserProfile`—Contains feed items created when a user changes records that can be tracked in a feed. Contains feed items whose parent is the user and feed items that @mention the user. This feed is different than the news feed, which returns more feed items, including group updates. You can get another user's user profile feed.	
`feedUrl`	String	Connect REST API resource URL for a specific feed	30.0
`keyPrefix`	String	A *key prefix* is the first three characters of a record ID, which specifies the object type. For filter feeds, this value is the key prefix associated with the object type used to filter this feed. All feed items in this feed have a parent whose object type matches this key prefix value. For non-filter feeds, this value is `null`.	30.0
`label`	String	Localized label of the feed	30.0

SEE ALSO:

ConnectApi.FeedDirectory

ConnectApi.FeedElement

Feed elements are the top-level items that a feed contains. Feeds are feed element containers.

This class is abstract.

Superclass of:

- ConnectApi.FeedItem
- ConnectApi.GenericFeedElement

Property Name	Type	Description	Available Version
`body`	`ConnectApi.FeedBody`	Information about the feed element. **Important:** Use the `header.text` property as the default value for rendering text because the `body.text` property can be `null`.	22.0
`capabilities`	`ConnectApi.FeedElementCapabilities`	A container for all capabilities that can be included with a feed element.	31.0
`createdDate`	Datetime	ISO 8601 format date string, for example, 2011-02-25T18:24:31.000Z.	31.0
`feedElementType`	`ConnectApi.FeedElementType`	Feed elements are the top-level objects that a feed contains. The feed element type describes the characteristics of that feed element. One of these values: • `Bundle`—A container of feed elements. A bundle also has a body made up of message segments that can always be gracefully degraded to text-only values. • `FeedItem`—A feed item has a single parent and is scoped to oneExperience Cloud site or across all Experience Cloud sites. A feed item can have capabilities such as bookmarks, canvas, content, comment, link, poll. Feed items have a body made up of message segments that can always be gracefully degraded to text-only values. • `Recommendation`—A recommendation is a feed element with a recommendations capability. A recommendation suggests records to follow, groups to join, or applications that are helpful to the context user.	31.0
`header`	`ConnectApi.MessageBody`	The header is the title of the post. This property contains renderable plain text for all the segments of the message. If a client doesn't know how to render a feed element type, it should render this text.	31.0
`id`	String	18-character ID of the feed element.	22.0
`modifiedDate`	Datetime	ISO 8601 format date string, for example, 2011-02-25T18:24:31.000Z.	31.0
`parent`	`ConnectApi.ActorWithId`	Feed element's parent	28.0

Property Name	Type	Description	Available Version
relativeCreatedDate	String	The created date formatted as a relative, localized string, for example, "17m ago" or "Yesterday."	31.0
url	String	Connect REST API URL to this feed element.	22.0

SEE ALSO:

- ConnectApi.Announcement
- ConnectApi.FeedElementPage
- ConnectApi.PinnedFeedElements
- ConnectApi.QuestionAndAnswersSuggestions

ConnectApi.FeedElementCapabilities

A container for all capabilities that can be included with a feed element.

Property Name	Type	Description	Available Version
approval	ConnectApi.ApprovalCapability	If a feed element has this capability, it includes information about an approval.	32.0
associatedActions	ConnectApi.AssociatedActionsCapability	If a feed element has this capability, it has platform actions associated with it.	33.0
banner	ConnectApi.BannerCapability	If a feed element has this capability, it has a banner motif and style.	31.0
bookmarks	ConnectApi.BookmarksCapability	If a feed element has this capability, the context user can bookmark it.	31.0
bundle	ConnectApi.BundleCapability	If a feed element has this capability, it has a container of feed elements called a *bundle*.	31.0
callCollaboration	ConnectApi.CallCollaborationCapability	If a feed element has this capability, it has a recording comment.	51.0
canvas	ConnectApi.CanvasCapability	If a feed element has this capability, it renders a canvas app.	32.0
caseComment	ConnectApi.CaseCommentCapability	If a feed element has this capability, it has a case comment on the case feed.	32.0
chatterLikes	ConnectApi.ChatterLikesCapability	If a feed element has this capability, the context user can like it. Exposes information about existing likes.	31.0
close	ConnectApi.CloseCapability	If a feed element has this capability, users with permission can close it.	43.0

Property Name	Type	Description	Available Version
comments	ConnectApi.CommentsCapability	If a feed element or comment has this capability, the context user can add a comment to it.	31.0
content	ConnectApi.ContentCapability	If a comment has this capability, it has a file attachment. Most ConnectApi.ContentCapability properties are null if the content has been deleted from the feed element or if the access has changed to private. **Important:** In version 36.0 and later, use the files property.	32.0–35.0
dashboardComponentSnapshot	ConnectApi.DashboardComponentSnapshotCapability	If a feed element has this capability, it has a dashboard component snapshot. A snapshot is a static image of a dashboard component at a specific point in time.	32.0
directMessage	ConnectApi.DirectMessageCapability	If a feed element has this capability, it's a direct message.	39.0
edit	ConnectApi.EditCapability	If a feed element has this capability, users who have permission can edit it.	34.0
emailMessage	ConnectApi.EmailMessageCapability	If a feed element has this capability, it has an email message from a case.	32.0
enhancedLink	ConnectApi.EnhancedLinkCapability	If a feed element has this capability, it has a link that may contain supplemental information like an icon, a title, and a description.	32.0
extensions	ConnectApi.ExtensionsCapability	If a feed element has this capability, it has one or more extension attachments.	40.0
feedEntityShare	ConnectApi.FeedShareCapability	If a feed element or comment has this capability, a feed entity is shared with it.	39.0
files	ConnectApi.FilesCapability	If a feed element has this capability, it has one or more file attachments.	36.0
interactions	ConnectApi.InteractionsCapability	If a feed element has this capability, it has information about user interactions.	37.0
link	ConnectApi.LinkCapability	If a feed element has this capability, it has a link.	32.0
mediaReferences	ConnectApi.MediaReferenceCapability	If a feed element has this capability, it has one or more media references.	41.0

Property Name	Type	Description	Available Version
moderation	ConnectApi.ModerationCapability	If a feed element has this capability, users in an Experience Cloud site can flag it for moderation.	31.0
mute	ConnectApi.MuteCapability	If a feed element has this capability, users can mute it.	35.0
origin	ConnectApi.OriginCapability	If a feed element has this capability, it was created by a feed action.	33.0
pin	ConnectApi.PinCapability	If a feed element has this capability, users who have permission can pin it to a feed.	41.0
poll	ConnectApi.PollCapability	If a feed element has this capability, it includes a poll.	31.0
questionAndAnswers	ConnectApi.QuestionAndAnswersCapability	If a feed element has this capability, it has a question and comments on the feed element are answers to the question.	31.0
readBy	ConnectApi.ReadByCapability	If a feed element has this capability, the context user can mark it as read.	40.0
recommendations	ConnectApi.RecommendationsCapability	If a feed element has this capability, it has a recommendation.	32.0
recordSnaphot	ConnectApi.RecordSnapshotCapability	If a feed element has this capability, it contains all the snapshotted fields of a record for a single create record event.	32.0
socialPost	ConnectApi.SocialPostCapability	If a feed element has this capability, it can interact with a social post on a social network.	36.0
status	ConnectApi.StatusCapability	If a feed post or comment has this capability, it has a status that determines its visibility.	37.0
topics	ConnectApi.TopicsCapability	If a feed element has this capability, the context user can add topics to it. Topics help users organize and discover conversations.	31.0
trackedChanges	ConnectApi.TrackedChangesCapability	If a feed element has this capability, it contains all changes to a record for a single tracked change event.	32.0

Property Name	Type	Description	Available Version
`upDownVote`	`ConnectApi.UpDownVoteCapability`	If a feed post or comment has this capability, users can upvote or downvote it.	41.0

SEE ALSO:

- ConnectApi.FeedElement
- ConnectApi.FeedItemSummary

ConnectApi.FeedElementCapability

A feed element capability, which defines the characteristics of a feed element.

In API version 30.0 and earlier, most feed items can have comments, likes, topics, and so on. In version 31.0 and later, every feed item (and feed element) can have a unique set of *capabilities*. If a capability property exists on a feed element, that capability is available, even if the capability property doesn't have a value. For example, if the `ChatterLikes` capability property exists on a feed element (with or without a value), the context user can like that feed element. If the capability property doesn't exist, it isn't possible to like that feed element. A capability can also contain associated data. For example, the `Moderation` capability contains data about moderation flags.

This class is abstract.

This class is a superclass of:

- ConnectApi.AssociatedActionsCapability
- ConnectApi.ApprovalCapability
- ConnectApi.BannerCapability
- ConnectApi.BookmarksCapability
- ConnectApi.BundleCapability
- ConnectApi.CallCollaborationCapability
- ConnectApi.CanvasCapability
- ConnectApi.CaseCommentCapability
- ConnectApi.ChatterLikesCapability
- ConnectApi.CloseCapability
- ConnectApi.CommentsCapability
- ConnectApi.ContentCapability
- ConnectApi.DashboardComponentSnapshotCapability
- ConnectApi.DirectMessageCapability
- ConnectApi.EmailMessageCapability
- ConnectApi.EnhancedLinkCapability
- ConnectApi.ExtensionsCapability
- ConnectApi.FeedEntityShareCapability
- ConnectApi.FilesCapability
- ConnectApi.InteractionsCapability
- ConnectApi.LinkCapability

- ConnectApi.MediaReferenceCapability
- ConnectApi.ModerationCapability
- ConnectApi.MuteCapability
- ConnectApi.OriginCapability
- ConnectApi.PinCapability
- ConnectApi.PollCapability
- ConnectApi.QuestionAndAnswersCapability
- ConnectApi.ReadByCapability
- ConnectApi.RecommendationsCapability
- ConnectApi.RecordCapability
- ConnectApi.RecordSnapshotCapability
- ConnectApi.SocialPostCapability
- ConnectApi.StatusCapability
- ConnectApi.TopicsCapability
- ConnectApi.TrackedChangesCapability
- ConnectApi.UpDownVoteCapability
- ConnectApi.VerifiedCapability

This class doesn't have any properties.

ConnectApi.FeedElementPage

A paged collection of `ConnectApi.FeedElement` objects.

Property Name	Type	Description	Available Version
`currentPageToken`	String	Token identifying the current page.	31.0
`currentPageUrl`	String	Connect REST API URL identifying the current page.	31.0
`elements`	List<`ConnectApi.FeedElement`>	Collection of feed elements.	31.0
`isModifiedToken`	String	Opaque polling token to use in the *`since`* parameter of the `ChatterFeeds.isModified` method. The token describes when the feed was last modified. **Important:** This feature is available through a Feed Polling pilot program. This pilot program is closed and not accepting new participants.	31.0
`isModifiedUrl`	String	Connect REST API URL with a *`since`* request parameter that contains an opaque token that describes when the feed was last modified. Returns `null` if the feed isn't a news feed. Use this URL to poll a news feed for updates.	31.0

Property Name	Type	Description	Available Version
		Important: This feature is available through a Feed Polling pilot program. This pilot program is closed and not accepting new participants.	
`nextPageToken`	String	Token identifying the next page, or `null` if there isn't a next page.	31.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	31.0
`updatesToken`	String	A token to use in a request to the `ConnectApi.ChatterFeeds.getFeedElementsUpdatedSince` method.	31.0
`updatesUrl`	String	Connect REST API feed resource containing the feed elements that have been updated since the feed was refreshed. If the feed doesn't support this feature, the value is `null`.	31.0

SEE ALSO:

ConnectApi.BundleCapability

ConnectApi.Feed

ConnectApi.FeedEnabledEntity

An entity that can have feeds associated with it.

Property Name	Type	Description	Available Version
`id`	String	The 18-character ID of the record.	39.0
`motif`	`ConnectApi.Motif`	Small, medium, and large icons indicating the record's type.	39.0
`name`	String	The localized name of the record.	39.0
`type`	String	The type of the record.	39.0
`url`	String	URL to the record.	39.0

SEE ALSO:

ConnectApi.ChatterStream

ConnectApi.FeedEntityIsEditable

Indicates if the context user can edit a feed element or comment.

Property Name	Type	Description	Available Version
`areAttachmentsEditableByMe`	Boolean	`true` if the context user can add and remove attachments on the feed element or comment, `false` otherwise.	36.0
`feedEntityUrl`	String	URL of the feed element or comment.	34.0
`isEditableByMe`	Boolean	`true` if the context user can edit the feed element or comment, `false` otherwise.	34.0

ConnectApi.FeedEntityNotAvailableSummary

A summary when the feed entity isn't available.

Subclass of ConnectApi.FeedEntitySummary.

No additional properties.

ConnectApi.FeedEntityReadSummary

Summary of the feed post or comment that was read.

Subclass of ConnectApi.UserFeedEntityActivitySummary.

No additional properties.

ConnectApi.FeedEntityShareCapability

If a feed element or comment has this capability, a feed entity is shared with it.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
`feedEntity`	`ConnectApi.FeedEntitySummary`	The summary of the feed entity that is shared with the feed element or comment.	39.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.FeedEntitySummary

The summary of a feed entity that is shared with a feed element.

This class is abstract.

Superclass of:

- ConnectApi.FeedItemSummary
- ConnectApi.FeedEntityNotAvailableSummary

Property Name	Type	Description	Available Version
`actor`	`ConnectApi.Actor`	Entity that created the feed entity.	39.0
`body`	`ConnectApi.FeedBody`	Information about the feed entity.	39.0
`createdDate`	Datetime	ISO 8601 date string, for example, 2011-02-25T18:24:31.000Z, when the entity was created.	39.0
`feedElementType`	`ConnectApi.FeedElementType`	Type of feed entity. • `Bundle`—A container of feed elements. A bundle also has a body made up of message segments that can always be gracefully degraded to text-only values. • `FeedItem`—A feed item has a single parent and is scoped to oneExperience Cloud site or across all Experience Cloud sites. A feed item can have capabilities such as bookmarks, canvas, content, comment, link, poll. Feed items have a body made up of message segments that can always be gracefully degraded to text-only values. • `Recommendation`—A recommendation is a feed element with a recommendations capability. A recommendation suggests records to follow, groups to join, or applications that are helpful to the context user.	39.0
`id`	String	18-character ID of the feed entity.	39.0
`isEntityAvailable`	Boolean	Specifies whether the entity is available. If `false`, either the user doesn't have access to the entity or the entity was deleted.	39.0
`parent`	`ConnectApi.ActorWithId`	Parent of the feed entity.	39.0
`relativeCreatedDate`	String	Relative created date, for example, "2h ago."	39.0
`url`	String	URL to the feed entity.	39.0

SEE ALSO:

ConnectApi.FeedEntityShareCapability

ConnectApi.FeedFavorite

Feed favorite.

Name	Type	Description	Available Version
`community`	`ConnectApi.Reference`	Information about the Experience Cloud site that contains the favorite.	28.0
`createdBy`	`ConnectApi.UserSummary`	Favorite's creator.	28.0
`feedUrl`	String	Connect REST API URL identifying the feed item for this favorite.	28.0
`id`	String	Favorite's 18–character ID.	28.0
`lastViewDate`	Datetime	ISO 8601 date string, for example, 2011-02-25T18:24:31.000Z.	28.0
`name`	String	Favorite's name.	28.0
`searchText`	String	If the favorite is from a search, contains the search text, otherwise, an empty string.	28.0
`target`	`ConnectApi.Reference`	A reference to the topic if applicable, `null` otherwise.	28.0
`type`	`ConnectApi.FeedFavoriteType` Enum	An empty string or one of the following values: • `ListView` • `Search` • `Topic`	28.0
`url`	String	Connect REST API URL to this favorite.	28.0
`user`	`ConnectApi.UserSummary`	Information about the user who saved this favorite.	28.0

SEE ALSO:

ConnectApi.FeedDirectory

ConnectApi.FeedFavorites

ConnectApi.FeedFavorites

Feed favorites.

Name	Type	Description	Available Version
`favorites`	`List<ConnectApi.FeedFavorite>`	Complete list of favorites.	28.0
`total`	Integer	Total number of favorites.	28.0

ConnectApi.FeedItem

Feed item.

Subclass of ConnectApi.FeedElement Class as of version 31.0.

Name	Type	Description	Available Version
`actor`	`ConnectApi.Actor`	The entity that created the feed item.	28.0
`attachment`	`ConnectApi.FeedItemAttachment`	Information about the attachment. If there is no attachment, returns `null`. **Important:** As of version 32.0, use the inherited `capabilities` property.	28.0–31.0
`canShare`	Boolean	Indicates whether the feed item can be shared. If a feed item has multiple file attachments and at least one attachment has been deleted or is inaccessible, the feed item can't be shared. The `canShare` value is incorrectly set to `true` in these cases. **Important:** As of version 39.0, use the `isSharable` property.	28.0–38.0
`clientInfo`	`ConnectApi.ClientInfo`	Information about the connected app used to authenticate the connection.	28.0
`comments`	`ConnectApi.CommentPage`	First page of comments for this feed item. **Important:** As of version 32.0, use the inherited `capabilities.comments.page` property.	28.0–31.0
`event`	Boolean	`true` if feed item is created due to an event change, `false` otherwise.	22.0
`hasVerifiedComment`	Boolean	`true` if the feed item has a verified comment, otherwise `false`.	41.0
`isBookmarkedByCurrentUser`	Boolean	`true` if the context user has bookmarked this feed item, otherwise, `false`. **Important:** As of version 32.0, use the inherited `capabilities.bookmarks.isBookmarkedByCurrentUser` property.	28.0–31.0
`isDeleteRestricted`	Boolean	If this property is `true` the comment cannot be deleted by the context user. If it is `false`, it might be possible for the context user to delete the comment, but it is not guaranteed.	28.0

Name	Type	Description	Available Version
`isLikedBy CurrentUser`	Boolean	`true` if the context user has liked this feed item, otherwise, `false`. **Important:** As of version 32.0, use the inherited `capabilities.chatterLikes.isLikedByCurrentUser` property.	28.0–31.0
`isSharable`	Boolean	Indicates whether the feed item can be shared.	39.0
`likes`	`ConnectApi.ChatterLikePage`	First page of likes for this feed item. **Important:** As of version 32.0, use the inherited `capabilities.chatterLikes.page` property.	28.0–31.0
`likesMessage`	`ConnectApi.MessageBody`	A message body the describes who likes the feed item. **Important:** As of version 32.0, use the inherited `capabilities.chatterLikes.likesMessage` property.	28.0–31.0
`moderationFlags`	`ConnectApi.ModerationFlags`	Information about the moderation flags on a feed item. If `ConnectApi.Features.communityModeration` is `false`, this property is `null`. **Important:** As of version 31.0, use the inherited `capabilities.moderation.moderationFlags` property.	29.0–30.0
`myLike`	`ConnectApi.Reference`	If the context user has liked the feed item, this property is a reference to the specific like, otherwise, `null`. **Important:** As of version 32.0, use the inherited `capabilities.chatterLikes.myLike` property.	28.0–31.0
`originalFeedItem`	`ConnectApi.Reference`	A reference to the original feed item if this feed item is a shared feed item, otherwise, `null`.	28.0
`originalFeed ItemActor`	`ConnectApi.Actor`	If this feed item is a shared feed item, returns information about the original poster of the feed item, otherwise, returns `null`.	28.0

Name	Type	Description	Available Version
`photoUrl`	String	URL of the photo associated with the feed item	28.0
`preamble`	`ConnectApi.MessageBody`	A collection of message segments, including the unformatted text of the message that you can use as the title of a feed item. Message segments include name, link, and motif icon information for the actor that created the feed item. **Important:** For API versions 29.0 and 30.0, use the `ConnectApi.FeedItem.preamble.text` property as the default case to render text. For API versions 31.0 and later, use the `ConnectApi.FeedElement.header.text` property as the default case to render text.	28.0-30.0
`topics`	`ConnectApi.FeedItemTopicPage`	Topics for this feed item. **Important:** As of version 31.0, use the inherited `capabilities.topics.items` property.	28.0–31.0
`type`	`ConnectApi.FeedItemType`	Type of feed item. **Important:** As of API version 32.0, use the `capabilities` property to determine what can be done with a feed item. See Working with Feeds and Feed Elements. One of these values: • `ActivityEvent`—Feed item generated in Case Feed when an event or task associated with a parent record with a feed enabled is created or updated. • `AdvancedTextPost`—A feed item with advanced text formatting, such as a group announcement post. • `ApprovalPost`—Feed item with an approval capability. Approvers can act on the feed item parent. • `AttachArticleEvent`—Feed item generated when an article is attached to a case in Case Feed. • `BasicTemplateFeedItem`—Feed item with an enhanced link capability.	28.0

Name	Type	Description	Available Version
		• `CallLogPost`—Feed item generated when a call log is saved to a case in Case Feed. • `CanvasPost`—Feed item generated by a canvas app in the publisher or from Connect REST API or Connect in Apex. The post itself is a link to a canvas app. • `CaseCommentPost`—Feed item generated when a case comment is saved in Case Feed. • `ChangeStatusPost`—Feed item generated when the status of a case is changed in Case Feed. • `ChatTranscriptionPost`—Feed item generated in Case Feed when a Live Agent chat transcript is saved to a case. • `CollaborationGroupCreated`—Feed item generated when a new public group is created. Contains a link to the new group. • ~~`CollaborationGroupUnarchived`—Deprecated.~~ Feed item generated when an archived group is activated. • `ContentPost`—Feed item with a content capability. • `CreateRecordEvent`—Feed item that describes a record created in the publisher. • `DashboardComponentAlert`—Feed item with a dashboard alert. • `DashboardComponentSnapshot`—Feed item with a dashboard component snapshot capability. • `EmailMessageEvent`—Feed item generated when an email is sent from a case in Case Feed. • `FacebookPost`—Deprecated. Feed item generated when a Facebook post is created from a case in Case Feed. • `LinkPost`—Feed item with a link capability. • `MilestoneEvent`—Feed item generated when a case milestone is either completed or reaches a violation status. Contains a link to the case milestone. • `PollPost`—Feed item with a poll capability. Viewers of the feed item are allowed to vote on the options in the poll.	

Name	Type	Description	Available Version
		• `ProfileSkillPost`—Feed item generated when a skill is added to a user's profile. • `QuestionPost`—Feed item generated when a question is asked. As of API version 33.0, a feed item of this type can have a content capability and a link capability. • `ReplyPost`—Feed item generated by a Chatter Answers reply. • `RypplePost`—Feed item generated when a user posts thanks. • `SocialPost`—Feed item generated when a social post is created from a case in Case Feed. • `TextPost`—Feed item containing text only. • `TrackedChange`—Feed item created when one or more fields on a record have been changed. • `UserStatus`—Deprecated. A user's post to their own profile.	
`visibility`	`ConnectApi.FeedItemVisibilityType`	Type of users who can see a feed item. • `AllUsers`—Visibility is not limited to internal users. • `InternalUsers`—Visibility is limited to internal users.	28.0

ConnectApi.FeedItemSummary

A feed item summary.

Subclass of ConnectApi.FeedEntitySummary.

Property Name	Type	Description	Available Version
`capabilities`	`ConnectApi.FeedElementCapabilities`	Container for all capabilities that can be included with a feed item.	39.0
`header`	`ConnectApi.MessageBody`	Title of the post. This property contains renderable plain text for all the message segments. If a client doesn't know how to render a feed element type, it should render this text.	39.0

Property Name	Type	Description	Available Version
`modifiedDate`	Datetime	When the feed item was modified in the form of an ISO 8601 date string, for example, 2011-02-25T18:24:31.000Z.	39.0
`originalFeedItem`	`ConnectApi.Reference`	Reference to the original feed item if this feed item is a shared feed item; otherwise, `null`.	39.0
`originalFeedItemActor`	`ConnectApi.Actor`	If this feed item is a shared feed item, information about the original poster of the feed item; otherwise, `null`.	39.0
`photoUrl`	String	URL of the photo associated with the feed item.	39.0
`visibility`	`ConnectApi.FeedItemVisibility`	Specifies who can see a feed item. • `AllUsers`—Visibility is not limited to internal users. • `InternalUsers`—Visibility is limited to internal users.	39.0

ConnectApi.FeedModifiedInfo

Feed modified information.

Important: This feature is available through a Feed Polling pilot program. This pilot program is closed and not accepting new participants.

Name	Type	Description	Available Version
`isModified`	Boolean	`true` if the news feed has been modified since the last time it was polled; `false` otherwise. Returns `null` if the feed is not a news feed.	28.0
`isModifiedToken`	String	Opaque polling token to use in the *`since`* parameter of the `ChatterFeeds.isModified` method. The token describes when the feed was last modified.	28.0
`nextPollUrl`	String	Connect REST API URL with a *`since`* request parameter that contains an opaque token that describes when the feed was last modified. Returns `null` if the feed isn't a news feed. Use this URL to poll a news feed for updates.	28.0

ConnectApi.FeedPollChoice

Feed poll choice.

Name	Type	Description	Available Version
`id`	String	Poll choice ID.	28.0

Name	Type	Description	Available Version
`position`	Integer	The location in the poll where this poll choice exists. The first poll choice starts at 1.	28.0
`text`	String	Label text associated with the poll choice.	28.0
`voteCount`	Integer	Total number of votes for this poll choice.	28.0
`voteCountRatio`	Double	The ratio of total number of votes for this poll choice to all votes cast in the poll. Multiply the ratio by 100 to get the percentage of votes cast for this poll choice.	28.0

SEE ALSO:

ConnectApi.PollCapability

ConnectApi.FeedPostSummary

Summary of the post.

Subclass of ConnectApi.UserActivitySummary.

Property Name	Type	Description	Available Version
`feedItemId`	String	ID of the post.	42.0

ConnectApi.FeedReadSummary

Summary of the feed that was read.

Subclass of ConnectApi.UserActivitySummary.

Property Name	Type	Description	Available Version
`containerId`	String	ID of the parent of the feed.	42.0
`feedType`	`ConnectApi.FeedType`	Type of feed. • `Bookmarks`—Contains all feed items saved as bookmarks by the context user. • `Company`—Contains all feed items except feed items of type `TrackedChange`. To see the feed item, the user must have sharing access to its parent. • `DirectMessageModeration`—Contains all direct messages that are flagged for moderation. The Direct Message Moderation feed is available only to users with Moderate Experiences Chatter Messages permissions. • `DirectMessages`—Contains all feed items of the context user's direct messages.	42.0

Property Name	Type	Description	Available Version
		• `Draft`—Contains all the feed items that the context user drafted. • `Files`—Contains all feed items that contain files posted by people or groups that the context user follows. • `Filter`—Contains the news feed filtered to contain feed items whose parent is a specified object type. • `Groups`—Contains all feed items from all groups the context user either owns or is a member of. • `Home`—Contains all feed items associated with any managed topic in an Experience Cloud site. • `Isolated`—Contains all the feed items and comments that are isolated. • `Landing`—Contains all feed items that best drive user engagement when the feed is requested. Allows clients to avoid an empty feed when there aren't many personalized feed items. • `Moderation`—Contains all feed items that are flagged for moderation, except direct messages. The moderation feed is available only to users with Moderate Experiences Feeds permissions. • `Mute`—Contains all feed items that the context user muted. • `News`—Contains all updates for people the context user follows, groups the user is a member of, and files and records the user is following. Contains all updates for records whose parent is the context user. • `PendingReview`—Contains all feed items and comments that are pending review. • `People`—Contains all feed items posted by all people the context user follows. • `Record`—Contains all feed items whose parent is a specified record, which could be a group, user, object, file, or any other standard or custom object. When the record is a group, the feed also contains feed items that mention the group. When the record is a user, the feed contains only feed items on that user. You can get another user's record feed.	

Property Name	Type	Description	Available Version
		• `Streams`—Contains all feed items for any combination of up to 25 feed-enabled entities that the context user subscribes to in a stream. Examples of feed-enabled entities include people, groups, and records, • `To`—Contains all feed items with mentions of the context user. Contains feed items the context user commented on and feed items created by the context user that are commented on. • `Topics`—Contains all feed items that include the specified topic. • `UserProfile`—Contains feed items created when a user changes records that can be tracked in a feed. Contains feed items whose parent is the user and feed items that @mention the user. This feed is different than the news feed, which returns more feed items, including group updates. You can get another user's user profile feed.	

ConnectApi.FieldChangeNameSegment

Field change name segment.

Subclass of ConnectApi.MessageSegment.

No additional properties.

ConnectApi.FieldChangeSegment

Field change segment.

Subclass of ConnectApi.ComplexSegment.

No additional properties.

SEE ALSO:

ConnectApi.MoreChangesSegment

ConnectApi.FieldChangeValueSegment

Field change value segment.

Subclass of ConnectApi.MessageSegment.

Name	Type	Description	Available Version
`valueType`	`ConnectApi.FieldChangeValueType Enum`	Value type of a field change. • `NewValue`—A new value • `OldValue`—An old value	28.0
`url`	String	URL value if the field change is to a URL field (such as a web address)	28.0

ConnectApi.FieldMetadata

Search metadata for the field of an object.

Property Name	Type	Description	Available Version
`domain`	String	The object that the field is associated with.	63.0
`field`	String	Field path through the object.	63.0
`fieldApiName`	String	API name of the field.	63.0
`fieldType`	`ConnectApi.FieldType`	Field type. Values are: • `Address` • `AnyType` • `Base64` • `Boolean` • `Combobox` • `ComplexValue` • `Currency` • `DataCategoryGroupReference` • `Date` • `DateTime` • `Double` • `Email` • `EncryptedString` • `ExtensionEntityLookup` • `ExternalLookup` • `FloatArray` • `Id` • `ImageUrl` • `IndirectLookup` • `Integer` • `Json` • `Location`	63.0

Property Name	Type	Description	Available Version
		• `Long` • `MultiPicklist` • `Percent` • `PersonName` • `Phone` • `Picklist` • `PlainTextArea` • `Reference` • `RichTextArea` • `Sobject` • `String` • `SwitchablePersonName` • `TextArea` • `Time` • `Url`	
`filterable`	Boolean	Specifies whether the field is filterable (`true`) or not (`false`).	63.0
`highlightable`	Boolean	Specifies whether the field is highlightable (`true`) or not (`false`).	63.0
`label`	String	Label of the field.	63.0
`sortable`	Boolean	Specifies whether the field is sortable (`true`) or not (`false`).	63.0

SEE ALSO:

ConnectApi.ObjectMetadata

ConnectApi.FieldValue

Field's value in product search results.

Property Name	Type	Description	Available Version
`value`	String	Value of the field.	52.0

SEE ALSO:

ConnectApi.ProductSummary

ConnectApi.File

File.

This class is abstract.

Subclass of ConnectApi.ActorWithId.

Superclass of ConnectApi.FileSummary.

Name	Type	Description	Available Version
`checksum`	String	MD5 checksum for the file.	28.0
`contentModifiedDate`	Datetime	ISO 8601 format date string, for example, 2011-02-25T18:24:31.000Z. File-specific modified date, which is updated only for direct file operations, such as rename. Modifications to the file from outside of Salesforce can update this date.	32.0
`contentSize`	Integer	Size of the file in bytes.	28.0
`contentUrl`	String	If the file is a link, returns the URL, otherwise, the string `null`.	28.0
`createdDate`	Datetime	ISO 8601 date string when the file was created.	41.0
`description`	String	Description of the file.	28.0
`downloadUrl`	String	URL to the file, that can be used for downloading the file.	28.0
`fileExtension`	String	Extension of the file.	28.0
`fileType`	String	Type of file, such as PDF, PowerPoint.	28.0
`flashRenditionStatus`	String	Specifies if a flash preview version of the file has been rendered. Note: Flash renditions were retired on July 16, 2021.	28.0
`isFileAsset`	Boolean	Specifies whether the file is an asset.	46.0
`isInMyFileSync`	Boolean	`true` if the file is synced with Salesforce Files Sync; `false` otherwise. Note: Salesforce Files Sync was retired on May 25, 2018.	28.0
`isMajorVersion`	Boolean	`true` if the file is a major version; `false` if the file is a minor version. Major versions can't be replaced.	31.0
`mimeType`	String	File's MIME type.	28.0
`moderationFlags`	`ConnectApi.ModerationFlags`	Information about the moderation flags on a file. If `ConnectApi.Features.communityModeration` is `false`, this property is `null`.	30.0

Name	Type	Description	Available Version
`modifiedDate`	Datetime	ISO 8601 format date string, for example, 2011-02-25T18:24:31.000Z. Modifications to the file from within Salesforce update this date.	28.0
`name`	String	Name of the file.	28.0
`origin`	String	Specifies the file source. Valid values are: • `Chatter`—file came from Chatter • `Content`—file came from content	28.0
`owner`	`ConnectApi.UserSummary`	File's owner.	28.0
`pdfRenditionStatus`	String	Specifies if a PDF preview version of the file has been rendered.	28.0
`publishStatus`	`ConnectApi.FilePublishStatus`	Specifies the publish status of the file. • `PendingAccess`—File is pending publishing. • `PrivateAccess`—File is private. • `PublicAccess`—File is public.	28.0
`renditionUrl`	String	URL to the rendition for the file.	28.0
`renditionUrl240By180`	String	URL to the 240 x 180 rendition resource for the file. For shared files, renditions process asynchronously after upload. For private files, renditions process when the first file preview is requested, and aren't available immediately after the file is uploaded.	29.0
`renditionUrl720By480`	String	URL to the 720 x 480 rendition resource for the file. For shared files, renditions process asynchronously after upload. For private files, renditions process when the first file preview is requested, and aren't available immediately after the file is uploaded.	29.0
`sharingOption`	`ConnectApi.FileSharingOption`	Sharing option of the file. Values are: • `Allowed`—Resharing of the file is allowed. • `Restricted`—Resharing of the file is restricted.	35.0
`sharingPrivacy`	`ConnectApi.FileSharingPrivacy`	Sharing privacy of a file. Values are: • `None`—File is visible to anyone with record access. • `PrivateOnRecords`—File is private on records.	41.0
`sharingRole`	`ConnectApi.FileSharingType`	Sharing role of the file. • `Admin`—Owner permission, but doesn't own the file. • `Collaborator`—Viewer permission, and can edit, change permissions, and upload a new version of a file.	28.0

Name	Type	Description	Available Version
		• `Owner`—Collaborator permission, and can make a file private, and delete a file. • `Viewer`—Can view, download, and share a file. • `WorkspaceManaged`—Permission controlled by the library.	
`systemModstamp`	Datetime	ISO 8601 date string indicating when a user or any automated system process, such as a trigger, updated the file.	41.0
`textPreview`	String	Text preview of the file if available; `null` otherwise.	30.0
`thumb120By90 RenditionStatus`	String	Specifies the rendering status of the 120 x 90 preview image of the file. One of these values: • `Processing`—Image is being rendered. • `Failed`—Rendering process failed. • `Success`—Rendering process was successful. • `Na`—Rendering is not available for this image.	28.0
`thumb240By180 RenditionStatus`	String	Specifies the rendering status of the 240 x 180 preview image of the file. One of these values: • `Processing`—Image is being rendered. • `Failed`—Rendering process failed. • `Success`—Rendering process was successful. • `Na`—Rendering is not available for this image.	28.0
`thumb720By480 RenditionStatus`	String	Specifies the rendering status of the 720 x 480 preview image of the file. One of these values: • `Processing`—Image is being rendered. • `Failed`—Rendering process failed. • `Success`—Rendering process was successful. • `Na`—Rendering is not available for this image.	28.0
`title`	String	Title of the file.	28.0
`versionNumber`	String	File's version number.	28.0

ConnectApi.FileAsset

An asset file.

Property Name	Type	Description	Available Version
`baseAssetUrl`	String	Base download URL of the asset.	45.0

Property Name	Type	Description	Available Version
`baseUnauthenticatedAssetUrl`	String	Base download URL of the asset for unauthenticated users if `isVisibleByExternalUsers` is `true`, otherwise `null`.	45.0
`id`	String	ID of the asset.	45.0
`isVisibleByExternalUsers`	Boolean	Indicates whether unauthenticated users can see the asset file (`true`) or not (`false`).	45.0
`masterLabel`	String	Label of the asset.	45.0
`name`	String	Unique name of the asset.	45.0
`namespacePrefix`	String	Namespace prefix of the package containing the asset.	45.0
`type`	String	Type of asset.	45.0

SEE ALSO:

ConnectApi.Recommendation

ConnectApi.NBANativeRecommendation

ConnectApi.FilePreview

A file preview.

Property Name	Type	Description	Available Version
`format`	`ConnectApi.FilePreviewFormat`	The format of the preview. Values are: • `Jpg`—Preview format is JPG. • `Pdf`—Preview format is PDF. • `Svg`—Preview format is compressed SVG. • `Thumbnail`—Preview format is 240 x 180 PNG. • `ThumbnailBig`—Preview format is 720 x 480 PNG. • `ThumbnailTiny`—Preview format is 120 x 90 PNG.	39.0
`previewUrlCount`	Integer	The total number of preview URLs for this preview format.	39.0
`previewUrls`	List<`ConnectApi.FilePreviewUrl`>	A list of file preview URLs.	39.0
`status`	`ConnectApi.FilePreviewStatus`	The availability status of the preview. Values are: • `Available`—Preview is available. • `InProgress`—Preview is being processed.	39.0

Property Name	Type	Description	Available Version
		• `NotAvailable`—Preview is unavailable. • `NotScheduled`—Generation of the preview isn't scheduled yet.	
`url`	String	The URL for the file preview.	39.0

SEE ALSO:

ConnectApi.FilePreviewCollection

ConnectApi.FilePreviewCollection

A collection of file previews.

Property Name	Type	Description	Available Version
`fileId`	String	ID of the file.	39.0
`previews`	List<`ConnectApi.FilePreview`>	Previews supported for the file.	39.0
`url`	String	URL to the current page of file previews.	39.0
`versionNumber`	String	Version number of the file.	40.0

SEE ALSO:

ConnectApi.InlineImageSegment

ConnectApi.FilePreviewUrl

A URL to a file preview.

Property Name	Type	Description	Available Version
`pageNumber`	Integer	Preview page number starting from zero, or `null` for PDF files.	39.0
`previewUrl`	String	File preview URL.	39.0

SEE ALSO:

ConnectApi.FilePreview

ConnectApi.FilesCapability

If a feed element has this capability, it has one or more file attachments.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
items	List<ConnectApi.Content>	Collection of files.	36.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.FileSummary

A file summary.

Subclass of ConnectApi.File.

No additional properties.

ConnectApi.FindRoutesWithFewestSplitsOutputRepresentation

A list of inventory location combinations that can fulfill an order without exceeding the maximum number of shipments.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
targetLocations	List<ConnectApi.AvailableLocationOutputRepresentation>	Each element of the list is a set of inventory locations that together can fulfill the order being routed.	51.0

SEE ALSO:

findRoutesWithFewestSplits(findRoutesWithFewestSplitsInputRepresentation)

ConnectApi.FindRoutesWithFewestSplitsUsingOCIOutputRepresentation

A list of order fulfillment routes with inventory availability information.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
results	List<ConnectApi.FindRoutesWithFewestSplitsWithInventoryOutputRepresentation>	Each element of the list is the response for one element of the input list.	54.0

SEE ALSO:

findRoutesWithFewestSplitsUsingOCI(findRoutesWithFewestSplitsUsingOCIInput)

ConnectApi.FindRoutesWithFewestSplitsWithInventoryOutputRepresentation

Sets of inventory locations that can combine to fulfill an order, with availability data for those locations.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
`inventory`	`ConnectApi.OCIGetInventoryAvailabilityOutputRepresentation`	Inventory availability data for the location groups and locations specified in the input.	54.0
`targetLocations`	`List<ConnectApi.AvailableLocationOutputRepresentation>`	Each entry in the list is a set of inventory locations that can combine to fulfill an order.	54.0

SEE ALSO:

findRoutesWithFewestSplitsUsingOCI(findRoutesWithFewestSplitsUsingOCIInput)

ConnectApi.FindRoutesWithFewestSplitsUsingOCIOutputRepresentation

ConnectApi.FollowerPage

Page of followers.

Name	Type	Description	Available Version
`currentPageUrl`	String	Connect REST API URL identifying the current page.	28.0
`followers`	`List<ConnectApi.Subscription>`	List of subscriptions.	28.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	28.0
`previousPageUrl`	String	Connect REST API URL identifying the previous page, or `null` if there isn't a previous page.	28.0
`total`	Integer	Total number of followers across all pages.	28.0

ConnectApi.FollowingCounts

Following counts.

Name	Type	Description	Available Version
`people`	Integer	Number of people user is following.	28.0
`records`	Integer	Number of records user is following. Topics are a type of record that can be followed as of version 29.0.	28.0

Name	Type	Description	Available Version
`total`	Integer	Total number of items user is following.	28.0

SEE ALSO:

ConnectApi.UserDetail

ConnectApi.FollowingPage

Page of following subscriptions.

Name	Type	Description	Available Version
`currentPageUrl`	String	Connect REST API URL identifying the current page.	28.0
`following`	`List<ConnectApi.Subscription>`	List of subscriptions.	28.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	28.0
`previousPageUrl`	String	Connect REST API URL identifying the previous page, or `null` if there isn't a previous page.	28.0
`total`	Integer	Total number of records being followed across all pages.	28.0

ConnectApi.FollowIntents

A list of follow intents for a social persona.

Property Name	Type	Description	Available Version
`follows`	`List<ConnectApi.FollowSocialPersonaIntent>`	List of follow intents for the social persona.	45.0

SEE ALSO:

ConnectApi.SocialPostIntents

ConnectApi.FollowSocialPersonaIntent

Follow intent on a social persona.

Property Name	Type	Description	Available Version
`managedSocialAccount`	`ConnectApi.ManagedSocialAccount`	Managed social account that follows the social persona.	45.0

Property Name	Type	Description	Available Version
socialPersonaId	String	ID of the social persona to follow.	45.0

SEE ALSO:

ConnectApi.FollowIntents

ConnectApi.Form

Marketing integration form.

Property Name	Type	Description	Available Version
dataExtensionId	String	ID of the data extension associated with the marketing integration form.	53.0
formFieldsList	ConnectApi.FormFields	List of form fields associated with the marketing integration form.	53.0
formId	String	ID of the marketing integration form.	53.0
formName	String	Name of the marketing integration form.	53.0

ConnectApi.FormField

Marketing integration form field.

Property Name	Type	Description	Available Version
name	String	Name of the marketing integration form field.	53.0
type	ConnectApi.FormFieldType	Type of marketing integration form field. Values are: • Boolean • Date • EmailAddress • Number • Text	53.0

SEE ALSO:

ConnectApi.FormFields

ConnectApi.FormFields

List of marketing integration form fields.

Property Name	Type	Description	Available Version
formFields	List<ConnectApi.FormField>	List of form fields associated with the marketing integration form.	53.0

SEE ALSO:

ConnectApi.Form

ConnectApi.FormSubmission

Marketing integration form submission.

Property Name	Type	Description	Available Version
formSubmissionId	String	ID of the form submission, representing the submission data that was saved.	53.0

ConnectApi.FormulaScope

Formula scope for a target.

Property Name	Type	Description	Available Version
contextValues	Map<String, String>	Map of context values for the scope. Note: In version 52.0 and later, use the contextValuesMap property.	50.0–51.0
contextValuesMap	Map<String, Object>	Map of context values for the scope.	52.0
fields	List<String>	List of fields of the scope.	50.0
formula	String	Formula of the scope.	50.0

SEE ALSO:

ConnectApi.Target

ConnectApi.FulfillmentGroupOutputRepresentation

Information about one FulfillmentOrder from a request to create fulfillment orders from multiple OrderDeliveryGroupSummaries. If the FulfillmentOrder was created, then its ID is returned. If it failed, then data from the input is returned so you can resubmit it.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
fulfilledFromLocationId	String	(Creation failed) The input FulfilledFromLocationId.	50.0

Property Name	Type	Description	Available Version
`fulfillmentOrderId`	String	The FulfillmentOrderId from the successfully created FulfillmentOrder.	50.0
`fulfillmentType`	String	(Creation failed) The input FulfillmentType.	50.0
`orderDeliveryGroupSummaryId`	String	(Creation failed) The input OrderDeliveryGroupSummaryId.	50.0
`orderItemSummaries`	List<`ConnectApi.OrderItemSummaryOutputRepresentation`>	(Creation failed) The input list of OrderItemSummaries.	50.0
`orderSummaryId`	String	(Creation failed) The input OrderSummaryId.	50.0
`referenceId`	String	The referenceId from the FulfillmentGroup input representation. Use this value to troubleshoot a failure.	50.0

ConnectApi.FulfillmentOrderCancelLineItemsOutputRepresentation

Wraps the base output.

Subclass of ConnectApi.BaseOutputRepresentation.

No additional properties.

ConnectApi.FulfillmentOrderInvoiceOutputRepresentation

ID of the created invoice.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
`invoiceId`	String	ID of the created invoice.	48.0

ConnectApi.FulfillmentOrderOutputRepresentation

A list of IDs of the created FulfillmentOrders.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
`fulfillmentOrderIds`	List<String>	A list of IDs of created Fulfillment Orders.	48.0

ConnectApi.GatewayLogResponse

Gateway log output.

Property Name	Type	Description	Available Version
`createdDate`	Datetime	Date when the gateway log was created.	50.0
`gatewayResultCode`	String	Result codes that show the status of a transaction as it is passed to the financial institution and then returned to the client.	50.0
`id`	String	ID of the gateway log record.	50.0
`interactionStatus`	String	Gateway interaction status. It can be SUCCESS, FAILED, or TIMEOUT.	50.0

ConnectApi.GenericBundleCapability

If a feed element has this capability, the feed element has a group of other feed elements condensed into one feed element. This group is called a *bundle*.

Subclass of ConnectApi.BundleCapability.

ConnectApi.GenericFeedElement

A concrete implementation of the abstract `ConnectApi.FeedElement` class.

Subclass of ConnectApi.FeedElement.

ConnectApi.GetFOCapacityValuesOutputRepresentation

Response to a request for fulfillment order capacity values for one or more locations.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
`locations`	List<`ConnectApi.LocationCapacityOutputRepresentation`>	List of fulfillment order capacity values for one or more locations.	55.0

ConnectApi.GlobalInfluence

Chatter influence.

Name	Type	Description	Available Version
`percentile`	String	Percentile value for the user's influence rank within the org or Experience Cloud site.	28.0

Name	Type	Description	Available Version
`rank`	Integer	Number indicating the user's influence rank, relative to all other users within the org or Experience Cloud site.	28.0

SEE ALSO:

ConnectApi.UserDetail

ConnectApi.GroupChatterSettings

A user's Chatter settings for a specific group.

Name	Type	Description	Available Version
`emailFrequency`	`ConnectApi.GroupEmailFrequency` on page 2366	The frequency with which a group member receives email from a group.	28.0

ConnectApi.GroupInformation

Describes the Information section of the group. If the group is private, this section is visible only to members.

Name	Type	Description	Available Version
`text`	String	The text of the "Information" section of the group.	28.0
`title`	String	The title of the "Information" section of the group.	28.0

SEE ALSO:

ConnectApi.ChatterGroupDetail

ConnectApi.GroupMember

Member of a group.

Name	Type	Description	Available Version
`id`	String	User's 18-character ID.	28.0
`lastFeedAccessDate`	Datetime	The date and time at which the group member last accessed the group feed.	31.0
`role`	`ConnectApi.GroupMembershipType` Enum	Type of membership the user has with the group. • `GroupOwner` • `GroupManager` • `NotAMember` • `NotAMemberPrivateRequested`	28.0

Name	Type	Description	Available Version
		• StandardMember	
url	String	Connect REST API URL to this membership.	28.0
user	ConnectApi.UserSummary	Information about the user who is subscribed to this group.	28.0

SEE ALSO:

ConnectApi.GroupMemberPage

ConnectApi.GroupMemberPage

Page of group members.

Name	Type	Description	Available Version
currentPageUrl	String	Connect REST API URL identifying the current page.	28.0
members	List<ConnectApi.GroupMember>	List of group members.	28.0
myMembership	ConnectApi.Reference	If the context user is a member of this group, returns information about that membership, otherwise, null.	28.0
nextPageUrl	String	Connect REST API URL identifying the next page, or null if there isn't a next page.	28.0
previousPageUrl	String	Connect REST API URL identifying the previous page, or null if there isn't a previous page.	28.0
totalMemberCount	Integer	Total number of group members across all pages.	28.0

ConnectApi.GroupMembershipRequest

Request to become a group member.

Name	Type	Description	Available Version
createdDate	Datetime	ISO 8601 date string, for example, 2011-02-25T18:24:31.000Z.	28.0
id	String	ID for the group membership request object.	28.0
lastUpdateDate	Datetime	ISO 8601 date string, for example, 2011-02-25T18:24:31.000Z.	28.0
requestedGroup	ConnectApi.Reference	Information about the group the context user is requesting to join.	28.0

Name	Type	Description	Available Version
`responseMessage`	String	A message for the user if their membership request is declined. The value of this property is used only when the value of the `status` property is `Declined`. The maximum length is 756 characters.	28.0
`status`	`ConnectApi.GroupMembershipRequestStatus` Enum	Status of a request to join a private group. Values are: • `Accepted` • `Declined` • `Pending`	28.0
`url`	String	URL of the group membership request object.	28.0
`user`	`ConnectApi.UserSummary`	Information about the user requesting membership in a group.	28.0

SEE ALSO:

ConnectApi.GroupMembershipRequests

ConnectApi.GroupMembershipRequests

Requests to become group members.

Name	Type	Description	Available Version
`requests`	`List<ConnectApi.GroupMembershipRequest>`	Information about group membership requests.	28.0
`total`	Integer	The total number of requests.	28.0

ConnectApi.GroupRecord

A record associated with a group.

Property	Type	Description	Available Version
`id`	String	Record's 18-character ID.	33.0
`record`	`ConnectApi.ActorWithId`	Information about the record associated with the group.	33.0
`url`	String	Record URL.	33.0

SEE ALSO:

ConnectApi.GroupRecordPage

ConnectApi.GroupRecordPage

A paginated list of `ConnectApi.GroupRecord` objects.

Property	Type	Description	Available Version
`currentPageUrl`	String	Connect REST API URL identifying the current page.	33.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	33.0
`previousPageUrl`	String	Connect REST API URL identifying the previous page, or `null` if there isn't a previous page.	33.0
`records`	`List<ConnectApi.GroupRecord>`	List of records on the current page.	33.0
`totalRecordCount`	Integer	Total number of records associated with the group.	33.0

ConnectApi.HashtagSegment

Hashtag segment.

Subclass of ConnectApi.MessageSegment.

Name	Type	Description	Available Version
`tag`	String	Text of the topic without the hash symbol (#).	28.0
`topicUrl`	String	Connect REST API Topics resource that searches for the topic: `/services/data/v63.0/chatter/topics?exactMatch=true&q=topic`	28.0
`url`	String	Connect REST API Feed Items resource URL that searches for the topic in all feed items in an organization: `/services/data/v63.0/chatter/feed-items?q=topic`	28.0

ConnectApi.HideSocialPostIntent

Hide intent for a social post.

Property Name	Type	Description	Available Version
`isHidden`	Boolean	Specifies whether the managed social account hid the social post (`true`) or not (`false`).	45.0

Property Name	Type	Description	Available Version
managedSocialAccount	ConnectApi.ManagedSocialAccount	Managed social account that hides the social post.	45.0

SEE ALSO:

ConnectApi.SocialPostIntents

ConnectApi.HoldFOCapacityOutputRepresentation

Response to a request to hold fulfillment order capacity at one or more locations. Can correspond to one action call.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
holdFOCapacityResponses	List<ConnectApi.HoldFOCapacityResponseOutputRepresentation>	List of responses to the requests to hold fulfillment order capacity at one or more locations.	55.0

ConnectApi.HoldFOCapacityResponseOutputRepresentation

Response to a request to hold fulfillment order capacity at one or more locations.

Property Name	Type	Description	Available Version
capacityResponses	List<ConnectApi.CapacityResponseOutputRepresentation>	List of responses to the requests to hold fulfillment order capacity at individual locations.	55.0

ConnectApi.HttpHeaderOutputRepresentation

HTTP header with information about a text classification

Property Name	Type	Description	Available Version
name	String	Name of the HTTP header.	59.0
value	String	Value of the HTTP header.	59.0

ConnectApi.Icon

Icon.

Property	Type	Description	Available Version
`height`	Integer	The height of the icon in pixels.	28.0
`width`	Integer	The width of the icon in pixels.	28.0
`url`	String	The URL of the icon. This URL is available to unauthenticated users. This URL does not expire.	28.0

SEE ALSO:

- ConnectApi.CanvasCapability
- ConnectApi.EnhancedLinkCapability
- ConnectApi.SocialPostCapability

ConnectApi.InlineImageSegment

An inline image in the feed body.

Subclass of ConnectApi.MessageSegment.

Property Name	Type	Description	Available Version
`altText`	String	Alt text for the inline image.	35.0
`contentSize`	Integer	Size of the file in bytes.	35.0
`fileExtension`	String	Extension of the file, such as gif.	37.0
`thumbnails`	`ConnectApi.FilePreviewCollection`	Information about the available thumbnails for the image.	35.0
`url`	String	URL to the latest version of the inline image.	35.0

ConnectApi.InteractionsCapability

If a feed element has this capability, it has information about user interactions.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
`count`	Long	The number of individual views, likes, and comments on a feed post. Note: This count appears in the UI under the feed post as the number of views, for example, "5 views."	37.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.RelatedQuestion

ConnectApi.Invitation

An invitation.

Property Name	Type	Description	Available Version
`email`	String	Email address of the user.	39.0
`status`	`ConnectApi.GroupViralInvitationsStatus`	Specifies the status of an invitation to join a group. Values are: • `ActedUponUser`—The user was added to the group. An email was sent asking the user to visit the group. • `Invited`—An email was sent asking the user to sign up for the org. • `MaxedOutUsers`—The group has the maximum allowed members. • `MultipleError`—The user wasn't invited due to multiple errors. • `NoActionNeededUser`—The user is already a member of the group. • `NotVisibleToExternalInviter`—The user is not accessible to the user sending the invitation. • `Unhandled`—The user couldn't be added to the group for an unknown reason.	39.0
`userId`	String	ID of the user.	39.0

SEE ALSO:

ConnectApi.Invitations

ConnectApi.Invitations

A collection of invitations.

Property Name	Type	Description	Available Version
`invitations`	List<`ConnectApi.Invitation`>	Collection of invitations.	39.0

ConnectApi.KnowledgeArticleVersion

A knowledge article version.

Property Name	Type	Description	Available Version
`articleType`	String	Type of the knowledge article.	36.0
`id`	String	ID of the knowledge article version.	36.0
`knowledgeArticleId`	String	ID of the corresponding knowledge article.	36.0
`lastPublishedDate`	Datetime	Last published date of the knowledge article.	36.0
`summary`	String	Summary of the knowledge article contents.	36.0
`title`	String	Title of the knowledge article.	36.0
`urlName`	String	URL name of the knowledge article.	36.0

SEE ALSO:

ConnectApi.KnowledgeArticleVersionCollection

ConnectApi.KnowledgeArticleVersionCollection

A collection of knowledge article versions.

Property Name	Type	Description	Available Version
`items`	List<ConnectApi.KnowledgeArticleVersion>	A collection of knowledge article versions.	36.0

ConnectApi.LabeledRecordField

Record field containing a label and a text value.

This class is abstract.

Subclass of ConnectApi.AbstractRecordField.

Superclass of:

- ConnectApi.CompoundRecordField
- ConnectApi.CurrencyRecordField
- ConnectApi.DateRecordField

- ConnectApi.PercentRecordField
- ConnectApi.PicklistRecordField
- ConnectApi.RecordField
- ConnectApi.ReferenceField
- ConnectApi.ReferenceWithDateRecordField

Important: The composition of a feed can change between releases. Write your code to handle instances of unknown subclasses.

Name	Type	Description	Available Version
`label`	String	Localized string describing the record field.	29.0
`text`	String	Text value of the record field. All record fields have a text value. To ensure that all clients can consume new content, inspect the record field's `type` property. If it isn't recognized, render the text value as the default case.	29.0

ConnectApi.LightningExtensionInformation

Lightning extension information.

Subclass of ConnectApi.AbstractExtensionInformation.

Property Name	Type	Description	Available Version
`compositionComponent`	String	Component to use in compose state.	40.0
`headerTextLabel`	String	Label for the extension's header.	40.0
`hoverTextLabel`	String	Label for hovering over the extension.	40.0
`renderComponent`	String	Component to use in render or preview state.	40.0

SEE ALSO:

ConnectApi.ExtensionDefinition

ConnectApi.LikeIntent

Like intent for a social post.

Property Name	Type	Description	Available Version
`isLiked`	Boolean	Specifies whether the managed social account liked the social post (`true`) or not (`false`).	45.0

Property Name	Type	Description	Available Version
managedSocialAccount	ConnectApi.ManagedSocialAccount	Managed social account that likes the social post.	45.0

SEE ALSO:

ConnectApi.LikeIntents

ConnectApi.LikeIntents

List of like intents for a social post.

Property Name	Type	Description	Available Version
likes	List<ConnectApi.LikeIntent>	List of like intents for the social post.	45.0

SEE ALSO:

ConnectApi.SocialPostIntents

ConnectApi.LikeSocialPostIntent

Like intent on a social post.

Property Name	Type	Description	Available Version
socialAccountId	String	ID of the social account that likes the social post.	46.0
socialPostId	String	ID of the social post to like.	46.0

ConnectApi.LikeSummary

Summary of a like.

Subclass of ConnectApi.UserFeedEntityActivitySummary.

Property Name	Type	Description	Available Version
likeId	String	ID of the like.	42.0

ConnectApi.LineItemResponse

Response class that stores information about a list of one or more line items on which the tax engine has calculated tax.

Property Name	Type	Description	Available Version
addresses	ConnectApi.TaxAddressesResponse	The Ship From, Ship To, and Sold To addresses used during tax calculation.	55.0
amountDetails	ConnectApi.TaxAmountDetailsResponse	Information about tax amount values on the line item.	55.0
effectiveDate	Datetime	The date that the tax calculation takes effect.	55.0
lineNumber	String	System-generated number used to identify the tax line.	55.0
productCode	String	Product code for the product related to the taxed line item.	55.0
quantity	Double	Quantity of the taxed line item.	55.0
taxCode	String	Tax code for the taxed line item.	55.0
taxes	List<ConnectApi.TaxDetailsResponse>	Tax details for each line item in a tax line item output.	55.0

ConnectApi.LinkCapability

If a feed element has this capability, it has a link.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
url	String	Link URL. The URL can be to an external site.	32.0
urlName	String	Description of the link.	32.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.LinkMetadata

Metadata for a link.

Property Name	Type	Description	Available Version
description	String	Description of the link.	42.0
frameSource	String	HTML required to display the resource.	42.0
height	Integer	Height required to display the HTML.	42.0

Property Name	Type	Description	Available Version
`originalUrl`	String	Original URL that was used to request the metadata.	42.0
`providerUrl`	String	URL of the provider that the information is retrieved from.	42.0
`source`	`ConnectApi.LinkMetadataSource`	Source of the link metadata. Values are: • `None`—Link metadata wasn't retrieved. • `Sfdc`—Salesforce is the source.	42.0
`thumbnailUrl`	String	Thumbnail of the resource.	42.0
`title`	String	Title of the link.	42.0
`type`	`ConnectApi.LinkMetadataType`	Type of link that the metadata represents. Values are: • `Error`—Link metadata couldn't be retrieved. • `Link`—Represents a link. • `None`—Link metadata wasn't retrieved because the link isn't an allowed domain. • `Photo`—Represents a photo. • `Rich`—Represents rich content, typically HTML content. • `Unknown`—Link metadata was retrieved, but the type is unknown. • `Video`—Represents a video.	42.0
`url`	String	URL of the image to display, if one is available.	42.0
`width`	Integer	Width required to display the HTML.	42.0

SEE ALSO:

ConnectApi.LinkMetadataCollection

ConnectApi.LinkMetadataCollection

Collection of link metadata.

Property Name	Type	Description	Available Version
`linkMetadataList`	List<`ConnectApi.LinkMetadata`>	List of metadata for links.	42.0

ConnectApi.LinkSegment

Link segment.

Subclass of ConnectApi.MessageSegment.

Name	Type	Description	Available Version
`url`	String	The link URL.	28.0

ConnectApi.LocationCapacityOutputRepresentation

Fulfillment order capacity values for a location.

Property Name	Type	Description	Available Version
`assigned`	Integer	Value of the location's Assigned Fulfillment Order Count.	55.0
`capacity`	Integer	Value of the location's Fulfillment Order Capacity. This property represents the location's maximum capacity.	55.0
`error`	`ConnectApi.ErrorResponse`	Error returned by the request, if any.	55.0
`heldCapacity`	Integer	Number of fulfillment orders that the location is holding capacity for.	55.0
`locationId`	String	ID of the location.	55.0

ConnectApi.LocationOutputRepresentation

An inventory location's distance to an order recipient.

Property Name	Type	Description	Available Version
`distance`	Double	The distance from the location to the order recipient.	51.0
`locationIdentifier`	String	The location identifier.	51.0

ConnectApi.MaintenanceInfo

Information about the upcoming scheduled maintenance for the organization.

Property Name	Type	Description	Available Version
`description`	String	Description of the maintenance.	34.0
`maintenanceTitle`	String	Title of the maintenance.	34.0
`maintenanceType`	`ConnectApi.MaintenanceType`	Type of maintenance. Values are: • `Downtime`—Downtime maintenance. • `GenerallyAvailable`—Generally available mode.	34.0

Property Name	Type	Description	Available Version
		• `MaintenanceAndAvailable`—Maintenance with available mode. • `MaintenanceWithDowntime`—Scheduled maintenance with downtime. • `ReadOnly`—Maintenance with read-only mode.	
`messageEffectiveTime`	Datetime	Effective time when users start seeing the maintenance message.	34.0
`messageExpirationTime`	Datetime	Expiration time of the maintenance message.	34.0
`scheduledEndDowntime`	Datetime	Scheduled end of downtime. `null` for `GenerallyAvailable` and `ReadOnly` maintenance types.	34.0
`scheduledEndMaintenanceTime`	Datetime	Scheduled end of maintenance. `null` for `Downtime` maintenance type.	34.0
`scheduledStartDowntime`	Datetime	Scheduled start of downtime. `null` for `GenerallyAvailable` and `ReadOnly` maintenance types.	34.0
`scheduledStartMaintenanceTime`	Datetime	Scheduled start time of maintenance. `null` for `Downtime` maintenance type.	34.0

SEE ALSO:

ConnectApi.OrganizationSettings

ConnectApi.ManagedContentAssociations

Content topics associated with managed content.

Property Name	Type	Description	Available Version
`topics`	List<`ConnectApi.TopicSummary`>	A collection of topics associated with the managed content.	47.0

SEE ALSO:

ConnectApi.ManagedContentVersion

ConnectApi.ManagedContentChannel

Managed content channel.

Subclass of ConnectApi.AbstractManagedContentChannelRepresentation in version 62.0 and later

Property Name	Type	Description	Available Version
`cacheControlMaxAge`	Long	HTTP cache control max age response header for content delivered from the channel.	55.0
`channelId`	String	ID of the managed content channel.	48.0–61.0
`channelName`	String	Name of the managed content channel.	48.0–61.0
`channelType`	`ConnectApi.ManagedContentChannelType`	Type of managed content channel. Values are: • `CloudToCloud`—Cloud-to-Cloud integrated channel. • `Community`—Experience Cloud site channel. • `ConnectedApp`—Channel served by a connected app. • `PublicUnauthenticated`—Public channel. All published content is publicly available. • `UserPermission`—Channel backed by a system permission. All published content is available only to users with the permission.	48.0–61.0
`domain`	String	ID or name of the domain assigned to the channel.	52.0–61.0
`domainId`	String	ID of the domain assigned to the channel. In version 52.0 and later, this information is returned in the `domain` property.	50.0–51.0
`domainName`	String	Name of the domain assigned to the channel.	50.0–61.0
`id`	String	ID of the managed content channel.	62.0
`isChannelSearchable`	Boolean	Specifies whether the text contents of the channel are searchable (`true`) or not (`false`).	48.0–61.0
`isDedicatedContentDelivery`	Boolean	Specifies whether the channel has off-core dedicated content delivery enabled (`true`) or not (`false`). Orgs hosted on Hyperforce use off-core dedicated content delivery to deliver content in public channels with high performance and low latency.	63.0
`isDomainLocked`	Boolean	Specifies whether the domain is locked and can't be changed (`true`) or not (`false`).	50.0–61.0
`isSearchable`	Boolean	Specifies whether the text contents of the channel are searchable (`true`) or not (`false`).	62.0
`managedContentChannelDomain`	`ConnectApi.ManagedContentChannelDomainRepresentation`	Domain associated with the channel.	62.0

Property Name	Type	Description	Available Version
`mediaCacheControlMaxAge`	Long	HTTP cache control max age response header for media delivered from the channel.	57.0
`name`	String	Name of the managed content channel.	62.0
`targetId`	String	ID of the target associated with the channel.	62.0
`type`	`ConnectApi.ManagedContentChannelType`	Type of managed content channel. Values are: • `CloudToCloud`—Cloud-to-Cloud integrated channel. • `Community`—Experience Cloud site channel. • `ConnectedApp`—Channel served by a connected app. • `PublicUnauthenticated`—Public channel. All published content is publicly available. • `UserPermission`—Channel backed by a system permission. All published content is available only to users with the permission.	62.0

SEE ALSO:

ConnectApi.ManagedContentChannelCollection

postManagedContentChannel(ManagedContentCreateInputParam)

getManagedContentChannel(channelId)

patchManagedContentChannel(channelId, ManagedContentChannelInput)

ConnectApi.ManagedContentChannelCollection

Collection of managed content channels.

Property Name	Type	Description	Available Version
`channels`	List<`ConnectApi.ManagedContentChannel`>	List of managed content channels.	48.0–61.0
`currentPageUrl`	String	Connect REST API URL identifying the current page.	48.0–61.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	48.0–61.0
`previousPageUrl`	String	Connect REST API URL identifying the previous page, or `null` if there isn't a previous page.	48.0–61.0
`totalChannels`	Integer	Total number of managed content channels.	48.0–61.0

ConnectApi.ManagedContentChannelDetail

Managed content channel detail.

Property Name	Type	Description	Available Version
`channelId`	String	ID of the managed content channel.	54.0–61.0
`channelName`	String	Name of the managed content channel.	54.0–61.0
`channelType`	`ConnectApi.ManagedContentChannelType`	Type of managed content channel. Values are: • `CloudToCloud`—Cloud-to-Cloud integrated channel. • `Community`—Experience Cloud site channel. • `ConnectedApp`—Channel served by a connected app. • `PublicUnauthenticated`—Public channel. All published content is publicly available. • `UserPermission`—Channel backed by a system permission. All published content is available only to users with the permission.	54.0–61.0
`domain`	String	Domain assigned to the managed content channel.	54.0–61.0
`domainName`	String	Name of the domain assigned to the managed content channel.	54.0–61.0
`isChannelSearchable`	Boolean	Specifies whether the text of the channel's contents is searchable (`true`) or not (`false`).	54.0–61.0
`isDomainLocked`	Boolean	Specifies whether the channel's domain is locked and can't be changed (`true`) or not (`false`).	54.0–61.0

ConnectApi.ManagedContentChannelDomainRepresentation

Domain associated with a managed content channel.

Property Name	Type	Description	Available Version
`isLocked`	Boolean	Specifies whether the domain is locked and can't be changed (`true`) or not (`false`).	62.0
`name`	String	Name of the domain assigned to the channel.	62.0
`value`	String	Value of the domain (name or ID) associated with the channel.	62.0

SEE ALSO:

ConnectApi.ManagedContentChannel

ConnectApi.ManagedContentChannelsRepresentation

Collection of managed content channels.

Property Name	Type	Description	Available Version
`channels`	List<`ConnectApi.AbstractManagedContentChannelRepresentation`>	List of managed content channels.	62.0
`currentPageUrl`	String	Connect REST API URL identifying the current page.	62.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	62.0
`previousPageUrl`	String	Connect REST API URL identifying the previous page, or `null` if there isn't a previous page.	62.0
`totalChannels`	Integer	Total number of managed content channels.	62.0

SEE ALSO:

getManagedContentChannels(pageParam, pageSize, showDetails)

ConnectApi.ManagedContentChannelSummary

Managed content channel.

Subclass of ConnectApi.AbstractManagedContentChannelRepresentation in version 62.0 and later

Property Name	Type	Description	Available Version
`domainUrl`	String	Domain URL of the channel.	55.0–61.0
`id`	String	ID of the managed content channel.	62.0
`name`	String	Name of the managed content channel.	54.0
`resourceUrl`	String	Resource URL to complete information of the channel.	54.0–61.0
`target`	`ConnectApi.ManagedContentChannelTargetSummary`	Target site associated with the channel.	54.0–61.0
`type`	`ConnectApi.ManagedContentChannelType`	Type of managed content channel. Values are: • `CloudToCloud`—Cloud-to-Cloud integrated channel. • `Community`—Experience Cloud site channel. • `ConnectedApp`—Channel served by a connected app.	62.0

Property Name	Type	Description	Available Version
		• `PublicUnauthenticated`—Public channel. All published content is publicly available. • `UserPermission`—Channel backed by a system permission. All published content is available only to users with the permission.	
`url`	String	URL to the channel resource.	62.0

SEE ALSO:

ConnectApi.ManagedContentDeliveryDocument

ConnectApi.ManagedContentCollectionItems

ConnectApi.ManagedContentDeliveryDocumentCollection

ConnectApi.ManagedContentChannelTargetSummary

Target site associated with the channel.

Property Name	Type	Description	Available Version
`id`	String	ID of the site associated with the channel.	54.0

SEE ALSO:

ConnectApi.ManagedContentChannelSummary

ConnectApi.ManagedContentDeliveryChannelSummaryRepresentation

ConnectApi.ManagedContentCloneStatus

Information about managed content's clone status.

Property Name	Type	Description	Available Version
`label`	String	Localized label for the status.	61.0
`status`	`ConnectApi.ManagedContentCloneStatus`	Status of the managed content clone. Values are: • `PartialSuccess` • `Success`	61.0

SEE ALSO:

ConnectApi.ManagedContentDocumentClone

ConnectApi.ManagedContentClonedVariants

Information about clone variants.

Property Name	Type	Description	Available Version
`language`	String	Language of the translated clone variant.	61.0
`managedContentVariantId`	String	ID of the clone variant.	61.0
`resourceURL`	String	Resource URL of the clone variant.	61.0

SEE ALSO:

ConnectApi.ManagedContentDocumentClone

ConnectApi.ManagedContentCollectionItem

Managed content collection item.

Property Name	Type	Description	Available Version
`body`	Map<String, Object>	Map of properties of the collection item with their values.	56.0
`contentType`	`ConnectApi.ManagedContentCollectionItemTypeSummary`	Type of collection item.	56.0
`id`	String	ID of the collection item.	56.0
`name`	String	Name or title for collection item.	56.0

SEE ALSO:

ConnectApi.ManagedContentCollectionItems

ConnectApi.ManagedContentCollectionItems

Managed content collection Items.

Property Name	Type	Description	Available Version
`channelInfo`	`ConnectApi.ManagedContentChannelSummary`	Information about the managed content channel.	56.0–61.0

Property Name	Type	Description	Available Version
channelSummary	ConnectApi.ManagedContentDeliveryChannelSummaryRepresentation	Summary information about the managed content delivery channel.	62.0
collectionKey	String	Unique identifier for the collection.	56.0
collectionType	ConnectApi.ManagedContentTypeSummary	Type of collection.	56.0
currentPageUrl	String	URL to the current page.	63.0
id	String	ID of the collection.	56.0
items	List<ConnectApi.ManagedContentCollectionItem>	List of collection Items.	56.0
language	String	Language locale of the collection.	56.0
nextPageUrl	String	URL to the next page.	63.0
previousPageUrl	String	URL to the previous page.	63.0
publishedDate	Datetime	Most recent publish date of the collection.	56.0
title	String	Title of the collection.	56.0
total	Integer	Total number of items in the current collection detail page.	56.0
urlName	String	URL name of the collection.	56.0

SEE ALSO:

getCollectionItemsForChannel(channelId, collectionKeyOrId, language)

getCollectionItemsForSite(siteId, collectionKeyOrId, language)

ConnectApi.ManagedContentCollectionItemTypeSummary

Summary of a collection item type.

Property Name	Type	Description	Available Version
fullyQualifiedName	String	Fully qualified name of the collection item type.	56.0
name	String	Name of the collection item type.	56.0

SEE ALSO:

ConnectApi.ManagedContentCollectionItem

ConnectApi.ManagedContentDateAndTimeNodeValue

Managed content node of date and time type.

Subclass of ConnectApi.ManagedContentNodeValue.

Property Name	Type	Description	Available Version
`dateTimeValue`	Datetime	UTC date and time value of the managed content node.	48.0
`timeZone`	String	Time zone in which the date and time is authored.	48.0

ConnectApi.ManagedContentDateNodeValue

Managed content node of date type.

Subclass of ConnectApi.ManagedContentNodeValue.

Property Name	Type	Description	Available Version
`value`	Datetime	Date value of the managed content node.	48.0

ConnectApi.ManagedContentDeliveryChannelRepresentation

Managed content delivery channel.

Property Name	Type	Description	Available Version
`domain`	String	Domain assigned to the managed content channel.	62.0
`domainName`	String	Name of the domain assigned to the managed content channel.	62.0
`id`	String	ID of the managed content channel.	62.0
`isDedicated ContentDelivery`	Boolean	Specifies whether the channel has off-core dedicated content delivery enabled (`true`) or not (`false`). Orgs hosted on Hyperforce use off-core dedicated content delivery to deliver content in public channels with high performance and low latency.	63.0
`isDomainLocked`	Boolean	Specifies whether the channel's domain is locked and can't be changed (`true`) or not (`false`).	62.0
`isSearchable`	Boolean	Specifies whether the text of the channel's contents is searchable (`true`) or not (`false`).	62.0
`name`	String	Name of the managed content channel.	62.0

Property Name	Type	Description	Available Version
`type`	`ConnectApi.ManagedContentChannelType`	Type of managed content channel. Values are: • `CloudToCloud`—Cloud-to-Cloud integrated channel. • `Community`—Experience Cloud site channel. • `ConnectedApp`—Channel served by a connected app. • `PublicUnauthenticated`—Public channel. All published content is publicly available. • `UserPermission`—Channel backed by a system permission. All published content is available only to users with the permission.	62.0

SEE ALSO:

getManagedContentDeliveryChannel(channelId)

ConnectApi.ManagedContentDeliveryChannelsRepresentation

Collection of managed content delivery channels.

Property Name	Type	Description	Available Version
`channels`	List<`ConnectApi.ManagedContentDeliveryChannelSummaryRepresentation`>	List of managed content delivery channels.	62.0
`currentPageUrl`	String	Connect REST API URL identifying the current page.	62.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	62.0
`previousPageUrl`	String	Connect REST API URL identifying the previous page, or `null` if there isn't a previous page.	62.0
`totalChannels`	Integer	Total number of managed content delivery channels.	62.0

SEE ALSO:

getAllDeliveryChannels(pageParam, pageSize)

ConnectApi.ManagedContentDeliveryChannelSummaryRepresentation

Summary information of a managed content delivery channel.

Property Name	Type	Description	Available Version
`domainUrl`	String	Domain URL of the channel.	62.0
`id`	String	ID of the managed content delivery channel.	62.0
`name`	String	Name of the managed content channel.	62.0
`resourceUrl`	String	Resource URL to complete information of the channel.	62.0
`target`	`ConnectApi.ManagedContentChannelTargetSummary`	Target site associated with the channel.	62.0
`type`	`ConnectApi.ManagedContentChannelType`	Type of managed content channel. Values are: • `CloudToCloud`—Cloud-to-Cloud integrated channel. • `Community`—Experience Cloud site channel. • `ConnectedApp`—Channel served by a connected app. • `PublicUnauthenticated`—Public channel. All published content is publicly available. • `UserPermission`—Channel backed by a system permission. All published content is available only to users with the permission.	62.0

SEE ALSO:

ConnectApi.ManagedContentDeliveryDocumentCollection

ConnectApi.ManagedContentDeliveryDocument

ConnectApi.ManagedContentCollectionItems

ConnectApi.ManagedContentDeliveryDocument

Managed content in delivery scope.

Subclass of ConnectApi.AbstractManagedContentDeliveryDocument in version 55.0 and later. Properties with an available version of 54.0 only are included in ConnectApi.AbstractManagedContentDeliveryDocument in version 55.0 and later.

Property Name	Type	Description	Available Version
`channelInfo`	`ConnectApi.ManagedContentChannelSummary`	Information about the managed content channel.	54.0–61.0

Property Name	Type	Description	Available Version
channelSummary	ConnectApi.ManagedContentDeliveryChannelSummaryRepresentation	Summary information about the managed content delivery channel.	62.0
contentBody	Map<String, Object>	Map of properties of the managed content with their values.	54.0
contentKey	String	Globally unique identifier (GUID) for the managed content.	54.0 only
contentType	ConnectApi.ManagedContentTypeSummary	Type of managed content.	54.0 only
language	String	Language locale of the managed content.	54.0 only
managedContentId	String	ID of the managed content.	54.0 only
publishedDate	Datetime	Most recent publish date of the managed content.	54.0 only
references	Map<String, ConnectApi.AbstractManagedContentReference>	Map of references with `contentKey` as the key.	54.0
referencesList	List<ConnectApi.AbstractManagedContentReference>	List of references.	54.0
title	String	Title of the managed content.	54.0 only
unauthenticatedUrl	String	Public URL for the managed content.	54.0 only
urlName	String	URL name of the managed content.	54.0 only

SEE ALSO:

ConnectApi.ManagedContentDeliveryDocumentCollection

ConnectApi.ManagedContentDeliveryDocumentCollection

Managed content delivery document collection.

Property Name	Type	Description	Available Version
channelInfo	ConnectApi.ManagedContentChannelSummary	Information about the managed content channel.	55.0–61.0

Property Name	Type	Description	Available Version
channelSummary	ConnectApi.ManagedContentDeliveryChannelSummaryRepresentation	Summary information about the managed content delivery channel.	62.0
contents	List<ConnectApi.AbstractManagedContentDeliveryDocument>	List of managed content delivery documents.	55.0
currentPageUrl	String	URL to the current page of managed content records.	55.0
nextPageUrl	String	URL to the next page of managed content records.	55.0
previousPageUrl	String	URL to the previous page of managed content records.	55.0
references	Map<String, ConnectApi.AbstractManagedContentReference>	Map of references with contentKey as the key.	55.0
referencesList	List<ConnectApi.AbstractManagedContentReference>	List of references.	55.0

ConnectApi.ManagedContentDeliveryDocumentSummary

Managed content delivery document summary.

Subclass of ConnectApi.AbstractManagedContentDeliveryDocument.

No additional properties.

ConnectApi.ManagedContentDocument

Information about a piece of managed content in an authoring space.

Property Name	Type	Description	Available Version
apiName	String	API name of the managed content.	61.0
contentBody	Map<String, Object>	Map of properties of the managed content with their values.	60.0
contentKey	String	Globally unique identifier (GUID) for the managed content.	60.0
contentSpace	ConnectApi.ManagedContentSpaceSummary	Content space of the managed content.	60.0

Property Name	Type	Description	Available Version
contentType	ConnectApi.ManagedContentTypeSummary	Type of managed content.	60.0
createdBy	ConnectApi.ManagedContentUserSummary	User who created the managed content.	60.0
createdDate	Datetime	Date when the managed content was created.	60.0
externalId	String	External ID of the managed content.	60.0
folder	ConnectApi.ManagedContentFolderSummary	Folder of the managed content.	60.0
isPublished	Boolean	Specifies whether the managed content variant is published to a channel (`true`) or not (`false`).	60.0
language	String	Language locale of the managed content.	60.0
lastModifiedBy	ConnectApi.ManagedContentUserSummary	User who last modified the managed content.	60.0
lastModifiedDate	Datetime	Date when the managed content was last modified.	60.0
managedContentId	String	ID of the managed content.	60.0
managedContentVariantId	String	Managed content variant ID.	60.0
managedContentVersionId	String	Managed content version ID.	60.0
status	ConnectApi.ManagedContentVariantStatusOutput	Status of the managed content variant.	60.0
title	String	Title of the managed content.	60.0
urlName	String	URL name of the managed content.	60.0
versionNumber	String	Version number of the managed content.	60.0

ConnectApi.ManagedContentDocumentClone

Managed content document clone.

Property Name	Type	Description	Available Version
apiName	String	API name of the cloned content.	61.0

Property Name	Type	Description	Available Version
cloneStatus	ConnectApi.ManagedContentCloneStatus	Status of the cloned content.	61.0
contentKey	String	Globally unique identifier (GUID) for the cloned content.	61.0
errorMessage	String	Error message if the primary variant failed to clone.	61.0
failedVariants	List<ConnectApi.ManagedContentFailedVariants>	Information about failed cloned variants if cloning was partially successful.	61.0
folder	ConnectApi.ManagedContentFolderSummary	Folder of the cloned content.	61.0
managedContentId	String	ID of the cloned content in the authoring workspace.	61.0
resourceURL	String	Resource URL of the cloned content.	61.0
sourceContentKeyOrId	String	ID or content key of the source managed content in the authoring workspace.	61.0
title	String	Title of cloned content.	61.0
variants	List<ConnectApi.ManagedContentClonedVariants>	List of cloned variants.	61.0

SEE ALSO:

cloneManagedContentDocument(contentKeyOrId, ManagedContentCloneInputParam)

ConnectApi.ManagedContentFailedVariants

Information about failed cloned managed content variants if cloning was partially successful.

Property Name	Type	Description	Available Version
errorMessage	String	Error message for the failure.	61.0
language	String	Language of the failed translated variant.	61.0
sourceManagedContentVariantId	String	ID of the managed content variant that failed to clone.	61.0

SEE ALSO:

ConnectApi.ManagedContentDocumentClone

ConnectApi.ManagedContentFolderSummary

Information about the managed content folder.

Property Name	Type	Description	Available Version
`id`	String	ID of the managed content folder in the authoring space.	60.0
`resourceUrl`	String	Resource URL of the managed content folder.	60.0

SEE ALSO:

- ConnectApi.ManagedContentDocument
- ConnectApi.ManagedContentVariant
- ConnectApi.ManagedContentDocumentClone

ConnectApi.ManagedContentMediaNodeValue

Managed content node of media type.

Subclass of ConnectApi.ManagedContentNodeValue.

Property Name	Type	Description	Available Version
`altText`	String	Alternative text for the managed content node.	47.0
`altUrl`	String	Alternative URL to the managed content node. In version 49.0 and later, this information is returned in the `thumbnailUrl` property.	47.0–48.0
`contentKey`	String	Content key of the managed content.	51.0
`fileName`	String	File name of the managed content node.	48.0
`mediaType`	`ConnectApi.ManagedContentMediaType`	Type of managed content media. Value is `Image`.	47.0
`mimeType`	String	MIME type of the managed content node.	47.0
`resourceUrl`	String	Resource URL to the image. In version 48.0, the resource URL is available if referencing asset files and is `null` if referencing media. In version 49.0 and later, the resource URL is available if referencing asset files or media.	48.0
`thumbnailUrl`	String	URL to the thumbnail of the media.	49.0
`title`	String	Title of the managed content node.	47.0
`unauthenticatedUrl`	String	Unauthenticated URL to the image or `null` if the image isn't visible to external users.	48.0

Property Name	Type	Description	Available Version
`url`	String	URL to the image.	47.0

ConnectApi.ManagedContentMediaSourceNodeValue

Source of managed content media.

Property Name	Type	Description	Available Version
`fileName`	String	File name of the media source.	49.0
`isExternal`	Boolean	Specifies whether the media source is referenced via an external URL (`true`) or uploaded (`false`).	49.0
`mediaType`	`ConnectApi.ManagedContentMediaType`	Type of managed content media. Values are: • `Document` • `Image`	49.0
`mimeType`	String	MIME type of the media source.	49.0
`referenceId`	String	Reference ID of the uploaded media source.	49.0
`resourceUrl`	String	Resource URL of the media source.	49.0
`unauthenticatedUrl`	String	URL to the media source for unauthenticated users, or `null` if the media source isn't available to external users.	49.0
`url`	String	URL to the media source for authenticated users.	49.0

ConnectApi.ManagedContentNodeType

Managed content node type.

Property Name	Type	Description	Available Version
`label`	String	Label of the managed content node type.	47.0
`name`	String	Developer name of the managed content node type.	47.0

Property Name	Type	Description	Available Version
nodeType	ConnectApi.ManagedContentNodeTypeEnum	Type of managed content node. Values are: • Date • DateTime • Media • MediaSource • MultilineText • NameField • RichText • Text • Url	47.0

SEE ALSO:

ConnectApi.ManagedContentType

ConnectApi.ManagedContentNodeValue

Managed content node.

This class is abstract.

Superclass of:

- ConnectApi.ManagedContentDateAndTimeNodeValue
- ConnectApi.ManagedContentDateNodeValue
- ConnectApi.ManagedContentMediaNodeValue
- ConnectApi.ManagedContentMediaSourceNodeValue
- ConnectApi.ManagedContentTextNodeValue

Property Name	Type	Description	Available Version
nodeType	ConnectApi.ManagedContentNodeType	Type of managed content node. Values are: • Date • DateTime • Media • MediaSource • MultilineText • NameField • RichText • Text • Url	47.0

SEE ALSO:

ConnectApi.ManagedContentVersion

ConnectApi.ManagedContentPublishOutput

Information about a Publish action

Property Name	Type	Description	Available Version
deploymentId	String	ID of the managed content deployment.	60.0
description	String	Publish description.	60.0
publishDate	Datetime	Publish date.	60.0

ConnectApi.ManagedContentReference

Managed content reference.

Subclass of ConnectApi.AbstractManagedContentReference.

Property Name	Type	Description	Available Version
contentBody	Map<String, Object>	Map of properties of the managed content reference with their values.	54.0
title	String	Title of the managed content reference.	54.0

SEE ALSO:

ConnectApi.ManagedContentReferenceSummary

ConnectApi.ManagedContentReferenceSummary

Summary of the managed content reference.

Subclass of ConnectApi.AbstractManagedContentReference.

Property Name	Type	Description	Available Version
`title`	String	Title of the managed content reference.	54.0

ConnectApi.ManagedContentSpace

Managed content space.

Property Name	Type	Description	Available Version
`apiName`	String	API name of the managed content space.	61.0
`createdBy`	String	ID of the user who created the managed content space.	55.0
`createdDate`	Datetime	Date when the managed content space was created.	55.0
`defaultLanguage`	String	Default language of the managed content space.	55.0
`description`	String	Description of the managed content space.	55.0
`fullyQualifiedName`	String	Fully qualified name of the managed content space.	63.0
`id`	String	ID of the managed content space.	55.0
`isEnhancedSpace`	Boolean	Specifies whether the space is enhanced (`true`) or not (`false`).	60.0
`lastModifiedBy`	String	ID of the user who last modified the managed content space.	55.0
`lastModifiedDate`	Datetime	Date when the managed content space was last modified.	55.0
`name`	String	Name of the managed content space.	55.0
`rootFolderId`	String	ID of the root folder of the managed content space.	55.0

ConnectApi.ManagedContentSpaceChannelRepresentation

Managed content space channel.

Property Name	Type	Description	Available Version
`channelSummary`	`ConnectApi.ManagedContentChannelSummary`	Information about the managed content space channel.	62.0

Property Name	Type	Description	Available Version
`createdBy`	`ConnectApi.ManagedContentUserSummary`	Information about the user who created the managed content space channel.	62.0
`createdDate`	Datetime	Date when the managed content space channel was created.	62.0
`status`	`ConnectApi.ManagedContentSpaceChannelStatus` on page 2369	Status of the add or remove operation for a channel and managed content space. • `Added`—Channel was added to the managed content space. • `Failed`—Add or remove operation failed. • `Pending`—Add or remove operation is pending. • `Removed`—Channel was removed from the managed content space.	62.0

SEE ALSO:

ConnectApi.ManagedContentSpaceChannelsRepresentation

ConnectApi.ManagedContentSpaceChannelsRepresentation

List of managed content space channels.

Property Name	Type	Description	Available Version
`currentPageUrl`	String	URL to the current page of managed content space channels.	62.0
`nextPageUrl`	String	URL to the next page of Mmanaged content space channels.	62.0
`previousPageUrl`	String	URL to the previous page of managed content space channels.	62.0
`spaceChannels`	List<`ConnectApi.ManagedContentSpaceChannelRepresentation`>	List of managed content space channels.	62.0
`totalSpaceChannels`	Integer	Total count of managed content space channels returned for the request.	62.0

SEE ALSO:

patchManagedContentSpaceChannels(contentSpaceId, spaceChannels)

getManagedContentSpaceChannels(contentSpaceId, pageParam, pageSize)

ConnectApi.ManagedContentSpaceSummary

Information about the managed content space.

Property Name	Type	Description	Available Version
`id`	String	ID of the managed content space.	60.0
`resourceUrl`	String	Resource URL of the managed content space.	60.0

SEE ALSO:

- ConnectApi.ManagedContentDocument
- ConnectApi.ManagedContentVariant

ConnectApi.ManagedContentTextNodeValue

Managed content node of text type.

Subclass of ConnectApi.ManagedContentNodeValue.

Property Name	Type	Description	Available Version
`value`	String	Text value of the managed content node.	47.0

ConnectApi.ManagedContentType

Managed content type.

Property Name	Type	Description	Available Version
`label`	String	Label of the managed content type.	47.0
`name`	String	Developer name of the managed content type.	47.0
`nodeTypes`	Map<String, `ConnectApi.ManagedContentNodeType`>	Map of node types for the managed content type.	47.0

SEE ALSO:

- ConnectApi.ManagedContentVersionCollection

ConnectApi.ManagedContentTypeSummary

Managed content type.

Property Name	Type	Description	Available Version
`fullyQualifiedName`	String	Fully qualified name of the managed content type.	54.0
`name`	String	Reserved for future use.	55.0

SEE ALSO:

- ConnectApi.ManagedContentDeliveryDocument
- ConnectApi.ManagedContentCollectionItems
- ConnectApi.ManagedContentDocument
- ConnectApi.ManagedContentVariant

ConnectApi.ManagedContentUnpublishOutput

Managed content unpublish action.

Property Name	Type	Description	Available Version
`deploymentId`	String	ID of the Managed content deployment.	60.0
`description`	String	Unpublish description.	60.0
`unpublishDate`	Datetime	Unpublish date.	60.0

ConnectApi.ManagedContentUserSummary

Information about the user who created or modified the content.

Property Name	Type	Description	Available Version
`id`	String	ID of the user.	60.0
`name`	String	Reserved for future use.	60.0
`resourceUrl`	String	Resource URL of the user.	60.0

SEE ALSO:

- ConnectApi.ManagedContentDocument
- ConnectApi.ManagedContentVariant

ConnectApi.ManagedContentVariant

Managed content variant.

Property Name	Type	Description	Available Version
`apiName`	String	API name of the managed content variant.	63.0

Property Name	Type	Description	Available Version
contentBody	Map<String, Object>	Map of properties of the managed content with their values.	60.0
contentKey	String	Globally unique identifier (GUID) for the managed content.	60.0
contentSpace	ConnectApi.ManagedContentSpaceSummary	Content space of the managed content.	60.0
contentType	ConnectApi.ManagedContentTypeSummary	Type of managed content.	60.0
createdBy	ConnectApi.ManagedContentUserSummary	User who created the managed content variant.	60.0
createdDate	Datetime	Date when the managed content variant was created.	60.0
externalId	String	External ID of the managed content.	60.0
folder	ConnectApi.ManagedContentFolderSummary	Folder of the managed content.	60.0
isPublished	Boolean	Specifies whether the managed content variant is published to a channel (true) or not (false).	60.0
language	String	Language locale of the managed content.	60.0
lastModifiedBy	ConnectApi.ManagedContentUserSummary	User who last modified the managed content variant.	60.0
lastModifiedDate	Datetime	Date when the managed content variant was last modified.	60.0
managedContentId	String	ID of the managed content.	60.0
managedContentVariantId	String	ID of the managed content variant.	60.0
managedContentVersionId	String	Managed content version ID.	60.0
status	ConnectApi.ManagedContentVariantStatusOutput	Information about a managed content variant's status in the authoring space.	60.0
title	String	Title of the managed content.	60.0
urlName	String	URL name of the managed content.	60.0

ConnectApi.ManagedContentVariantStatusOutput

Information about a managed content variant's status in the authoring space.

Property Name	Type	Description	Available Version
`label`	String	Localized label for the status.	60.0
`status`	`ConnectApi.ManagedContentVariantStatus`	Status of the managed content variant. Values are: • `Draft`—Content isn't published. • `Published`—Content is published and available for use in your live sites. • `Revised`—Content that's published and edited. Publish this content to make the changes available for use in your live sites.	60.0

SEE ALSO:

ConnectApi.ManagedContentDocument

ConnectApi.ManagedContentVariant

ConnectApi.ManagedContentVersion

Managed content version.

Property Name	Type	Description	Available Version
`associations`	`ConnectApi.ManagedContentAssociations`	Content topics associated with the managed content.	47.0
`contentKey`	String	Content key of the managed content.	51.0
`contentNodes`	Map<String, `ConnectApi.ManagedContentNodeValue`>	Map of content nodes.	47.0
`contentUrlName`	String	Content URL name of the managed content version.	48.0
`language`	String	Language of the managed content version.	48.0
`managedContentId`	String	ID of the managed content.	47.0
`publishedDate`	Datetime	Date when the managed content version was last published.	47.0
`title`	String	Title of the managed content version.	47.0
`type`	String	Type of managed content version.	47.0
`typeLabel`	String	Type label of the managed content type.	47.0

Property Name	Type	Description	Available Version
`unauthenticatedUrl`	String	Unauthenticated delivery URL.	50.0

SEE ALSO:

ConnectApi.ManagedContentVersionCollection

ConnectApi.ManagedContentVersionCollection

Collection of managed content versions.

Property Name	Type	Description	Available Version
`currentPageUrl`	String	Connect REST API URL identifying the current page.	47.0
`items`	List<`ConnectApi.ManagedContentVersion`>	List of managed content versions.	47.0
`managedContentTypes`	Map<String, `ConnectApi.ManagedContentType`>	Map of managed content types.	47.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	47.0
`total`	Integer	Total number of managed content versions.	47.0
`totalTypes`	Integer	Total number of managed content types.	47.0

ConnectApi.ManagedSocialAccount

Information describing a managed social account or fan page of a social network.

Subclass of ConnectApi.BaseManagedSocialAccount

No additional properties.

ConnectApi.ManagedSocialAccounts

A list of managed social accounts.

Property Name	Type	Description	Available Version
`managedSocialAccounts`	List<`ConnectApi.ManagedSocialAccount`>	List of managed social accounts.	44.0

ConnectApi.ManagedTopic

Represents a managed topic in an Experience Cloud site.

Property Name	Type	Description	Available Version
children	List<ConnectApi.ManagedTopic>	Children managed topics of the managed topic; null if the depth request parameter isn't specified or is 1.	35.0
id	String	ID of managed topic.	32.0
managedTopicType	ConnectApi.ManagedTopicType	Type of managed topic. • Content—Topics that are associated with native content. • Featured—Topics that are featured, for example, on the Experience Cloud site home page, but don't provide overall navigation. • Navigational—Topics that display in a navigational menu in the Experience Cloud site.	32.0
parent	ConnectApi.Reference	Parent managed topic of the managed topic.	35.0
topic	ConnectApi.Topic	Information about the topic.	32.0
url	String	Connect REST API URL to the managed topic.	32.0

SEE ALSO:

ConnectApi.ManagedTopicCollection

ConnectApi.ManagedTopicCollection

A collection of managed topics.

Property Name	Type	Description	Available Version
currentPageUrl	String	Connect REST API URL identifying the current page.	32.0
managedTopics	List<ConnectApi.ManagedTopic>	List of managed topics.	32.0
nextPageUrl	String	Connect REST API URL identifying the next page, or null if there isn't a next page.	44.0

ConnectApi.MarkupBeginSegment

The beginning of rich text markup.

Subclass of ConnectApi.MessageSegment.

Property Name	Type	Description	Available Version
altText	String	Alternative text for the segment, if available.	45.0
htmlTag	String	The HTML tag for this markup.	35.0

Property Name	Type	Description	Available Version
markupType	ConnectApi.MarkupType	Type of rich text markup. • Bold—Bold tag. • Code—Code tag. • Hyperlink—Hyperlink anchor tag. • Italic—Italic tag. • ListItem—List item tag. • OrderedList—Ordered list tag. • Paragraph—Paragraph tag. • Strikethrough—Strikethrough tag. • Underline—Underline tag. • UnorderedList—Unordered list tag.	35.0
url	String	URL to the segment, if available.	45.0

ConnectApi.MarkupEndSegment

The end of rich text markup.

Subclass of ConnectApi.MessageSegment.

Property Name	Type	Description	Available Version
htmlTag	String	The HTML tag for this markup.	35.0
markupType	ConnectApi.MarkupType	Type of rich text markup. • Bold—Bold tag. • Code—Code tag. • Hyperlink—Hyperlink anchor tag. • Italic—Italic tag. • ListItem—List item tag. • OrderedList—Ordered list tag. • Paragraph—Paragraph tag. • Strikethrough—Strikethrough tag. • Underline—Underline tag. • UnorderedList—Unordered list tag.	35.0

ConnectApi.MatchInfo

Search information related to the search result.

Property Name	Type	Description	Available Version
`isPromoted`	Boolean	Specifies whether search promotion affected the result (`true`) or not (`false`).	63.0
`isSpellCorrected`	Boolean	Specifies whether spell correction affected the result (`true`) or not (`false`).	63.0

SEE ALSO:

ConnectApi.SearchResult

ConnectApi.MCSFolderShare

Target that a managed content space folder is shared with.

Property Name	Type	Description	Available Version
`canUserUnshare`	Boolean	Specifies whether the user has permission to unshare with the target space (`true`) or not (`false`). Content Manager or higher role in the target workspace has permission to unshare.	63.0
`shareStatus`	`ConnectApi.MCSFolderShareStatus`	Status of sharing a managed content space folder. Values are: • `PendingShare` • `PendingUnshare` • `Shared`	63.0
`targetId`	String	ID of the share target.	63.0
`targetLabel`	String	Label of the share target.	63.0

SEE ALSO:

ConnectApi.MCSFolderShareCollection

ConnectApi.MCSFolderShareCollection

Collection of targets that a managed content space folder is shared with.

Property Name	Type	Description	Available Version
`folderId`	String	ID of the managed content space folder.	63.0

Property Name	Type	Description	Available Version
shares	List<ConnectApi.MCSFolderShare>	List of targets that a managed content space folder is shared with.	63.0

SEE ALSO:

patchMCSFolderShares(folderId, mcsFolderShareCollectionUpdateInput)

getMCSFolderShares(folderId)

ConnectApi.MCSFolderShareTarget

Target that a managed content space folder can be shared with.

Property Name	Type	Description	Available Version
id	String	ID of the share target.	63.0
label	String	Label of the share target.	63.0
resourceUrl	String	Resource URL of the share target that provides target details.	63.0

SEE ALSO:

ConnectApi.MCSFolderShareTargetCollection

ConnectApi.MCSFolderShareTargetCollection

Collection of targets that a managed content space folder can be shared with.

Property Name	Type	Description	Available Version
shareTargets	List<ConnectApi.MCSFolderShareTarget>	List of targets that a managed content space folder can be shared with.	63.0

SEE ALSO:

getMCSFolderShareTargets(folderId)

ConnectApi.MediaReference

A media reference.

Property Name	Type	Description	Available Version
mediaUrl	String	URL to stream or download the media.	41.0

Property Name	Type	Description	Available Version
thumbnailUrl	String	If one exists, URL of the media's thumbnail.	41.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.MediaReferenceCapability

ConnectApi.MediaReferenceCapability

If a feed element has this capability, it has one or more media references.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
media	List<ConnectApi.MediaReference>	Collection of media references.	41.0

ConnectApi.MentionCompletion

Information about a record that could be used to @mention a user or group.

Name	Type	Description	Available Version
additionalLabel	String	If one exists, an additional label for the record represented by this completion, for example, "(Customer)" or "(Acme Corporation)".	29.0
description	String	A description of the record represented by this completion.	29.0
name	String	The name of the record represented by this completion. The name is localized, if possible.	29.0
outOfOffice	ConnectApi.OutOfOffice	If the record represented by this completion is a user, an additional out-of-office message, if one exists, for the user.	40.0
photoUrl	String	A URL to the photo or icon of the record represented by this completion.	29.0
recordId	String	The ID of the record represented by this completion.	29.0

Name	Type	Description	Available Version
`userType`	`ConnectApi.UserType` Enum	If the record represented by this completion is a user, this value is the user type associated with that user; otherwise the value is `null`. One of these values: • `ChatterGuest`—User is an external user in a private group. • `ChatterOnly`—User is a Chatter Free customer. • `Guest`—User is unauthenticated. • `Internal`—User is a standard org member. • `Portal`—User is an external user in an Experience Cloud site. • `System`—User is Chatter Expert or a system user. • `Undefined`—User is a user type that is a custom object.	30.0

SEE ALSO:

ConnectApi.MentionCompletionPage

ConnectApi.MentionCompletionPage

Paginated list of Mention Completion response bodies.

Name	Type	Description	Available Version
`currentPageUrl`	String	Connect REST API URL identifying the current page.	29.0
`mentionCompletions`	`List<ConnectApi.MentionCompletion>`	A list of mention completion proposals. Use these proposals to build a feed post body.	29.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	29.0
`previousPageUrl`	String	Connect REST API URL identifying the previous page, or `null` if there isn't a previous page.	29.0

ConnectApi.MentionSegment

Mention segment.

Subclass of ConnectApi.MessageSegment.

Name	Type	Description	Available Version
`accessible`	Boolean	Specifies whether the mentioned user or group can see the post in which they are mentioned (`true`) or not (`false`).	28.0
`name`	String	Name of the mentioned user or group.	28.0

Name	Type	Description	Available Version
`record`	`ConnectApi.ActorWithId`	Information about the mentioned user or group.	29.0
`user`	`ConnectApi.UserSummary`	Information about the mentioned user. **Important:** In versions 29.0 and later, use the `record` property.	28.0 only In versions before 29.0, if the mention is not a user, the mention is in a `ConnectApi.TextSegment` object.

ConnectApi.MentionValidation

Information about whether a proposed mention is valid for the context user.

Name	Type	Description	Available Version
`recordId`	String	The ID of the mentioned record.	29.0
`validationStatus`	`ConnectApi.MentionValidationStatus` Enum	Type of validation error for a proposed mention, if any. • `Disallowed`—The proposed mention is invalid and is rejected because the context user is trying to mention something that is not allowed. For example, a user who is not a member of a private group is trying to mention the private group. • `Inaccessible`—The proposed mention is allowed, but the user or record being mentioned isn't notified. They don't have access to the parent record that's being discussed. • `Ok`—There is no validation error for this proposed mention.	29.0

SEE ALSO:

ConnectApi.MentionValidations

ConnectApi.MentionValidations

Information about whether a set of mentions is valid for the context user.

Name	Type	Description	Available Version
`hasErrors`	Boolean	Indicates whether at least one of the proposed mentions has an error (`true`), or not (`false`). For example, context users can't mention private groups	29.0

Name	Type	Description	Available Version
		they don't belong to. If such a group is included in the list of mention validations, `hasErrors` is `true` and the group has a `validationStatus` of `Disallowed` in its mention validation.	
`mentionValidations`	`List<ConnectApi.MentionValidation>`	List of mention validation information in the same order as the provided record IDs.	29.0

ConnectApi.MessageBody

Message body.

Subclass of ConnectApi.AbstractMessageBody.

No additional properties.

SEE ALSO:

- ConnectApi.ChatterLikesCapability
- ConnectApi.ChatterMessage
- ConnectApi.Comment
- ConnectApi.FeedElement
- ConnectApi.FeedItemSummary

ConnectApi.MessageSegment

Message segment.

This class is abstract.

Superclass of:

- ConnectApi.ComplexSegment
- ConnectApi.EntityLinkSegment
- ConnectApi.FieldChangeSegment
- ConnectApi.FieldChangeNameSegment
- ConnectApi.FieldChangeValueSegment
- ConnectApi.HashtagSegment
- ConnectApi.InlineImageSegment
- ConnectApi.LinkSegment
- ConnectApi.MarkupBeginSegment
- ConnectApi.MarkupEndSegment
- ConnectApi.MentionSegment
- ConnectApi.MoreChangesSegment
- ConnectApi.ResourceLinkSegment
- ConnectApi.TextSegment

Message segments in a feed item are typed as `ConnectApi.MessageSegment`. Feed item capabilities are typed as `ConnectApi.FeedItemCapability`. Record fields are typed as `ConnectApi.AbstractRecordField`. These classes are all abstract and have several concrete subclasses. At runtime you can use `instanceof` to check the concrete types of these objects and then safely proceed with the corresponding downcast. When you downcast, you must have a default case that handles unknown subclasses.

Important: The composition of a feed can change between releases. Write your code to handle instances of unknown subclasses.

Name	Type	Description	Available Version
`text`	String	Text-only rendition of this segment. If a client encounters an unknown message segment type, it can render this value.	28.0
`type`	`ConnectApi.MessageSegmentType` Enum	The message segment type. One of these values: • `EntityLink` • `FieldChange` • `FieldChangeName` • `FieldChangeValue` • `Hashtag` • `InlineImage` • `Link` • `MarkupBegin` • `MarkupEnd` • `Mention` • `MoreChanges` • `ResourceLink` • `Text`	28.0

SEE ALSO:

ConnectApi.AbstractMessageBody

ConnectApi.ModerationCapability

If a feed element has this capability, users in an Experience Cloud site can flag it for moderation.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
`moderationFlags`	`ConnectApi.ModerationFlags`	The moderation flags for this feed element. Moderators can view and take action on flagged items.	31.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.ModerationFlagItemDetail

Flag details on a feed item, comment, or file.

Property Name	Type	Description	Available Version
`createdBy`	String	ID of the user who flagged the item.	40.0
`createdDate`	Datetime	Date when the item was flagged.	40.0
`id`	String	ID of the moderation flag.	40.0
`moderationType`	`ConnectApi.CommunityFlagType`	Type of moderation flag. Values are: • `FlagAsInappropriate`—Flag for inappropriate content. • `FlagAsSpam`—Flag for spam.	40.0
`note`	String	Note from user who flagged the item.	40.0
`visibility`	`ConnectApi.CommunityFlagVisibility`	Visibility behavior of a flag for various user types. Values are: • `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item. • `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.	40.0

SEE ALSO:

ConnectApi.ModerationFlagsCollection

ConnectApi.ModerationFlags

Information about the moderation flags on a feed item, comment, or file.

Name	Type	Description	Available Version
`flagCount`	Integer	Number of moderation flags on this feed item, comment, or file. If the context user is not a moderator, the property is `null`.	29.0
`flagCountByReason`	`Map<ConnectApi.CommunityFlagReasonType, Integer>`	Number of moderation flags categorized by reason. Values for `ConnectApi.CommunityFlagReasonType` are: • `FlaggedByRule`—Moderation rule flagged the item. • `FlaggedBySystem`—Einstein flagged the item. • `FlaggedByUserAsInappropriate`—User flagged the item as inappropriate. • `FlaggedByUserAsSpam`—User flagged the item as spam.	40.0

Name	Type	Description	Available Version
`flaggedByMe`	Boolean	`true` if the context user flagged the feed item, comment, or file for moderation; `false` otherwise.	29.0
`flags`	[illegible]	Collection of flags.	40.0

SEE ALSO:

- ConnectApi.Comment
- ConnectApi.File
- ConnectApi.ModerationCapability

ConnectApi.ModerationFlagsCollection

A collection of flags on a feed item, comment, or file.

Property Name	Type	Description	Available Version
`currentPageToken`	String	Token identifying the current page.	40.0
`currentPageUrl`	String	Connect REST API URL identifying the current page.	40.0
`flags`	List<`ConnectApi.ModerationFlagItemDetail`>	List of flag details.	40.0
`nextPageToken`	String	Token identifying the next page, or `null` if there isn't a next page.	40.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	40.0
`pageSize`	Integer	Number of items per page.	40.0

SEE ALSO:

- ConnectApi.ModerationFlags

ConnectApi.MoreChangesSegment

In feed items with a large number of tracked changes, the message is formatted as: "changed A, B, and made X more changes." The `MoreChangesSegment` contains the "X more changes."

Subclass of ConnectApi.MessageSegment.

Name	Type	Description	Available Version
`moreChanges`	List<`ConnectApi.FieldChangeSegment`>	Complete list of tracked changes.	29.0

Name	Type	Description	Available Version
moreChangesCount	Integer	Number of additional changes.	28.0

ConnectApi.Motif

The motif properties contain URLs for small, medium, and large icons that indicate the Salesforce record type. Common record types are files, users, and groups, but all record types have a set of motif icons. Custom object records use their tab style icon. All icons are available to unauthenticated users so that, for example, you can display the motif icons in an email. The motif can also contain the record type's base color.

The motif images are icons, not user uploaded images or photos. For example, every user has the same set of motif icons.

Custom object records use their tab style icon, for example, the following custom object uses the "boat" tab style:

```
"motif": {
   "color: "8C004C",
   "largeIconUrl": "/img/icon/custom51_100/boat64.png",
   "mediumIconUrl": "/img/icon/custom51_100/boat32.png",
   "smallIconUrl": "/img/icon/custom51_100/boat16.png",
   "svgIconUrl": null
},
```

Users use the following icons:

```
"motif": {
   "color: "1797C0",
   "largeIconUrl": "/img/icon/profile64.png",
   "mediumIconUrl": "/img/icon/profile32.png",
   "smallIconUrl": "/img/icon/profile16.png",
   "svgIconUrl": null
},
```

Groups use the following icons:

```
"motif": {
   "color: "1797C0",
   "largeIconUrl": "/img/icon/groups64.png",
   "mediumIconUrl": "/img/icon/groups32.png",
   "smallIconUrl": "/img/icon/groups16.png",
   "svgIconUrl": null
},
```

Files use the following icons:

```
"motif": {
    "color: "1797C0",
    "largeIconUrl": "/img/content/content64.png",
    "mediumIconUrl": "/img/content/content32.png",
    "smallIconUrl": "/img/icon/files16.png",
    "svgIconUrl": null
},
```

Note: To view the icons in the previous examples, preface the URL with `https://instance_name`. For example, `https://instance_name/img/icon/profile64.png`.

Name	Type	Description	Available Version
`color`	String	A hex value representing the base color of the record type, or `null`.	29.0
`largeIconUrl`	String	A large icon indicating the record type.	28.0
`mediumIconUrl`	String	A medium icon indicating the record type.	28.0
`smallIconUrl`	String	A small icon indicating the record type.	28.0
`svgIconUrl`	String	An icon in SVG format indicating the record type, or `null` if the icon doesn't exist.	34.0

ConnectApi.MultipleAsyncOutputRepresentation

IDs of the asynchronous background operations. This output only includes the operation IDs, regardless of whether calls are made to an external payment gateway. It doesn't include any errors from the operations.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
`asyncOutputs`	List<`ConnectApi.AsyncOutputRepresentation`>	List of IDs of background operations.	56.0

SEE ALSO:

multipleEnsureFundsAsync(multipleEnsureFundsInput)

ConnectApi.MultipleFulfillmentOrderInvoicesOutputRepresentation

IDs of the created Invoices.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
`invoiceIds`	List<String>	List of IDs of the created Invoices.	52.0

ConnectApi.MultipleFulfillmentOrderOutputRepresentation

List of responses for the individual FulfillmentOrder creation attempts from a create multiple fulfillment orders request.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
`fulfillmentOrders`	List<`ConnectApi.FulfillmentGroupOutputRepresentation`>	A list of response data for created and failed FulfillmentOrders.	50.0

ConnectApi.MuteCapability

If a feed element has this capability, users can mute it. Muted feed elements are visible in the muted feed, and invisible in all other feeds that respect mute.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
`isMutedByMe`	Boolean	Indicates whether the context user muted the feed element.	35.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.MuteSummary

Summary of a mute.

Subclass of ConnectApi.UserFeedEntityActivitySummary.

No additional properties.

ConnectApi.NamedCredential

Named credential associated with an external credential.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Property Name	Type	Description	Available Version
`calloutOptions`	`ConnectApi.NamedCredentialCalloutOptions`	Callout options for the named credential.	58.0
`calloutStatus`	`ConnectApi.CalloutStatus`	Indicates whether a named credential is enabled for callout. Values are: • `Disabled` • `Enabled`	59.0
`calloutUrl`	String	URL of the named credential in a callout.	58.0
`createdByNamespace`	String	Namespace of the package that created the named credential.	59.0
`customHeaders`	List<`ConnectApi.CredentialCustomHeader`>	Custom HTTP headers for the named credential.	58.0
`developerName`	String	Fully qualified developer name of the named credential.	56.0

Property Name	Type	Description	Available Version
`externalCredentials`	List<`ConnectApi.ExternalCredential`>	External credentials used by the named credential.	58.0
`id`	String	Named credential ID.	58.0
`masterLabel`	String	Named credential label.	56.0
`networkConnection`	`ConnectApi.NetworkConnection`	PrivateConnect outbound network connection for the named credential.	58.0
`parameters`	List<`ConnectApi.NamedCredentialParameter`>	Named credential parameters.	58.0
`type`	`ConnectApi.NamedCredentialType`	Type of named credential. Values are: • `PrivateEndpoint` • `SecuredEndpoint`	58.0
`url`	String	Connect REST API URL of the named credential.	58.0

SEE ALSO:

ConnectApi.ExternalCredential

ConnectApi.NamedCredentialList

ConnectApi.NamedCredentialCalloutOptions

Named credential callout options.

Property Name	Type	Description	Available Version
`allowMergeFieldsInBody`	Boolean	Specifies whether to allow merge fields in the HTTP body (`true`) or not (`false`).	58.0
`allowMergeFieldsInHeader`	Boolean	Specifies whether to allow merge fields in the HTTP header (`true`) or not (`false`).	58.0
`generateAuthorizationHeader`	Boolean	Specifies whether to generate an authorization header (`true`) or not (`false`).	58.0

SEE ALSO:

ConnectApi.NamedCredential

ConnectApi.NamedCredentialList

List of named credentials.

Property Name	Type	Description	Available Version
`namedCredentials`	List<`ConnectApi.NamedCredential`>	List of named credentials.	58.0

ConnectApi.NamedCredentialParameter

Named credential parameter.

Property Name	Type	Description	Available Version
`id`	String	ID of the parameter.	58.0
`parameterName`	String	Name of the parameter.	58.0
`parameterType`	`ConnectApi.NamedCredentialParameterType`	Type of named credential parameter. Values are: • `AllowedManagedPackageNamespaces` • `ClientCertificate`	58.0
`parameterValue`	String	Value of the parameter.	58.0

SEE ALSO:

ConnectApi.NamedCredential

ConnectApi.NavigationMenuItem

Navigation menu item.

Property Name	Type	Description	Available Version
`actionType`	`ConnectApi.NavigationMenuItemActionType`	Event, URL type, or modal navigation menu item. Values are: • `Event`—Event-based navigation. Note: `Event` is internal only and can't be used in custom components. • `ExternalLink`—URL outside of your Experience Cloud site. • `InternalLink`—Relative URL inside your Experience Cloud site. • `Modal`—Modal, such as Account Switcher.	52.0
`actionValue`	String	For `Event` action type, the event fully qualified name for the navigation menu item. For `ExternalLink` and `InternalLink` action types, the route URL for the navigation menu item.	52.0

Property Name	Type	Description	Available Version
		For `Modal` action type, the component fully qualified name for the navigation menu item.	
`imageUrl`	String	URL to the image of the navigation menu item.	52.0
`label`	String	Label for the navigation menu item.	52.0
`pageReference`	`ConnectApi.NavigationMenuPageReference`	Page reference for the navigation menu item. Page reference is returned only for the Storefront Categories data source.	59.0
`subMenu`	List<`ConnectApi.NavigationMenuItem`>	Submenu for the navigation menu item.	52.0
`target`	`ConnectApi.NavigationMenuItemOpenTarget`	Target for the navigation menu item. Values are: • `CurrentWindow`—Navigation menu item opens in the current window. • `NewWindow`—Navigation menu item opens in a new window.	52.0

SEE ALSO:

ConnectApi.NavigationMenuItemCollection

ConnectApi.NavigationMenuItemCollection

Collection of navigation menu items.

Property Name	Type	Description	Available Version
`menuItems`	List<`ConnectApi.NavigationMenuItem`>	Collection of navigation menu items.	52.0

ConnectApi.NavigationMenuPageReference

Navigation menu item page reference.

Property Name	Type	Description	Available Version
`attributes`	Map<String, String>	Attributes for the navigation menu item page reference.	59.0
`state`	Map<String, String>	State for the navigation menu item page reference.	59.0
`type`	String	Type for the navigation menu item page reference.	59.0

SEE ALSO:

ConnectApi.NavigationMenuItem

ConnectApi.NBAActionParameter

A parameter for an action.

Property Name	Type	Description	Available Version
`name`	String	Name of the parameter.	45.0
`type`	String	Type of the parameter.	45.0
`value`	String	Value of the parameter.	45.0

ConnectApi.NBAFlowAction

A recommended flow.

Subclass of ConnectApi.AbstractNBAAction.

Property Name	Type	Description	Available Version
`flowLabel`	String	Label of the recommended flow.	47.0
`flowType`	`ConnectApi.NBAFlowType`	Type of recommended flow. Values are: • `AutoLaunchedFlow`—Autolaunched flow that runs in the background. • `Flow`—Screen flow that accepts user inputs.	47.0
`id`	String	ID of the flow.	45.0
`name`	String	Name of the flow.	45.0

ConnectApi.NBANativeRecommendation

A record the user is recommended to take action on.

Subclass of ConnectApi.AbstractNBATarget.

Property Name	Type	Description	Available Version
`id`	String	ID of the recommendation.	45.0
`name`	String	Name of the recommendation.	45.0
`url`	String	URL to the recommendation.	45.0

ConnectApi.NBARecommendation

A recommendation returned by a recommendation strategy.

Property Name	Type	Description	Available Version
`aiModel`	String	Reserved for future use.	47.0

Property Name	Type	Description	Available Version
acceptanceLabel	String	Text indicating user acceptance of the recommendation.	45.0
description	String	Description of the recommendation.	45.0
externalId	String	External ID of the recommendation. This ID doesn't need to be a Salesforce 18-character ID. For example, it can be a product number from an external system.	46.0
imageUrl	String	URL to the asset file to display.	45.0
recommendationMode	String	Reserved for future use.	46.0
recommendationScore	Double	Reserved for future use.	46.0
rejectionLabel	String	Text indicating user rejection of the recommendation.	45.0
target	ConnectApi.AbstractNBATarget	Target to act on.	45.0
targetAction	ConnectApi.AbstractNBAAction	Action to recommend.	45.0

SEE ALSO:

ConnectApi.NBARecommendations

ConnectApi.NBARecommendations

Recommendations returned by a recommendation strategy.

Property Name	Type	Description	Available Version
debug	String	Runtime debug information recorded during recommendation strategy execution.	45.0
errors	String	Runtime errors that occurred during recommendation strategy execution.	45.0
executionId	String	ID of the recommendation strategy execution.	45.0
onBehalfOfId	String	ID of the user or entity for which the recommendation strategy was executed.	45.0
recommendations	List<ConnectApi.NBARecommendation>	List of recommendations returned by a recommendation strategy.	45.0
trace	ConnectApi.StrategyTrace	Trace information for the recommendation strategy execution, if requested.	45.0

ConnectApi.NetworkConnection

External network connection.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Property Name	Type	Description	Available Version
`developerName`	String	Name of the network connection.	58.0
`id`	String	ID of the network connection.	58.0
`masterLabel`	String	Label of the network connection.	58.0
`namespacePrefix`	String	Namespace prefix of the network connection.	58.0

SEE ALSO:

ConnectApi.NamedCredential

ConnectApi.NewUserAudienceCriteria

Criteria for the new members type of custom recommendation audience.

Subclass of ConnectApi.AudienceCriteria.

Property Name	Type	Description	Available Version
`maxDaysInCommunity`	Double	The maximum number of days since a user became a site member.	36.0

ConnectApi.OAuthCredentialAuthUrl

OAuth authentication URL for a credential.

Property Name	Type	Description	Available Version
`authenticationUrl`	String	Authentication URL for the user external credential. Authentication URLs have encoded and escaped special characters. Before using the URL, undo the encoded and escaped characters.	56.0
`external Credential`	String	Fully qualified developer name of the external credential.	56.0
`principalName`	String	Name of the external credential named principal.	56.0
`principalType`	`ConnectApi.CredentialPrincipalType`	Type of credential principal. Values are: • `AwsStsPrincipal` • `NamedPrincipal` • `PerUserPrincipal`	56.0

ConnectApi.OauthProviderInfo

OAuth provider information.

Name	Type	Description	Available Version
`authorizationUrl`	String	The URL used for authorization.	37.0
`name`	String	The name of the OAuth service provider.	37.0

SEE ALSO:

ConnectApi.UserOauthInfo

ConnectApi.ObjectMetadata

Search metadata related to the object.

Property Name	Type	Description	Available Version
`dataCategories`	Map<String, `ConnectApi.DataCategoryMetadata`>	Metadata on each data category for the object.	63.0
`fields`	Map<String, `ConnectApi.FieldMetadata`>	Metadata on each field of the object.	63.0
`label`	String	Name of the object.	63.0
`labelPlural`	String	Plural name of the object.	63.0
`objectApiName`	String	API name of the object	63.0
`themeInfo`	`ConnectApi.ThemeInfo`	Theme related to the object.	63.0

SEE ALSO:

ConnectApi.SearchResultGroups

ConnectApi.SearchAnswer

ConnectApi.ScopedSearchResults

ConnectApi.OCIBaseOutputRepresentation

Base Omnichannel Inventory output class.

This class is abstract.

Superclass of:

- ConnectApi.OCIGetInventoryAvailabilityOutputRepresentation
- ConnectApi.OCIPublishLocationStructureOutputRepresentation

- ConnectApi.OCIPublishLocationStructureStatusOutputRepresentation
- ConnectApi.OCIUploadInventoryAvailabilityOutputRepresentation
- ConnectApi.OCIUploadInventoryAvailabilityStatusOutputRepresentation

Property Name	Type	Description	Available Version
errors	List<ConnectApi.ErrorResponse>	Any errors that were returned.	51.0
success	Boolean	Indicates whether the request was successful.	51.0

ConnectApi.OCICreateReservationErrorOutputRepresentation

Error returned from an attempt to create an Omnichannel Inventory reservation.

Property Name	Type	Description	Available Version
errorCode	String	The error code.	51.0
message	String	Details of the error, if available.	51.0

ConnectApi.OCICreateReservationOutputRepresentation

Result of an Omnichannel Inventory reservation creation request.

Property Name	Type	Description	Available Version
details	List<ConnectApi.OCICreateReservationSingleOutputRepresentation>	Details for each product in the reservation.	51.0
errors	List<ConnectApi.OCICreateReservationErrorOutputRepresentation>	Any errors returned by the reservation request.	51.0
expirationTime	String	The time at which the reservation would expire.	51.0
reservationTime	String	The time when the reservation was recorded.	51.0
success	Boolean	Indicates whether the reservation was successfully created.	51.0

ConnectApi.OCICreateReservationSingleOutputRepresentation

Details of an inventory reservation for one product.

Property Name	Type	Description	Available Version
errorCode	String	The error code, if any.	51.0

Property Name	Type	Description	Available Version
locationGroupIdentifier	String	Identifier of the location group where the inventory is reserved.	51.0
locationIdentifier	String	Identifier of the location where the inventory is reserved.	51.0
quantity	Double	The reserved quantity of the product.	51.0
stockKeepingUnit	String	The SKU of the reserved product.	51.0

ConnectApi.OCIFulfillReservationErrorOutputRepresentation

Response to a request to fulfill one inventory reservation.

Property Name	Type	Description	Available Version
details	ConnectApi.OCIFulfillReservationSingleOutputRepresentation	Details of the fulfilled reservation, if successful.	51.0
errorCode	String	Error code, if any.	51.0
message	String	Details of the error, if available.	51.0

ConnectApi.OCIFulfillReservationOutputRepresentation

Response to a request to fulfill one or more inventory reservations.

Property Name	Type	Description	Available Version
errors	List<ConnectApi.OCIFulfillReservationErrorOutputRepresentation>	Responses for the individual reservations in the fulfillment request.	51.0
success	Boolean	Indicates whether the request was successful.	51.0

ConnectApi.OCIFulfillReservationSingleOutputRepresentation

Details of a single fulfilled reservation.

Property Name	Type	Description	Available Version
actionRequestId	String	The UUID that identifies the original fulfill reservation request.	51.0
externalRefId	String	The external reference ID of the location that fulfilled the reservation.	51.0
locationIdentifier	String	The identifier of the location that fulfilled the reservation.	51.0

Property Name	Type	Description	Available Version
`quantity`	Double	The fulfilled quantity.	51.0
`stockKeepingUnit`	String	The SKU of the fulfilled product.	51.0

ConnectApi.OCIFutureInventoryOutputRepresentation

An expected future inventory restock.

Property Name	Type	Description	Available Version
`expectedDate`	Datetime	Date when the future inventory is expected.	51.0
`quantity`	Double	Quantity of the future inventory.	51.0

ConnectApi.OCIGetInventoryAvailabilityOutputRepresentation

Response to a request for inventory availability data.

Subclass of ConnectApi.OCIBaseOutputRepresentation.

Property Name	Type	Description	Available Version
`locationGroups`	List<`ConnectApi.OCILocationGroupAvailabilityOutputRepresentation`>	A list of inventory availability data for individual location groups.	51.0
`locations`	List<`ConnectApi.OCILocationAvailabilityOutputRepresentation`>	A list of inventory availability data for individual locations.	51.0

SEE ALSO:

getInventoryAvailability(inventoryAvailabilityInputRepresentation)

findRoutesWithFewestSplitsUsingOCI(findRoutesWithFewestSplitsUsingOCIInput)

ConnectApi.FindRoutesWithFewestSplitsUsingOCIOutputRepresentation

ConnectApi.OCIInventoryRecordOutputRepresentation

Inventory availability data for a product.

Property Name	Type	Description	Available Version
`availableToFulfill`	Double	The Available To Fulfill quantity.	51.0
`availableToOrder`	Double	The Available To Order quantity.	51.0
`effectiveDate`	Datetime	The effective date of the inventory. Indicates if the SKU exists in the inventory.	51.0

Property Name	Type	Description	Available Version
`exists`	Boolean	Indicates if the SKU exists in the inventory.	62.0
`futures`	List<`ConnectApi.OCIFutureInventoryOutputRepresentation`>	A list of any expected future inventory restocks.	51.0
`onHand`	Double	The On Hand quantity.	51.0
`reserved`	Double	The Reserved quantity.	51.0
`safetyStockCount`	Double	The Safety Stock Count.	51.0
`stockKeepingUnit`	String	The SKU of the product.	51.0

ConnectApi.OCILocationAvailabilityOutputRepresentation

A set of inventory availability data for one inventory location.

Property Name	Type	Description	Available Version
`inventoryRecords`	List<`ConnectApi.OCIInventoryRecordOutputRepresentation`>	A list of availability data for individual products at this location.	51.0
`locationIdentifier`	String	The identifier of the location.	51.0

ConnectApi.OCILocationGroupAvailabilityOutputRepresentation

A set of inventory availability data for one inventory location group.

Property Name	Type	Description	Available Version
`inventoryRecords`	List<`ConnectApi.OCIInventoryRecordOutputRepresentation`>	A list of availability data for individual products. The data combines the quantities for all locations belonging to this location group.	51.0
`locationGroupIdentifier`	String	The identifier of the location group.	51.0

ConnectApi.OCIPublishLocationStructureOutputRepresentation

Response to a publish location structure request.

Subclass of ConnectApi.OCIBaseOutputRepresentation.

Property Name	Type	Description	Available Version
`uploadId`	String	Identifier of the publish job. Use this value to retrieve the status of the job.	51.0

ConnectApi.OCIPublishLocationStructureStatusOutputRepresentation

Detailed status of a publish location structure job.

Subclass of ConnectApi.OCIBaseOutputRepresentation.

Property Name	Type	Description	Available Version
`endTimeUTC`	String	The UTC time when the job finished. (for example: "2020-07-06T22:54:08.012Z")	51.0
`recordsProcessedCount`	Integer	The number of records processed by the job.	51.0
`recordsReadCount`	Integer	The number of records read by the job.	51.0
`recordsSkippedCount`	Integer	The number of records skipped by the job.	51.0
`startTimeUTC`	String	The UTC time when the job started. (for example: "2020-07-06T22:53:06.788Z")	51.0
`status`	String	The status of the job. (e.g., "PENDING," "COMPLETED," etc.).	51.0
`uploadId`	String	Identifier of the job.	51.0
`validationErrors`	List<`String`>	List of any validation errors returned by the job.	51.0
`validationStatus`	String	The validation status of the job.	51.0

ConnectApi.OCIReleaseReservationErrorOutputRepresentation

Response to a request to release one inventory reservation.

Property Name	Type	Description	Available Version
`details`	`ConnectApi.OCIReleaseReservationSingleOutputRepresentation`	Details of the released reservation, if successful.	51.0
`errorCode`	String	Error code, if any.	51.0
`message`	String	Details of the error, if available.	51.0

ConnectApi.OCIReleaseReservationOutputRepresentation

Response to a request to release one or more inventory reservations.

Property Name	Type	Description	Available Version
`errors`	List<`ConnectApi.OCIReleaseReservationErrorOutputRepresentation`>	Responses for the individual reservations in the release request.	51.0
`success`	Boolean	Indicates whether the request was successful.	51.0

ConnectApi.OCIReleaseReservationSingleOutputRepresentation

Details of a single released reservation.

Property Name	Type	Description	Available Version
`actionRequestId`	String	The UUID that identifies the original release reservation request.	51.0
`externalRefId`	String	The external reference ID of the location that released the reservation.	51.0
`locationGroupIdentifier`	String	The identifier of the location group that released the reservation.	51.0
`locationIdentifier`	String	The identifier of the location that released the reservation.	51.0
`quantity`	Double	The released quantity.	51.0
`stockKeepingUnit`	String	The SKU of the released product.	51.0

ConnectApi.OCITransferReservationErrorOutputRepresentation

Response to a request to fulfill one inventory reservation.

Property Name	Type	Description	Available Version
`details`	`ConnectApi.OCITransferReservationSingleOutputRepresentation`	Details of the transferred reservation, if successful.	51.0
`errorCode`	String	Error code, if any.	51.0
`message`	String	Details of the error, if available.	51.0

ConnectApi.OCITransferReservationOutputRepresentation

Response to a request to transfer one or more inventory reservations.

Property Name	Type	Description	Available Version
`errors`	List<`ConnectApi.OCITransferReservationErrorOutputRepresentation`>	Responses for the individual reservations in the transfer request.	51.0
`success`	Boolean	Indicates whether the request was successful.	51.0

ConnectApi.OCITransferReservationSingleOutputRepresentation

Details of a single transferred reservation.

Property Name	Type	Description	Available Version
actionRequestId	String	The UUID that identifies the original transfer reservation request.	51.0
externalRefId	String	The external reference ID of the location that received the reservation.	51.0
fromLocationGroupIdentifier	String	The identifier of the location group that sent the reservation.	51.0
fromLocationIdentifier	String	The identifier of the location that sent the reservation.	51.0
ignoreAvailabilityCheck	Boolean	Whether this call ignored availability data at the location that received the reservation.	52.0
quantity	Double	The quantity of transferred inventory.	51.0
stockKeepingUnit	String	The SKU of the transferred product.	51.0
toLocationGroupIdentifier	String	The identifier of the location group that received the reservation.	51.0
toLocationIdentifier	String	The identifier of the location that received the reservation.	51.0

ConnectApi.OCIUpdateReservationErrorOutputRepresentation

Error output representation for the update inventory reservation.

Property Name	Type	Description	Available Version
errorCode	String	The error code.	61.0
message	String	Details of the error, if available.	61.0

ConnectApi.OCIUpdateReservationOutputRepresentation

Result of an Omnichannel Inventory update request for reserved inventory.

Property Name	Type	Description	Available Version
details	List<ConnectApi.OCIUpdateReservationDetailOutputRepresentation>[]	Details for each product in the reservation.	61.0
errors	List<ConnectApi.OCIUpdateReservationErrorOutputRepresentation>[]	Any errors returned by the reservation update request.	61.0
reservationTime	String	The time when the reservation was updated.	61.0
success	Boolean	Indicates whether the reservation was successfully updated.	61.0

ConnectApi.OCIUpdateReservationSingleOutputRepresentation

Details of an updated reservation for one product.

Property Name	Type	Description	Available Version
`adjustment`	Double	The total reservation adjustment.	61.0
`errorCode`	String	The error code, if any.	61.0
`locationGroupIdentifier`	String	Identifier of the location group where the inventory is reserved.	61.0
`locationIdentifier`	String	Identifier of the location where the inventory is reserved.	61.0
`quantity`	Double	The total reservation quantity of the product.	61.0
`stockKeepingUnit`	String	The SKU of the updated product.	61.0

ConnectApi.OCIUploadInventoryAvailabilityOutputRepresentation

Response to an upload inventory availability job.

Subclass of ConnectApi.OCIBaseOutputRepresentation.

Property Name	Type	Description	Available Version
`uploadId`	String	Identifier of the upload job. Use this value to retrieve the status of the job.	51.0

ConnectApi.OCIUploadInventoryAvailabilityStatusOutputRepresentation

Detailed status of an upload inventory availability job.

Subclass of ConnectApi.OCIBaseOutputRepresentation.

Property Name	Type	Description	Available Version
`endTimeUTC`	String	The UTC time when the job finished. (for example: "2020-07-06T22:54:08.012Z")	51.0
`recordsProcessedCount`	Integer	The number of records processed by the job.	51.0
`recordsReadCount`	Integer	The number of records read by the job.	51.0
`recordsSkippedCount`	Integer	The number of records skipped by the job.	51.0
`startTimeUTC`	String	The UTC time when the job started. (for example: "2020-07-06T22:53:06.788Z")	51.0
`status`	String	The overall status of the inventory availability upload (e.g. "PENDING", "COMPLETED").	51.0
`uploadId`	String	Identifier of the job.	51.0

Property Name	Type	Description	Available Version
validationErrors	List<String>	List of any validation errors returned by the job.	51.0
validationStatus	String	The validation status of the job.	51.0

ConnectApi.OrchestrationInstance

Orchestration instance.

Property Name	Type	Description	Available Version
flowDefinitionDeveloperName	String	Developer name of the flow definition.	54.0
flowDefinitionId	String	ID of the flow definition.	54.0
flowDefinitionName	String	Name of the flow definition.	54.0
id	String	ID of the orchestration instance.	54.0
interviewId	String	ID of the interview to resume.	54.0
stageInstances	List<ConnectApi.OrchestrationStageInstance>	Orchestration stage instances.	54.0
status	ConnectApi.OrchestrationStatus	Status of the orchestration instance. Values are: • Canceled • Completed • Discontinued • Error • InProgress • NotStarted • Suspended	54.0

SEE ALSO:

ConnectApi.OrchestrationInstanceCollection

ConnectApi.OrchestrationInstanceCollection

Collection of orchestration instances.

Property Name	Type	Description	Available Version
instances	List<ConnectApi.OrchestrationInstance>	Collection of orchestration instances.	54.0

ConnectApi.OrchestrationStageInstance

Orchestration stage instance.

Property Name	Type	Description	Available Version
`completionTime`	String	The duration of the stage in seconds.	63.0
`id`	String	ID of the orchestration stage instance.	54.0
`label`	String	Orchestration stage instance label.	54.0
`name`	String	Orchestration stage instance name.	54.0
`status`	`ConnectApi.OrchestrationStatus`	Status of the orchestration instance. Values are: • `Canceled` • `Completed` • `Discontinued` • `Error` • `InProgress` • `NotStarted` • `Suspended`	54.0
`stepInstances`	List<`ConnectApi.OrchestrationStepInstance`>	Orchestration stage instance steps.	54.0

SEE ALSO:

ConnectApi.OrchestrationInstance

ConnectApi.OrchestrationStepInstance

Orchestration step instance.

Property Name	Type	Description	Available Version
`assignedTo`	String	The ID of the user, group, or queue that's assigned to a work item.	63.0
`assigneeType`	String	The assignee type associated with a work item. Valid values are: • Group • Invalid • Queue • User	63.0

Property Name	Type	Description	Available Version
`comments`	String	The string stored in an output variable with the API name of Comments from a flow called by a completed orchestration step.	63.0
`completedBy`	String	The user ID of the user who completed the work item.	63.0
`completionTime`	String	The duration of the step in seconds.	63.0
`description`	String	The description associated with the orchestration step.	63.0
`id`	String	ID of the orchestration step instance.	54.0
`label`	String	Orchestration step instance label.	54.0
`name`	String	Orchestration step instance name.	54.0
`status`	`ConnectApi.OrchestrationStatus`	Status of the orchestration instance. Values are: • `Canceled` • `Completed` • `Discontinued` • `Error` • `InProgress` • `NotStarted` • `Suspended`	54.0
`stepType`	`ConnectApi.OrchestrationStepType`	Type of orchestration step. Values are: • `AsynchronousBackgroundStep` • `ApprovalStep` • `BackgroundStep` • `InteractiveStep` • `ManagedContentRoleInteractiveStep` • `ManagedContentVariantAutoPublishBackgroundStep` • `ManagedContentVariantAutoUnpublishBackgroundStep` • `ManagedContentVariantSetLockBackgroundStep` • `ManagedContentVariantSetReadyBackgroundStep` • `MuleSoftStep`	54.0

Property Name	Type	Description	Available Version
workItems	List<ConnectApi.OrchestrationWorkItem>	Orchestration step instance work items.	54.0

SEE ALSO:

ConnectApi.OrchestrationStageInstance

ConnectApi.OrchestrationWorkItem

Orchestration work item.

Property Name	Type	Description	Available Version
assigneeId	String	ID of the assignee for the orchestration work item.	54.0
createdDate	Datetime	Date when the orchestration work item was created.	61.0
description	String	Description of the orchestration work item.	54.0
flowType	String	Flow type of the orchestration that created the orchestration work item.	62.0
id	String	ID of the orchestration work item.	54.0
label	String	Label key for the orchestration work item.	54.0
lastModifiedDate	Datetime	Date when the work item was last modified.	61.0
relatedRecordId	String	ID of the record the orchestration work item is related to.	54.0
screenFlowDeveloperName	String	Developer name of the screen flow to start when assignees work on the orchestration work item.	54.0
screenFlowId	String	ID of the screen flow to start when assignees work on the orchestration work item.	54.0
screenFlowInputs	String	Input parameters for the screen flow.	54.0
status	ConnectApi.OrchestrationWorkItemStatus	Status of the orchestration work item. Values are: • Assigned • Completed	54.0

SEE ALSO:

ConnectApi.OrchestrationStepInstance

ConnectApi.OrderDeliveryGroupSummary

Order delivery group summary.

Property Name	Type	Description	Available Version
fields	Map<String, ConnectApi.RecordField>	Map of fields from order delivery group summary and other related objects that were queried.	51.0

SEE ALSO:

ConnectApi.OrderDeliveryGroupSummaryCollection

ConnectApi.OrderDeliveryGroupSummaryCollection

Collection of order delivery group summaries.

Property Name	Type	Description	Available Version
currentPageToken	String	Token identifying the current page of order delivery group summaries.	51.0
currentPageUrl	String	URL to the current page of order delivery group summaries.	51.0
nextPageToken	String	Token identifying the next page of order delivery group summaries.	51.0
nextPageUrl	String	URL to the next page of order delivery group summaries.	51.0
orderDeliveryGroups	List<ConnectApi.OrderDeliveryGroupSummary>	Collection of order delivery group summaries.	51.0
previousPageToken	String	Token identifying previous page of order delivery group summaries.	51.0
previousPageUrl	String	URL to the previous page of order delivery group summaries.	51.0

ConnectApi.OrderDeliveryGroupSummaryLookupOutput

Order delivery group summary lookup output.

Property Name	Type	Description	Available Version
currencyIsoCode	String	Three-letter ISO 4217 currency code associated with the order delivery group summary record.	58.0
deliveryMethod	ConnectApi.OrderDeliveryMethodLookupOutput	Delivery method associated with order the delivery group summary.	58.0
fields	Map<String, ConnectApi.RecordFieldCollection on page 2270>	Map of requested order delivery group summary fields.	58.0

Property Name	Type	Description	Available Version
id	String	ID of the order delivery group summary.	58.0
lineItems	List<ConnectApi.OrderSummaryLookupOutput>	Line items associated with the order delivery group summary.	58.0

ConnectApi.OrderDeliveryMethodLookupOutput

Order delivery method lookup output.

Property Name	Type	Description	Available Version
fields	Map<String, ConnectApi.RecordField>	Map of requested order delivery method fields.	58.0
id	String	ID of the order delivery method.	58.0

ConnectApi.OrderItemSummary

Order item summary.

Property Name	Type	Description	Available Version
adjustmentAggregates	ConnectApi.OrderItemSummaryAdjustmentAggregates	Adjustment aggregates associated with an order item summary.	55.0
fields	Map<String, ConnectApi.RecordField>	Map of fields from order item summary and other related objects that were queried.	51.0
orderItemSummaryId	String	ID of the order item summary.	51.0
orderSummaryId	String	ID of the order summary.	51.0
product	ConnectApi.OrderItemSummaryProduct	Associated product item information.	51.0

SEE ALSO:

ConnectApi.OrderItemSummaryCollection

ConnectApi.OrderItemSummaryAdjustmentAggregates

Adjustment aggregates associated with an order item summary.

Property Name	Type	Description	Available Version
`available`	Boolean	Indicates whether adjustment aggregates are available (`true`) or not (`false`).	55.0
`status`	`ConnectApi.OrderSummaryAdjustmentAggregatesStatus`	Order summary adjustment aggregate job status. • `Failed`—The adjustment aggregate data job for the order summary failed. • `InProgress`—The adjustment aggregate data job for the order summary is in progress. • `NotInitiated`—The adjustment aggregate data job for the order summary is not initiated. • `Submitted`—The adjustment aggregate data job for the order summary is submitted.	55.0
`totalLinePromotionAmount`	String	Total of all line item promotions applied to this specific product.	55.0
`totalPromotionDistAmount`	String	Total of all order level promotions applied to this specific product.	55.0

ConnectApi.OrderItemSummaryAdjustmentCollection

Collection of adjustments for order item summaries.

Property Name	Type	Description	Available Version
`orderItemSummaries`	Map<String, `ConnectApi.OrderItemSummaryAdjustmentList`>	Order item summaries and their associated adjustments.	53.0

ConnectApi.OrderItemSummaryAdjustmentList

Representation for list of adjustments for an Order Item Summary.

Property Name	Type	Description	Available Version
`adjustments`	List<`ConnectApi.OrderSummaryAdjustment`>	Adjustments associated with an order item summary.	53.0

SEE ALSO:

ConnectApi.OrderItemSummaryAdjustmentCollection

ConnectApi.OrderItemSummaryCollection

Collection of order item summaries.

Property Name	Type	Description	Available Version
currentPageToken	String	Token identifying the current page of items.	51.0
currentPageUrl	String	URL to the current page of items.	51.0
items	List<ConnectApi.OrderItemSummary>	Collection of order item summaries.	51.0
nextPageToken	String	Token identifying the next page of items.	51.0
nextPageUrl	String	URL to the next page of items.	51.0
previousPageToken	String	Token identifying the previous page of items.	51.0
previousPageUrl	String	URL to the previous page of items.	51.0

ConnectApi.OrderItemSummaryLookupOutput

Order item summary lookup output.

Property Name	Type	Description	Available Version
adjustmentAggregates	[illegible]	Adjustment aggregates for the order item summary.	58.0
adjustments	[illegible]	Adjustments associated with the order item summary.	58.0
currencyIsoCode	String	Three-letter ISO 4217 currency code associated with the order item summary record.	58.0
fields	Map<String, ConnectApi.RecordField>	Map of requested order item summary fields.	58.0
id	String	ID of the order item summary.	58.0
product	[illegible]	Details of the product associated with order item summary.	58.0

ConnectApi.OrderItemSummaryOutputRepresentation

Details of an OrderItemSummary from a failed FulfillmentOrder in a create multiple fulfillment orders request.

Property Name	Type	Description	Available Version
errors	List<ConnectApi.ErrorResponse>	List of errors specific to the OrderItemSummary, if any.	50.0
orderItemSummaryId	String	ID of the OrderItemSummary.	50.0
quantity	Double	Quantity of the OrderItemSummary.	50.0

ConnectApi.OrderItemSummaryProduct

Product item mapped to the order item summary.

Property Name	Type	Description	Available Version
`canViewProduct`	Boolean	Specifies whether the context user can view the product (`true`) or not (`false`).	51.0
`errorCode`	String	Error code for the product with errors.	51.0
`errorMessage`	String	Error message for the product with errors.	51.0
`fields`	Map<String, `ConnectApi.RecordField`>	Map of the product fields queried.	51.0
`media`	`ConnectApi.ProductMedia`	Associated product media.	51.0
`productAttributes`	`ConnectApi.ProductAttributeSetSummary`	Summary of the product attributes.	51.0
`productId`	String	ID of the product.	51.0

SEE ALSO:

ConnectApi.OrderItemSummary

ConnectApi.OrderShipment

Order shipment.

Property Name	Type	Description	Available Version
`expectedDeliveryDate`	Datetime	Expected delivery date for the shipment.	52.0
`fields`	Map<String, `ConnectApi.RecordField`>	Map of requested fields.	52.0
`orderSummaryId`	String	ID of the order summary.	52.0
`shipmentId`	String	ID of the shipment.	52.0
`shipmentNumber`	String	Number of the shipment.	52.0
`status`	String	Status of the shipment.	52.0

SEE ALSO:

ConnectApi.OrderShipmentCollection

ConnectApi.OrderShipmentCollection

Collection of order shipments.

Property Name	Type	Description	Available Version
`count`	Integer	Total number of records returned in the collection.	52.0
`currentPageToken`	String	Token identifying the current page of order shipments.	52.0
`currentPageUrl`	String	URL to the current page of order shipments.	52.0
`nextPageToken`	String	Token identifying the next page of order shipments.	52.0
`nextPageUrl`	String	URL to the next page of order shipments.	52.0
`previousPageToken`	String	Token identifying the previous page of order shipments.	52.0
`previousPageUrl`	String	URL to the previous page of order shipments.	52.0
`shipments`	List<`ConnectApi.OrderShipment`>	Collection of order shipments.	52.0
`sortOrder`	`ConnectApi.OrderShipmentSort`	Sort order for order shipments. Values are: • `ExpectedDeliveryDateAsc`—Sorts by the oldest expected delivery date. • `ExpectedDeliveryDateDesc`—Sorts by the most recent expected delivery date. • `ShipmentNumberAsc`—Sorts by shipment number in ascending order (0–9). • `ShipmentNumberDesc`—Sorts by shipment number in descending order (9–0).	52.0

ConnectApi.OrderShipmentItem

Shipment item.

Property Name	Type	Description	Available Version
`fields`	Map<String, `ConnectApi.RecordField`>	Map of requested fields.	52.0
`orderItemSummaryId`	String	ID of the order item summary.	52.0
`product`	`ConnectApi.OrderItemSummaryProduct`	Product mapped to an order item summary.	52.0
`productId`	String	ID of the product.	52.0

Property Name	Type	Description	Available Version
quantity	Double	Quantity of the product.	52.0
shipmentId	String	ID of the shipment.	52.0
shipmentItemId	String	ID of the shipment item.	52.0

ConnectApi.OrderShipmentItemCollection

Collection of order shipment items.

Property Name	Type	Description	Available Version
count	Integer	Total number of records returned in a page.	52.0
currentPageToken	String	Token identifying the current page of order shipment items.	52.0
currentPageUrl	String	URL to the current page of order shipment items.	52.0
items	List<ConnectApi.OrderShipmentItem>	Collection of order shipment items.	52.0
nextPageToken	String	Token identifying the next page of order shipment items.	52.0
nextPageUrl	String	URL to the next page of order shipment items.	52.0
previousPageToken	String	Token identifying the previous page of order shipment items.	52.0
previousPageUrl	String	URL to the previous page of order shipment items.	52.0
sortOrder	ConnectApi.OrderShipmentItemSort	Sort order for order shipment items. Values are: • IdAsc—Sorts by ID in ascending alphanumeric order (A–Z, 0–9). • IdDesc—Sorts by ID in descending alphanumeric order (Z–A, 9–0).	52.0

ConnectApi.OrderSummaryAdjustment

Adjustment associated with an order summary.

Property Name	Type	Description	Available Version
amount	String	Amount associated with the adjustment.	53.0
basisReferenceDisplayName	String	Display name for secondary cause of the adjustment (for example, Null or the CouponCode that's associated with a Coupon)	54.0

Property Name	Type	Description	Available Version
currencyIsoCode	String	Three-letter ISO 4217 currency code associated with the adjustment.	53.0
displayName	String	Display name for the primary cause of the adjustment (for example, Display name of the Promotion)	53.0
targetType	ConnectApi.OrderSummaryAdjustmentTargetType	Type of price adjustment in promotions. Values are: • SplitLine—Price adjustment on an order item. • Header—Price adjustment on the entire order.	56.0
type	String	Type of adjustment (for example, Promotion, Other).	53.0

SEE ALSO:

ConnectApi.OrderSummaryAdjustmentCollection

ConnectApi.OrderItemSummaryAdjustmentList

ConnectApi.OrderSummaryAdjustmentAggregates

Adjustment aggregates associated with an order summary.

Property Name	Type	Description	Available Version
available	Boolean	Indicates if adjustment aggregate values are available (true) or not (false).	55.0
status	ConnectApi.OrderSummaryAdjustmentAggregatesStatus	Order summary adjustment aggregate job status. Values are: • Failed—The adjustment aggregate data job for the order summary failed. • InProgress—The adjustment aggregate data job for the order summary is in progress. • NotInitiated—The adjustment aggregate data job for the order summary is not initiated. • Submitted—The adjustment aggregate data job for the order summary is submitted.	55.0
totalDeliveryPromotionDistAmount	String	Total distributed delivery promotion amounts associated with an order summary.	55.0
totalDeliveryPromotionLineAmount	String	Total delivery promotion line amounts associated with an order summary.	55.0
totalDeliveryPromotionTotalAmount	String	Total delivery promotion amount associated with an order summary.	55.0

Property Name	Type	Description	Available Version
`totalProductPromotionDistAmount`	String	Total distributed product promotion amounts associated with an order summary.	55.0
`totalProductPromotionLineAmount`	String	Total product promotion line amount associated with an order summary.	55.0
`totalProductPromotionTotalAmount`	String	Total product promotion amount associated with an order summary.	55.0

ConnectApi.OrderSummaryAdjustmentAggregatesAsyncOutput

Async adjustment aggregates output.

Property Name	Type	Description	Available Version
`statusURL`	String	Status URL.	55.0

ConnectApi.OrderSummaryAdjustmentCollection

Collection of adjustments for an order summary.

Property Name	Type	Description	Available Version
`adjustments`	List<`ConnectApi.OrderSummaryAdjustment`>	Collection of adjustments for an order summary.	53.0

ConnectApi.OrderSummaryCollectionRepresentation

Collection of order summaries.

Property Name	Type	Description	Available Version
`count`	Integer	Total count of order summaries returned on the current page.	51.0
`currentPageToken`	String	Token identifying the current page.	51.0
`currentPageUrl`	String	Connect REST API URL identifying the current page.	51.0
`nextPageToken`	String	Token identifying the next page, or `null` if there isn't a next page.	51.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	51.0

Property Name	Type	Description	Available Version
orderSummaries	List<ConnectApi.OrderSummaryRepresentation>	Collection of order summaries.	51.0
previousPageToken	String	Token identifying the previous page, or null if there isn't a previous page.	51.0
previousPageUrl	String	Connect REST API URL identifying the previous page, or null if there isn't a previous page.	51.0
sortOrder	ConnectApi.OrderSummarySortOrder	Sort order for order summaries. Values are: • CreatedDateAsc—Sorts by the oldest created date. • CreatedDateDesc—Sorts by the most recent created date. • OrderedDateAsc—Sorts by the oldest ordered date. • OrderedDateDesc—Sorts by the most recent ordered date.	51.0

ConnectApi.OrderSummaryLookupOutput

Order summary lookup output.

Property Name	Type	Description	Available Version
adjustmentAggregates	ConnectApi.OrderSummaryAdjustmentAggregates	Adjustment aggregates associated with the order summary.	58.0
currencyIsoCode	String	Three-letter ISO 4217 currency code associated with the order summary.	58.0
deliveryGroups	List<ConnectApi.OrderDeliveryGroupSummaryLookupOutput>	Delivery groups associated with the order summary.	58.0
fields	Map<String, ConnectApi.RecordField>	Map of requested order summary fields.	58.0
id	String	ID of the order summary.	58.0
orderNumber	String	Reference number of the order summary.	58.0
status	String	Status associated with the order summary.	58.0

ConnectApi.OrderSummaryOutputRepresentation

ID of the created Order Summary.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
orderSummaryId	String	ID of the Order Summary.	48.0

ConnectApi.OrderSummaryProductLookupOutput

Order summary product lookup output.

Property Name	Type	Description	Available Version
canViewProduct	Boolean	Specifies whether the context user can view the product (true) or not (false).	58.0
errorCode	String	Error code captured during product load.	58.0
errorMessage	String	Error message captured during product load.	58.0
fields	Map<String, ConnectApi.RecordField>	Map of requested product fields.	58.0
id	String	Id of the product	58.0
media	ConnectApi.ProductMedia	Associated product media.	58.0
variationAttributes	Map<String, ConnectApi.OrderSummaryProductAttribute>	Variation attributes (color, size, and so on) associated with the product.	58.0

ConnectApi.OrderSummaryRepresentation

Order summary.

Property Name	Type	Description	Available Version
adjustmentAggregates	ConnectApi.OrderSummaryAdjustmentAggregates	Adjustment aggregates associated with the order summary.	55.0
createdDate	Datetime	Created date of the order summary.	51.0
fields	Map< String, ConnectApi.RecordField >	Map of requested order summary fields.	51.0
orderNumber	String	Order number of the order summary.	51.0
orderSummaryId	String	ID of the order summary.	51.0
orderedDate	Datetime	Ordered date of the order summary.	51.0
ownerId	String	ID of the owner of the order summary.	51.0
status	String	Status of the order summary.	51.0

Property Name	Type	Description	Available Version
totalAmount	String	Total amount of the order summary.	51.0

SEE ALSO:

ConnectApi.OrderSummaryCollectionRepresentation

ConnectApi.OrderSummaryProductAttribute

Order summary product attribute representation.

Property Name	Type	Description	Available Version
label	String	Label or display name of the attribute.	58.0
sequence	Integer	Sequence of the attribute set with regard to the product.	58.0
value	String	Display value of the attribute.	58.0

ConnectApi.OrderToCartFailedProduct

Product that could not be added to the cart from an order, with error information.

Property Name	Type	Description	Available Version
errorCode	String	Error code.	57.0
errorMessage	String	Error message about the cause of the failure.	57.0
productId	String	ID of the product.	57.0
productName	String	Name of the product.	57.0
productSKU	String	SKU of the product.	57.0

ConnectApi.OrderToCartResult

Result of action adding an order to a cart.

Property Name	Type	Description	Available Version
cartId	String	ID of the cart.	57.0
totalFailedProductCount	Integer	Number of products that could not be successfully added to the cart from the order.	57.0
totalSucceededProductCount	Integer	Number of products successfully added to the cart from the order.	57.0
unaddedProducts	List<ConnectApi.OrderToCartFailedProduct>	List of products not successfully added to the cart.	57.0

ConnectApi.OrganizationSettings

Org settings.

Name	Type	Description	Available Version
`accessTimeout`	Integer	Amount of time after which the system prompts users who have been inactive to log out or continue working.	28.0
`features`	`ConnectApi.Features`	Information about features available in the org.	28.0
`maintenanceInfo`	`List<ConnectApi.MaintenanceInfo>`	Information about a list of upcoming scheduled maintenances for the org.	34.0
`name`	String	Org name.	28.0
`orgId`	String	18-character ID for the org.	28.0
`userSettings`	`ConnectApi.UserSettings`	Information about the org permissions for the user.	28.0

ConnectApi.OriginCapability

If a feed element has this capability, it was created by a feed action.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
`actor`	ConnectApi.UserSummary	The user who executed the feed action.	33.0
`originRecord`	ConnectApi.Reference	A reference to the feed element containing the feed action.	33.0

SEE ALSO:

- ConnectApi.FeedElementCapabilities

ConnectApi.OutOfOffice

User's out-of-office message.

Property Name	Type	Description	Available Version
`message`	String	Out-of-office message for the user.	40.0

SEE ALSO:

- ConnectApi.User
- ConnectApi.MentionCompletion

ConnectApi.PageInfo

Page position information for the object search.

Property Name	Type	Description	Available Version
`hasNextPage`	Boolean	Specifies whether the search has more results to return (`true`) or not (`false`).	63.0
`offset`	Integer	Search page offset position.	63.0
`pageSize`	Integer	Number of results per page.	63.0

SEE ALSO:

ConnectApi.SearchObject

ConnectApi.PardotBusinessUnitContextItem

Pardot business unit context item.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Property Name	Type	Description	Available Version
`id`	String	ID of the PardotTenant record.	55.0
`isCurrent`	Boolean	Specifies whether the business unit is selected as the context user's current business unit context in the business unit switcher of the Pardot Lightning app (`true`) or not (`false`).	55.0
`name`	String	Name of the Pardot business unit as it is specified in the `MasterLabel` of the PardotTenant record.	55.0

SEE ALSO:

ConnectApi.PardotBusinessUnitContextOutput

ConnectApi.PardotBusinessUnitContextOutput

Pardot business unit context.

Property Name	Type	Description	Available Version
`businessUnits`	List<`ConnectApi.PardotBusinessUnitContextItem`>	List of the Pardot business unit context items that the context user has access to.	55.0
`isSuccess`	Boolean	Indicates whether the requested resource was successfully provided.	55.0

Property Name	Type	Description	Available Version
`totalBusinessUnits`	Integer	Indicates the total number of Pardot business units that the context user has access to.	55.0

ConnectApi.PaymentAuthAdjustmentResponse

Authorization Adjustment output representation.

Property Name	Type	Description	Available Version
`accountId`	String	ID of the account containing the payment authorization being adjusted.	51.0
`amount`	Double	Amount of the payment authorization adjustment.	51.0
`currencyIsoCode`	String	Three-letter ISO 4217 currency code associated with the payment authorization adjustment.	51.0
`effectiveDate`	Datetime	Date when the authorization adjustment becomes effective.	51.0
`id`	String	ID of the PaymentAuthAdjustment record.	51.0
`paymentAuthAdjustmentNumber`	String	System-defined reference number.	51.0
`requestDate`	Datetime	Date when the authorization adjustment transaction occurred.	51.0
`status`	String	Status of the payment authorization adjustment.Possible values are: • `Canceled`: The payment authorization reversal has been canceled. The parent authorization has returned to its pre-reversal balance. • `Draft`: The payment authorization reversal can be edited before applying it against the parent authorization. • `Processed`: The payment authorization reversal has been finalized. Users can change the status as follows: • Draft to Processed • Processed to Canceled • Draft to Canceled	51.0

ConnectApi.PaymentAuthorizationResponse

Payment authorization output representation.

Property Name	Type	Description	Available Version
accountId	String	Salesforce account for the payment authorization.	51.0
amount	Double	Amount that the gateway authorized for the payment transaction.	51.0
currencyIsoCode	String	Three-letter ISO 4217 currency code associated with the payment group record.	51.0
effectiveDate	Datetime	Date that the authorization becomes effective.	51.0
expirationDate	Datetime	Date that the authorization expires.	51.0
id	String	ID of the payment authorization record.	51.0
paymentAuthorizationNumber	String	System-defined number for the payment authorization record.	51.0
requestDate	Datetime	Date that the authorization occurred.	51.0
status	String	Status of the payment authorization as returned by the gateway.	51.0

ConnectApi.PaymentGroupResponse

Payment group.

Property Name	Type	Description	Available Version
currencyIsoCode	String	Three-letter ISO 4217 currency code associated with the payment group record.	50.0
id	String	ID of the payment group record.	50.0
sourceObjectId	String	Source object ID of the payment group record. Supports only OrderId.	50.0

ConnectApi.PaymentMethodDetails

Details about the payment method.

Property Name	Type	Description	Available Version
alternativePaymentMethod	ConnectApi.AlternativePaymentMethodOutput	Alternative Payment Method details.	56.0
cardPaymentMethod	ConnectApi.CardPaymentMethodOutput	Card Payment Method details.	56.0

ConnectApi.PaymentMethodResponse

Payment method information response.

Property Name	Type	Description	Available Version
`accountId`	String	Salesforce Payments account to which this payment method is linked.	51.0
`id`	String	ID of the payment method.	51.0
`paymentMethodDetails`	ConnectApi.PaymentMethodDetails	Details about the payment method.	
`status`	String	Status of the payment method.	51.0

ConnectApi.PaymentMethodTokenizationGatewayResponse

Payment method tokenization gateway response representation.

Subclass of ConnectApi.AbstractGatewayResponse.

Property Name	Type	Description	Available Version
`gatewayToken`	String	The payment method token sent from the gateway.	52.0

ConnectApi.PaymentMethodTokenizationResponse

Payment method tokenization output representation.

Property Name	Type	Description	Available Version
`error`	`ConnectApi.ErrorResponse`	Error representation for the payment method tokenization process. Sent only if the tokenization process encounters an error in the gateway.	52.0
`gatewayResponse`	`ConnectApi.PaymentMethodTokenizationGatewayResponse`	Response containing the tokenized payment method value from the payment gateway.	52.0
`paymentGatewayLogs`	List<`ConnectApi.GatewayLogResponse`>	Logs showing more details about the tokenization process that occurred in the gateway.	52.0
`paymentMethod`	`ConnectApi.PaymentMethodResponse`	Object representation of the payment method object that was tokenized.	52.0

ConnectApi.PaymentResponse

Payment output.

Property Name	Type	Description	Available Version
`accountId`	String	ID of the account related the payment record.	50.0
`amount`	Double	Total amount of the payment transaction performed in the payment request.	50.0
`currencyIsoCode`	String	Three-letter ISO 4217 currency code associated with the payment output.	50.0
`effectiveDate`	Datetime	Date that the payment becomes effective.	50.0
`id`	String	ID of the payment record.	50.0
`paymentNumber`	String	Number of the payment record created as a result of the request processing.	50.0
`requestDate`	Datetime	Date when the payment transaction occurred.	50.0
`status`	String	Status of the new payment record. Can be DRAFT, PROCESSED or CANCELLED.	50.0

ConnectApi.PercentRecordField

Record field containing a percentage value.

Subclass of ConnectApi.LabeledRecordField.

Name	Type	Description	Available Version
`value`	Double	Value of the percentage.	29.0

ConnectApi.PhoneNumber

Phone number.

Name	Type	Description	Available Version
`label`	String	A localized string indicating the phone type.	30.0
`phoneNumber`	String	Phone number.	28.0
`phoneType`	String	Phone type. Values are: • `Fax` • `Mobile` • `Work` These values are not localized.	30.0

Name	Type	Description	Available Version
`type`	String	Note: This property is not available after version 29.0. Use the `phoneType` property instead. Values are: • `Fax` • `Mobile` • `Work` These values are not localized.	28.0–29.0

SEE ALSO:

ConnectApi.DatacloudCompany
ConnectApi.DatacloudContact
ConnectApi.UserDetail

ConnectApi.Photo

Profile photo.

Name	Type	Description	Available Version
`fullEmailPhotoUrl`	String	A temporary URL to the large profile picture. The URL expires after 30 days and is available to unauthenticated users.	28.0
`largePhotoUrl`	String	URL to the large profile picture. The default width is 200 pixels, and the height is scaled so the original image proportions are maintained. If a user hasn't uploaded a photo, this URL points to a default photo. If the user hasn't uploaded a photo and the request header included `X-Connect-Theme: Salesforce1`, this URL points to a default photo based on a theme that the admin selected for the org.	28.0
`mediumPhotoUrl`	String	URL to the medium profile picture. The default width is 160 pixels, and the height is scaled so the original image proportions are maintained. If a user hasn't uploaded a photo, this URL points to a default photo. If the user hasn't uploaded a photo and the request header included `X-Connect-Theme: Salesforce1`, this URL points to a default photo based on a theme that the admin selected for the org.	37.0
`photoVersionId`	String	18–character ID to that version of the photo	28.0

Name	Type	Description	Available Version
`smallPhotoUrl`	String	URL to the small profile picture. The default size is 64x64 pixels. If a user hasn't uploaded a photo, this URL points to a default photo. If the user hasn't uploaded a photo and the request header included `X-Connect-Theme: Salesforce1`, this URL points to a default photo based on a theme that the admin selected for the org.	28.0
`standardEmailPhotoUrl`	String	A temporary URL to the small profile. The URL expires after 30 days and is available to unauthenticated users.	28.0
`url`	String	A resource that returns a Photo object: for example, `/services/data/v63.0/chatter/users/005D0000001LL8OIAW/photo`	28.0

SEE ALSO:

- ConnectApi.ChatterGroup
- ConnectApi.RecommendationDefinition
- ConnectApi.User

ConnectApi.PicklistRecordField

Record field containing an enumerated value.

Subclass of ConnectApi.LabeledRecordField.

ConnectApi.PinCapability

If a feed element has this capability, users who have permission can pin it to a feed.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
`isPinnableByMe`	Boolean	Specifies whether the context user can pin or unpin the entity to the feed (`true`) or not (`false`).	41.0
`isPinned`	Boolean	Specifies whether the entity is pinned (`true`) or not pinned (`false`) to the feed.	41.0

SEE ALSO:

- ConnectApi.FeedElementCapabilities

ConnectApi.PinnedFeedElements

List of pinned feed elements for a feed.

Property Name	Type	Description	Available Version
`elements`	List<`ConnectApi.FeedElement`>	List of pinned feed elements. Note: In the UI, pinned feed elements don't show all auxiliary information, such as comments, likes, interaction counts, or read by information. As a result, the `ConnectApi.PinnedFeedElements` output class doesn't include all the information for these capabilities.	41.0

ConnectApi.PlatformAction

A platform action instance with state information for the context user.

Property Name	Type	Description	Available Version
`actionUrl`	String	For action links of `subtype` `Ui` or `Download`, direct the user to download or visit the UI from this link. Salesforce issues a Javascript redirect for the link in this format: `/action-link-redirect/`***`communityId`***`/`***`actionLinkId`***`?_bearer=`***`bearerToken`***. For `Api` action links and for all platform actions, this value is `null` and Salesforce handles the call.	33.0
`apiName`	String	The API name. The value may be `null`.	33.0
`confirmationMessage`	String	If this action requires a confirmation and has a status of `NewStatus`, this is a default localized message that should be shown to an end user prior to invoking the action. Otherwise, this is `null`.	33.0
`executingUser`	`ConnectApi.UserSummary`	The user who initiated execution of this platform action.	33.0
`groupDefault`	Boolean	`true` if this platform action is the default or primary platform action in the platform action group; `false` otherwise. There can be only one default platform action per platform action group.	33.0
`iconUrl`	String	The URL of the icon for the platform action. This value may be `null`.	33.0
`id`	String	The ID for the platform action. If the `type` is `QuickAction` and the `subtype` is `Create`, this value is `null`.	33.0
`label`	String	The localized label for this platform action.	33.0

Property Name	Type	Description	Available Version
`modifiedDate`	Datetime	ISO 8601 format date string, for example, 2011-02-25T18:24:31.000Z.	33.0
`platformActionGroup`	`ConnectApi.Reference`	A reference to the platform action group containing this platform action.	33.0
`status`	`ConnectApi.PlatformActionStatus`	The execution status of the platform action. Values are: • `FailedStatus`—The action link execution failed. • `NewStatus`—The action link is ready to be executed. Available for `Download` and `Ui` action links only. • `PendingStatus`—The action link is executing. Choosing this value triggers the API call for `Api` and `ApiAsync` action links. • `SuccessfulStatus`—The action link executed successfully.	33.0
`subtype`	String	The subtype of a platform action or `null`. If the `type` property is `ActionLink`, possible values are: • `Api`—The action link calls a synchronous API at the action URL. Salesforce sets the status to `SuccessfulStatus` or `FailedStatus` based on the HTTP status code returned by your server. • `ApiAsync`—The action link calls an asynchronous API at the action URL. The action remains in a `PendingStatus` state until a third party makes a request to `/connect/action-links/`***`actionLinkId`*** to set the status to `SuccessfulStatus` or `FailedStatus` when the asynchronous operation is complete. • `Download`—The action link downloads a file from the action URL. • `Ui`—The action link takes the user to a web page at the action URL. Note: Invoking `ApiAsync` action links from an app requires a call to set the status. However, there isn't currently a way to set the status of an action link using Apex. To set the status, use Connect REST API. See the Action	33.0

Property Name	Type	Description	Available Version
		Link resource in the Connect REST API Developer Guidefor more information.	
`type`	`ConnectApi.PlatformActionType`	The type of platform action. Values are: • `ActionLink`—An indicator on a feed element that targets an API, a web page, or a file, represented by a button in the Salesforce UI. • `CustomButton`—When clicked, opens a URL or a Visualforce page in a window or executes JavaScript. • `ProductivityAction`—Productivity actions are predefined and attached to a limited set of objects. Productivity actions include Send Email, Call, Map, View Website, and Read News. Except for the Call action, you can't edit productivity actions. • `QuickAction`—A global or object-specific action. • `StandardButton`—A predefined Salesforce button such as New, Edit, or Delete.	33.0
`url`	String	The URL for this platform action. If the `type` is `QuickAction` and the `subtype` is `Create`, this value is `null`.	33.0

SEE ALSO:

ConnectApi.PlatformActionGroup

ConnectApi.PlatformActionGroup

A platform action group instance with state appropriate for the context user.

Action link groups are one type of platform action group and are therefore represented as `ConnectApi.PlatformActionGroup` output classes.

Property Name	Type	Description	Available Version
`category`	`ConnectApi.PlatformActionGroupCategory`	Indicates the priority and relative locations of platform actions. Values are: • `Primary`—The action link group is displayed in the body of the feed element. • `Overflow`—The action link group is displayed in the overflow menu of the feed element.	33.0

Property Name	Type	Description	Available Version
`id`	String	The 18-character ID or an opaque string ID of the platform action group. If the `ConnectApi.PlatformAction` `type` is `QuickAction` and the `subtype` is `Create`, this value is `null`.	33.0
`modifiedDate`	Datetime	ISO 8601 date string, for example, 2014-02-25T18:24:31.000Z.	33.0
`platformActions`	List<`ConnectApi.PlatformAction`>	The platform action instances for this group. Within an action link group, action links are displayed in the order listed in the `actionLinks` property of the `ConnectApi.ActionLinkGroup DefinitionInput` class. Within a feed item, action link groups are displayed in the order specified in the `actionLinkGroupIds` property of the `ConnectApi.AssociatedActions CapabilityInput` class.	33.0
`url`	String	The URL for this platform action group. If the `ConnectApi.PlatformAction` `type` is `QuickAction` and the `subtype` is `Create`, this value is `null`.	33.0

SEE ALSO:

ConnectApi.AbstractRecommendation
ConnectApi.AssociatedActionsCapability

ConnectApi.PollCapability

If a feed element has this capability, it includes a poll.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
`choices`	List<`ConnectApi.FeedPollChoice`>	Collection of poll choices that make up the poll.	32.0
`myChoiceId`	String	18-character ID of the poll choice that the context user has voted for in this poll. Returns `null` if the context user has not voted.	32.0

Property Name	Type	Description	Available Version
totalVoteCount	Integer	Total number of votes cast on the feed poll element.	32.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.PostAuthGatewayResponse

Gateway response after confirmation that the merchant is ready to capture payment of an existing pre-authorized transaction.

Sublass of ConnectApi.AbstractGatewayResponse.

Property Name	Type	Description	Available Version
gatewayAuthorizationCode	String	Code used to authorize the payment that the payment gateway is processing.	54.0
paymentMethodDetails	ConnectApi.PaymentMethodDetails	Details about the payment method.	54.0

ConnectApi.PostAuthorizationResponse

Gateway response following a post authorization request.

Property Name	Type	Description	Available Version
error	ConnectApi.ErrorResponse	Information about errors that occurred in the payment gateway while evaluating the post authorization request.	54.0
gatewayResponse	ConnectApi.PostAuthGatewayResponse	Payment gateway's response to the post authorization request.	54.0
paymentAuthorization	ConnectApi.PaymentAuthorizationResponse	Payment gateway's response to the original payment authorization request.	54.0
paymentGatewayLogs	List<ConnectApi.GatewayLogResponse>	Stores information exchanged between the Salesforce payments platform and external payment gateways. Gateway logs can also record payloads from external payment entities.	54.0
paymentGroup	ConnectApi.PaymentGroupResponse	Payment group, consisting of one or more payments, sent to the gateway for the post authorization request.	54.0
paymentMethod	ConnectApi.PaymentMethodResponse	Payment method used in the post authorization request.	54.0

ConnectApi.PreserveCart

Represents a preserved cart.

Property Name	Type	Description	Available Version
cartId	String	ID of the authenticated cart.	60.0
currencyIsoCode	String	Currency ISO code for the authenticated cart.	60.0
failedCartItems	List<ConnectApi.CartItemBasicResult>	List of products that weren't successfully transferred from the guest cart to the authenticated cart.	60.0
numberOfProducts	Integer	Total number of products in the guest cart.	60.0
numberOfProductsWithError	Integer	Total number of products that weren't successfully transferred from the guest cart to the authenticated cart.	60.0
numberOfProductsWithSuccess	Integer	Total number of products successfully transferred from the guest cart to the authenticated cart.	60.0
succeededCartItems	List<ConnectApi.CartItemBasicResult>	List of products successfully transferred from the guest cart to the authenticated cart.	60.0

ConnectApi.PreviewCancelOutputRepresentation

Expected financial values for a proposed cancel action.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
changeBalances	ConnectApi.ChangeItemOutputRepresentation	Expected financial values for the proposed cancel action.	48.0
orderSummaryId	String	ID of the OrderSummary.	48.0

ConnectApi.PreviewCartToExchangeOrderOutputRepresentation

Expected change order financial values for the preview cart to exchange order action.

Property Name	Type	Description	Available Version
balanceStateExchangeWebCart	ConnectApi.BalanceStatePreviewOutput on page 1985	The balance state preview for the exchange web cart.	Big, 61.0
balanceStateOriginalOrderSummary	ConnectApi.BalanceStatePreviewOutput on page 1985	The balance state preview for the original order summary.	Big, 61.0
balanceStateReturnOrder	ConnectApi.BalanceStatePreviewOutput on page 1985	The balance state preview for the return order.	Big, 61.0

Property Name	Type	Description	Available Version
changeBalances	ConnectApi.ChangeItemOutputRepresentation on page 2027	Change order financial values for a preview order action.	Big, 60.0
errors	List<ConnectApi.ErrorResponse>	Any errors that were returned.	Big, 60.0
orderSummaryId	String	ID of the order summary.	Big, 60.0
success	Boolean	Indicates whether the transaction was successful.	Big, 60.0

ConnectApi.PreviewReturnOutputRepresentation

Expected financial values for a proposed return action.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
changeBalances	ConnectApi.ChangeItemOutputRepresentation	Expected financial values for the proposed return action.	48.0
orderSummaryId	String	ID of the OrderSummary.	48.0

ConnectApi.PriceAdjustmentSchedule

Price adjustment schedule.

Property Name	Type	Description	Available Version
adjustmentMethod		Reserved for future use.	59.0
id	String	ID of the price adjustment schedule.	49.0
priceAdjustmentTiers	List<ConnectApi.PriceAdjustmentTier>	List of price adjustment tiers.	49.0
scheduleType		Reserved for future use.	59.0

SEE ALSO:

ConnectApi.ProductPrice

ConnectApi.PriceAdjustmentTier

Price adjustment tier.

Property Name	Type	Description	Available Version
adjustmentType	ConnectApi.PriceAdjustmentTierType	Type of price adjustment for the tier. Values are: • AmountBasedAdjustment—Price is adjusted by a specified amount. • PercentageBasedAdjustment—Price is adjusted by a specified percentage.	49.0
adjustmentValue	String	Adjustment value of the tier.	49.0
id	String	ID of the price adjustment tier.	49.0
lowerBound	String	Lower limit of the tier.	49.0
tierUnitPrice	String	Unit price of the tier.	49.0
upperBound	String	Upper limit of the tier.	49.0

SEE ALSO:

ConnectApi.PriceAdjustmentSchedule

ConnectApi.PricingResult

Product pricing result.

Property Name	Type	Description	Available Version
currencyIsoCode	String	Three-letter ISO 4217 currency code associated with the product.	49.0
error	ConnectApi.ErrorResponse	Error code and message.	49.0
pricingLineItemResults	List<ConnectApi.PricingResultLineItem>	Pricing result line Items.	49.0
success	Boolean	Specifies whether the execution was successful (true) or not (false).	49.0

ConnectApi.PricingResultLineItem

Pricing result line item.

Property Name	Type	Description	Available Version
error	ConnectApi.ErrorResponse	Error code and message.	49.0
listPrice	String	List price for the product.	49.0
lowestUnitPrice	String	Lowest unit price for the product.	49.0

Property Name	Type	Description	Available Version
`pricebookEntryId`	String	ID of the pricebook entry.	49.0
`productId`	String	ID of the product to price.	49.0
`success`	Boolean	Specifies whether the execution was successful (`true`) or not (`false`).	49.0
`unitPrice`	String	Unit price for the product.	49.0

SEE ALSO:

ConnectApi.PricingResult

ConnectApi.ProductSummary

ConnectApi.ProductAttributeInfo

Product attribute information.

Property Name	Type	Description	Available Version
`allowableValues`	List<String>	Active attribute picklist values that can be used to create variations. These values are determined by the order of the picklist values in Object Manager.	50.0
`apiName`	String	API name of the attribute.	50.0
`availableValues`	List<String>	Attribute picklist values that are available for the product in the store. These values are sorted by the order of values in the `allowableValues` property.	50.0
`fieldEnumOrId`	String	Field ID for custom fields or enumeration value for standard fields.	50.0
`label`	String	Label of the attribute.	50.0
`objectName`	String	Name of the object that contains the field.	50.0
`options`	List<`ConnectApi.ProductAttributeValueMetadataRepresentation`>	List of product attribute value metadata.	63.0
`sequence`	Integer	Sequence value determined by the order of the attributes under Commerce Setup for the attribute set.	50.0

Property Name	Type	Description	Available Version
viewType	ConnectApi.ProductAttributeViewType	View type for product attributes. Values are: • ColorSwatch • Dropdown • Pill	63.0

SEE ALSO:

ConnectApi.ProductDetail
ConnectApi.ProductAttributeSetInfo
ConnectApi.ProductVariationInfo

ConnectApi.ProductAttributeSelectionInfo

Product attribute.

Property Name	Type	Description	Available Version
apiName	String	API name of the attribute.	50.0
label	String	Label of the attribute.	50.0
sequence	Integer	Sequence value determined by the order of the attributes under Commerce Setup for the attribute set.	50.0
value	String	Display value of the attribute.	50.0

SEE ALSO:

ConnectApi.ProductAttributesToProductEntry

ConnectApi.ProductAttributeSet

Product attribute set data.

Property Name	Type	Description	Available Version
attributes	Map<String, String>	Map of the attributes that are members of the attribute set.	50.0
developerName	String	Name of the attribute set.	50.0
id	String	ID of the product attribute record that represents the product attribute set.	50.0

ConnectApi.ProductAttributeSetInfo

Attribute set metadata.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Property Name	Type	Description	Available Version
`attributeInfo`	Map<String, `ConnectApi.ProductAttributeInfo`>	Map of the API name of the attribute field to the attribute metadata.	50.0
`description`	String	Description of the attribute set.	50.0
`developerName`	String	Developer name of the attribute set.	50.0
`id`	String	ID of the attribute set.	50.0
`masterLabel`	String	Label of the attribute set.	50.0
`sequence`	Integer	Sequence of the attribute set for the product.	50.0

SEE ALSO:

ConnectApi.ProductDetail

ConnectApi.ProductAttributeSetSummary

Summary of a product attribute set.

Property Name	Type	Description	Available Version
`apiName`	String	API name of the attribute set.	51.0
`attributes`	List<`ConnectApi.ProductAttributeSummary`>	List of attributes in the attribute set.	51.0
`label`	String	Display label of the attribute set.	51.0

SEE ALSO:

ConnectApi.OrderItemSummaryProduct

ConnectApi.ProductSummary

ConnectApi.ProductAttributeSummary

Summary of a product attribute.

Property Name	Type	Description	Available Version
`apiName`	String	API name of the attribute.	51.0

Property Name	Type	Description	Available Version
`label`	String	Display label of the attribute.	51.0
`sequence`	Integer	Sequence of the attribute in the attribute set.	51.0
`value`	String	Display value of the attribute.	51.0

SEE ALSO:

ConnectApi.ProductAttributeSetSummary

ConnectApi.ProductAttributesToProductEntry

Mapping of an attribute value combination to a variation product ID.

Property Name	Type	Description	Available Version
`canonicalKey`	String	Attribute API values concatenated with an underscore (_) based on the sequence number of the attributes in the attribute set.	50.0
`productId`	String	Variation product ID for the selection of attributes.	50.0
`selectedAttributes`	List<`ConnectApi.ProductAttributeSelectionInfo`>	Ordered list of attribute values and metadata that can be used to form a key that maps to product ID.	50.0
`urlName`	String	Variant URL name for the selection of attributes.	59.0

SEE ALSO:

ConnectApi.ProductVariationInfo

ConnectApi.ProductAttributeValueMetadataRepresentation

Metadata for a product attribute value.

Property Name	Type	Description	Available Version
`apiName`	String	API Name of the product attribute picklist value. For example, `blue__c`.	63.0
`colorHexCode`	String	Hex code value of a color attribute. For example, `#0000FF`.	63.0
`label`	String	Label of the picklist value for a custom field product attribute. For example, `Red`.	63.0
`variantAvailable`	Boolean	Specifies whether a variation product exists (`true`) or not (`false`).	63.0

ConnectApi.ProductCartItem

Cart items of a specific product type.

Property Name	Type	Description	Available Version
`cartItems`	List<`ConnectApi.CartItemResult` on page 2005>	Items in a cart.	60.0
`product`	`ConnectApi.CartItemProduct on page 2004`	Product summary for a cart item.	60.0

ConnectApi.ProductCartItemCollection

Items in the cart, grouped by product type.

Property Name	Type	Description	Available Version
`count`	Integer	Number of cart items returned on the current page	60.0
`currentPage`	Integer	Current page of cart items. The value matches the requested page number, unless the requested page number exceeds the total number of pages. In this scenario, the current page is the highest available page number.	60.0
`hasErrors`	Boolean	Indicates whether at least one of the results contains an error (`True`) or not (`False`).	60.0
`products`	List<`ConnectApi.ProductCartItem` on page 2240>	Products in the cart.	60.0
`totalItemCount`	Integer	Total number of unique products in the cart.	60.0
`totalNumberOfPages`	Integer	Total number of pages for the given page size.	60.0

ConnectApi.ProductCategoryData

Product category.

Property Name	Type	Description	Available Version
`description`	String	Description of the category.	49.0
`id`	String	ID of the category.	49.0
`name`	String	Name of the category.	49.0

Property Name	Type	Description	Available Version
`urlSlug`	String	SEO-friendly URL slug of the category.	59.0

SEE ALSO:

ConnectApi.ProductCategoryPath

ConnectApi.SearchCategory

ConnectApi.ProductCategoryDetail

Details of a product category.

Property Name	Type	Description	Available Version
`bannerImage`	`ConnectApi.ProductCategoryMedia`	Banner image of the product category.	49.0
`fields`	Map<String, String>	List of fields for the product category.	49.0
`id`	String	ID of the product category.	49.0
`mediaGroups`	List<`ConnectApi.ProductCategoryMediaGroup`>	List of media groups of the product category.	49.0
`tileImage`	`ConnectApi.ProductCategoryMedia`	Tile image of the product category.	49.0
`urlName`	String	SEO-friendly URL name of the product category.	59.0

SEE ALSO:

ConnectApi.ProductCategoryDetailCollection

ConnectApi.ProductCategoryDetailCollection

Collection of product category details.

Property Name	Type	Description	Available Version
`productCategories`	List<`ConnectApi.ProductCategoryDetail`>	List of product category details.	52.0

ConnectApi.ProductCategoryMedia

Media associated with a product category.

Property Name	Type	Description	Available Version
`alternateText`	String	Alternative text for the product category media.	49.0
`contentVersionId`	String	ID of the latest published content version if the media is stored as a ContentDocument. If the image is a customer-provided external URL, the value is `null`. Not supported in enhanced CMS workspaces.	49.0
`id`	String	ID of the product category image.	49.0
`mediaType`	`ConnectApi.ProductMediaType`	Type of product media. Values are: • `Document` • `Image` • `Video`	49.0
`sortOrder`	Integer	Sort order of a media item inside a media group.	49.0
`thumbnailUrl`	String	URL of the thumbnail for product media. If a value exists, it should be used for the thumbnail whether the image is natively uploaded or hosted externally. Not supported in enhanced CMS workspaces.	49.0
`title`	String	Title of the product category media.	49.0
`url`	String	URL of the product category media.	49.0

SEE ALSO:

ConnectApi.ProductCategoryMediaGroup

ConnectApi.ProductCategoryDetail

ConnectApi.ProductCategoryMediaGroup

Media group associated with a product category.

Property Name	Type	Description	Available Version
`developerName`	String	API name of the product category media group.	49.0
`id`	String	ID of the product category media group.	49.0
`mediaItems`	List<`ConnectApi.ProductCategoryMedia`>	List of media items within a product category media group.	49.0
`name`	String	Name of the product category media group.	49.0
`usageType`	`ConnectApi.ProductMediaUsageType`	Usage type of a product media item within a media group. Values are: • `Attachment`—Product media group with product documents as attachments.	49.0

Property Name	Type	Description	Available Version
		• `Banner`—Product category media group with banner images of the product. • `Listing`—Product media group with listing images of the product. • `Standard`—Product media group with standard images and videos of the product. • `Tile`—Product category media group with tile images of the product.	

ConnectApi.ProductCategoryPath

List of product categories in a path.

Property Name	Type	Description	Available Version
`path`	List<`ConnectApi.ProductCategoryData`>	List of product categories in a path.	49.0

SEE ALSO:

ConnectApi.ProductDetail

ConnectApi.ProductChild

Child product related to a parent product.

Property Name	Type	Description	Available Version
`defaultQuantity`	String	Default quantity of child products to be ordered.	57.0
`isEntitled`	Boolean	Specifies whether the child product can be viewed on the product detail page (`true`) or not (`false`).	62.0
`productInfo`	`ConnectApi.ProductDetail`	Product details of the child product.	57.0

ConnectApi.ProductChildCollection

Collection of child products related to a parent product.

Property Name	Type	Description	Available Version
`count`	Integer	Number of child products returned on this page.	57.0
`currentPageToken`	String	Current page token, if any.	57.0
`currentPageUrl`	String	URL of the current page in the response.	57.0

Property Name	Type	Description	Available Version
`items`	List<`ConnectApi.ProductChild`>	List of child products related to the parent product. The child products are sorted by their configured sequence values, in ascending order, with null values sorted last. If there are no configured sequence values, the child products are sorted by `createdDate`, in ascending order.	57.0
`nextPageToken`	String	Token for the next page, if any. A value is included in the response only if a value is returned for `nextPageUrl`.	57.0
`nextPageUrl`	String	URL of the next page, if any.	57.0
`previousPageToken`	String	Token for the previous page, if any. A value is included in the response only if a value is returned for `previousPageUrl`.	57.0
`previousPageUrl`	String	URL of the previous page, if any.	57.0
`productClass`	`ConnectApi.ProductClass`	Class of product. Values are: • `Bundle` • `Set` • `Simple` • `Variation` • `VariationParent`	62.0
`total`	Integer	Total number of child products in the collection.	57.0

ConnectApi.ProductDeliverEstimationOutputRepresentation

Delivery estimation information for each product.

Property Name	Type	Description	Available Version
`estimatedDeliveryDate`	DateConnectApi.ProductDeliverEstimationOutputRepresentation on page 2084	Estimated delivery date.	63.0
`estimatedShipDate`	Datetime	Estimated shipping date.	63.0
`quantity`	Double	Product quantity.	63.0
`routingCalculationType`	String	Routing calculation type.	63.0
`stockKeepingUnit`	String	Product stock keeping unit (SKU).	63.0

ConnectApi.ProductDetail

Details of a product.

Property Name	Type	Description	Available Version
`attributeSetInfo`	Map<String, `ConnectApi.ProductAttributeSetInfo`>	Map of the attribute set developer name to its metadata.	50.0
`defaultImage`	`ConnectApi.ProductMedia`	Default image of the product.	49.0
`entitlement`	`ConnectApi.ProductEntitlement`	Entitlement details for the product. To get pricing information for products in version 57 and later, use the CommerceStorePricing Class.	49.0–56.0
`fields`	Map<String, String>	List of fields for the product.	49.0
`id`	String	ID of the product.	49.0
`mediaGroups`	List<`ConnectApi.ProductMediaGroup`>	List of media groups of the product.	49.0
`primaryProductCategoryPath`	`ConnectApi.ProductCategoryPath`	Primary category path of the product.	49.0
`productClass`	`ConnectApi.ProductClass`	Class of product. Values are: • `Bundle` • `Set` • `Simple` • `Variation` • `VariationParent`	50.0
`productSellingModels`	List<`ConnectApi.ProductSellingModel`>	List of product selling models for the product.	56.0
`purchaseQuantityRule`	`ConnectApi.PurchaseQuantityRule`	If one exists, purchase quantity rule for the product.	52.0
`urlName`	String	SEO-friendly URL name for the product.	59.0
`variationAttributeSet`	`ConnectApi.ProductAttributeSet`	Variation attribute set for the product.	50.0
`variationInfo`	`ConnectApi.ProductVariationInfo`	Available and allowable values for variation attributes and a map to resolve variation product IDs from attribute value combinations.	50.0
`variationParentId`	String	ID of the variation parent.	50.0

ConnectApi.ProductDetailsOutputRepresentation

Details about a product.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
attributes	List<ConnectApi.ProductVariationAttributeOutputRepresentation>	List of variation attributes that define variations of the product.	55.0
currencyIsoCode	String	Currency ISO code.	55.0
description	String	Description of the product.	55.0
fields	Map<String, String>	List of the product's fields.	55.0
imageGroups	List<ConnectApi.ProductImageGroupOutputRepresentation>	List of the product's image groups.	55.0
listPrice	Double	List price.	55.0
name	String	Name.	55.0
productQuantityRule	Purchase Quantity Rule	If one exists, purchase quantity rule for the product.	55.0
productId	String	Product ID.	55.0
stockKeepingUnit	String	Stock keeping unit.	55.0
unitPrice	Double	Unit price.	55.0
variants	List<ConnectApi.ProductVariantOutputRepresentation>	List of variations of the product.	55.0

ConnectApi.ProductEntitlement

Entitlements for a product.

Property Name	Type	Description	Available Version
canViewPrice	Boolean	Specifies whether the product's price can be viewed (`true`) or not (`false`).	49.0

SEE ALSO:

ConnectApi.ProductDetail

ConnectApi.ProductExpandOutputRepresentation

Product expand information with return reasons.

Property Name	Type	Description	Available Version
`returnReasons`	List<`String`>	Return reasons for products.	59.0

ConnectApi.ProductImageGroupOutputRepresentation

Details about a product image group.

Property Name	Type	Description	Available Version
`images`	List<`ConnectApi.ProductImageOutputRepresentation`>	List of product images in the group.	55.0
`viewType`	String	The type of product images in the group.	55.0

ConnectApi.ProductImageOutputRepresentation

Details about a product image.

Property Name	Type	Description	Available Version
`alternateText`	String	Alternate text for accessibility.	55.0
`mediaType`	String	Media type.	55.0
`thumbnailUrl`	String	URL of the thumbnail version of the product image.	55.0
`title`	String	Title.	55.0
`url`	String	URL of the product image.	55.0

ConnectApi.ProductsListOutputRepresentation

Output representation of products with product data along with expand details.

Property Name	Type	Description	Available Version
`product`	List<ConnectApi.ProductOutputRepresentation>	The product's identifier.	**59.0**

ConnectApi.ProductMedia

Media associated with a product.

Property Name	Type	Description	Available Version
`alternateText`	String	Alternative text for the product media.	49.0
`contentVersionId`	String	ID of the latest published content version if the media is stored as a ContentDocument. If the image is a customer-provided external URL, the value is `null`. Not supported in enhanced CMS workspaces.	49.0
`id`	String	ID of the product image.	49.0
`mediaType`	`ConnectApi.ProductMediaType`	Type of product media. Values are: • `Document` • `Image` • `Video`	49.0
`sortOrder`	Integer	Sort order of a media item within a media group.	49.0
`thumbnailUrl`	String	URL of the thumbnail for product media. If a value exists, it should be used for the thumbnail whether the image is natively uploaded or hosted externally. Not supported in enhanced CMS workspaces.	49.0
`title`	String	Title of the product media.	49.0
`url`	String	URL of the product media.	49.0

SEE ALSO:

- ConnectApi.CartItemProduct
- ConnectApi.ProductDetail
- ConnectApi.ProductMediaGroup
- ConnectApi.OrderItemSummaryProduct
- ConnectApi.ProductSummary

ConnectApi.ProductMediaGroup

Media group associated with a product.

Property Name	Type	Description	Available Version
`developerName`	String	API name of the product media group.	49.0
`id`	String	ID of the product media group.	49.0
`mediaItems`	List<`ConnectApi.ProductMedia`>	List of media items within a product media group.	49.0
`name`	String	Name of the product media group.	49.0

Property Name	Type	Description	Available Version
usageType	ConnectApi.ProductMediaUsageType	Usage type of a product media item within a media group. Values are: • Attachment—Product media group with product documents as attachments. • Banner—Product category media group with banner images of the product. • Listing—Product media group with listing images of the product. • Standard—Product media group with standard images and videos of the product. • Tile—Product category media group with tile images of the product.	49.0

SEE ALSO:

ConnectApi.ProductDetail

ConnectApi.ProductCategoryDetail

ConnectApi.ProductOutputRepresentation

Output representation for product data.

Property Name	Type	Description	Available Version
expand	[illegible]	Output representation for expand feature.	**59.0**
products	String	Product data.	**59.0**

ConnectApi.ProductOverview

Overview of a product, with summary information about prices, selected fields, and the product's default image.

Property Name	Type	Description	Available Version
defaultImage	ConnectApi.ProductMedia	Media representation of the product's default image.	54.0
error	ConnectApi.ErrorResponse	Error code and error message.	54.0
fields	Map<String, String>	Map of fields belonging to the product.	54.0
id	String	ID of the product.	54.0
prices	ConnectApi.PricingResultLineItem	Price of the product. To get pricing information for products in version 58 and later, use the CommerceStorePricing Class.	54.0–57.0

Property Name	Type	Description	Available Version
sku	String	SKU of the product.	54.0
success	Boolean	Represents whether execution was successful and product overview information was retrieved without error.	54.0

ConnectApi.ProductOverviewCollection

Collection of product overviews.

Property Name	Type	Description	Available Version
products	List<ConnectApi.ProductOverview>	Collection of product overview.	54.0
total	Integer	Total number of products returned.	54.0

ConnectApi.ProductPrice

Pricing information for a product.

Property Name	Type	Description	Available Version
currencyIsoCode	String	Three-letter ISO 4217 currency code associated with the product. Products are priced using the currency for the buyer account or guest buyer profile. If your store doesn't support the currency for the buyer account or guest buyer profile, products are priced using the default currency for your store.	49.0
listPrice	String	List price for the product.	49.0
lowestUnitPrice	String	Lowest unit price for the product.	49.0
priceAdjustment	ConnectApi.PriceAdjustmentSchedule	Price adjustment schedule for the product. If a product selling model ID is specified in a request parameter, this property is empty.	49.0
pricebookEntryId	String	ID of the price book entry. If a product selling model ID is specified in a request parameter, this property is empty.	49.0
productPriceEntries	List<ConnectApi.ProductPriceEntry>	List of line item prices for the product.	56.0
unitPrice	String	Unit price for the product. If a product selling model ID is specified in a request parameter, this property is empty.	49.0

ConnectApi.ProductPriceEntry

Line item price for the product.

Property Name	Type	Description	Available Version
`error`	`ConnectApi.ErrorResponse`	Error code and error message.	56.0
`listPrice`	String	List price for the product entry.	56.0
`priceAdjustment`	`ConnectApi.PriceAdjustmentSchedule`	Price adjustment schedule.	56.0
`pricebookEntryId`	String	ID of the pricebook entry.	56.0
`productSellingModelId`	String	ID of the product selling model. If no product selling model ID is specified in a request parameter, this property is empty.	56.0
`success`	Boolean	Specifies whether execution was successful (`true`) or not (`false`).	56.0
`unitPrice`	String	Unit price for the product entry.	56.0

ConnectApi.ProductReturnRateListOutputRepresentation

Products with corresponding return rates.

Property Name	Type	Description	Available Version
`productReturnRateList`	[illegible]	List of product return rates.	59.0

ConnectApi.ProductReturnRateOutputRepresentation

Return rate of a product (units returned divided by units sold).

Property Name	Type	Description	Available Version
`productId`	String	ID of the product.	59.0
`returnRate`	Double	Return rate of specified product. Values range from 0 to 1.	59.0

ConnectApi.ProductSearchFacetOutputRepresentation

Product search facet value.

Property Name	Type	Description	Available Version
`attributeType`	String	Attribute type of the facet value.	59.0

Property Name	Type	Description	Available Version
displayName	String	Display name of the facet value.	59.0
displayRank	Integer	Display rank of the facet value.	59.0
displayType	String	Display type of the facet value.	59.0
facetType	String	Type of the facet.	59.0
nameOrId	String	Name or ID of the facet.	59.0
values	[illegible]	A list of facet values for the search.	59.0

ConnectApi.ProductSearchFacetValueOutputRepresentation

Output representation of a product search facet value.

Property Name	Type	Description	Available Version
displayName	String	Display name of the search facet.	59.0
nameOrId	String	Unique name or ID of the search facet.	59.0
productCount	Integer	Number of products found with the search facet.	59.0
type	String	Type of the search facet.	59.0

ConnectApi.ProductSearchImageOutputRepresentation

Output representation of the product search image.

Property Name	Type	Description	Available Version
alternateText	String	Alternate text for the product image.	59.0
mediaType	String	Media type of the product image.	59.0
sortOrder	Integer	Sort order of the product image.	59.0
title	String	Title of the product image.	59.0
url	String	URL of the product image.	59.0

ConnectApi.ProductSearchOutputRepresentation

Output representation of the product search response

Property Name	Type	Description	Available Version
currencyIsoCode	String	Three-letter ISO 4217 currency code associated with the product.	59.0

Property Name	Type	Description	Available Version
facets	List<ConnectApi.ProductSearchFacetOutputRepresentation>	A list of facet names to filter the search.For example, ["size_medium", "color_red"].	59.0
locale	String	Locale used for the product search.	59.0
pageNumber	Integer	Maximum number of search results pages to return. If you don't specify a value, the default is 1.	59.0
pageSize	Integer	Number of items per page. Valid values are from 1 through 100. If you don't specify a value, the default size is 20.	59.0
products	List<ConnectApi.ProductSearchProductOutputRepresentation>	List of products found by the search.	59.0
totalRecordsFound	Integer	Total products found.	59.0

ConnectApi.ProductSearchProductOutputRepresentation

Product found by a product search.

Property Name	Type	Description	Available Version
description	String	Description of the product.	59.0
image	ConnectApi.ProductSearchImageOutputRepresentation	Image of the product.	59.0
name	String	Name of the product.	59.0
productClass	String	Class of the product.	59.0
stockKeepingUnit	String	Stock Keeping Unit (SKU) of the product.	59.0
variationAttributeSet	List<ConnectApi.ProductSearchVariationAttributeOutputRepresentation>	Variation attribute set of the product.	59.0

ConnectApi.ProductSearchResults

Product search results.

Property Name	Type	Description	Available Version
categories	ConnectApi.SearchCategory	Categories from the search results.	52.0
correlationId	String	Reserved for future use.	55.0
facets	List<ConnectApi.SearchFacet>	Facets from the search results.	52.0
locale	String	Locale of the search results.	52.0
productsPage	ConnectApi.ProductSummaryPage	Page of products from the search results.	52.0

ConnectApi.ProductSearchSuggestionsResults

Product search suggestions results.

Property Name	Type	Description	Available Version
`recentSearchSuggestions`	List<`ConnectApi.SearchSuggestion`>	Suggestions based on the user's recent searches.	52.0

ConnectApi.ProductSellingModel

Product selling model for Commerce subscriptions.

Property Name	Type	Description	Available Version
`description`	String	Description of the product selling model displayed on the UI.	60.0
`displayName`	String	Name of the product selling model displayed on the UI.	60.0
`id`	String	ID of the product selling model.	56.0
`name`	String	Name of the product selling model.	56.0
`pricingTerm`	Integer	Number of pricing term units in the pricing term. Used with `pricingTermUnit` to define the length of the pricing term. For example, if `pricingTermUnit` is `Months` and this property is 1, the subscription is priced monthly. However, if the `sellingModelType` property is set to `OneTime`, the `pricingTerm` property is empty, because the product isn't sold as a subscription. The only allowed value for this property is 1.	56.0
`pricingTermUnit`	`ConnectApi.PricingTermUnit`	Unit of time used to define a pricing term. Value is: • `Months`—Product is priced on a monthly basis. • `Annual`—Product is priced on an annual basis. This unit of time is combined with a number (specified by the `pricingTerm` property) to define the full term of the subscription. For example, if the unit of time is `Months` and the `pricingTerm` property is set to 1, the subscription is priced monthly. However, if the `sellingModelType` property is set to `OneTime`, the `pricingTermUnit` property is empty, because the product isn't sold as a subscription.	56.0

Property Name	Type	Description	Available Version
sellingModelType	ConnectApi.SellingModelType	Type of product selling model. Values are: • Evergreen—A subscription without an end date. An evergreen subscription continues until the shopper affirmatively cancels it. • OneTime—A product that isn't sold as a subscription. • TermDefined—A subscription with a defined end date. The subscription continues for a specified time period. When the term ends, the subscription ends.	56.0
subscriptionTermRule	ConnectApi.SubscriptionTermRule	Rules for the subscription term.	59.0

ConnectApi.ProductSummary

Product summary.

Property Name	Type	Description	Available Version
defaultImage	ConnectApi.ProductMedia	Default image of the product.	52.0
fields	Map<String, ConnectApi.FieldValue>	Map of fields belonging to the product.	52.0
id	String	ID of the product.	52.0
name	String	Name of the product.	52.0
prices	ConnectApi.PricingResultLineItem	Prices of the product.	52.0
productClass	ConnectApi.ProductClass	Class of product. Values are: • Bundle • Set • Simple • Variation • VariationParent	52.0
productSellingModelInformation	ConnectApi.CommerceProductSellingModel	Product selling model information.	59.0

Property Name	Type	Description	Available Version
purchaseQuantityRule	ConnectApi.PurchaseQuantityRule	If one exists, purchase quantity rule for the product.	52.0
urlName	String	SEO-friendly URL name for the product.	59.0
variationAttributeSet	ConnectApi.ProductAttributeSetSummary	Variation attribute set that's associated with the product.	52.0

SEE ALSO:

ConnectApi.ProductSummaryPage

ConnectApi.ProductSummaryPage

Page of product summaries.

Property Name	Type	Description	Available Version
currencyIsoCode	String	Three-letter ISO 4217 currency code associated with the product.	52.0
pageSize	Integer	Number of products per page in the search results.	52.0
products	List<ConnectApi.ProductSummary>	Collection of product summaries.	52.0
total	Long	Total number of products in the search results.	52.0

SEE ALSO:

ConnectApi.ProductSearchResults

ConnectApi.ProductVariantOutputRepresentation

Details about a product variation.

Property Name	Type	Description	Available Version
listPrice	Double	List price.	55.0
price	Double	Price.	55.0
productId	String	Product ID.	55.0
stockKeepingUnit	String	Stock Keeping Unit.	55.0
unitPrice	Double	Unit price.	55.0
variationValues	Map<String, String>	The variation attribute values that define the variation.	55.0

ConnectApi.ProductVariationAttributeOutputRepresentation

Details about a product variation attribute.

Property Name	Type	Description	Available Version
`apiName`	String	API name of the attribute.	55.0
`label`	String	Label of the attribute.	55.0
`variationAttributeValues`	List<`ConnectApi.ProductVariationAttributeValueOutputRepresentation`>	List of valid values for the variation attribute.	55.0

ConnectApi.ProductVariationAttributeValueOutputRepresentation

Valid value for a product variation attribute.

Property Name	Type	Description	Available Version
`name`	String	API Name of the attribute this value belongs to.	55.0
`orderable`	Boolean	Whether the value defines an orderable product variation.	55.0
`value`	String	Value of the value.	55.0

ConnectApi.ProductVariationInfo

Product variation attributes, metadata, and mappings of attribute combinations to variation product IDs.

Property Name	Type	Description	Available Version
`attributesToProductMappings`	List<`ConnectApi.ProductAttributesToProductEntry`>	List ordered by `ProductAttribute.Sequence` values that map the attribute value combinations to the variation product ID.	50.0
`variationAttributeInfo`	Map<String, `ConnectApi.ProductAttributeInfo`>	Map of field API name to product attribute information.	51.0

SEE ALSO:

ConnectApi.ProductDetail

ConnectApi.PromotionCart

A cart, its items, and its adjustment groups.

Property Name	Type	Description	Available Version
`cartAdjustmentGroups`	List<ConnectApi.PromotionCartAdjustmentGroup>	Cart adjustment groups belonging to the cart.	57.0
`cartItems`	List<ConnectApi.PromotionCartItem>	Cart items belonging to the cart.	57.0
`currencyIsoCode`	String	Currency code of the cart.	57.0
`id`	String	ID of the cart.	57.0
`totalAdjustmentBaseAmount`	String	Total adjustment base amount for the cart.	57.0
`totalNetAmount`	String	Total price of the cart, including adjustments.	57.0
`totalProductBaseAmount`	String	Total price of all cart items in the cart.	57.0

SEE ALSO:

ConnectApi.PromotionEvaluation

evaluate(salesTransaction)

ConnectApi.PromotionCartAdjustmentGroup

Adjustment group associated with a cart.

Property Name	Type	Description	Available Version
`adjustmentBasisReference`	String	ID of the associated coupon, if applicable.	57.0
`adjustmentDescription`	String	Description of the adjustment.	57.0
`adjustmentType`	`ConnectApi.AdjustmentType`	How the price adjustment amount is calculated. Values are: • `AdjustmentAmount`—The adjustment is a fixed amount. • `AdjustmentPercentage`—The adjustment is a percentage.	57.0
`adjustmentValue`	String	Price value of the adjustment.	57.0
`baseAmount`	String	Total amount of the adjustment.	57.0
`cartId`	String	ID of the cart.	57.0
`id`	String	ID of the cart adjustment group.	57.0
`priceAdjustmentCauseId`	String	ID of the related promotion.	57.0

Property Name	Type	Description	Available Version
`priority`	Integer	Where in the sequence of adjustments this adjustment was applied.	57.0

SEE ALSO:

- ConnectApi.PromotionCart
- ConnectApi.PromotionEvaluation
- evaluate(salesTransaction)

ConnectApi.PromotionCartDeliveryMethodAdjustment

Adjustment for a cart delivery method in a promotion.

Property Name	Type	Description	Available Version
`adjustmentType`	String	Type of price adjustment. Valid values are: • `AdjustmentAmount`—The adjustment is a fixed amount. • `AdjustmentPercentage`—The adjustment is a percentage.	60.0
`adjustmentValue`	String	Price value of the adjustment.	60.0
`baseAmount`	String	Price value of the adjustment.	60.0
`priceAdjustmentCauseId`	String	ID of the related promotion.	60.0
`priority`	Integer	Where in the sequence of adjustments this adjustment was applied.	60.0

ConnectApi.PromotionCartItem

A cart item and its adjustments.

Property Name	Type	Description	Available Version
`cartDeliveryGroupId`	String	ID of the associated cart delivery group.	57.0
`cartId`	String	ID of the associated cart.	57.0
`cartItemPriceAdjustments`	List<ConnectApi.PromotionCartItemPriceAdjustment>	List of price adjustments applied to the cart item.	57.0
`id`	String	ID of the cart item.	57.0
`itemDescription`	String	Description of the cart item.	57.0

Property Name	Type	Description	Available Version
`itemName`	String	Name of the cart item.	57.0
`listPrice`	String	Unit list price of the cart item.	57.0
`product2Id`	String	ID of the product.	57.0
`quantity`	String	Quantity of the cart item.	57.0
`salesPrice`	String	Unit sales price of the cart item.	57.0
`sku`	String	Stock keeping unit of the cart item.	57.0
`totalAdjustmentBaseAmount`	String	Total adjustment amount for the cart item.	57.0
`totalLineBaseAmount`	String	Total amount for the cart item, based on sales price and quantity, not including adjustments.	57.0
`totalListBaseAmount`	String	Total amount for the cart item, based on list price and quantity, not including adjustments.	57.0
`totalNetAmount`	String	Total amount for the cart item, based on list price and quantity, including adjustments.	57.0
`type`	`ConnectApi.CartItemType`	Type of item in a cart. Values are: • `DeliveryCharge` • `Product`	57.0

SEE ALSO:

ConnectApi.PromotionCart

ConnectApi.PromotionEvaluation

evaluate(salesTransaction)

ConnectApi.PromotionCartItemPriceAdjustment

Price adjustments applied to a cart item.

Property Name	Type	Description	Available Version
`adjustmentAmountScope`	`ConnectApi.AdjustmentAmountScope`	Scope of the price adjustment amount. Values are: • `Total`—The adjustment scope is the total price. • `Unit`—The adjustment scope is the unit price. • `UnproratedTotal`—The adjustment scope is the unprorated total price.	57.0
`adjustmentBasisReference`	String	ID of the associated coupon, if applicable.	57.0

Property Name	Type	Description	Available Version
adjustmentDescription	String	Description of the adjustment.	57.0
adjustmentTargetType	ConnectApi.CartPromotionType	Level of the promotion target. Values are: • Cart—The target is cart-level. • Item—The target is item-level.	57.0
adjustmentType	ConnectApi.AdjustmentType	How the price adjustment amount is calculated. Values are: • AdjustmentAmount—The adjustment is a fixed amount. • AdjustmentPercentage—The adjustment is a percentage.	57.0
adjustmentValue	String	Value of the price adjustment.	57.0
baseAmount	String	Total adjustment amount.	57.0
cartAdjustmentGroupId	String	ID of the associated cart adjustment group.	57.0
cartItemId	String	ID of the associated cart item.	57.0
id	String	ID of the cart item price adjustment.	57.0
priceAdjustmentCauseId	String	ID of the associated promotion.	57.0
priority	Integer	Where in the sequence of adjustments this adjustment was applied.	57.0

SEE ALSO:

ConnectApi.PromotionCartItem
ConnectApi.PromotionCart
ConnectApi.PromotionEvaluation
evaluate(salesTransaction)

ConnectApi.PromotionCoupon

A coupon used in a promotion.

Property Name	Type	Description	Available Version
couponCode	String	Coupon code.	57.0
couponErrorCode	String	Error code returned if the coupon is invalid.	57.0
id	String	ID of the coupon.	57.0

Property Name	Type	Description	Available Version
`isValidCoupon`	Boolean	Indicates whether the coupon is valid (`true`) or invalid (`false`).	57.0

SEE ALSO:

- ConnectApi.PromotionEvaluation
- evaluate(salesTransaction)

ConnectApi.PromotionEvaluation

Results of a promotion evaluation.

Property Name	Type	Description	Available Version
`cart`	`ConnectApi.PromotionCart`	Cart and its items.	57.0
`coupons`	List<`ConnectApi.PromotionCoupon`>	Collection of coupon codes to enable promotions. A customer can apply a maximum of two coupons.	57.0

SEE ALSO:

- evaluate(salesTransaction)

ConnectApi.PurchaseQuantityRule

Rule that restricts the quantity of a product that can be purchased.

Property Name	Type	Description	Available Version
`increment`	String	Increment value of the quantity that can be purchased.	52.0
`maximum`	String	Maximum quantity that can be purchased.	52.0
`minimum`	String	Minimum quantity that can be purchased.	52.0

SEE ALSO:

- ConnectApi.CartItemProduct
- ConnectApi.ProductDetail
- ConnectApi.ProductSummary

ConnectApi.QuestionAndAnswersCapability

If a feed element has this capability, it has a question and comments on the feed element are answers to the question.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
bestAnswer	ConnectApi.Comment	Comment selected as the best answer for the question.	32.0
bestAnswerSelectedBy	ConnectApi.UserSummary	User who selected the best answer for the question.	32.0
canCurrentUserSelectOrRemoveBestAnswer	Boolean	Indicates whether the context user can select or remove a best answer (true) or not (false).	32.0
candidateAnswers	ConnectApi.CandidateAnswersStatus	Status of candidate answers for the question.	41.0
escalatedCase	ConnectApi.Reference	If a question post is escalated, this is the case to which it was escalated.	33.0
questionTitle	String	Title for the question.	32.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.QuestionAndAnswersSuggestions

Question and answers suggestions.

Property Name	Type	Description	Available Version
articles	List<ConnectApi.ArticleItem>	List of articles.	32.0
questions	List<ConnectApi.FeedElement>	List of questions.	32.0

ConnectApi.RankAverageDistanceOutputRepresentation

The results of calculating the average distances from sets of inventory locations to an order recipient.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
distanceUnit	String	The specified unit of distance (miles or kilometers).	51.0
results	List<ConnectApi.AverageDistanceResultOutputRepresentation>	The results of the shipping distance calculations.	51.0

ConnectApi.ReadBy

Information about who read the feed element and when.

Property Name	Type	Description	Available Version
`lastReadDateByUser`	Datetime	When the user last read the feed element in ISO 8601 format.	40.0
`user`	`ConnectApi.UserSummary`	Information about the user who read the feed element.	40.0

SEE ALSO:

ConnectApi.ReadByPage

ConnectApi.ReadByCapability

If a feed element has this capability, the context user can mark it as read.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
`isReadByMe`	Boolean	Specifies whether the feed element has been read (`true`) or not (`false`) by the context user.	40.0
`lastReadDateByMe`	Datetime	Last date when the feed element was marked read for the context user in ISO 8601 format. Otherwise, `null`.	40.0
`page`	`ConnectApi.ReadByPage`	First page of information about who read the feed element and when.	40.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.ReadByPage

A collection of information about who read the feed element and when.

Property Name	Type	Description	Available Version
`currentPageToken`	String	Token identifying the current page.	40.0
`currentPageUrl`	String	Connect REST API URL identifying the current page. The default is 25 items per page.	40.0
`items`	List<`ConnectApi.ReadBy`>	Collection of read-by information, including users and when they last read the feed element.	40.0
`nextPageToken`	String	Token identifying the next page, or `null` if there isn't a next page.	40.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	40.0

Property Name	Type	Description	Available Version
previousPageToken	String	Reserved for future use.	40.0
previousPageUrl	String	Reserved for future use.	40.0
total	Integer	Total number of users who read the feed element. Note: This count appears in the UI under feed posts in private and unlisted Chatter groups as the "seen by" number, for example, "Seen by 48."	40.0

SEE ALSO:

ConnectApi.ReadByCapability

ConnectApi.Recommendation

A Next Best Action recommendation object.

Property Name	Type	Description	Available Version
acceptanceLabel	String	Text indicating user acceptance of the recommendation.	45.0
actionReference	String	Reference to the action to perform, for example, launching a flow.	45.0
description	String	Description of the recommendation.	45.0
externalId	String	External ID of the recommendation. This ID doesn't need to be a Salesforce 18-character ID. For example, it can be a product number from an external system.	46.0
id	String	ID of the recommendation.	45.0
image	ConnectApi.FileAsset	Image to display.	45.0
name	String	Name of the recommendation.	45.0
rejectionLabel	String	Text indicating user rejection of the recommendation.	45.0
url	String	URL to the recommendation.	45.0

ConnectApi.RecommendationAudience

A custom recommendation audience.

Property Name	Type	Description	Available Version
criteria	ConnectApi.AudienceCriteria	The criteria for the custom recommendation audience type.	36.0
id	String	18-character ID of the custom recommendation audience.	35.0
memberCount	Integer	**Important:** This property is available only in version 35.0. In version 36.0 and later, this property is available in ConnectApi.CustomListAudienceCriteria. Number of members in the custom recommendation audience.	35.0 only
members	ConnectApi.UserReferencePage	**Important:** This property is available only in version 35.0. In version 36.0 and later, this property is available in ConnectApi.CustomListAudienceCriteria. Members of the custom recommendation audience.	35.0 only
modifiedBy	ConnectApi.User	User who last modified the custom recommendation audience.	36.0
modifiedDate	Datetime	ISO 8601 format date string, for example, 2011-02-25T18:24:31.000Z.	36.0
name	String	Name of the custom recommendation audience.	35.0
url	String	URL to the custom recommendation audience.	35.0

SEE ALSO:

ConnectApi.RecommendationAudiencePage

ConnectApi.RecommendationAudiencePage

A list of custom recommendation audiences.

Property Name	Type	Description	Available Version
audienceCount	Integer	The total number of custom recommendation audiences.	35.0
currentPageUrl	String	URL to the current page.	35.0
nextPageUrl	String	URL to the next page.	35.0
previousPageUrl	String	URL to the previous page.	35.0

Property Name	Type	Description	Available Version
recommendationAudiences	List<ConnectApi.RecommendationAudience>	A list of custom recommendation audiences.	35.0

ConnectApi.RecommendationCollection

A list of Chatter, custom, and static recommendations.

Property Name	Type	Description	Available Version
recommendations	List<ConnectApi.AbstractRecommendation>	Collection of Chatter, custom, and static recommendations.	33.0

ConnectApi.RecommendationDefinition

Represents a custom recommendation definition.

Property Name	Type	Description	Available Version
actionUrl	String	The URL for acting on this custom recommendation.	35.0
actionUrlName	String	The text label for the action URL in the user interface.	35.0
explanation	String	Explanation of the custom recommendation definition.	35.0
id	String	18-character ID of the custom recommendation definition.	35.0
name	String	Name of the custom recommendation definition. The name is displayed in Setup.	35.0
photo	ConnectApi.Photo	Photo of the custom recommendation definition.	35.0
title	String	Title of the custom recommendation definition.	35.0
url	String	URL to the Connect REST API resource for the custom recommendation definition.	35.0

SEE ALSO:

- ConnectApi.RecommendationDefinitionPage
- ConnectApi.ScheduledRecommendation

ConnectApi.RecommendationDefinitionPage

A list of custom recommendation definitions.

Property Name	Type	Description	Available Version
recommendationDefinitions	List<ConnectApi.RecommendationDefinition>	A list of custom recommendation definitions.	35.0
url	String	URL to the Connect REST API resource for the recommendation definition collection.	35.0

ConnectApi.RecommendationExplanation

Explanation for a Chatter recommendation.

Subclass of ConnectApi.AbstractRecommendationExplanation.

Property Name	Type	Description	Available Version
detailsUrl	String	URL to explanation details or `null` if the Chatter recommendation doesn't have a detailed explanation.	32.0

SEE ALSO:

ConnectApi.AbstractRecommendation

ConnectApi.RecommendationReaction

A reaction to a recommendation produced by a recommendation strategy

Property Name	Type	Description	Available Version
aiModel	String	Reserved for future use.	47.0
contextRecord	ConnectApi.Reference	Reference to the context record.	45.0
createdBy	ConnectApi.Reference	Reference to the reaction creator.	45.0
createdDate	Datetime	Reaction creation date.	45.0
externalId	String	External target ID of the recommendation reacted on. This ID doesn't need to be a Salesforce 18-character ID. For example, it can be a product number from an external system.	46.0
id	String	Reaction record ID.	45.0
onBehalfOf	ConnectApi.Reference	Reference to the user or record that is indirectly reacting to the recommendation.	45.0
reactionType	ConnectApi.RecommendationReactionType	Type of reaction to a recommendation. Values are: • `Accepted` • `Rejected`	45.0

Property Name	Type	Description	Available Version
recommendationMode	String	Reserved for future use.	46.0
recommendationScore	Double	Reserved for future use.	46.0
strategy	ConnectApi.RecordSnapshot	Strategy that recommended the target record.	45.0
targetAction	ConnectApi.RecordSnapshot	Target action that is recommended.	45.0
targetRecord	ConnectApi.Reference	Reference to the target record.	45.0
url	String	URL to the recommendation reaction.	45.0

SEE ALSO:

ConnectApi.RecommendationReactions

ConnectApi.RecommendationReactions

A list of recommendation reactions.

Property Name	Type	Description	Available Version
currentPageUrl	String	URL to the current page of reactions in the collection.	45.0
nextPageUrl	String	URL to the next page of reactions in the collection.	45.0
reactions	List<ConnectApi.RecommendationReaction>	Collection of recommendation reactions.	45.0

ConnectApi.RecommendationsCapability

If a feed element has this capability, it has a recommendation.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
items	List<ConnectApi.AbstractRecommendation>	A list of recommendations.	32.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.RecommendedObject

A recommended object, such as a custom or static recommendation.

Subclass of ConnectApi.Actor

Property Name	Type	Description	Available Version
`idOrEnum`	String	ID of a recommendation definition for a custom recommendation or the enum value `Today` for static recommendations that don't have an ID (version 35.0 and later).	34.0
`motif`	`ConnectApi.Motif`	Motif of the recommended object.	34.0

ConnectApi.RecordCapability

If a comment has this capability, it has a record attachment.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
`recordId`	String	ID of the record.	42.0
`url`	String	URL to the record.	42.0

ConnectApi.RecordField

Generic record field containing a label and text value.

Subclass of ConnectApi.LabeledRecordField.

No additional properties.

SEE ALSO:

- ConnectApi.CompoundRecordField
- ConnectApi.OrderItemSummary
- ConnectApi.OrderItemSummaryProduct
- ConnectApi.OrderDeliveryGroupSummary
- ConnectApi.OrderSummaryRepresentation

ConnectApi.RecordFieldValue

Field value.

Property Name	Type	Description	Available Version
`displayValue`	String	Field display value.	63.0
`highlight`	String	Represents field highlighting in the results.	63.0

Property Name	Type	Description	Available Version
`value`	Object	Field raw value.	63.0

SEE ALSO:

ConnectApi.SearchResult

ConnectApi.RecordSnapshot

A record snapshot in a recommendation reaction.

Property Name	Type	Description	Available Version
`id`	String	ID of the record.	45.0
`nameAtSnapshot`	String	Name of the record when the ID was recorded.	45.0

SEE ALSO:

ConnectApi.RecommendationReaction

ConnectApi.RecordSnapshotCapability

If a feed element has this capability, it contains all the snapshotted fields of a record for a single create record event.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
`recordView`	`ConnectApi.RecordView`	A record representation that includes metadata and data so you can display the record easily.	32.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.RecordSummary

Record summary.

Subclass of ConnectApi.AbstractRecordView.

Property Name	Type	Description	Available Version
entityLabel	ConnectApi.EntityLabel	Label of the record's entity.	40.0

SEE ALSO:

- ConnectApi.EmailAddress
- ConnectApi.EmailAttachment
- ConnectApi.ReferenceRecordField
- ConnectApi.ReferenceWithDateRecordField

ConnectApi.RecordSummaryList

Summary information about a list of records in the organization including custom objects.

Name	Type	Description	Available Version
records	List<ConnectApi.ActorWithId>	A list of records.	30.0
url	String	The URL to this list of records.	30.0

ConnectApi.RecordView

A view of any record in the org, including a custom object record. This object is used if a specialized object, such as User or ChatterGroup, isn't available for the record type.

Subclass of ConnectApi.AbstractRecordView.

Name	Type	Description	Available Version
sections	List<ConnectApi.RecordViewSection>	List of record view sections.	29.0

SEE ALSO:

- ConnectApi.RecordSnapshotCapability

ConnectApi.RecordViewSection

Section of record fields and values on a record detail.

Name	Type	Description	Available Version
columnCount	Integer	Number of columns to use to lay out the fields in a record section.	29.0

Name	Type	Description	Available Version
`columnOrder`	`ConnectApi.RecordColumnOrder` Enum	Order of the fields to use in the `fields` property to lay out the fields in a record section. • `LeftRight`—Fields are rendered from left to right. • `TopDown`—Fields are rendered from the top down.	29.0
`fields`	`ConnectApi.AbstractRecordField`	Fields and values for the record contained in this section.	29.0
`heading`	String	Localized label to display when rendering this section of fields.	29.0
`isCollapsible`	Boolean	Indicates whether the section can be collapsed to hide all the fields (`true`) or not (`false`).	29.0

SEE ALSO:

ConnectApi.RecordView

ConnectApi.RecordsetFilterCriteria

Recordset filter criteria and the filtered records.

Property Name	Type	Description	Available Version
`criteriaId`	String	Recordset filter criteria ID.	53.0
`recordIds`	List<String>	List of filtered record IDs.	53.0

SEE ALSO:

ConnectApi.RecordsetFilterCriteriaCollection

ConnectApi.RecordsetFilterCriteriaCollection

List of the recordset filters and records.

Property Name	Type	Description	Available Version
`recordsetFilters`	List<`ConnectApi.RecordsetFilterCriteria`>	Collection of recordset filter criteria IDs and filtered record IDs.	53.0

ConnectApi.Reference

Reference to a record.

Name	Type	Description	Available Version
id	String	The ID of the record being referenced, which could be an 18-character ID or some other string identifier.	28.0
url	String	The URL to the resource endpoint.	28.0

ConnectApi.ReferenceRecordField

Record field with a label and text value.

Subclass of ConnectApi.LabeledRecordField.

Name	Type	Description	Available Version
reference	ConnectApi.RecordSummary	Object referenced by the record field.	29.0

ConnectApi.ReferencedRefundResponse

Refund comprehensive output.

Property Name	Type	Description	Available Version
error	ConnectApi.ErrorResponse	Error response representation for the refund.	50.0
gatewayResponse	ConnectApi.RefundGatewayResponse	Gateway response received for the processed refund request.	50.0
paymentGatewayLogs	List<ConnectApi.GatewayLogResponse>	Gateway log collection representation for the refund.	50.0
paymentGroup	ConnectApi.PaymentGroupResponse	Payment group associated with the refund.	50.0
refund	ConnectApi.RefundResponse	Refund response representation.	50.0

ConnectApi.ReferenceWithDateRecordField

Record field containing a referenced object that acted at a specific time, for example, "Created By...".

Subclass of ConnectApi.LabeledRecordField.

Name	Type	Description	Available Version
dateValue	Datetime	Time at which the referenced object acted.	29.0
reference	ConnectApi.RecordSummary	Object referenced by the record field.	29.0

ConnectApi.RefundGatewayResponse

Refund gateway response.

Subclass of ConnectApi.AbstractGatewayResponse.

No additional properties.

ConnectApi.RefundResponse

Refund output.

Property Name	Type	Description	Available Version
`accountId`	String	ID of the account related to the refund record.	50.0
`amount`	Double	Total amount of the refund transaction performed in the payment request.	50.0
`currencyIsoCode`	String	Three-letter ISO 4217 currency code associated with the payment group record.	50.0
`effectiveDate`	Datetime	Date that the refund becomes effective.	50.0
`id`	String	ID of the refund record.	50.0
`refundNumber`	String	Number of the refund record that was created as a result of the request processing.	50.0
`requestDate`	Datetime	Date when the refund occurred.	50.0
`status`	String	Indicates the results of processing the refund transaction in the gateway. Can be DRAFT, PROCESSED or CANCELLED.	50.0

ConnectApi.RegisterGuestBuyerOutputRepresentation

Indicates success or failure of a register guest buyer action.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
`errors`	List<`ConnectApi.ErrorResponse`>	Any errors that were returned.	48.0
`success`	Boolean	Indicates whether the transaction was successful.	48.0

ConnectApi.RelatedFeedPost

This class is abstract.

Subclass of ConnectApi.ActorWithId.

Superclass of: ConnectApi.RelatedQuestion.

Property Name	Type	Description	Available Version
score	Double	Score of the related feed post that indicates how closely related it is to the context feed post.	37.0
title	String	Title of the related feed post.	37.0

SEE ALSO:

ConnectApi.RelatedFeedPosts

ConnectApi.RelatedFeedPosts

A collection of related feed posts.

Property Name	Type	Description	Available Version
relatedFeedPosts	List<ConnectApi.RelatedFeedPost>	Collection of related feed posts.	37.0

ConnectApi.RelatedQuestion

A related question.

Subclass of ConnectApi.RelatedFeedPost.

Property Name	Type	Description	Available Version
hasBestAnswer	Boolean	Indicates whether the question has a best answer.	37.0
interactions	ConnectApi.InteractionCapability	The number of individual views, likes, and comments on a question.	38.0

ConnectApi.ReleaseHeldFOCapacityOutputRepresentation

Response to a request to confirm held fulfillment order capacity at one or more locations. Can correspond to one action call.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
releaseHeldFOCapacityResponses	List<ConnectApi.ReleaseHeldFOCapacityResponseOutputRepresentation>	List of responses to the requests to confirm held fulfillment order capacity at one or more locations.	55.0

ConnectApi.ReleaseHeldFOCapacityResponseOutputRepresentation

Response to a request to release held fulfillment order capacity at one or more locations.

Property Name	Type	Description	Available Version
capacityResponses	List<ConnectApi.CapacityResponseOutputRepresentation>	List of responses to the requests to release held fulfillment order capacity at individual locations.	55.0

ConnectApi.ReplyIntent

Reply intent for a social post.

Property Name	Type	Description	Available Version
managedSocialAccount	ConnectApi.ManagedSocialAccount	Managed social account that replies to the social post.	45.0

SEE ALSO:

ConnectApi.ReplyIntents

ConnectApi.ReplyIntents

List of reply intents for a social post.

Property Name	Type	Description	Available Version
replies	List<ConnectApi.ReplyIntent>	List of reply intents for the social post.	45.0

SEE ALSO:

ConnectApi.SocialPostIntents

ConnectApi.RepositoryFileDetail

A detailed description of a repository file.

Subclass of ConnectApi.AbstractRepositoryFile.

No additional properties.

ConnectApi.RepositoryFileSummary

A summary of a repository file.

Subclass of ConnectApi.AbstractRepositoryFile.

No additional properties.

SEE ALSO:

ConnectApi.RepositoryFolderItem

ConnectApi.RepositoryFolderDetail

A detailed description of a repository folder.

Subclass of ConnectApi.AbstractRepositoryFolder.

No additional properties.

ConnectApi.RepositoryFolderItem

A folder item.

Property Name	Type	Description	Available Version
`file`	`ConnectApi.RepositoryFileSummary`	If the folder item is a file, the file summary. If the folder item is a folder, `null`.	39.0
`folder`	`ConnectApi.RepositoryFolderSummary`	If the folder item is a folder, the folder summary. If the folder item is a file, `null`.	39.0
`type`	`ConnectApi.FolderItemType`	Type of item in a folder. Values are: • `file` • `folder`	39.0

SEE ALSO:

ConnectApi.RepositoryFolderItemsCollection

ConnectApi.RepositoryFolderItemsCollection

A collection of repository folder items.

Property Name	Type	Description	Available Version
`currentPageUrl`	String	URL to the current page of items.	39.0
`items`	List<`ConnectApi.RepositoryFolderItem`>	Collection of items in a repository folder.	39.0
`nextPageUrl`	String	URL to the next page of items, or `null` if there isn't a next page.	39.0
`previousPageUrl`	String	URL to the previous page of items, or `null` if there isn't a previous page.	39.0

ConnectApi.RepositoryFolderSummary

A summary of a repository folder.

Subclass of ConnectApi.AbstractRepositoryFolder.

No additional properties.

SEE ALSO:

ConnectApi.RepositoryFolderItem

ConnectApi.RepositoryGroupSummary

A group summary.

Subclass of ConnectApi.AbstractDirectoryEntrySummary.

Property Name	Type	Description	Available Version
`groupType`	`ConnectApi.ContentHubGroupType`	Type of group. Values are: • `Everybody`—Group is public to everybody. • `EverybodyInDomain`—Group is public to everybody in the same domain. • `Unknown`—Group type is unknown.	39.0
`name`	String	Name of the group.	39.0

SEE ALSO:

ConnectApi.ExternalFilePermissionInformation

ConnectApi.RepositoryUserSummary

A user summary.

Subclass of ConnectApi.AbstractDirectoryEntrySummary.

Property Name	Type	Description	Available Version
`firstName`	String	First name of the user.	39.0
`lastName`	String	Last name of the user.	39.0

ConnectApi.Reputation

Reputation for a user.

Property Name	Type	Description	Available Version
`reputationLevel`	`ConnectApi.ReputationLevel`	User's reputation level.	32.0
`reputationPoints`	Double	User's reputation points, which can be earned by performing different activities.	32.0

Property Name	Type	Description	Available Version
`url`	String	Connect REST API URL to the reputation.	32.0

SEE ALSO:

ConnectApi.User

ConnectApi.ReputationLevel

Reputation level for a user.

Property Name	Type	Description	Available Version
`levelImageUrl`	String	URL to the reputation level image.	32.0
`levelName`	String	Name of the reputation level.	32.0
`levelNumber`	Integer	Reputation level number, which is the numerical rank of the level, with the lowest level at 1. Administrators define the reputation level point ranges.	32.0

SEE ALSO:

ConnectApi.Reputation

ConnectApi.RequestHeader

An HTTP request header name and value pair.

Property Name	Type	Description	Available Version
`name`	String	The name of the request header.	33.0
`value`	String	The value of the request header.	33.0

SEE ALSO:

ConnectApi.ActionLinkDefinition

ConnectApi.ResourceLinkSegment

Resource link segment.

Name	Type	Description	Available Version
`url`	String	URL to a resource not otherwise identified by an ID field, for example, a link to a list of users.	28.0

ConnectApi.ReturnItemsOutputRepresentation

Output of Return Items. Includes the ID of the generated change order for items and delivery charges being returned, as well as the ID of the generated change order for any charged return fees. Also includes information about any ReturnOrderLineItems that were created to represent remaining return quantities.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
`changeOrderId`	String	ID of the change order created by processing the ReturnOrderLineItems representing returned items and delivery charges. Use this change order to create a credit memo.	52.0
`feeChangeOrderId`	String	ID of the change order created by processing the ReturnOrderLineItems representing return fees. Use this change order to create an invoice.	56.0
`returnLineItemSplits`	List<`ConnectApi.ReturnOrderItemSplitLineOutputRepresentation`>	List of properties representing any remaining quantities from partial returns processed by this call. It includes order items, delivery charges, and return fees. Each element of the list includes the ID of a split ReturnOrderLineItem and the ID of the partially processed ReturnOrderLineItem whose remaining quantity it holds.	52.0

SEE ALSO:

returnItems(returnOrderId, returnItemsInput)

ConnectApi.ReturnOrderItemSplitLineOutputRepresentation

After a change order is created for a ReturnOrderLineItem, that ReturnOrderLineItem is read-only. If the Return Items API is used to return a partial quantity, it creates a new "split" ReturnOrderLineItem to hold the remaining quantity to be returned. In that case, it returns this output property, which contains the IDs of the original and split ReturnOrderLineItems.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
`newReturnOrderItemId`	String	ID of the new ReturnOrderLineItem that holds the remaining return quantity.	52.0
`originalReturnOrderItemId`	String	ID of the original ReturnOrderLineItem.	52.0

SEE ALSO:

ConnectApi.ReturnItemsOutputRepresentation

returnItems(returnOrderId, returnItemsInput)

ConnectApi.ReturnOrderOutputRepresentation

ID of the created ReturnOrder.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
`returnOrderId`	String	ID of the created ReturnOrder.	50.0

SEE ALSO:

createReturnOrder(returnOrderInput)

ConnectApi.SaleGatewayResponse

Sale gateway response.

Subclass of ConnectApi.AbstractGatewayResponse.

No additional properties.

ConnectApi.SaleResponse

Payment sale response.

Property Name	Type	Description	Available Version
`error`	`ConnectApi.ErrorResponse`	Error representation for the payment sale.	54.0
`gatewayResponse`	`ConnectApi.SaleGatewayResponse`	Information from the payment gateway following the sale request.	54.0
`payment`	`ConnectApi.PaymentResponse`	Information about the payment used in the sale request.	54.0
`paymentGatewayLogs`	List<`ConnectApi.GatewayLogResponse`>	Collection of responses from the gateway following the sale request.	54.0
`paymentGroup`	`ConnectApi.PaymentGroupResponse`	Payment group used in the sale request.	54.0
`paymentMethod`	`ConnectApi.PaymentMethodResponse`	Payment method used in the sale request.	54.0

ConnectApi.ScheduledRecommendation

Represents a scheduled custom recommendation.

Property Name	Type	Description	Available Version
`channel`	`ConnectApi.RecommendationChannel`	A way to tie custom recommendations together. For example, display recommendations in specific places in the UI or show recommendations based on time of day or geographic locations. Values are: • `CustomChannel1`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels. For example, community managers can use Experience Builder to determine where recommendations appear. • `CustomChannel2`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels. • `CustomChannel3`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels. • `CustomChannel4`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels. • `CustomChannel5`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels. • `DefaultChannel`—Default recommendation channel. Recommendations appear by default on the Home and Question Detail pages of Customer Service and Partner Central Experience Builder templates. They also appear in the feed in the Salesforce mobile web and anywhere community managers add recommendations using Experience Builder.	36.0
`enabled`	Boolean	Indicates whether scheduling is enabled. If `true`, the custom recommendation is enabled and appears in Experience Cloud sites. If `false`, custom recommendations in feeds in Salesforce mobile web aren't removed, but no new custom recommendations appear. In Customer Service and Partner Central sites, disabled custom recommendations no longer appear.	35.0
`id`	String	18-character ID of the scheduled custom recommendation.	35.0

Property Name	Type	Description	Available Version
`rank`	Integer	The rank determining the order of this scheduled custom recommendation.	35.0
`recommendationAudienceId`	String	ID of the audience for the scheduled custom recommendation.	35.0
`recommendationDefinitionRepresentation`	`ConnectApi.RecommendationDefinition`	Custom recommendation definition that this scheduled recommendation schedules.	35.0
`url`	String	URL to the Connect REST API resource for the scheduled custom recommendation.	35.0

SEE ALSO:

ConnectApi.ScheduledRecommendationPage

ConnectApi.ScheduledRecommendationPage

A list of scheduled custom recommendations.

Property Name	Type	Description	Available Version
`scheduledRecommendations`	List<`ConnectApi.ScheduledRecommendation`>	A list of scheduled custom recommendations.	35.0
`url`	String	URL to the Connect REST API resource for the scheduled custom recommendation collection.	35.0

ConnectApi.Scope

Scope information for a target.

Property Name	Type	Description	Available Version
`name`	String	Name of the scope for the target.	48.0–49.0
`value`	String	Value of the scope for the target.	48.0–49.0

SEE ALSO:

ConnectApi.Target

ConnectApi.ScopedSearchResults

Keyword search results for an object.

Property Name	Type	Description	Available Version
metadata	ConnectApi.ObjectMetadata	All search metadata associated with the object.	63.0
searchObject	ConnectApi.SearchObject	Record results for the keyword search.	63.0

SEE ALSO:

find(objectApiName, request)

ConnectApi.SearchAnswer

Results of searching objects using a natural language query.

Property Name	Type	Description	Available Version
content	String	AI generated response.	63.0
llmGenerationId	String	LLM generation ID used to track any feedback on the conversation.	63.0
metadata	Map<String, ConnectApi.ObjectMetadata>	All search related metadata associated with the objects found in the results.	63.0
searchObjects	List<ConnectApi.SearchObject>	Record results for the natural language search.	63.0

SEE ALSO:

answer(q, objectApiName)

answer(q, objectApiName, displayFields)

answer(q)

ConnectApi.SearchCategory

Search category.

Property Name	Type	Description	Available Version
category	ConnectApi.ProductCategoryData	Information about the category.	52.0
children	List<ConnectApi.SearchCategory>	First-level child categories of the category searched with non-empty search results.	52.0

Property Name	Type	Description	Available Version
`productCount`	Long	Number of products in the search results that belong to the category.	52.0

SEE ALSO:

ConnectApi.ProductSearchResults

ConnectApi.SearchFacet

Search facet.

This class is abstract and is a superclass of ConnectApi.DistinctValueSearchFacet.

Property Name	Type	Description	Available Version
`attributeType`	`ConnectApi.CommerceSearchAttributeType`	Search attribute type. Values are: • `Custom` • `ProductAttribute` • `ProductCategory` • `Product2` • `Standard`	52.0
`displayName`	String	Display name of the facet.	52.0
`displayRank`	Integer	Display rank of the facet. Valid values are from 1 through 50.	52.0
`displayType`	`ConnectApi.CommerceSearchFacetDisplayType`	Display type of the facet. Values are: • `CategoryTree` • `DatePicker` • `MultiSelect` • `SingleSelect`	52.0
`facetType`	`ConnectApi.CommerceSearchFacetType`	Search facet type. Value is: • `DistinctValue`	52.0
`nameOrId`	String	Developer name of the attribute. In version 52.0 and later, the ID of the attribute isn't supported.	52.0

SEE ALSO:

ConnectApi.ProductSearchResults

ConnectApi.SearchObject

Record results for the keyword search.

Property Name	Type	Description	Available Version
displayFields	List<String>	Fields to display from the response.	63.0
objectApiName	String	Object API name.	63.0
orderBy	List<ConnectApi.SearchAppliedOrderBy>	Applied order for object search.	63.0
pageInfo	ConnectApi.PageInfo	Page position information for the object search.	63.0
searchResults	List<ConnectApi.SearchResult>	Search results from the query.	63.0
spellCorrectionInfo	ConnectApi.SpellCorrectionInfo	Spell correction information for the object search.	63.0
status	ConnectApi.SearchStatus	Provides status on the object search such as error messages and warnings.	63.0

SEE ALSO:

- ConnectApi.SearchResultGroups
- ConnectApi.SearchAnswer
- ConnectApi.ScopedSearchResults

ConnectApi.SearchResult

Results from searching an object using keywords.

Property Name	Type	Description	Available Version
fields	Map<String, ConnectApi.RecordFieldValue>	Field values by field API name.	63.0
id	String	ID of the record.	63.0
matchInfo	ConnectApi.MatchInfo	Search information related to the search result.	63.0

SEE ALSO:

- ConnectApi.SearchObject

ConnectApi.SearchResultGroups

Results of searching objects using keywords.

Property Name	Type	Description	Available Version
`metadata`	Map<String, `ConnectApi.ObjectMetadata`>	All related metadata associated with the objects found in the results.	63.0
`searchObjects`	List<`ConnectApi.SearchObject`>	Record results for the keyword search grouped by object.	63.0

SEE ALSO:

findAndGroup(q)

findAndGroup(q, configurationName)

findAndGroup(q, configurationName, highlights)

ConnectApi.SearchStatus

Provides status on the object search.

Property Name	Type	Description	Available Version
`code`	String	Search status code.	63.0
`message`	String	Search status message.	63.0

SEE ALSO:

ConnectApi.SearchObject

ConnectApi.SearchSuggestion

Search suggestion.

Property Name	Type	Description	Available Version
`value`	String	Search suggestion.	52.0

SEE ALSO:

ConnectApi.ProductSearchSuggestionsResults

ConnectApi.ServiceAppointmentOutput

Output of the create service appointment request.

Property Name	Type	Description	Available Version
`result`	ConnectApi.ServiceAppointmentResult	Result of the create or update service appointment request.	53.0

ConnectApi.ServiceAppointmentResult

Contains result of the service appointment.

Property Name	Type	Description	Available Version
`assignedResourceIds`	List<String>	The IDs of the assigned resources.	53.0
`parentRecordId`	String	The ID of the parent record.	53.0
`serviceAppointmentId`	String	The ID of the service appointment record.	53.0

ConnectApi.ShiftsFromPattern

Shifts created from a pattern.

Property Name	Type	Description	Available Version
`count`	Integer	Total count of created shifts.	51.0
`error`	`ConnectApi.ShiftsFromPatternError`	Error details for shifts from a pattern.	53.0
`isSuccess`	Boolean	Indicates if the request is successful (`true`) or not (`false`).	53.0
`recordIds`	List<String>	Collection of created shift IDs.	51.0

ConnectApi.ShiftsFromPatternError

Shifts from pattern error response.

Property Name	Type	Description	Available Version
`code`	String	Error code.	53.0
`invalidServiceResourceId`	String	ID of invalid service resource.	53.0
`message`	String	Error message.	53.0

ConnectApi.SiteSearchItem

Site search result item.

Property Name	Type	Description	Available Version
`contentReference`	String	Content reference field, which is the route developer name for a site page or a content key for a content detail page.	54.0

Property Name	Type	Description	Available Version
`contentType DeveloperName`	String	Developer name of the content type of the site search result item.	54.0
`highlightedSnippet`	String	Text snippet that contains the query term.	54.0
`id`	String	ID of the site search result item.	54.0
`pageType`	`ConnectApi. SitesPageType`	Type of site search result item. Values are: • `ContentPage` • `SitePage`	54.0
`title`	String	Title of the site search result item.	54.0

SEE ALSO:

ConnectApi.SiteSearchResult

ConnectApi.SiteSearchResult

Site search result.

Property Name	Type	Description	Available Version
`currentPageToken`	String	Token for the current page of search results.	54.0
`currentPageUrl`	String	URL to the current page of search results.	54.0
`items`	List<`ConnectApi. SiteSearchItem`>	Collection of search result items.	54.0
`language`	String	Language of the search results.	54.0
`nextPageToken`	String	Token for the next page of search results.	54.0
`nextPageUrl`	String	URL to the next page of search results, or `null` if there isn't a next page.	54.0
`pageSize`	Integer	Number of items per page in search results.	54.0
`previousPageToken`	String	Token for the previous page of search results.	54.0
`previousPageUrl`	String	URL to the previous page of search results, or `null` if there isn't a previous page.	54.0
`totalItems`	Integer	Total number of items in the search results across all pages.	54.0

ConnectApi.SocialAccount

A social account on a social network.

Property Name	Type	Description	Available Version
`externalSocialAccountId`	String	ID of the external social account, if available.	38.0
`handle`	String	Social handle, screen name, or alias that identifies this account.	36.0
`name`	String	Name of the account as defined by the account's owner.	36.0
`profileUrl`	String	URL to the account's profile.	36.0
`socialPersonaId`	String	ID of the social persona account, if the external social account ID isn't available.	39.0

SEE ALSO:

ConnectApi.SocialPostCapability

ConnectApi.SocialAccountRelationship

Follow relationship between a managed social account and a social persona.

Property Name	Type	Description	Available Version
`isFollowed`	Boolean	Specifies whether the social account is followed by the social persona.	46.0
`isFollowing`	Boolean	Specifies whether the social account is following the social persona.	46.0
`socialAccountId`	String	ID of the social account.	46.0
`socialPersonaId`	String	ID of the social persona.	46.0

ConnectApi.SocialPostCapability

If a feed element has this capability, it can interact with a social post on a social network.

Subclass of ConnectApi.FeedElementCapabilities.

Property Name	Type	Description	Available Version
`author`	`ConnectApi.SocialAccount`	Social account that authored the social post.	36.0
`content`	String	Content body of the social post.	36.0
`deletedBy`	`ConnectApi.UserSummary`	User who deleted the social post.	38.0
`hiddenBy`	`ConnectApi.UserSummary`	User who hid the social post.	41.0
`icon`	`ConnectApi.Icon`	Icon of the social network.	36.0
`id`	String	ID associated with the social post Salesforce record.	36.0

Property Name	Type	Description	Available Version
`isOutbound`	Boolean	If `true`, the social post originated from the Salesforce application.	36.0
`likedBy`	String	External social account who liked the social post.	40.0
`messageType`	`ConnectApi.SocialPostMessageType`	Message type of the social post. Values are: • `Comment` • `Direct` • `Post` • `PrivateMessage` • `Reply` • `Retweet` • `Tweet`	38.0
`name`	String	Title or heading of the social post.	36.0
`postUrl`	String	External URL to the social post on the social network.	36.0
`provider`	`ConnectApi.SocialNetworkProvider`	Social network that this social post belongs to. Values are: • `Facebook` • `GooglePlus` • `Instagram` • `InstagramBusiness` • `KakaoTalk` • `Kik` • `Line` • `LinkedIn` • `Messenger` • `Other` • `Pinterest` • `QQ` • `Rypple` • `SinaWeibo` • `SMS` • `Snapchat` • `Telegram` • `Twitter` • `VKontakte` • `WeChat` • `WhatsApp` • `YouTube`	36.0

Property Name	Type	Description	Available Version
recipient	ConnectApi.SocialAccount	Social account that is the recipient of the social post.	36.0
recipientId	String	ID of the recipient of the social post.	38.0
reviewScale	Double	Review scale of the social post.	40.0
reviewScore	Double	Review score of the social post.	40.0
status	ConnectApi.SocialPostStatus	Status of the social post.	36.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.SocialPostIntents

Intents available for a social post.

Property Name	Type	Description	Available Version
approvalIntent	ConnectApi.ApprovalIntent	Approval intent for the social post.	45.0
deleteIntent	ConnectApi.DeleteIntents	Delete intents for the social post.	45.0
followIntent	ConnectApi.FollowIntents	Follow intents for the social persona.	45.0
hideIntent	ConnectApi.HideSocialPostIntent	Hide intent for the social post.	45.0
likeIntent	ConnectApi.LikeIntents	Like intents for the social post.	45.0
replyIntent	ConnectApi.ReplyIntents	Reply intents for the social post.	45.0

ConnectApi.SocialPostMassApprovalOutput

Approval or rejection of a large number of social posts.

Property Name	Type	Description	Available Version
isApproved	Boolean	Specifies whether the social posts were approved (`true`) or rejected (`false`) for publishing.	46.0

ConnectApi.SocialPostStatus

The status of a social post.

Property Name	Type	Description	Available Version
`message`	String	Status message.	36.0
`type`	`ConnectApi.SocialPostStatusType`	Status type. Values are: • `ApprovalPending` • `ApprovalRecalled` • `ApprovalRejected` • `Deleted` • `Failed` • `Hidden` • `Pending` • `Sent` • `Unknown`	36.0

SEE ALSO:

ConnectApi.SocialPostCapability

ConnectApi.SortRule

Sort rule.

Property Name	Type	Description	Available Version
`direction`	`ConnectApi.CommerceSearchSortRuleDirection`	Direction of the sort rule. Values are: • `Ascending`—Sorts in ascending alphanumeric order (A–Z, 0–9). • `Default`—When no direction is defined, sorts by relevance. • `Descending`—Sorts in descending alphanumeric order (Z–A, 9–0).	52.0
`label`	String	Label of the sort rule.	52.0
`labelSuffix`	`ConnectApi.CommerceSearchSortRuleLabelSuffix`	Label suffix of the sort rule.Values are: • `Ascen`—Label suffix for 'Asc' • `Ascending`—Label suffix for 'Ascending' • `Az`—Label suffix for 'A-Z' • `Descen`—Label suffix for 'Desc' • `Descending`—Label suffix for 'Descending' • `FewMany`—Label suffix for 'Few-Many'	54.0

Property Name	Type	Description	Available Version
		• `HeavyLight`—Label suffix for 'Heavy-Light' • `HighLow`—Label suffix for 'High-Low' • `HighestLowest`—Label suffix for 'Highest-Lowest' • `LightHeavy`—Label suffix for 'Light-Heavy' • `LowHigh`—Label suffix for 'Low-High' • `LowestHighest`—Label suffix for 'Lowest-Highest' • `ManyFew`—Label suffix for 'Many-Few' • `NewOld`—Label suffix for 'New-Old' • `Newest`—Label suffix for 'Newest' • `NewestOldest`—Label suffix for 'Newest-Oldest' • `NineZero`—Label suffix for '9-0' • `OldNew`—Label suffix for 'Old-New' • `Oldest`—Label suffix for 'Oldest' • `OldestNewest`—Label suffix for 'Oldest-Newest' • `PriceDecreasing`—Label suffix for '$$-$' • `PriceIncreasing`—Label suffix for '$-$$' • `ThickThin`—Label suffix for 'Thick-Thin' • `ThinThick`—Label suffix for 'Thin-Thick' • `Za`—Label suffix for 'Z-A' • `ZeroNine`—Label suffix for '0-9'	
`nameOrId`	String	Name of the sort rule field or, if the sort rule is based on a custom field, ID.	52.0
`sortOrder`	Integer	Sort order for the rule. A lower number has higher precedence. The first sort option is called when no other option is selected.	54.0
`sortRuleId`	String	ID of the sort rule.	52.0

Property Name	Type	Description	Available Version
`type`	`ConnectApi.CommerceSearchSortRuleType`	Type of sort rule. Values are: • `ProductAttributeBased`—Sorts by product attribute fields. • `ProductBased`—Sorts by product field data. • `Relevancy`—Sorts by product and catalog term frequency. • `SortByPricebook`—Sorts by product prices defined in the specified pricebook (version 55.0 and later).	52.0

SEE ALSO:

ConnectApi.SortRulesCollection

ConnectApi.SortRulesCollection

Collection of sort rules.

Property Name	Type	Description	Available Version
`sortRules`	List<`ConnectApi.SortRule`>	Collection of sort rules.	52.0

ConnectApi.SpellCorrectionInfo

Spell correction information for object search.

Property Name	Type	Description	Available Version
`correctedQuery`	String	Specifies corrected query.	63.0
`hasNonCorrectedResults`	Boolean	Specifies whether some non-corrected results were returned (`true`) or not (`false`).	63.0

SEE ALSO:

ConnectApi.SearchObject

ConnectApi.Stamp

A user stamp.

Property Name	Type	Description	Available Version
`description`	String	Description of the stamp.	39.0–43.0

Property Name	Type	Description	Available Version
`id`	String	ID of the stamp.	39.0–43.0
`imageUrl`	String	Image URL of the stamp.	39.0–43.0
`label`	String	Label of the stamp.	39.0–43.0

SEE ALSO:

ConnectApi.User

ConnectApi.StatusCapability

If a feed post or comment has this capability, it has a status that determines its visibility.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
`feedEntityStatus`	`ConnectApi.FeedEntityStatus`	Status of the feed post or comment. Values are: • `Draft`—The feed post isn't published but is visible to the author and users with Modify All Data or View All Data permission. Comments can't be drafts. • `Isolated`—The feed post or comment is isolated, and only admins can see it. • `PendingReview`—The feed post or comment isn't approved yet and therefore isn't published or visible. • `Published`—The feed post or comment is approved and visible.	37.0
`isApprovableByMe`	Boolean	Specifies whether the context user can change the status of the feed post or comment.	37.0

SEE ALSO:

ConnectApi.CommentCapabilities

ConnectApi.FeedElementCapabilities

ConnectApi.StrategyTrace

Messages and trace nodes for a recommendation strategy execution.

Property Name	Type	Description	Available Version
`messages`	List<String>	Messages and errors from the strategy execution.	45.0

Property Name	Type	Description	Available Version
nodes	List<ConnectApi.StrategyTraceNode>	Nodes of the strategy execution used for debugging.	45.0

SEE ALSO:

ConnectApi.NBARecommendations

ConnectApi.StrategyTraceNode

A trace node for a recommendation strategy execution.

Property Name	Type	Description	Available Version
inputCount	Integer	Number of items put into the node.	45.0
messages	List<String>	Messages that occurred during node execution.	45.0
nodeName	String	Name of the node.	45.0
nodeTime	Long	Time spent processing inside the node.	45.0
nodeType	String	Type of node.	45.0
outputCount	Integer	Number of items returned from the node.	45.0
outputs	List<String>	Recommendations that are returned from the node.	45.0
totalTime	Long	Total time spent processing.	45.0

SEE ALSO:

ConnectApi.StrategyTrace

ConnectApi.SubmitCancelOutputRepresentation

ID of the change order created for a cancel action, and a set of its financial values.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
changeBalances	ConnectApi.ChangeItemOutputRepresentation	Financial values resulting from the cancel.	48.0
changeOrderId	String	ID of the change order created for the canceled order items and shipping charges. Use this change order to create a credit memo.	48.0
feeChangeOrderId	String	ID of the change order created by canceling order items with associated cancel fees. Use this change order to create an invoice.	57.0

ConnectApi.SubmitCartToExchangeOrderOutputRepresentation

Exhange order summary resulting from a submit cart to exchange order action.

Property Name	Type	Description	Available Version
balanceStateExchangeWebCart	ConnectApi.BalanceStatePreviewOutput on page 1985	The balance state preview for the exchange web cart.	Big, 61.0
balanceStateOriginalOrderSummary	ConnectApi.BalanceStatePreviewOutput on page 1985	The balance state preview for the original order summary.	Big, 61.0
balanceStateReturnOrder	ConnectApi.BalanceStatePreviewOutput on page 1985	The balance state preview for the return order.	Big, 61.0
changeBalances	ConnectApi.ChangeItemOutputRepresentation on page 2027	Change order financial values for a preview order action.	Big, 60.0
errors	List<ConnectApi.ErrorResponse>	Any errors that were returned.	Big, 60.0
exchangeOrderSummaryId	String	Exchange order summary ID.	Big, 60.0
orderSummaryId	String	ID of the order summary.	Big, 60.0
success	Boolean	Indicates whether the transaction was successful.	Big, 60.0

ConnectApi.SubmitReturnOutputRepresentation

ID of the change order created for a return action, and a set of its financial values.

Subclass of ConnectApi.BaseOutputRepresentation.

Property Name	Type	Description	Available Version
changeBalances	ConnectApi.ChangeItemOutputRepresentation	Financial values resulting from the return.	48.0
changeOrderId	String	ID of the change order created for the returned order items and shipping charges. Use this change order to create a credit memo.	48.0
feeChangeOrderId	String	ID of the change order created by returning order items with associated return fees. Use this change order to create an invoice.	57.0

ConnectApi.Subscription

Subscription.

Name	Type	Description	Available Version
`community`	`ConnectApi.Reference`	Information about the Experience Cloud site in which the subscription exists.	28.0
`id`	String	Subscription's 18–character ID.	28.0
`subject`	`ConnectApi.Actor`	Information about the parent, that is, the thing or person being followed.	28.0
`subscriber`	`ConnectApi.Actor`	Information about the subscriber, that is, the person following this item.	28.0
`url`	String	Connect REST API URL to this specific subscription.	28.0

SEE ALSO:

ConnectApi.FollowerPage

ConnectApi.FollowingPage

ConnectApi.SubscriptionTermRule

Subscription term rules.

Property Name	Type	Description	Available Version
`increment`	Integer	Number of pricing term units that can be used to increase the subscription term.	59.0
`maximum`	Integer	Maximum number of pricing term units per subscription term.	59.0
`minimum`	Integer	Minimum number of pricing term units per subscription term.	59.0

ConnectApi.SupportedEmojis

A collection of supported emoji.

Property Name	Type	Description	Available Version
`supportedEmojis`	`ConnectApi.EmojiCollection`	A collection of supported emoji.	39.0

ConnectApi.SurveyInvitationEmailOutput

Survey invitation email.

Property Name	Type	Description	Available Version
`errorCode`	Integer	Error code for the failed call.	50.0

Property Name	Type	Description	Available Version
errorMessage	String	Details explaining why the call failed.	50.0
status	ConnectApi.SurveyEmailStatusEnum	Status of a survey invitation email. Values are: • Failed—The survey invitation email wasn't sent. • Queued—The survey invitation email is queued for sending.	50.0

ConnectApi.Target

Personalization target information.

Property Name	Type	Description	Available Version
audience	ConnectApi.AudienceTarget	Audience assigned to the target.	48.0
formulaScope	ConnectApi.FormulaScope	Formula scope of the target.	50.0
groupName	String	Group name of the target. Groups bundle related target and audience pairs.	48.0
id	String	ID of the target.	48.0
priority	Integer	Priority of the target. Within a group, priority determines which target is returned if the user matches more than one audience.	48.0
publishStatus	ConnectApi.PublishStatus	Publish status of the target. Values are: • Draft • Live	48.0
scope	List<ConnectApi.Scope>	List of scopes for the target. In version 50.0 and later, the formulaScope property returns this information.	48.0–49.0
targetType	String	Type of target, indicating the nature of the data being targeted.	48.0
targetValue	String	Value of the target.	48.0
url	String	URL to the target.	48.0

SEE ALSO:

ConnectApi.TargetCollection

ConnectApi.TargetCollection

List of personalization targets.

Property Name	Type	Description	Available Version
`targets`	List<`ConnectApi.Target`>	List of personalization targets.	48.0

ConnectApi.TaxAddressesResponse

The Ship From, Ship To, and Sold To addresses used during tax calculation.

Property Name	Type	Description	Available Version
`shipFrom`	`ConnectApi.TaxAddressResponse`	The Ship From address used in tax calculation.	55.0
`shipTo`	`ConnectApi.TaxAddressResponse`	The Ship To address used in tax calculation.	55.0
`soldTo`	`ConnectApi.TaxAddressResponse`	The Sold To address used in tax calculation.	55.0

ConnectApi.TaxAddressResponse

Location code of an address.

Property Name	Type	Description	Available Version
`locationCode`	String	Location code of an address.	55.0

ConnectApi.TaxAmountDetailsResponse

Information about tax amount values on the line item.

Property Name	Type	Description	Available Version
`exemptAmount`	Double	Amount of the line item exempt from tax application.	55.0
`taxAmount`	Double	Tax amount for the line item.	55.0
`totalAmount`	Double	Total amount of the line item.	55.0
`totalAmountWithTax`	Double	The line item's total amount with tax.	55.0

ConnectApi.TaxDetailsResponse

Tax details for each line item in a tax line item output.

Property Name	Type	Description	Available Version
`exemptAmount`	Double	Amount of the line item that is exempt from taxation.	55.0
`exemptReason`	String	The reason that any tax exemption applied to the line item.	55.0
`imposition`	`ConnectApi.TaxImpositionResponse`	The business justification for applying tax to a line item.	55.0
`jurisdiction`	`ConnectApi.TaxJurisdictionResponse`	Business address used to calculate the tax rate for the line item.	55.0
`rate`	Double	Tax rate for the line item.	55.0
`tax`	Double	Total amount of tax on the line item.	55.0
`taxId`	String	ID for the type of tax applied to the line item.	55.0
`taxableAmount`	Double	Amount of line item that can be taxed.	55.0

ConnectApi.TaxEngineLogResponse

Shows the results of the tax calculation request to the tax engine.

Property Name	Type	Description	Available Version
`createdDate`	Datetime	The date that the gateway log was created.	55.0
`id`	String	ID of the tax engine log record.	55.0
`resultCode`	String	Result code sent from the external tax engine. Review the tax engine provider's documentation for more information about the code.	55.0

ConnectApi.TaxImpositionResponse

Tax imposition output representation.

Property Name	Type	Description	Available Version
`name`	String	Name of the tax imposition.	55.0
`type`	String	Type of the tax imposition.	55.0

ConnectApi.TaxJurisdictionResponse

Represents the address or jurisdiction of the primary business used for calculating tax.

Property Name	Type	Description	Available Version
`country`	String	Country of the tax jurisdiction address.	55.0
`id`	String	ID of the tax jurisdiction address.	55.0
`level`	String	Level of the tax jurisdiction address.	55.0
`name`	String	Name of the tax jurisdiction address.	55.0
`region`	String	Region of the tax jurisdiction address.	55.0
`stateAssignedNo`	String	State-assigned number of the tax jurisdiction address.	55.0

ConnectApi.TaxTransactionResponse

Tax transaction output representation

This class is abstract.

Superclass of ConnectApi.CalculateTaxResponse.

Property Name	Type	Description	Available Version
`addresses`	`ConnectApi.TaxAddressesResponse`	The Ship From, Ship To, and Sold To addresses used during tax calculation.	55.0
`amountDetails`	`ConnectApi.TaxAmountDetailsResponse`	Information about tax amount values on the line item.	55.0
`currencyIsoCode`	String	Three-letter ISO 4217 currency code associated with the payment group record.	55.0
`description`	String	Information about whether the tax transaction failed or was successful.	55.0
`documentCode`	String	Document code.	55.0
`effectiveDate`	Datetime	The date that tax is applied to the taxed entity.	55.0
`lineItems`	List<`ConnectApi.LineItemResponse`>	A list of line items on which tax was calculated.	55.0
`referenceDocumentCode`	String	The original document code. Used in case of subsequent transactions such as credit tax.	55.0
`referenceEntityId`	String	ID of the reference entity used during tax calculation.	55.0
`taxTransactionId`	String	ID of the tax transaction.	55.0

Property Name	Type	Description	Available Version
`transactionDate`	Datetime	The date that the tax transaction occurred.	55.0

ConnectApi.TextClassificationsBulkResultsOutputRepresentation

Text classification to get results for multiple text classification request IDs.

Property Name	Type	Description	Available Version
`resultsList`	[illegible]	List of results for passed request IDs.	**59.0**

ConnectApi.TextClassificationsOutputRepresentation

HTTP headers containing URLs associating text strings and classifications.

Property Name	Type	Description	Available Version
`httpHeaders`	[illegible]	HTTP headers for text classifications output. Each header provides a URL you can use to get the result of the classification. The URL takes a list of text strings and classifiers that each text string can be classified in.	**59.0**

ConnectApi.TextClassificationsResultOutputRepresentation

Text classifications result.

Property Name	Type	Description	Available Version
`classifications`	[illegible]	List of classifications that each text string was given after analysis.	**59.0**
`classificationsId`	String	Response ID to receive feedback for classification.	**59.0**

ConnectApi.TextClassificationsResultWithIdOutputRepresentation

Classified text with status and text classification request IDs.

Property Name	Type	Description	Available Version
`id`	String	Request ID for text classifications.	**59.0**
`result`	[illegible]	Result for text classifications.	**59.0**
`status`	String	Request status for text classification.	**59.0**

ConnectApi.TextSegment

Text segment.

Subclass of ConnectApi.MessageSegment.

No additional properties.

ConnectApi.ThemeInfo

Theme information related to an object.

Property Name	Type	Description	Available Version
`color`	String	Color that represents the object.	63.0
`iconUrl`	String	Icon that represents the object.	63.0

SEE ALSO:

ConnectApi.ObjectMetadata

ConnectApi.TimeZone

User's time zone as selected in the user's personal settings in Salesforce. This value doesn't reflect a device's current location.

Name	Type	Description	Available Version
`gmtOffset`	Double	Signed offset, in hours, from GMT.	30.0
`name`	String	Display name of this time zone.	30.0

SEE ALSO:

ConnectApi.UserSettings

ConnectApi.Topic

Topic.

Name	Type	Description	Available Version
`createdDate`	Datetime	ISO 8601 date string, for example, 2011-02-25T18:24:31.000Z.	29.0
`description`	String	Description of the topic.	29.0
`id`	String	18-character ID.	29.0
`images`	`ConnectApi.TopicImages`	Images associated with the topic.	32.0

Name	Type	Description	Available Version
`isBeingDeleted`	Boolean	`true` if the topic is currently being deleted; `false` otherwise. After the topic is deleted, when attempting to retrieve the topic, the output is NOT_FOUND.	33.0
`name`	String	Name of the topic.	29.0
`nonLocalizedName`	String	Non-localized name of the topic.	36.0
`talkingAbout`	Integer	Number of people talking about this topic over the last two months, based on factors such as topic additions and comments on posts with the topic.	29.0
`url`	String	URL to the topic detail page.	29.0

SEE ALSO:

- ConnectApi.ManagedTopic
- ConnectApi.TopicPage
- ConnectApi.TopicEndorsement
- ConnectApi.TopicEndorsementCollection
- ConnectApi.TopicSuggestion
- ConnectApi.TopicsCapability

ConnectApi.TopicEndorsement

Represents one user endorsing another user for a single topic.

Name	Type	Description	Available Version
`endorsee`	`ConnectApi.UserSummary`	User being endorsed.	30.0
`endorsementId`	String	18-character ID of the endorsement record.	30.0
`endorser`	`ConnectApi.UserSummary`	User performing the endorsement.	30.0
`topic`	`ConnectApi.Topic`	Topic the user is being endorsed for.	30.0
`url`	String	URL to the resource for the endorsement record.	30.0

ConnectApi.TopicEndorsementCollection

Collection of topic endorsement response bodies.

Name	Type	Description	Available Version
currentPageUrl	String	Connect REST API URL identifying the current page.	30.0
nextPageUrl	String	Connect REST API URL identifying the next page, or null if there isn't a next page.	30.0
previousPageUrl	String	Connect REST API URL identifying the previous page, or null if there isn't a previous page.	30.0
topicEndorsements	List<ConnectApi.Topic>	List of topic endorsements.	30.0

ConnectApi.TopicEndorsementSummary

Topic endorsement summary.

Subclass of ConnectApi.UserActivitySummary.

Property Name	Type	Description	Available Version
endorsementId	ID	ID of the topic endorsement.	42.0

ConnectApi.TopicImages

Images associated with a topic.

Property Name	Type	Description	Available Version
coverImageUrl	String	URL to a topic's cover image, which appears on the topic page. Both topics and managed topics can have cover images.	32.0
featuredImageUrl	String	URL to a managed topic's featured image, which appears wherever you feature it, for example, on your Experience Cloud site home page.	32.0

SEE ALSO:

ConnectApi.Topic

ConnectApi.TopicPage

Page of topics.

Name	Type	Description	Available Version
currentPageUrl	String	Connect REST API URL identifying the current page.	29.0

Name	Type	Description	Available Version
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	29.0
`topics`	`List<ConnectApi.Topic>`	List of topics.	29.0

ConnectApi.TopicsCapability

If a feed element has this capability, the context user can add topics to it. Topics help users organize and discover conversations.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
`canAssignTopics`	Boolean	`true` if a topic can be assigned to the feed element, `false` otherwise.	32.0
`items`	`List<ConnectApi.Topic>`	A collection of topics associated with this feed element.	32.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.TopicSuggestion

Topic suggestion.

Name	Type	Description	Available Version
`existingTopic`	`ConnectApi.Topic`	Topic that already exists or `null` for a new topic	29.0
`name`	String	Topic name	29.0

SEE ALSO:

ConnectApi.TopicSuggestionPage

ConnectApi.TopicSuggestionPage

Page of topic suggestions.

Name	Type	Description	Available Version
`TopicSuggestions`	`List<ConnectApi.TopicSuggestion>`	List of topic suggestions.	29.0

ConnectApi.TopicSummary

Summary of a topic.

Property Name	Type	Description	Available Version
`id`	String	ID of the topic.	47.0
`name`	String	Name of the topic.	47.0

SEE ALSO:

ConnectApi.ManagedContentAssociations

ConnectApi.TrackedChangeBundleCapability

If a feed element has this capability, it has a group of other feed elements aggregated into one feed element called a *bundle*. This type of bundle aggregates feed tracked changes.

Subclass of ConnectApi.BundleCapability.

Property Name	Type	Description	Available Version
`changes`	List<`ConnectApi.TrackedChangeItem`>	Collection of feed tracked changes.	31.0

ConnectApi.TrackedChangeItem

Tracked change item.

Name	Type	Description	Available Version
`fieldName`	String	The name of the field that was updated.	28.0
`newValue`	String	The new value of the field or `null` if the field length is long.	28.0
`oldValue`	String	The old value of the field or `null` if the field length is long.	28.0

SEE ALSO:

ConnectApi.TrackedChangesCapability

ConnectApi.TrackedChangeBundleCapability

ConnectApi.TrackedChangesCapability

If a feed element has this capability, it contains all changes to a record for a single tracked change event.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
changes	List<ConnectApi.TrackedChangeItem>	Collection of feed tracked changes.	32.0

SEE ALSO:

ConnectApi.FeedElementCapabilities

ConnectApi.UnauthenticatedUser

Unauthenticated user.

Subclass of ConnectApi.Actor.

No additional properties.

Instances of this class are used as the actor for feed items and comments posted by Chatter customers.

ConnectApi.UnreadConversationCount

Unread count for a conversation.

Name	Type	Description	Available Version
hasMore	Boolean	Specifies if there are more than 50 unread messages (true) or not (false).	29.0
unreadCount	Integer	The total number of unread messages.	29.0

ConnectApi.UpDownVoteCapability

If a feed post or comment has this capability, users can upvote or downvote it.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
downVoteCount	Long	Number of downvotes.	41.0
myVote	ConnectApi.UpDownVoteValue	Specifies the context user's vote. Values are: • Down • None • Up	41.0

Property Name	Type	Description	Available Version
`upVoteCount`	Long	Number of upvotes.	41.0

SEE ALSO:

- ConnectApi.CommentCapabilities
- ConnectApi.FeedElementCapabilities

ConnectApi.UpVoteSummary

Summary of an upvote.

Subclass of ConnectApi.UserFeedEntityActivitySummary.

No additional properties.

ConnectApi.User

User.

This class is abstract.

Subclass of ConnectApi.ActionWithId.

Superclass of:

- ConnectApi.UserDetail
- ConnectApi.UserSummary

Name	Type	Description	Available Version
`additionalLabel`	String	If one exists, an extra label for the user, for example, “Customer,” “Partner,” or “Acme Corporation.”	30.0
`communityNickname`	String	User’s nickname in the site.	32.0
`companyName`	String	Name of the company. If your Experience Cloud site allows access without logging in, the value is `null` for guest users.	28.0
`displayName`	String	User’s name that is displayed in the site. If nicknames are enabled, the nickname is displayed. If nicknames aren’t enabled, the full name is displayed.	32.0
`firstName`	String	User's first name. In version 39.0 and later, if nicknames are enabled, `firstName` is `null`.	28.0
`isChatterGuest`	Boolean	`true` if user is a Chatter customer; `false` otherwise.	28.0
`isInThisCommunity`	Boolean	`true` if user is in the same site as the context user; `false` otherwise.	28.0
`lastName`	String	User's last name. In version 39.0 and later, if nicknames are enabled, `lastName` is `null`.	28.0

Name	Type	Description	Available Version
outOfOffice	ConnectApi.OutOfOffice	If one exists, extra out-of-office message for the user.	40.0
photo	ConnectApi.Photo	Information about the user's photos.	28.0
reputation	ConnectApi.Reputation	Reputation of the user.	32.0
stamps	List<ConnectApi.Stamp>	Collection of the user's stamps. In version 44.0 and later, use SOQL to get a user's stamps.	39.0–43.0
title	String	User's title. If your Experience Cloud site allows access without logging in, the value is `null` for guest users.	28.0
userType	ConnectApi.UserType Enum	Type of user. • ChatterGuest—User is an external user in a private group. • ChatterOnly—User is a Chatter Free customer. • Guest—User is unauthenticated. • Internal—User is a standard org member. • Portal—User is an external user in an Experience Cloud site. • System—User is Chatter Expert or a system user. • Undefined—User is a user type that is a custom object.	28.0

SEE ALSO:

ConnectApi.RecommendationAudience

ConnectApi.UserActivitiesJob

User activities job.

Property Name	Type	Description	Available Version
jobToken	String	Token that identifies the user activities job.	42.0
jobType	String	Type of user activities job. Value is `export` or `purge`.	42.0

Property Name	Type	Description	Available Version
`message`	String	Message describing the status and expected outcome of the job. When the job completes, you receive an email with information about the Salesforce file that contains `ConnectApi.UserActivityCollection`.	42.0

SEE ALSO:

exportUserActivities(communityId, userId)

purgeUserActivities(communityId, userId)

ConnectApi.UserActivityCollection

User activity collection.

Property Name	Type	Description	Available Version
`activityType`	String	Type of user activity. Values are: • `Bookmark`—User bookmarked a post. • `ChatterActivity`—Total counts of posts and comments made and likes and comments received for a user. • `ChatterLike`—User liked a post or comment. • `Comment`—User commented on a post. • `CompanyVerify`—User verified comment. • `DownVote`—User downvoted a post or comment. • `FeedEntityRead`—User read a post. • `FeedRead`—User read a feed. • `Mute`—User muted a post. • `Post`—User made a post. • `TopicEndorsement`—User endorsed another user on a topic or received endorsement on a topic. • `UpVote`—User upvoted a post or comment.	42.0
`userActivities`	List<`ConnectApi.UserActivitySummary`>	Collection of user activities.	42.0

SEE ALSO:

ConnectApi.UserActivitiesJob

ConnectApi.UserActivitySummary

User activity summary.

This class is abstract.

Superclass of:

- ConnectApi.CommentSummary
- ConnectApi.FeedPostSummary
- ConnectApi.FeedReadSummary
- ConnectApi.TopicEndorsementSummary
- ConnectApi.UserFeedEntityActivitySummary

Property Name	Type	Description	Available Version
`activityDate`	Datetime	Date of the user activity.	42.0
`activityType`	String	Type of user activity. Values are: • `Bookmark`—User bookmarked a post. • `ChatterActivity`—Total counts of posts and comments made and likes and comments received for a user. • `ChatterLike`—User liked a post or comment. • `Comment`—User commented on a post. • `CompanyVerify`—User verified comment. • `DownVote`—User downvoted a post or comment. • `FeedEntityRead`—User read a post. • `FeedRead`—User read a feed. • `Mute`—User muted a post. • `Post`—User made a post. • `TopicEndorsement`—User endorsed another user on a topic or received endorsement on a topic. • `UpVote`—User upvoted a post or comment.	42.0
`activityUrl`	String	URL to the user activity.	42.0
`community`	`ConnectApi.CommunitySummary`	Experience Cloud site in which the user performed the activity.	42.0

SEE ALSO:

ConnectApi.UserActivityCollection

ConnectApi.UserCapabilities

Capabilities associated with a user profile.

Name	Type	Description	Available Version
`canChat`	Boolean	Specifies if the context user can use Chatter Messenger with the subject user (`true`) or not (`false`)	29.0
`canDirectMessage`	Boolean	Specifies if the context user can direct message the subject user (`true`) or not (`false`)	29.0
`canEdit`	Boolean	Specifies if the context user can edit the subject user's account (`true`) or not (`false`)	29.0
`canFollow`	Boolean	Specifies if the context user can follow the subject user's feed (`true`) or not (`false`)	29.0
`canViewFeed`	Boolean	Specifies if the context user can view the feed of the subject user (`true`) or not (`false`)	29.0
`canViewFullProfile`	Boolean	Specifies if the context user can view the full profile of the subject user (`true`) or only the limited profile (`false`)	29.0
`isModerator`	Boolean	Specifies if the subject user is a Chatter moderator or admin (`true`) or not (`false`)	29.0

SEE ALSO:

ConnectApi.UserProfile

ConnectApi.UserChatterSettings

User's global Chatter settings.

Name	Type	Description	Available Version
`defaultGroupEmailFrequency`	`ConnectApi.GroupEmailFrequency` Enum	The default frequency with which a user receives email from a group when they join it.	28.0

ConnectApi.UserDetail

Details about a user in an org.

Subclass of ConnectApi.User.

If the context user doesn't have permission to see a property, its value is set to `null`.

Name	Type	Description	Available Version
`aboutMe`	String	Text from user's profile.	28.0

Name	Type	Description	Available Version
address	ConnectApi.Address	User's address.	28.0
bannerPhoto	ConnectApi.BannerPhoto	User's banner photo.	36.0
chatterActivity	ConnectApi.ChatterActivity	Chatter activity statistics.	28.0
chatterInfluence	ConnectApi.GlobalInfluence	User's influence rank.	28.0
email	String	User's email address.	28.0
followersCount	Integer	Number of users following this user.	28.0
followingCounts	ConnectApi.FollowingCounts	Information about items the user is following.	28.0
groupCount	Integer	Number of groups user is following.	28.0
hasChatter	Boolean	`true` if user has access to Chatter; `false` otherwise.	31.0
isActive	Boolean	`true` if user is active; `false` otherwise.	28.0
managerId	String	18-character ID of the user's manager.	28.0
managerName	String	Locale-based concatenation of manager's first and last names.	28.0
phoneNumbers	List<ConnectApi.PhoneNumber>	Collection of user's phone numbers.	28.0
thanksReceived	Integer	The number of times the user has been thanked.	29.0
username	String	Username of the user, such as *Admin@mycompany.com*.	28.0

SEE ALSO:

- ConnectApi.UserPage
- ConnectApi.UserProfile

ConnectApi.UserFeedEntityActivitySummary

User feed entity activity summary.

This class is abstract.

Subclass of ConnectApi.UserActivitySummary.

Superclass of:

- ConnectApi.BookmarkSummary
- ConnectApi.ChatterActivitySummary
- ConnectApi.CompanyVerifySummary

- ConnectApi.DownVoteSummary
- ConnectApi.FeedEntityReadSummary
- ConnectApi.LikeSummary
- ConnectApi.MuteSummary
- ConnectApi.UpVoteSummary

Property Name	Type	Description	Available Version
`feedEntityId`	String	ID of the feed entity.	42.0

ConnectApi.UserGroupDetailPage

A page of groups that a user is a member of.

Property Name	Type	Description	Available Version
`currentPageUrl`	String	URL to the current page.	45.0
`groups`	List<`ConnectApi.ChatterGroupDetail`>	Collection of groups that the user is a member of.	45.0
`nextPageUrl`	String	URL to the next page, or `null` if there is no next page.	45.0
`previousPageUrl`	String	URL to the previous page, or `null` if there is no previous page.	45.0
`total`	Integer	Total number of groups that the user is a member of.	45.0

ConnectApi.UserGroupPage

A paginated list of groups the context user is a member of.

Name	Type	Description	Available Version
`currentPageUrl`	String	Connect REST API URL identifying the current page.	28.0
`groups`	`List<ConnectApi.ChatterGroupSummary>`	List of groups.	28.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	28.0
`previousPageUrl`	String	Connect REST API URL identifying the previous page, or `null` if there isn't a previous page.	28.0
`total`	Integer	Total number of groups across all pages.	28.0

ConnectApi.UserMission

Mission details for a user.

Subclass of ConnectApi.AbstractUserMissionActivity.

Property Name	Type	Description	Available Version
`missionName`	String	Name of the mission.	46.0
`missionThreshold`	Integer	Threshold of the mission. When a user reaches this activity count, the mission is achieved.	46.0

ConnectApi.UserMissionActivitiesJob

User mission activities job.

Property Name	Type	Description	Available Version
`jobToken`	String	Token that identifies the mission user activities job.	45.0
`jobType`	String	Type of user activities job, either `export` or `purge`.	45.0
`message`	String	Message describing the status and expected outcome of the job. When the job completes, you receive an email with information about the Salesforce file that contains `ConnectApi.UserMissionActivityCollection`.	45.0

ConnectApi.UserMissionActivity

User activity associated with missions.

Subclass of ConnectApi.AbstractUserMissionActivity.

No additional properties.

ConnectApi.UserMissionActivityCollection

List of mission activities for a user.

Property Name	Type	Description	Available Version
`community`	`ConnectApi.CommunitySummary`	Experience Cloud site in which the user performed activities.	45.0
`userId`	String	ID of the user.	45.0
`userMissionActivities`	List<`ConnectApi.AbstractUserMissionActivity`>	List of mission activities performed by the user.	45.0

Property Name	Type	Description	Available Version
`userName`	String	Name of the user.	45.0

SEE ALSO:

ConnectApi.UserMissionActivitiesJob

ConnectApi.UserMissionActivityStatus

Status of mission activity for a user.

Property Name	Type	Description	Available Version
`message`	String	Success or error message.	45.0
`status`	String	Status of mission activity for a user.	45.0

ConnectApi.UserOauthInfo

User OAuth information.

Name	Type	Description	Available Version
`availableExternalEmailService`	`Connect.OauthProviderInfo`	The available OAuth service provider.	37.0
`isAuthenticated`	Boolean	Specifies whether the user is authenticated (`true`) or not (`false`).	37.0

ConnectApi.UserPage

Page of users.

Name	Type	Description	Available Version
`currentPageToken`	Integer	Token identifying the current page.	28.0
`currentPageUrl`	String	Connect REST API URL identifying the current page.	28.0
`nextPageToken`	Integer	Token identifying the next page, or `null` if there isn't a next page.	28.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	28.0
`previousPageToken`	Integer	Token identifying the previous page, or `null` if there isn't a previous page.	28.0
`previousPageUrl`	String	Connect REST API URL identifying the previous page, or `null` if there isn't a previous page.	28.0

Name	Type	Description	Available Version
users	List<ConnectApi.UserDetail>	List of user detail information. If the context user doesn't have permission to see a property, the property is set to null.	28.0

ConnectApi.UserProfile

Details necessary to render a view of a user profile.

Name	Type	Description	Available Version
capabilities	ConnectApi.UserCapabilities	The context user's capabilities specific to the subject user's profile.	29.0
id	String	The ID of the user attached to the profile.	29.0
tabs	List<ConnectApi.UserProfileTab>	The tabs visible to the context user specific to the subject user's profile.	29.0
url	String	The URL of the user's profile.	29.0
userDetail	ConnectApi.UserDetail	The details about the user attached to the profile.	29.0

ConnectApi.UserProfileTab

Information about a profile tab.

Name	Type	Descriptio	Available Version
id	String	The tab's unique identifier or 18–character ID	29.0
isDefault	Boolean	Specifies if the tab appears first when clicking the user profile (true) or not (false)	29.0
tabType	ConnectApi.UserProfileTabType Enum	Specifies the type of tab • CustomVisualForce—Tab that displays data from a Visualforce page. • CustomWeb—Tab that displays data from any external web-based application or web page. • Element—Tab that displays generic content inline. • Feed—Tab that displays the Chatter feed. • Overview—Tab that displays user details.	29.0

Name	Type	Descriptio	Available Version
`tabUrl`	String	The current tab's content URL (for non built-in tab types)	29.0

SEE ALSO:

ConnectApi.UserProfile

ConnectApi.UserReferencePage

A list of user references.

Property Name	Type	Description	Available Version
`currentPageUrl`	String	URL to the current page.	35.0
`nextPageUrl`	String	URL to the next page.	35.0
`previousPageUrl`	String	URL to the previous page.	35.0
`userCount`	Integer	Number of users in the collection.	35.0
`users`	List<`ConnectApi.Reference`>	A collection of user references.	35.0

SEE ALSO:

ConnectApi.CustomListAudienceCriteria

ConnectApi.UserSettings

Settings specific to a user.

Property	Type	Description	Available Version
`approvalPosts`	Boolean	User can approve workflows from Chatter posts.	28.0
`canAccessPersonalStreams`	Boolean	User can access personal stream feeds.	40.0
`canFollow`	Boolean	User can follow users and records.	28.0
`canModifyAllData`	Boolean	User has Modify all Data permission.	28.0
`canOwnGroups`	Boolean	User can own groups.	28.0
`canViewAllData`	Boolean	User has View all Data permission.	28.0
`canViewAllGroups`	Boolean	User has View all Groups permission.	28.0

Property	Type	Description	Available Version
`canViewAllUsers`	Boolean	User has View all Users permission.	28.0
`canViewCommunitySwitcher`	Boolean	User can see the site switcher menu.	34.0
`canViewFullUserProfile`	Boolean	User can see other user's Chatter profile.	28.0
`canViewPublicFiles`	Boolean	User can see all files that are public.	28.0
`currencySymbol`	String	Currency symbol to use for displaying currency values. Applicable only when the `ConnectApi.Features.multiCurrency` property is `false`.	28.0
`externalUser`	Boolean	User is a Chatter customer.	28.0
`fileSyncLimit`	Integer	Maximum number of files user can sync.	32.0
`fileSyncStorageLimit`	Integer	Maximum storage for synced files, in megabytes (MB).	29.0
`folderSyncLimit`	Integer	Maximum number of folders user can sync.	32.0
`hasAccessToInternalOrg`	Boolean	User is a member of the internal org.	28.0
`hasChatter`	Boolean	User has access to Chatter.	31.0
`hasFileSync`	Boolean	User has Sync Files permission.	28.0
`hasFieldServiceLocationTracking`	Boolean	User has Field Service GPS tracking enabled.	41.0
`hasFieldServiceMobileAccess`	Boolean	User has access to the Field Service mobile app.	41.0
`hasFileSyncManagedClientAutoUpdate`	Boolean	Administrator for the user's org allows file sync clients to update automatically.	34.0
`hasRestDataApiAccess`	Boolean	User has access to REST API.	29.0
`timeZone`	`ConnectApi.TimeZone`	The user's time zone as selected in the user's personal settings in Salesforce. This value does not reflect a device's current location.	30.0
`userDefaultCurrencyIsoCode`	String	The ISO code for the default currency. Applicable only when the `ConnectApi.Features.multiCurrency` property is `true`.	28.0
`userId`	String	18-character ID of the user.	28.0

Property	Type	Description	Available Version
`userLocale`	String	Locale of user.	28.0

SEE ALSO:

ConnectApi.OrganizationSettings

ConnectApi.UserSummary

User summary.

Subclass of ConnectApi.User.

Name	Type	Description	Available Version
`isActive`	Boolean	`true` if user is active; `false` otherwise.	28.0

SEE ALSO:

ConnectApi.ChatterConversation
ConnectApi.ChatterConversationSummary
ConnectApi.ChatterGroup
ConnectApi.ChatterLike
ConnectApi.DashboardComponentSnapshot
ConnectApi.DirectMessageMemberPage
ConnectApi.GroupMembershipRequest
ConnectApi.GroupMember
ConnectApi.FeedFavorite
ConnectApi.OriginCapability
ConnectApi.PlatformAction
ConnectApi.DirectMessageMemberPage
ConnectApi.DirectMessageMemberActivity
ConnectApi.ChatterMessage
ConnectApi.Comment
ConnectApi.File
ConnectApi.MentionSegment
ConnectApi.QuestionAndAnswersCapability
ConnectApi.SocialPostCapability
ConnectApi.TopicEndorsement

ConnectApi.VerifiedCapability

If a comment has this capability, users with permission can mark it as verified or unverified.

Subclass of ConnectApi.FeedElementCapability.

Property Name	Type	Description	Available Version
`isVerifiableByMe`	Boolean	Specifies whether the context user has permission to mark comments as verified or unverified (`true`) or not (`false`).	41.0
`isVerified`	Boolean	Specifies whether the comment is marked as verified (`true`) or not (`false`).	41.0
`isVerifiedBy Anonymized`	Boolean	Specifies whether the comment is marked as verified by an anonymous user (`true`) or not (`false`). If the comment has never been marked as verified or unverified, `null`. Also `null` if the context user doesn't have permission to mark comments as verified or unverified.	43.0
`lastVerifiedByUser`	`ConnectApi.UserSummary`	User who last marked the comment as verified or unverified, otherwise `null`. Also `null` if the context user doesn't have permission to mark comments as verified or unverified.	41.0
`lastVerifiedDate`	Datetime	Date when the comment was last marked as verified or unverified, otherwise `null`. Also `null` if the context user doesn't have permission to mark comments as verified or unverified.	41.0

SEE ALSO:

ConnectApi.CommentCapabilities

ConnectApi.Vote

An upvote or downvote on a feed element or comment.

Property Name	Type	Description	Available Version
`type`	`ConnectApi. UpDownVoteValue`	Type of vote for a feed element or comment. • `Down` • `Up`	42.0
`user`	`ConnectApi. UserSummary`	User who voted on the feed element or comment.	42.0

Property Name	Type	Description	Available Version
votedItem	ConnectApi.Reference	Reference to the feed element or comment that was voted on.	42.0

SEE ALSO:

ConnectApi.VotePage

ConnectApi.VotePage

A page of upvotes or downvotes on a feed element or comment.

Property Name	Type	Description	Available Version
currentPageToken	Integer	Token identifying the current page.	42.0
currentPageUrl	String	Connect REST API URL identifying the current page.	42.0
items	List<ConnectApi.Vote>	Collection of users and their upvotes or downvotes. Upvotes include likes and upvotes. For example, if a post receives five likes and three upvotes, the number of upvotes is eight. For this reason, the collection of users and their upvotes also includes users who liked the post or comment. If a user both liked and upvoted a post, they appear only once in the collection.	42.0
nextPageToken	Integer	Token identifying the next page, or null if there isn't a next page.	42.0
nextPageUrl	String	Connect REST API URL identifying the next page, or null if there isn't a next page.	42.0
previousPageToken	Integer	Token identifying the previous page, or null if there isn't a previous page.	42.0
previousPageUrl	String	Connect REST API URL identifying the previous page, or null if there isn't a previous page.	42.0
total	Long	Total number of upvotes or downvotes for the feed element or comment. The number of upvotes includes the number of likes and upvotes. For example, if a post receives five likes and three upvotes, the total number of upvotes is eight. If a user both liked and upvoted a post, we count that as two upvotes.	42.0

ConnectApi.Wishlist

Wishlist, including summary and items.

Property Name	Type	Description	Available Version
page	ConnectApi.WishlistItemCollection	Page of wishlist items.	49.0
summary	ConnectApi.WishlistSummary	Summary of the wishlist.	49.0

SEE ALSO:

ConnectApi.WishlistsSummary

ConnectApi.WishlistItem

Item in a wishlist.

Property Name	Type	Description	Available Version
currencyIsoCode	String	Three-letter ISO 4217 currency code associated with the product.	49.0
error	ConnectApi.ErrorResponse	Error information.	49.0
listPrice	Double	List price of the wishlist item.	49.0
productSummary	ConnectApi.CartItemProduct	Product summary for the wishlist item.	49.0
salesPrice	Double	Sales price of the wishlist item.	49.0
wishlistItemId	String	ID of the wishlist item.	49.0

SEE ALSO:

ConnectApi.WishlistItemCollection

ConnectApi.WishlistItemCollection

Collection of wishlist items.

Property Name	Type	Description	Available Version
currencyIsoCode	String	Three-letter ISO 4217 currency code associated with the product.	49.0
currentPageToken	String	Token identifying the current page.	49.0
currentPageUrl	String	Connect REST API URL identifying the current page.	49.0
hasErrors	Boolean	Specifies whether at least one of the results contains an error (`true`) or not (`false`).	49.0

Property Name	Type	Description	Available Version
items	List<ConnectApi.WishlistItem>	Collection of wishlist items.	49.0
nextPageToken	String	Token identifying the next page, or null if there isn't a next page.	49.0
nextPageUrl	String	Connect REST API URL identifying the next page, or null if there isn't a next page.	49.0
previousPageToken	String	Token identifying the previous page, or null if there isn't a previous page.	49.0
previousPageUrl	String	Connect REST API URL identifying the previous page, or null if there isn't a previous page.	49.0

SEE ALSO:

ConnectApi.Wishlist

ConnectApi.WishlistsSummary

List of wishlist summaries and the displayed list for the context user.

Property Name	Type	Description	Available Version
displayedList	ConnectApi.Wishlist	Oldest wishlist displayed for the context user.	49.0
summaries	List<ConnectApi.WishlistSummary>	Summary of wishlists belonging to the context user.	49.0
wishlistCount	Integer	Total number of wishlists belonging to the context user.	49.0

ConnectApi.WishlistSummary

Summary of a wishlist.

Property Name	Type	Description	Available Version
createdDate	Datetime	Created date for the wishlist in ISO 8601 format, for example, 2011-02-25T18:24:31.000Z.	49.0
id	String	ID of the wishlist.	49.0
modifiedDate	Datetime	Last modified date of the wishlist in ISO 8601 format, for example, 2011-02-25T18:24:31.000Z.	49.0
name	String	Name of the wishlist.	49.0

Property Name	Type	Description	Available Version
`wishlistProductCount`	Integer	Unique product count in the wishlist.	49.0

SEE ALSO:

- ConnectApi.Wishlist
- ConnectApi.WishlistsSummary

ConnectApi.WishlistToCartResult

Result of adding a wishlist to a cart.

Property Name	Type	Description	Available Version
`cartId`	String	ID of the cart to which the products were added.	49.0
`failedWishlistToCartItems`	List<`ConnectApi.CartItemResult`>	Wishlist items that weren't successfully added to the cart.	49.0
`productsFailedCount`	Integer	Total number of products that weren't added to the cart.	49.0
`productsRequestedCount`	Integer	Total number of products requested to add to the cart.	49.0
`productsSucceededCount`	Integer	Total number of products that were successfully added to the cart.	49.0
`succeededWishlistToCartItems`	List<`ConnectApi.CartItemResult`>	Wishlist items that were successfully added to the cart.	49.0

ConnectApi.WrappedMapObject

Map of a parameter name and value.

Property Name	Type	Description	Available Version
`wrappedMap`	Map<String, Object>	Map of parameter name and value.	60.0

ConnectApi.Zone

Information about a Chatter Answers zone.

Name	Type	Description	Available Version
`description`	String	The description of the zone.	29.0
`id`	String	The zone ID.	29.0

Name	Type	Description	Available Version
`isActive`	Boolean	Indicates whether or not the zone is active.	29.0
`isChatterAnswers`	Boolean	Indicates whether or not the zone is available for Chatter Answers.	29.0
`name`	String	Name of the zone.	29.0
`url`	String	The URL of the zone.	30.0
`visibility`	`ConnectApi.ZoneShowIn`	Zone visibility type. • `Community`—Available in an Experience Cloud site. • `Internal`—Available internally only. • `Portal`—Available in a portal.	29.0
`visibilityId`	String	If the zone is available in a site, this property contains the ID of the site. If the zone is available to all sites, this property contains the value `All`.	29.0

SEE ALSO:

ConnectApi.ZonePage

ConnectApi.ZonePage

Page of zones.

Name	Type	Description	Available Version
`zones`	`List<ConnectApi.Zone>`	A list of one or more zones.	29.0
`currentPageUrl`	String	Connect REST API URL identifying the current page.	29.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	29.0

ConnectApi.ZoneSearchPage

Page of zone search results.

Name	Type	Description	Available Version
`currentPageToken`	String	Token identifying the current page.	29.0

Name	Type	Description	Available Version
`currentPageUrl`	String	Connect REST API URL identifying the current page.	29.0
`items`	`List<ConnectApi.ZoneSearchResult>`	List of search results.	29.0
`nextPageToken`	String	Token identifying the next page, or `null` if there isn't a next page.	29.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	29.0

ConnectApi.ZoneSearchResult

Information about a specific zone search result.

Name	Type	Description	Available Version
`hasBestAnswer`	Boolean	Indicates if the search result has a best answer.	29.0
`id`	String	ID of the search result. The search result can be a question or an article.	29.0
`title`	String	Title of the search result.	29.0
`type`	`ConnectApi.ZoneSearchResultType` Enum	Specifies the zone search result type. • `Article`—Search results contain only articles. • `Question`—Search results contain only questions.	29.0
`voteCount`	String	Number of votes given to the search result.	29.0

SEE ALSO:

ConnectApi.ZoneSearchPage

Retired `ConnectApi` Output Classes

These `ConnectApi` output classes are retired.

IN THIS SECTION:

ConnectApi.ApprovalAttachment
Attach an approval to a feed item.

ConnectApi.BasicTemplateAttachment
Attachments in feed items with type `BasicTemplate`.

ConnectApi.CanvasTemplateAttachment

Attachments in feed items with type `CanvasPost`.

ConnectApi.CaseComment

Attachments in feed items with type `CaseCommentPost`.

ConnectApi.ContentAttachment

Attachments in feed items with the type `ContentPost`.

ConnectApi.DashboardComponentAttachment

Attachments in feed items with type `DashboardSnapshot`.

ConnectApi.DatacloudCompany

Information about a Data.com company.

ConnectApi.DatacloudCompanies

Lists all companies that were purchased in a specific order, page URLs, and the number of companies in the order.

ConnectApi.DatacloudContact

Information about a Data.com contact.

ConnectApi.DatacloudContacts

Lists all contacts that were purchased in the specific order, page URLs, and the number of contacts in the order.

ConnectApi.DatacloudOrder

Represents a Datacloud order.

ConnectApi.DatacloudPurchaseUsage

Information about Data.com point usage for monthly and list pool users.

ConnectApi.EmailMessage

Email message from a case.

ConnectApi.FeedItemAttachment

Feed item attachment.

ConnectApi.FeedItemPage

A paged collection of `ConnectApi.FeedItem` objects.

ConnectApi.FeedItemTopicPage

Feed item topic page.

ConnectApi.FeedPoll

Attachment of `ConnectApi.FeedItem` objects where the `type` property is `PollPost`.

ConnectApi.LinkAttachment

Link attached to a feed item.

ConnectApi.NonEntityRecommendation

A recommendation for a non-Salesforce entity, such as an application.

ConnectApi.RecordSnapshotAttachment

Fields of a record at the point in time when the record was created.

ConnectApi.TrackedChangeAttachment

Tracked change attachment to a feed item.

ConnectApi.ApprovalAttachment

Attach an approval to a feed item.

Important: This class isn't available in version 32.0 and later. In version 32.0 and later, ConnectApi.ApprovalCapability is used.

Subclass of ConnectApi.FeedItemAttachment.

Name	Type	Description	Available Version
`id`	String	A work item ID.	28.0–31.0
`postTemplateFields`	`List<ConnectApi.ApprovalPostTemplateField>`	Collection of approval post template fields	28.0–31.0
`processInstanceStepId`	String	An approval step ID.	30.0–31.0
`status`	`ConnectApi.WorkflowProcessStatus` Enum	Status of a workflow process. • `Approved` • `Fault` • `Held` • `NoResponse` • `Pending` • `Reassigned` • `Rejected` • `Removed` • `Started`	28.0–31.0

ConnectApi.BasicTemplateAttachment

Attachments in feed items with type `BasicTemplate`.

Important: This class isn't available in version 32.0 and later. In version 32.0 and later, ConnectApi.EnhancedLinkCapability is used.

Subclass of ConnectApi.FeedItemAttachment.

Property	Type	Description	Available Version
`description`	String	An optional description with a 500 character limit.	28.0–31.0
`icon`	`ConnectApi.Icon`	An optional icon.	28.0–31.0
`linkRecordId`	String	If `linkURL` refers to a Salesforce record, `linkRecordId` contains the ID of the record.	28.0–31.0

Property	Type	Description	Available Version
`linkUrl`	String	An optional URL to a detail page if there is more content that can't be displayed inline. Do not specify `linkUrl` unless you specify a `title`.	28.0–31.0
`title`	String	An optional title to the detail page. If `linkUrl` is specified, the title links to `linkUrl`.	28.0–31.0

ConnectApi.CanvasTemplateAttachment

Attachments in feed items with type `CanvasPost`.

Important: This class isn't available in version 32.0 and later. In version 32.0 and later, ConnectApi.CanvasCapability is used.

Subclass of ConnectApi.FeedItemAttachment.

Property	Type	Description	Available Version
`description`	String	Optional. Description of the canvas app. The maximum length of this field is 500 characters.	29.0–31.0
`developerName`	String	Specifies the developer name (API name) of the canvas app.	29.0–31.0
`height`	String	Optional. The height of the canvas app in pixels. Default height is 200 pixels.	29.0–31.0
`icon`	ConnectApi.Icon	The canvas app icon.	29.0–31.0
`namespacePrefix`	String	Optional. The namespace prefix of the Developer Edition organization in which the canvas app was created.	29.0–31.0
`parameters`	String	Optional. Parameters passed to the canvas app in JSON format. Example: `{'isUpdated'='true'}`	29.0–31.0
`thumbnailUrl`	String	Optional. A URL to a thumbnail image for the canvas app. Maximum dimensions are 120x120 pixels.	29.0–31.0
`title`	String	Specifies the title of the link used to call the canvas app.	29.0–31.0

ConnectApi.CaseComment

Attachments in feed items with type `CaseCommentPost`.

Important: This class isn't available in version 32.0 and later. In version 32.0 and later, ConnectApi.CaseCommentCapability is used.

Subclass of ConnectApi.FeedItemAttachment.

Name	Type	Description	Available Version
`actorType`	`ConnectApi.CaseActorType` Enum	Type of user who made the comment. • `Customer`—if a Chatter customer made the comment • `CustomerService`—if a service representative made the comment	28.0–31.0
`createdBy`	`ConnectApi.UserSummary`	Comment's creator	28.0–31.0
`createdDate`	Datetime	ISO 8601 date string, for example, 2011-02-25T18:24:31.000Z	28.0–31.0
`id`	String	Comment's 18–character ID	28.0–31.0
`published`	Boolean	Specifies whether the comment has been published	28.0–31.0
`text`	String	Comment's text	28.0–31.0

ConnectApi.ContentAttachment

Attachments in feed items with the type `ContentPost`.

Important: This class isn't available in version 32.0 and later. In version 32.0 and later, ConnectApi.ContentCapability is used.

Subclass of ConnectApi.FeedItemAttachment.

Name	Type	Description	Available Version
`checkSum`	String	MD5 checksum for the file.	28.0–31.0
`contentUrl`	String	URL for link files and Google Docs; otherwise the value is `null`.	31.0–31.0
`description`	String	Description of the attachment.	28.0–31.0
`downloadUrl`	String	File's URL. This value is `null` if the content is a link or a Google Doc.	28.0–31.0
`fileExtension`	String	File's extension.	28.0–31.0
`fileSize`	String	Size of the file in bytes. If size cannot be determined, returns `unknown`.	28.0–31.0
`fileType`	String	Type of file.	28.0–31.0
`hasImagePreview`	Boolean	`true` if the file has a preview image available, otherwise ,`false`.	28.0–29.0
`hasPdfPreview`	Boolean	`true` if the file has a PDF preview available, otherwise, `false`.	28.0–31.0
`id`	String	Content's 18-character ID.	28.0–31.0
`isInMyFileSync`	Boolean	`true` if the file is synced with Salesforce Files Sync; `false` otherwise. **Note:** Salesforce Files Sync was retired on May 25, 2018.	28.0–31.0

Name	Type	Description	Available Version
`mimeType`	String	File's MIME type.	28.0–31.0
`renditionUrl`	String	URL to the file's rendition resource.	28.0–31.0
`renditionUrl240By180`	String	URL to the 240 x 180 rendition resource for the file.For shared files, renditions process asynchronously after upload. For private files, renditions process when the first file preview is requested, and aren't available immediately after the file is uploaded.	30.0–31.0
`renditionUrl720By480`	String	URL to the 720 x 480 rendition resource for the file.For shared files, renditions process asynchronously after upload. For private files, renditions process when the first file preview is requested, and aren't available immediately after the file is uploaded.	30.0–31.0
`textPreview`	String	Text preview of the file if available; `null` otherwise.	30.0–31.0
`thumb120By90RenditionStatus`	String	Specifies the rendering status of the 120 x 90 preview image of the file. One of these values: • `Processing`—Image is being rendered. • `Failed`—Rendering process failed. • `Success`—Rendering process was successful. • `Na`—Rendering is not available for this image.	30.0–31.0
`thumb240By180RenditionStatus`	String	Specifies the rendering status of the 240 x 180 preview image of the file. One of these values: • `Processing`—Image is being rendered. • `Failed`—Rendering process failed. • `Success`—Rendering process was successful. • `Na`—Rendering is not available for this image.	30.0–31.0
`thumb720By480RenditionStatus`	String	Specifies the rendering status of the 720 x 480 preview image of the file. One of these values: • `Processing`—Image is being rendered. • `Failed`—Rendering process failed. • `Success`—Rendering process was successful. • `Na`—Rendering is not available for this image.	30.0–31.0
`title`	String	Title of the file.	28.0–31.0
`versionId`	String	18-character ID for this version of the content.	28.0–31.0

ConnectApi.DashboardComponentAttachment

Attachments in feed items with type `DashboardSnapshot`.

Important: This class isn't available in version 32.0 and later. In version 32.0 and later, ConnectApi.DashboardComponentSnapshotCapability is used.

Subclass of ConnectApi.FeedItemAttachment.

Name	Type	Description	Available Version
componentId	String	Component's 18–character ID.	28.0–31.0
componentName	String	Name of the component. If no name is saved with the component, returns the localized string, "Untitled Component.".	28.0–31.0
dashboardBodyText	String	Text displayed next to the actor in the body of a feed item. This is used instead of the default body text. If no text is specified, and there is no default body text, returns null.	28.0–31.0
dashboardId	String	Dashboard's 18–character ID.	28.0–31.0
dashboardName	String	Name of the dashboard.	28.0–31.0
fullSizeImageUrl	String	URL of the full-sized dashboard image.	28.0–31.0
lastRefreshDate	Datetime	ISO8601 date string, for example, 2011-02-25T18:24:31.000Z, specifying when this dashboard was last refreshed.	28.0–31.0
lastRefreshDate DisplayText	String	The text of the last refresh date to be displayed, such as, "Last refreshed on October 31, 2011."	28.0–31.0
runningUser	ConnectApi. User Summary	The user running the dashboard.	28.0–31.0
thumbnailUrl	String	URL of the thumbnail-sized dashboard image.	28.0–31.0

ConnectApi.DatacloudCompany

Information about a Data.com company.

All company information is visible for companies that you purchased and own. If you haven't purchased a company, some of the fields are hidden. Hidden fields are fully or partially hidden by asterisks "***."

Property Name	Type	Description	Available Version
activeContacts	Integer	The number of active Data.com contacts who work in the company.	32.0
address	ConnectApi.Address	The postal address for the company. This is typically a physical address that can include the city, state, street, and postal code.	32.0
annualRevenue	Double	The amount of money that the company makes in one year. Annual revenue is measured in US dollars.	32.0
companyId	String	A unique numerical identifier for the company. This is the Data.com identifier for a company.	32.0

Property Name	Type	Description	Available Version
`description`	String	A brief synopsis of the company that provides a general overview of the company and what it does.	32.0
`dunsNumber`	String	A randomly generated nine-digit number that's assigned by Dun & Bradstreet (D&B) to identify unique business establishments.	32.0
`industry`	String	A description of the type of industry such as "Telecommunications," "Agriculture," or "Electronics."	32.0
`isInactive`	Boolean	Indicates whether this company is active (true) or not (false). Inactive companies have out-of-date information in Data.com.	32.0
`isOwned`	Boolean	• True: You or your organization owns this company. • False: Neither you nor your organization owns this company.	32.0
`naicsCode`	String	North American Industry Classification System (NAICS) codes were created to provide more details about a business's service orientation. The code descriptions are focused on what a business does.	32.0
`naicsDescription`	String	A description of the NAICS classification.	32.0
`name`	String	The name of the company.	32.0
`numberOf Employees`	Integer	The number of employees who are working for the company.	32.0
`ownership`	String	The type of ownership of the company: • Public • Private • Government • Other	32.0
`phoneNumbers`	ConnectApi.PhoneNumber	The list of telephone numbers for the company, including the type. Here are some possible types of telephone numbers. • Mobile • Work • Fax	32.0

Property Name	Type	Description	Available Version
`sic`	String	Standard Industrial Codes (SIC) is a numbering convention that indicates what type of service a business provides. It's a four-digit value.	32.0
`sicDescription`	String	A description of the SIC classification.	32.0
`site`	String	Company's site. For example, HQ, Single Location, or Branch. An organization status of the company. • Branch: a secondary location to a headquarter location. • Headquarter: the parent company has branches or subsidiaries. • Single Location: a single business with no subsidiaries or branches.	32.0
`tickerSymbol`	String	The symbol that uniquely identifies companies that are traded on public stock exchanges.	32.0
`tradeStyle`	String	A legal name under which a company conducts business.	32.0
`updatedDate`	Datetime	The date of the most recent change to this company's information.	32.0
`website`	String	The standard URL for the company's home page.	32.0
`yearStarted`	String	The year when the company was founded.	32.0

ConnectApi.DatacloudCompanies

Lists all companies that were purchased in a specific order, page URLs, and the number of companies in the order.

Property Name	Type	Description	Available Version
`companies`	ConnectApi.DatacloudCompany	A detailed list of companies that were part of a single order.	32.0
`currentPageUrl`	String	The URL for the current page of a list of companies.	32.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	32.0
`previousPageUrl`	String	The URL to the previous page of companies that were viewed before the current page. If this value is `null`, there's no previous page.	32.0

Property Name	Type	Description	Available Version
total	Integer	The number of companies in the order. You can calculate the number of pages to display by dividing this number by your page size. The default page size is 25.	32.0

ConnectApi.DatacloudContact

Information about a Data.com contact.

All contact information is visible for contacts that you purchased. If you have not purchased a contact, some of the fields will be hidden. Hidden fields are fully or partially hidden by asterisks "***."

Property Name	Type	Description	Available Version
address	ConnectApi.Address	The contact's business address.	32.0
companyId	String	A unique numerical identifier for the company where the contact works. This is the Data.com identifier for a company.	32.0
companyName	String	The company name where the contact works.	32.0
contactId	String	A unique numerical identifier for the contact. This is the Data.com identifier for a contact.	32.0
department	String	The department in the company where the contact works. Here are some possible departments. • Engineering • IT • Marketing • Sales	32.0
email	String	The most current business email address for the contact.	32.0
firstName	String	The first name of the contact.	32.0
isInactive	Boolean	Whether this contact is active (true) or not (false). Inactive contacts have out-of-date information in Data.com.	32.0
isOwned	Boolean	Whether this contact is owned (true) or not (false).	32.0
lastName	String	The last name of the contact.	32.0
level	String	A human resource label that designates a person's level in the company. Here are some possible levels. • C-Level	32.0

Property Name	Type	Description	Available Version
		• Director • Manager • Staff	
`phoneNumbers`	ConnectApi.PhoneNumber	Telephone numbers for the contact, which can include direct-dial business telephone numbers, mobile telephone numbers, and business headquarters telephone numbers. The type of telephone number is also indicated.	32.0
`title`	String	The title of the contact, such as CEO or Vice President.	32.0
`updatedDate`	Datetime	The date of the most recent change to this contact's information.	32.0

SEE ALSO:

ConnectApi.DatacloudContacts

ConnectApi.DatacloudContacts

Lists all contacts that were purchased in the specific order, page URLs, and the number of contacts in the order.

Property Name	Type	Description	Available Version
`contacts`	List<ConnectApi.DatacloudContact>	A detailed list of purchased contacts.	32.0
`currentPageUrl`	String	URL to the current page of contacts.	32.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	32.0
`previousPageUrl`	String	URL to the previous page of contacts. This value is null if there is no previous page.	32.0
`total`	Integer	Number of contacts that are associated with this order. Can be greater than the number of contacts that are shown on a single page.	32.0

ConnectApi.DatacloudOrder

Represents a Datacloud order.

Property Name	Type	Description	Available Version
`entityUrl`	String	URL to a list of contacts or companies that were purchased with this order.	32.0

Property Name	Type	Description	Available Version
`id`	String	Unique number that's used to track your order information.	32.0
`purchaseCount`	Integer	Number of contacts or companies that were purchased for this order.	32.0
`purchaseDate`	Datetime	Purchase date for this order.	32.0
`url`	String	GET request URL for this order.	32.0

ConnectApi.DatacloudPurchaseUsage

Information about Data.com point usage for monthly and list pool users.

Property Name	Type	Description	Available Version
`listpoolCreditsAvailable`	Integer	The points or credits that are available in a pool of credits for your organization. This pool of credits can be used by any List Pool User in your organization.	32.0
`listpoolCreditsUsed`	Integer	The points or credits that have been used from a pool of credits that are used by List Pool Users to purchase records.	32.0
`monthlyCreditsAvailable`	Integer	The total credits that are assigned to a Monthly User. Unused credits expire at the end of each month.	32.0
`monthlyCreditsUsed`	Integer	The credits that are used by a Monthly User for the current month.	32.0

ConnectApi.EmailMessage

Email message from a case.

Important: This class isn't available in version 32.0 and later. In version 32.0 and later, ConnectApi.EmailMessageCapability is used.

Subclass of ConnectApi.FeedItemAttachment.

Name	Type	Description	Available Version
`direction`	`ConnectApi.EmailMessageDirection` Enum	The direction of the email message. • `Inbound`—An inbound message (sent by a customer). • `Outbound`—An outbound message (sent to a customer by a support agent).	29.0–31.0
`emailMessageId`	String	The ID of the email message.	29.0–31.0

Name	Type	Description	Available Version
`subject`	String	The subject of the email message.	29.0–31.0
`textBody`	String	The body of the email message.	29.0–31.0
`toAddresses`	List<ConnectApi.EmailAddress>	A list of email addresses to send the message to.	29.0–31.0

ConnectApi.FeedItemAttachment

Feed item attachment.

Important: This class isn't available in version 32.0 and later. In version 32.0 and later, ConnectApi.FeedElementCapability is used.

This class is abstract.

Subclasses:

- ConnectApi.ApprovalAttachment
- ConnectApi.BasicTemplateAttachment
- ConnectApi.CanvasTemplateAttachment
- Connectapi.EmailMessage
- ConnectApi.CaseComment
- ConnectApi.ContentAttachment
- ConnectApi.DashboardComponentAttachment
- ConnectApi.FeedPoll
- ConnectApi.LinkAttachment
- ConnectApi.RecordSnapshotAttachment
- ConnectApi.TrackedChangeAttachment

Message segments in a feed item are typed as `ConnectApi.MessageSegment`. Feed item capabilities are typed as `ConnectApi.FeedItemCapability`. Record fields are typed as `ConnectApi.AbstractRecordField`. These classes are all abstract and have several concrete subclasses. At runtime you can use `instanceof` to check the concrete types of these objects and then safely proceed with the corresponding downcast. When you downcast, you must have a default case that handles unknown subclasses.

Important: The composition of a feed can change between releases. Write your code to handle instances of unknown subclasses.

ConnectApi.FeedItemPage

A paged collection of `ConnectApi.FeedItem` objects.

Important: This class isn't available in version 32.0 and later. In version 32.0 and later, ConnectApi.FeedElementPage is used.

Name	Type	Description	Available Version
`currentPageToken`	String	Token identifying the current page.	28.0–31.0
`currentPageUrl`	String	Connect REST API URL identifying the current page.	28.0–31.0

Name	Type	Description	Available Version
`isModifiedToken`	String	Opaque polling token to use in the *`since`* parameter of the `ChatterFeeds.isModified` method. The token describes when the feed was last modified. **Important:** This feature is available through a Feed Polling pilot program. This pilot program is closed and not accepting new participants.	28.0–31.0
`isModifiedUrl`	String	Connect REST API URL with a *`since`* request parameter that contains an opaque token that describes when the feed was last modified. Returns `null` if the feed isn't a news feed. Use this URL to poll a news feed for updates. **Important:** This feature is available through a Feed Polling pilot program. This pilot program is closed and not accepting new participants.	28.0–31.0
`items`	`List<ConnectApi.FeedItem>`	List of feed items	28.0–31.0
`nextPageToken`	String	Token identifying the next page, or `null` if there isn't a next page.	28.0–31.0
`nextPageUrl`	String	Connect REST API URL identifying the next page, or `null` if there isn't a next page.	28.0–31.0
`updatesToken`	String	Token to use in an `updatedSince` parameter, or `null` if not available.	30.0–31.0
`updatesUrl`	String	A Connect REST API resource with a query string containing the value of the `updatesToken` property. The resource returns the feed items that have been updated since the last request. Use the URL as it is—do not modify it. Property is `null` if not available.	30.0–31.0

ConnectApi.FeedItemTopicPage

Feed item topic page.

Important: This class isn't available in version 32.0 and later. In version 32.0 and later, ConnectApi.TopicsCapability is used.

Name	Type	Description	Available Version
`canAssignTopics`	Boolean	`true` if a topic can be assigned to the feed item, `false` otherwise.	28.0–31.0
`topics`	`List<ConnectApi.Topic>`	List of topics.	28.0–31.0

ConnectApi.FeedPoll

Attachment of `ConnectApi.FeedItem` objects where the `type` property is `PollPost`.

Important: This class isn't available in version 32.0 and later. In version 32.0 and later, ConnectApi.PollCapability is used.

Subclass of ConnectApi.FeedItemAttachment.

Name	Type	Description	Available Version
`choices`	`List<ConnectApi.FeedPollChoice>`	List of choices for poll.	28.0–31.0
`myChoiceId`	String	ID of the poll choice that the context user has voted for in this poll. Returns `null` if the context user hasn't voted.	28.0–31.0
`totalVoteCount`	Integer	Total number of votes cast on the feed poll item.	28.0–31.0

ConnectApi.LinkAttachment

Link attached to a feed item.

Important: This class isn't available in version 32.0 and later. In version 32.0 and later, ConnectApi.LinkCapability is used.

Subclass of ConnectApi.FeedItemAttachment.

Name	Type	Description	Available Version
`title`	String	Title given to the link if available, otherwise, `null`.	28.0–31.0
`url`	String	The link URL.	28.0–31.0

ConnectApi.NonEntityRecommendation

A recommendation for a non-Salesforce entity, such as an application.

Subclass of ConnectApi.AbstractRecommendation.

Important: ConnectApi.NonEntityRecommendation isn't used in version 34.0 and later. In version 34.0 and later, ConnectApi.EntityRecommendation is used for all recommendations.

Property Name	Type	Description	Available Version
`displayLabel`	String	Localized label of the non-entity object.	32.0
`motif`	`ConnectApi.Motif`	Motif for the non-entity object.	32.0

ConnectApi.RecordSnapshotAttachment

Fields of a record at the point in time when the record was created.

Important: This class isn't available in version 32.0 and later. In version 32.0 and later, ConnectApi.RecordSnapshotCapability is used.

Subclass of ConnectApi.FeedItemAttachment.

Name	Type	Description	Available Version
`recordView`	`ConnectApi.RecordView`	The representation of the record.	29.0–31.0

ConnectApi.TrackedChangeAttachment

Tracked change attachment to a feed item.

Important: This class isn't available in version 32.0 and later. In version 32.0 and later, ConnectApi.TrackedChangesCapability is used.

Name	Type	Description	Available Version
`changes`	`List<ConnectApi.TrackedChangeItem>`	A list of tracked changes.	28.0–31.0

`ConnectApi` Enums

Enums specific to the `ConnectApi` namespace.

`ConnectApi` enums inherit all properties and methods of Apex enums.

Enums are not versioned. Enum values are returned in all API versions. Clients should handle values they don't understand gracefully.

Enum	Description
`ConnectApi.ActionLinkExecutionsAllowed`	Number of times an action link can be executed. • `Once`—An action link can be executed only one time across all users. • `OncePerUser`—An action link can be executed only one time for each user. • `Unlimited`—An action link can be executed an unlimited number of times by each user. If the action link's `actionType` is `Api` or `ApiAsync`, you can't use this value.
`ConnectApi.ActionLinkType`	Type of action link. • `Api`—The action link calls a synchronous API at the action URL. Salesforce sets the status to `SuccessfulStatus` or `FailedStatus` based on the HTTP status code returned by your server. • `ApiAsync`—The action link calls an asynchronous API at the action URL. The action remains in a `PendingStatus` state until a third party makes a request to `/connect/action-links/`***`actionLinkId`*** to set the status to `SuccessfulStatus` or `FailedStatus` when the asynchronous operation is complete. • `Download`—The action link downloads a file from the action URL. • `Ui`—The action link takes the user to a web page at the action URL.

Enum	Description
`ConnectApi.ActivitySharingTypeEnum`	Type of sharing operation. • `Everyone`—The activity is shared with everyone. • `MyGroups`—The activity is shared only with a selection of the context user's groups. • `OnlyMe`—The activity is private.
`ConnectApi.AdjustmentAmountScope`	Scope of the price adjustment amount. • `Total`—The adjustment scope is the total price. • `Unit`—The adjustment scope is the unit price. • `UnproratedTotal`—The adjustment scope is the unprorated total price.
`ConnectApi.AdjustmentType`	How the price adjustment amount is calculated. • `AdjustmentAmount`—The adjustment is a fixed amount. • `AdjustmentPercentage`—The adjustment is a percentage.
`ConnectApi.ArticleTopicJobType`	Type of operation to perform on articles and topics. • `AssignTopicsToArticle`—Assign topics to articles in a data category. • `UnassignTopicsFromArticle`—Unassign topics from articles in a data category.
`ConnectApi.AsyncOperationStatus`	Asynchronous processing status of the cart, if asynchronous processing is enabled for the store. • `Completed` • `Errored` • `Processing`
`ConnectApi.AudienceCriteriaOperator`	Operator used in the personalization audience criterion. • `Contains` • `Equal` • `GreaterThan` • `GreaterThanOrEqual` • `Includes` • `LessThan` • `LessThanOrEqual` • `NotEqual` • `NotIncludes` • `StartsWith`
`ConnectApi.AudienceCriteriaType`	Type of personalization audience criterion. • `Audience`—Criterion based on audience. • `Default`—Audience has no criteria. • `Domain`—Criterion based on domain. • `FieldBased`—Criterion based on object fields.

Enum	Description
	• `GeoLocation`—Criterion based on location. • `Permission`—Criterion based on standard or custom permissions. • `Profile`—Criterion based on profile.
`ConnectApi.BannerStyle`	Decorates a feed item with a color and set of icons. • `Announcement`—An announcement displays in a designated location in the Salesforce UI until 11:59 p.m. on its expiration date, unless it's deleted or replaced by another announcement.
`ConnectApi.BillingFrequency`	Billing frequency for a subscription. • `Annual` • `Monthly`
`ConnectApi.BotVersion ActivationStatus`	Activation status of the bot version. • `Active` • `Inactive`
`ConnectApi.BundleType`	Type of bundle. • `GenericBundle`—A bundle that contains no additional information and is just a collection of feed elements. • `TrackedChanges`—A bundle that represents a collection of feed-tracked changes. The bundle includes summary information about the feed-tracked changes that make up the bundle.
`ConnectApi. CalculatedInsight DefinitionTypeEnum`	Definition type of the calculated insight. • `CALCULATED_METRIC` • `CALCULATED_METRIC` • `CALCULATED_METRIC`
`ConnectApi. CalculateTaxType`	Type of tax calculation. • `Actual`—Calculated tax represents the final taxed amount for the transaction. • `Estimated`—Calculated tax represents only an estimated value before the transaction is finalized.
`ConnectApi.CalloutStatus`	Indicates whether a named credential is enabled for callout. • `Disabled` • `Enabled`
`ConnectApi.CardCategory`	Indicates a credit card or debit card. • `CreditCard` • `DebitCard`

Enum	Description
`ConnectApi.CardType`	Credit card issuer. • `AmericanExpress` • `DinersClub` • `JCB` • `Maestro` • `MasterCard` • `Visa`
`ConnectApi.CartItemSortOrder`	Sort order for items in a cart. • `CreatedDateAsc`—Sorts by oldest creation date. • `CreatedDateDesc`—Sorts by most recent creation date. • `NameAsc`—Sorts by name in ascending alphabetical order (A–Z). • `NameDesc`—Sorts by name in descending alphabetical order (Z–A). • `SalesPriceAsc`—Sorts from lowest to highest negotiated price. • `SalesPriceDesc`—Sorts from highest to lowest negotiated price.
`ConnectApi.CartItemType`	Type of item in a cart. • `DeliveryCharge` • `Product`
`ConnectApi.CartMessageSeverity`	Severity of cart message. • `Error` • `Info` • `Warning`
`ConnectApi.CartPromotionType`	Level of the promotion target. • `Cart`—The target is cart-level. • `Item`—The target is item-level.
`ConnectApi.CartStatus`	Status of the cart. • `Active`—Cart is created and available for modifications, like adding or removing products or promotions. • `Checkout`—Cart is in checkout. If the customer modifies the cart, the current checkout session is canceled. • `Closed`—Checkout is complete and an order was created. The cart cannot be modified. • `PendingClosed`—Cart is marked to be closed, but the request isn't completed yet. The cart can't be modified. This value is available in API version 57.0 and later. • `PendingDelete`—Cart is marked for delete, but the request isn't completed yet. The cart can't be modified. • `Processing`—Cart is processing. For example, taxes are being calculated. The cart can't be modified.

Enum	Description
`ConnectApi.CartTaxType`	Tax type of the cart. • `Automatic`—Automatic taxation policy. • `Gross`—Gross taxation policy. • `Net`—Net taxation policy.
`ConnectApi.CartType`	Type of cart. • `Cart`—Cart created by a customer. • `PayNowReadOnly`—Clone of a Template cart that the customer can check out with using the Pay Now feature. • `Template`—Cart created by an internal user.
`ConnectApi.CaseActorType`	Type of user who made the comment. • `Customer`—if a Chatter customer made the comment • `CustomerService`—if a service representative made the comment
`ConnectApi.CaseCommentEventType`	Event type of a comment in Case Feed. • `NewInternal`—A case comment that has newly been marked Internal Only. • `NewPublished`—A newly published case comment. • `NewPublishedByCustomer`—A case comment by a customer that was newly published. • `PublishExisting`—An existing case comment that was republished. • `PublishExistingByCustomer`—An existing case comment by a customer that was republished. • `UnpublishExistingByCustomer`—An existing case comment by a customer that was unpublished. • `UnpublishExsiting`—An existing case comment that was unpublished. Note: Unfortunately, this typo is in the code, not the documentation. Use this spelling in your code.
`ConnectApi.CdpIdentityResolutionConfigurationType`	Source object for an identity resolution ruleset. • `Account` • `Individual`
`ConnectApi.CdpIdentityResolutionMatchMethodType`	Match method for a match rule criterion. • `Exact`—Exact match. • `ExactNormalized`—Exact normalized match. • `Fuzzy`—Fuzzy match with medium precision. • `FuzzyHigh`—Fuzzy match with high precision. • `FuzzyLow`—Fuzzy match with low precision.

Enum	Description
`ConnectApi.CdpIdentityResolutionReconciliationRuleType`	Default reconciliation rule applied to fields in the object the reconciliation rule applies to. • `LastUpdated` • `MostFrequent` • `SourceSequence`
`ConnectApi.CdpIdentityResolutionRunNowResultCode`	Result of an identity resolution ruleset job run. • `ExceededMaximumNumberOfSuccessfulRunsAllowedIn24Hours` • `IdentityResolutionJobIsAlreadyRunning` • `NoPendingChangesJobRunSkipped` • `SuccessfullySubmittedIdentityResolutionJobRunRequest`
`ConnectApi.CommentType`	Type of comment. • `ContentComment`—Comment holds a content capability. • `TextComment`—Comment contains only text.
`ConnectApi.CommerceAddressSort`	Sort order for Commerce addresses. • `CreatedDateAsc`—Sort in ascending order of created date. • `CreatedDateDesc`—Sort in descending order of created date. • `NameAsc`—Sort in ascending order of name. • `NameDesc`—Sort in descending order of name.
`ConnectApi.CommerceSearchAttributeType`	Search attribute type. • `Custom` • `ProductAttribute` • `ProductCategory` • `Product2` • `Standard`
`ConnectApi.CommerceSearchFacetDisplayType`	Display type of the facet. • `CategoryTree` • `DatePicker` • `MultiSelect` • `SingleSelect`
`ConnectApi.CommerceSearchFacetType`	Search facet type. • `DistinctValue`
`ConnectApi.CommerceSearchGroupingOption`	Grouping option for search results. • `BestMatch`—Search results are grouped by the best-match product of the variation group. • `NoGrouping`—Search results aren't grouped.

Enum	Description
	• `VariationParent`—Search results are grouped by the variation parent.
`ConnectApi.CommerceSearchIndexBuildType`	Build type of the index. • `Full` • `Incremental`
`ConnectApi.CommerceSearchIndexCreationType`	Creation type of the index. • `Manual` • `Scheduled`
`ConnectApi.CommerceSearchIndexStatus`	Status of the index. • `Completed` • `Failed` • `InProgress`
`ConnectApi.CommerceSearchIndexUsage`	Usage of the index. • `Live` • `OutOfUse`
`ConnectApi.CommerceSearchSortRuleDirection`	Direction of the sort rule. • `Ascending`—Sorts in ascending alphanumeric order (A–Z, 0–9). • `Default`—When no direction is defined, sorts by relevance. • `Descending`—Sorts in descending alphanumeric order (Z–A, 9–0).
`ConnectApi.CommerceSearchSortRuleLabelSuffix`	Label suffix of the sort rule. • `Ascen`—Label suffix for 'Asc' • `Ascending`—Label suffix for 'Ascending' • `Az`—Label suffix for 'A-Z' • `Descen`—Label suffix for 'Desc' • `Descending`—Label suffix for 'Descending' • `FewMany`—Label suffix for 'Few-Many' • `HeavyLight`—Label suffix for 'Heavy-Light' • `HighLow`—Label suffix for 'High-Low' • `HighestLowest`—Label suffix for 'Highest-Lowest' • `LightHeavy`—Label suffix for 'Light-Heavy' • `LowHigh`—Label suffix for 'Low-High' • `LowestHighest`—Label suffix for 'Lowest-Highest' • `ManyFew`—Label suffix for 'Many-Few' • `NewOld`—Label suffix for 'New-Old' • `Newest`—Label suffix for 'Newest'

Enum	Description
	• `NewestOldest`—Label suffix for 'Newest-Oldest' • `NineZero`—Label suffix for '9-0' • `OldNew`—Label suffix for 'Old-New' • `Oldest`—Label suffix for 'Oldest' • `OldestNewest`—Label suffix for 'Oldest-Newest' • `PriceDecreasing`—Label suffix for '$$-$' • `PriceIncreasing`—Label suffix for '$-$$' • `ThickThin`—Label suffix for 'Thick-Thin' • `ThinThick`—Label suffix for 'Thin-Thick' • `Za`—Label suffix for 'Z-A' • `ZeroNine`—Label suffix for '0-9'
`ConnectApi.CommerceSearchSortRuleType`	Type of sort rule. • `ProductAttributeBased`—Sorts by product attribute fields. • `ProductBased`—Sorts by product field data. • `Relevancy`—Sorts by product and catalog term frequency. • `SortByPricebook`—Sorts by product prices defined in the specified pricebook (version 55.0 and later).
`ConnectApi.CommerceSearchTopProductType`	Type of the top product to return for each product group in search results. • `VariationParent`
`ConnectApi.CommunityFlagReasonType`	Reason a post, comment, or file is flagged. • `FlaggedByRule`—Moderation rule flagged the item. • `FlaggedBySystem`—Einstein flagged the item. • `FlaggedByUserAsInappropriate`—User flagged the item as inappropriate. • `FlaggedByUserAsSpam`—User flagged the item as spam.
`ConnectApi.CommunityFlagType`	Type of moderation flag. • `FlagAsInappropriate`—Flag for inappropriate content. • `FlagAsSpam`—Flag for spam.
`ConnectApi.CommunityFlagVisibility`	Visibility behavior of a flag for various user types. • `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item. • `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.
`ConnectApi.CommunityStatus`	Status of the Experience Cloud site. • `Live` • `Inactive`

Enum	Description
	• `UnderConstruction`
`ConnectApi.ActivityType`	Type of activity. • `All` • `Event` • `Task`
`ConnectApi.ContentHub AuthenticationProtocol`	Authentication protocol used for the repository. • `NoAuthentication`—Repository doesn't require authentication. • `Oauth`—Repository uses OAuth authentication protocol. • `Password`—Repository uses user name and password authentication protocol.
`ConnectApi.ContentHub DirectoryEntryType`	Type of directory entry. • `GroupEntry` • `UserEntry`
`ConnectApi.ContentHub ExternalItemSharingType`	Sharing status for the external file. • `DomainSharing`—File is shared with the domain. • `PrivateSharing`—File is private or shared only with individuals. • `PublicSharing`—File is publicly shared.
`ConnectApi.ContentHub GroupType`	Type of group. • `Everybody`—Group is public to everybody. • `EverybodyInDomain`—Group is public to everybody in the same domain. • `Unknown`—Group type is unknown.
`ConnectApi.ContentHub ItemType`	Item types. • `Any`—Includes files and folders. • `FilesOnly`—Includes files only. • `FoldersOnly`—Includes folders only.
`ConnectApi.ContentHub StreamSupport`	Support for content streaming. • `ContentStreamAllowed` • `ContentStreamNotAllowed` • `ContentStreamRequired`
`ConnectApi.ContentHub VariableType`	Data type of the value of the field. • `BooleanType` • `DateTimeType` • `DecimalType` • `HtmlType`

Enum	Description
	• `IdType` • `IntegerType` • `StringType` • `UriType` • `XmlType`
`ConnectApi.ConversationApplicationIntegrationType`	Conversation application integration types. • `Api` • `Slack`
`ConnectApi.CreateCredentialAction`	Action to take when creating the credential. • `Refresh`
`ConnectApi.CredentialAuthenticationProtocol`	Authentication protocol of the external credential. • `AwsSv4` • `Basic` • `Custom` • `Jwt` • `OAuth`
`ConnectApi.CredentialAuthenticationProtocolVariant`	Authentication protocol variant of the external credential. • `AwsSv4_STS`—AWS Signature Version 4 with Security Token Service. • `ClientCredentialsClientSecret`—OAuth 2.0 Client Credentials client secret. Client secrets are sent in the callout's request body. • `ClientCredentialsClientSecretBasic`—OAuth 2.0 Client Credentials client secret. Client secrets are sent in the callout's authorization header, as with Basic authentication. • `ClientCredentialsJwtAssertion`—OAuth 2.0 Client Credentials JSON Web Token assertion. • `JwtBearer`—OAuth 2.0 JSON Web Token bearer flow. • `NoAuthentication`—No authentication. • `RolesAnywhere`—AWS Signature Version 4 with Identity and Access Management (IAM) Roles Anywhere.
`ConnectApi.CredentialAuthenticationStatus`	Status of the credential authentication. • `Configured`—Credential has all required credentials for at least one principal. • `NotConfigured`—Credential isn't configured. • `Unknown`—Credential status can't be determined because the authentication protocol is custom.
`ConnectApi.CredentialPrincipalType`	Type of credential principal. • `AwsStsPrincipal`

Enum	Description
	• `NamedPrincipal` • `PerUserPrincipal`
`ConnectApi.DataCategoryOperator`	Data category operator. • `Above`—Queries the data category and all of its parent categories. • `AboveOrBelow`—Queries the data category, all of its parent categories, and all of its subcategories. • `At`—Queries the data category. • `Below`—Queries the data category and all of its subcategories.
`ConnectApi.DatacloudUserType`	Type of user. • `Monthly`—A user type that's assigned monthly point limits for purchasing Data.com records. Only the assigned user can use monthly points. Points expire at the end of the month. Monthly is the default setting for `DatacloudUserType`. • `Listpool`—A user type that allows users to draw from a pool of points to purchase Data.com records.
`ConnectApi.DatacloudImportStatusTypeEnum`	Status of the import. • `Success`—Indicates that selected records were added to the org's CRM. • `Duplicate`—Marks a record that is already in the org's CRM. The API determines whether an org allows the addition of duplicate records in its CRM. • `Error`—Indicates that the selected records weren't added to the org's CRM.
`ConnectApi.DigestPeriod`	Time period that's included in a Chatter email digest. • `DailyDigest`—The email includes up to the 50 latest posts from the previous day. • `WeeklyDigest`—The email includes up to the 50 latest posts from the previous week.
`ConnectApi.EmailMessageAutomationType`	Automation type of the email message. • `aiAssisted`—The email message was created with the assistance of AI. • `aiAutomated`—The email message was created automatically by AI.
`ConnectApi.EmailMessageDirection`	Direction of an email message on a case. • `Inbound`—An inbound message (sent by a customer). • `Outbound`—An outbound message (sent to a customer by a support agent).
`ConnectApi.EmailMessageStatus`	Status of an email message on a case. • `DraftStatus` • `ForwardedStatus` • `NewStatus` • `ReadStatus` • `RepliedStatus` • `SentStatus`

Enum	Description
`ConnectApi.ExtensionInformationType`	Information type of the extension. • `Lightning`
`ConnectApi.ExternalAuthIdentityProviderParamType`	Parameter type for an external auth identity provider. • `AuthorizeRequestQueryParameter` • `IdentityProviderOptions` • `RefreshRequestBodyParameter` • `RefreshRequestHttpHeader` • `RefreshRequestQueryParameter` • `TokenRequestBodyParameter` • `TokenRequestHttpHeader` • `TokenRequestQueryParameter`
`ConnectApi.ExternalCredentialParameterType`	Parameter type of the external credential. • `AdditionalRefreshStatusCode` • `AuthParameter` • `AuthProvider` • `AuthProviderUrl` • `AuthProviderUrlQueryParameter` • `JwtBodyClaim` • `JwtHeaderClaim` • `SigningCertificate`
`ConnectApi.ExternalCredentialPrincipalAccessType`	Access type of the external credential principal. • `PermissionSet` • `PermissionSetGroup` • `Profile`
`ConnectApi.FeedCommentSortOrder`	Order of comments. • `CreatedDateLatestAsc`—Sorts by most recently created comments in ascending order. • `CreatedDateOldestAsc`—Sorts by oldest comments in ascending order. • `Relevance`—Sorts by most relevant content.
`ConnectApi.FeedDensity`	Density of the feed. • `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations. • `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. Also displays custom recommendations, but hides some system-generated updates from records.

Enum	Description
`ConnectApi.FeedElementCapabilityType`	Capabilities of a feed element in API versions 31.0 and later. If a capability exists on a feed element, the capability is available, even if the value doesn't exist or is `null`. If the capability doesn't exist, it isn't available. • `AssociatedActions`—The feed element includes information about actions associated with it. • `Approval`—The feed element includes information about an approval. • `Banner`—The body of the feed element has an icon and border. • `Bookmarks`—The context user can bookmark the feed element. Bookmarked feed elements are visible in the bookmarks feed. • `Bundle`—The feed element has a group of other feed elements that display as a bundle in the feed. The bundle type determines the additional data associated with the bundle. • `CallCollaboration`—The feed element has a recording comment. • `Canvas`—The feed element renders a canvas app. • `CaseComment`—The feed element has a case comment in the case feed. • `ChatterLikes`—The context user can like the feed element. • `Close`—The feed element can't be edited, commented on, or deleted. If the feed element is a poll, it can't be voted on. • `Comments`—The context user can add comments to the feed element. • `Content`—The feed element has a file. • `DashboardComponentSnapshot`—The feed element has a dashboard component snapshot. • `DirectMessage`—The feed element is a direct message. • `Edit`—Users who have permission can edit the feed element. • `EmailMessage`—The feed element has an email message from a case. • `EnhancedLink`—The feed element has a link that can contain supplemental information like an icon, a title, and a description. • `Extensions`—The feed element has one or more extension attachments. • `FeedEntityShare`—The feed element has a feed entity shared with it. • `Files`—The feed element has one or more file attachments. • `Interactions`—The feed element has information about user interactions. • `Link`—The feed element has a URL. • `MediaReferences`—The feed element has one or more media references. • `Moderation`—Users in an Experience Cloud site can flag the feed element for moderation. • `Mute`—The context user can mute the feed element. • `Origin`—A feed action created the feed element. • `Pin`—Users who have permission can pin the feed element. • `Poll`—The feed element has poll voting. • `QuestionAndAnswers`—The feed element has a question, and users can add answers to the feed element instead of comments. Users can also select the best answer. • `ReadBy`—The context user can mark the feed element as read.

Enum	Description
	• `Recommendations`—The feed element has a recommendation. • `Record`—The comment has a record attachment. • `RecordSnapshot`—The feed element has all the snapshot fields of a record for a single create record event. • `SocialPost`—The feed element can interact with a social post on a social network. • `Status`—The feed element has a status that determines its visibility. • `Topics`—The context user can add topics to the feed element. • `TrackedChanges`—The feed element has all changes to a record for a single tracked change event. • `UpDownVote`—Users can upvote or downvote the feed element. • `Verified`—Users who have permission can mark a comment as verified or unverified.
`ConnectApi.FeedElementType`	Feed elements are the top-level objects that a feed contains. The feed element type describes the characteristics of that feed element. • `Bundle`—A container of feed elements. A bundle also has a body made up of message segments that can always be gracefully degraded to text-only values. • `FeedItem`—A feed item has a single parent and is scoped to oneExperience Cloud site or across all Experience Cloud sites. A feed item can have capabilities such as bookmarks, canvas, content, comment, link, poll. Feed items have a body made up of message segments that can always be gracefully degraded to text-only values. • `Recommendation`—A recommendation is a feed element with a recommendations capability. A recommendation suggests records to follow, groups to join, or applications that are helpful to the context user.
`ConnectApi.FeedEntityStatus`	Status of the feed post or comment. • `Draft`—The feed post isn't published but is visible to the author and users with Modify All Data or View All Data permission. Comments can't be drafts. • `Isolated`—The feed post or comment is isolated, and only admins can see it. • `PendingReview`—The feed post or comment isn't approved yet and therefore isn't published or visible. • `Published`—The feed post or comment is approved and visible.
`ConnectApi.FeedFavoriteType`	Origin of the feed favorite. • `ListView` • `Search` • `Topic`
`ConnectApi.FeedFilter`	Filter value for a feed. • `AllQuestions`—Feed elements that are questions. • `AuthoredBy`—Feed elements authored by the user profile owner. This value is valid only for the `UserProfile` feed.

Enum	Description
	• `CommunityScoped`—Feed elements that are scoped to Experience Cloud sites. Currently, these feed elements have a User or a Group parent record. However, other parent record types could be scoped to sites in the future. Feed elements that are always visible in all sites are filtered out. This value is valid only for the `UserProfile` feed. • `QuestionsWithCandidateAnswers`—Feed elements that are questions that have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission. • `QuestionsWithCandidateAnswersReviewedPublished`—Feed elements that are questions that have candidate answers that have been reviewed or published. This value is valid only for users with the Access Einstein-Generated Answers permission. • `Read`—Feed elements that are older than 30 days or are marked as read for the context user. Includes existing feed elements when the context user joined the group. This value is valid only for the `Record` feed of a group. • `SolvedQuestions`—Feed elements that are questions and that have a best answer. • `UnansweredQuestions`—Feed elements that are questions and that don't have any answers. • `UnansweredQuestionsWithCandidateAnswers`—Feed elements that are questions that don't have answers but have candidate answers associated with them. This value is valid only for users with the Access Einstein-Generated Answers permission. • `Unread`—Feed elements that are created in the past 30 days and aren't marked as read for the context user. This value is valid only for the `Record` feed of a group. • `UnsolvedQuestions`—Feed elements that are questions and that don't have a best answer.
`ConnectApi.FeedItem AttachmentType`	Attachment type for feed item output objects. • `Approval`—A feed item requiring approval. • `BasicTemplate`—A feed item with a generic rendering of an image, link, and title. • `Canvas`—A feed item that contains the metadata to render a link to a canvas app. • `CaseComment`—A feed item created from a comment to a case record. • `CaseComment`—A feed item created from a comment to a case record. • `Content`—A feed item with a file attached. • `DashboardComponent`—A feed item with a dashboard attached. • `EmailMessage`—An email attached to a case record in Case Feed. • `Link`—A feed item with a URL attached. • `Poll`—A feed item with a poll attached. • `Question`—A feed item with a question attached. • `RecordSnapshot`—The feed item attachment contains a view of a record at a single `ConnectApi.FeedItemType.CreateRecordEvent`. • `TrackedChange`—All changes to a record for a single `ConnectApi.FeedItemType.TrackedChange` event.

Enum	Description
`ConnectApi.FeedItemType`	Type of feed item. • `ActivityEvent`—Feed item generated in Case Feed when an event or task associated with a parent record with a feed enabled is created or updated. • `AdvancedTextPost`—A feed item with advanced text formatting, such as a group announcement post. • `ApprovalPost`—Feed item with an approval capability. Approvers can act on the feed item parent. • `AttachArticleEvent`—Feed item generated when an article is attached to a case in Case Feed. • `BasicTemplateFeedItem`—Feed item with an enhanced link capability. • `CallLogPost`—Feed item generated when a call log is saved to a case in Case Feed. • `CanvasPost`—Feed item generated by a canvas app in the publisher or from Connect REST API or Connect in Apex. The post itself is a link to a canvas app. • `CaseCommentPost`—Feed item generated when a case comment is saved in Case Feed. • `ChangeStatusPost`—Feed item generated when the status of a case is changed in Case Feed. • `ChatTranscriptionPost`—Feed item generated in Case Feed when a Live Agent chat transcript is saved to a case. • `CollaborationGroupCreated`—Feed item generated when a new public group is created. Contains a link to the new group. • `CollaborationGroupUnarchived`—Deprecated. Feed item generated when an archived group is activated. • `ContentPost`—Feed item with a content capability. • `CreateRecordEvent`—Feed item that describes a record created in the publisher. • `DashboardComponentAlert`—Feed item with a dashboard alert. • `DashboardComponentSnapshot`—Feed item with a dashboard component snapshot capability. • `EmailMessageEvent`—Feed item generated when an email is sent from a case in Case Feed. • `FacebookPost`—Deprecated. Feed item generated when a Facebook post is created from a case in Case Feed. • `LinkPost`—Feed item with a link capability. • `MilestoneEvent`—Feed item generated when a case milestone is either completed or reaches a violation status. Contains a link to the case milestone. • `PollPost`—Feed item with a poll capability. Viewers of the feed item are allowed to vote on the options in the poll. • `ProfileSkillPost`—Feed item generated when a skill is added to a user's profile. • `QuestionPost`—Feed item generated when a question is asked. As of API version 33.0, a feed item of this type can have a content capability and a link capability. • `ReplyPost`—Feed item generated by a Chatter Answers reply.

Enum	Description
	• `RypplePost`—Feed item generated when a user posts thanks. • `SocialPost`—Feed item generated when a social post is created from a case in Case Feed. • `TextPost`—Feed item containing text only. • `TrackedChange`—Feed item created when one or more fields on a record have been changed. • `UserStatus`—Deprecated. A user's post to their own profile.
`ConnectApi.FeedItem VisibilityType`	Type of users who can see a feed item. • `AllUsers`—Visibility is not limited to internal users. • `InternalUsers`—Visibility is limited to internal users.
`ConnectApi. FeedSortOrder`	Order of feed items in the feed. • `CreatedDateAsc`—Sorts by oldest creation date. This sort order is available only for `DirectMessageModeration`, `Draft`, `Isolated`, `Moderation`, and `PendingReview` feeds. • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. • `MostViewed`—Sorts by most viewed content. This sort order is available only for `Home` feeds when the `ConnectApi.FeedFilter` is `UnansweredQuestions`. • `Relevance`—Sorts by most relevant content. This sort order is available only for `Company`, `Home`, and `Topics` feeds.
`ConnectApi.FeedType`	Type of feed. • `Bookmarks`—Contains all feed items saved as bookmarks by the context user. • `Company`—Contains all feed items except feed items of type `TrackedChange`. To see the feed item, the user must have sharing access to its parent. • `DirectMessageModeration`—Contains all direct messages that are flagged for moderation. The Direct Message Moderation feed is available only to users with Moderate Experiences Chatter Messages permissions. • `DirectMessages`—Contains all feed items of the context user's direct messages. • `Draft`—Contains all the feed items that the context user drafted. • `Files`—Contains all feed items that contain files posted by people or groups that the context user follows. • `Filter`—Contains the news feed filtered to contain feed items whose parent is a specified object type. • `Groups`—Contains all feed items from all groups the context user either owns or is a member of. • `Home`—Contains all feed items associated with any managed topic in an Experience Cloud site. • `Isolated`—Contains all the feed items and comments that are isolated.

Enum	Description
	• `Landing`—Contains all feed items that best drive user engagement when the feed is requested. Allows clients to avoid an empty feed when there aren't many personalized feed items. • `Moderation`—Contains all feed items that are flagged for moderation, except direct messages. The moderation feed is available only to users with Moderate Experiences Feeds permissions. • `Mute`—Contains all feed items that the context user muted. • `News`—Contains all updates for people the context user follows, groups the user is a member of, and files and records the user is following. Contains all updates for records whose parent is the context user. • `PendingReview`—Contains all feed items and comments that are pending review. • `People`—Contains all feed items posted by all people the context user follows. • `Record`—Contains all feed items whose parent is a specified record, which could be a group, user, object, file, or any other standard or custom object. When the record is a group, the feed also contains feed items that mention the group. When the record is a user, the feed contains only feed items on that user. You can get another user's record feed. • `Streams`—Contains all feed items for any combination of up to 25 feed-enabled entities that the context user subscribes to in a stream. Examples of feed-enabled entities include people, groups, and records, • `To`—Contains all feed items with mentions of the context user. Contains feed items the context user commented on and feed items created by the context user that are commented on. • `Topics`—Contains all feed items that include the specified topic. • `UserProfile`—Contains feed items created when a user changes records that can be tracked in a feed. Contains feed items whose parent is the user and feed items that @mention the user. This feed is different than the news feed, which returns more feed items, including group updates. You can get another user's user profile feed.
`ConnectApi.FieldChangeValueType`	Value type of a field change. • `NewValue`—A new value • `OldValue`—An old value
`ConnectApi.FieldType`	Field type. • `Address` • `AnyType` • `Base64` • `Boolean` • `Combobox` • `ComplexValue` • `Currency` • `DataCategoryGroupReference` • `Date`

Enum	Description
	• DateTime • Double • Email • EncryptedString • ExtensionEntityLookup • ExternalLookup • FloatArray • Id • ImageUrl • IndirectLookup • Integer • Json • Location • Long • MultiPicklist • Percent • PersonName • Phone • Picklist • PlainTextArea • Reference • RichTextArea • Sobject • String • SwitchablePersonName • TextArea • Time • Url
ConnectApi. FilePreviewFormat	Format of the file preview. • Jpg—Preview format is JPG. • Pdf—Preview format is PDF. • Svg—Preview format is compressed SVG. • Thumbnail—Preview format is 240 x 180 PNG. • ThumbnailBig—Preview format is 720 x 480 PNG. • ThumbnailTiny—Preview format is 120 x 90 PNG.
ConnectApi. FilePreviewStatus	Availability status of the file preview. • Available—Preview is available.

Enum	Description
	• `InProgress`—Preview is being processed. • `NotAvailable`—Preview is unavailable. • `NotScheduled`—Generation of the preview isn't scheduled yet.
`ConnectApi.FilePublishStatus`	Publish status of the file. • `PendingAccess`—File is pending publishing. • `PrivateAccess`—File is private. • `PublicAccess`—File is public.
`ConnectApi.FileSharingOption`	Sharing option of the file. • `Allowed`—Resharing of the file is allowed. • `Restricted`—Resharing of the file is restricted.
`ConnectApi.FileSharingPrivacy`	Sharing privacy of a file. • `None`—File is visible to anyone with record access. • `PrivateOnRecords`—File is private on records.
`ConnectApi.FileSharingType`	Sharing role of the file. • `Admin`—Owner permission, but doesn't own the file. • `Collaborator`—Viewer permission, and can edit, change permissions, and upload a new version of a file. • `Owner`—Collaborator permission, and can make a file private, and delete a file. • `Viewer`—Can view, download, and share a file. • `WorkspaceManaged`—Permission controlled by the library.
`ConnectApi.FilterOperator`	Filter operator. • `EqOp`—Equal • `ExcludesOp`—Excludes • `GtOp`—Greater than • `GteOp`—Greater than or equal • `InOp`—In • `IncludesOp`—Includes • `LikeOp`—Like • `LtOp`—Less than • `LteOp`—Less than or equal • `NeOp`—Not equal • `NinOp`—Not in
`ConnectApi.FolderItemType`	Type of item in a folder. • `file`

Enum	Description
	• `folder`
`ConnectApi.FormFieldType`	Type of marketing integration form field. • `Boolean` • `Date` • `EmailAddress` • `Number` • `Text`
`ConnectApi.FormulaFilterType`	Formula filter type for the personalization audience. • `AllCriteriaMatch`—All audience criteria are true (AND operation). • `AnyCriterionMatches`—Any audience criterion is true (OR operation). • `CustomLogicMatches`—Audience criteria match the custom formula (for example, (1 AND 2) OR 3).
`ConnectApi.GroupArchiveStatus`	Archive status of groups. • `All`—All groups, including groups that are archived and groups that aren't archived. • `Archived`—Groups that are archived. • `NotArchived`—Groups that aren't archived.
`ConnectApi.GroupEmailFrequency`	Frequency with which a user receives email. • `EachPost` • `DailyDigest` • `WeeklyDigest` • `Never` • `UseDefault`
`ConnectApi.GroupMembershipType`	Type of membership the user has with the group. • `GroupOwner` • `GroupManager` • `NotAMember` • `NotAMemberPrivateRequested` • `StandardMember`
`ConnectApi.GroupMembershipRequestStatus`	Status of a request to join a private group. • `Accepted` • `Declined` • `Pending`

Enum	Description
`ConnectApi.GroupViralInvitationsStatus`	Status of an invitation to join a group. • `ActedUponUser`—The user was added to the group. An email was sent asking the user to visit the group. • `Invited`—An email was sent asking the user to sign up for the org. • `MaxedOutUsers`—The group has the maximum allowed members. • `MultipleError`—The user wasn't invited due to multiple errors. • `NoActionNeededUser`—The user is already a member of the group. • `NotVisibleToExternalInviter`—The user is not accessible to the user sending the invitation. • `Unhandled`—The user couldn't be added to the group for an unknown reason.
`ConnectApi.GroupVisibilityType`	Group visibility type. • `PrivateAccess`—Only members of the group can see posts to this group. • `PublicAccess`—All users within the Experience Cloud site can see posts to this group. • `Unlisted`—Reserved for future use.
`ConnectApi.HttpRequestMethod`	HTTP method. • `HttpDelete`—Returns HTTP 204 on success. Response body or output class is empty. • `HttpGet`—Returns HTTP 200 on success. • `HttpHead`—Returns HTTP 200 on success. Response body or output class is empty. • `HttpPatch`—Returns HTTP 200 on success or HTTP 204 if the response body or output class is empty. • `HttpPost`—Returns HTTP 201 on success or HTTP 204 if the response body or output class is empty. Exceptions are the batch posting resources and methods, which return HTTP 200 on success. • `HttpPut`—Return HTTP 200 on success or HTTP 204 if the response body or output class is empty.
`ConnectApi.IdentityProviderAuthFlow`	Authentication flow to get tokens to call protected APIs. • `AuthorizationCode`
`ConnectApi.IdentityProviderAuthProtocol`	Authentication protocol required to access the external system. • `OAuth`
`ConnectApi.IdentityProviderClientAuth`	Client authentication method that describes how credentials are sent to the authorization server. • `ClientSecretBasic` • `ClientSecretPost`
`ConnectApi.LinkMetadataSource`	Source of the link metadata. • `None`—Link metadata wasn't retrieved. • `Sfdc`—Salesforce is the source.

Enum	Description
`ConnectApi.LinkMetadataType`	Type of link that the metadata represents. • `Error`—Link metadata couldn't be retrieved. • `Link`—Represents a link. • `None`—Link metadata wasn't retrieved because the link isn't an allowed domain. • `Photo`—Represents a photo. • `Rich`—Represents rich content, typically HTML content. • `Unknown`—Link metadata was retrieved, but the type is unknown. • `Video`—Represents a video.
`ConnectApi.MaintenanceType`	Type of maintenance. • `Downtime`—Downtime maintenance. • `GenerallyAvailable`—Generally available mode. • `MaintenanceAndAvailable`—Maintenance with available mode. • `MaintenanceWithDowntime`—Scheduled maintenance with downtime. • `ReadOnly`—Maintenance with read-only mode.
`ConnectApi.ManagedContentChannelType`	Type of managed content channel. • `CloudToCloud`—Cloud-to-Cloud integrated channel. • `Community`—Experience Cloud site channel. • `ConnectedApp`—Channel served by a connected app. • `PublicUnauthenticated`—Public channel. All published content is publicly available. • `UserPermission`—Channel backed by a system permission. All published content is available only to users with the permission.
`ConnectApi.ManagedContentCloneStatus`	Status of the managed content clone. • `PartialSuccess` • `Success`
`ConnectApi.ManagedContentMediaType`	Type of managed content media. • `Document` • `Image`
`ConnectApi.ManagedContentNodeType`	Type of managed content node. • `Date` • `DateTime` • `Media` • `MediaSource` • `MultilineText` • `NameField` • `RichText`

Enum	Description
	• `Text` • `Url`
`ConnectApi.ManagedContentSpaceChannelOperation`	Operation to perform on the channel and managed content space. • `Add`—Add a channel to a managed content space. • `Remove`—Remove a channel from a managed content space.
`ConnectApi.ManagedContentSpaceChannelStatus`	Status of the add or remove operation for a channel and managed content space. • `Added`—Channel was added to the managed content space. • `Failed`—Add or remove operation failed. • `Pending`—Add or remove operation is pending. • `Removed`—Channel was removed from the managed content space.
`ConnectApi.ManagedContentVariantStatus`	Status of the managed content variant. • `Draft`—Content isn't published. • `Published`—Content is published and available for use in your live sites. • `Revised`—Content that's published and edited. Publish this content to make the changes available for use in your live sites.
`ConnectApi.ManagedTopicType`	Type of managed topic. • `Content`—Topics that are associated with native content. • `Featured`—Topics that are featured, for example, on the Experience Cloud site home page, but don't provide overall navigation. • `Navigational`—Topics that display in a navigational menu in the Experience Cloud site.
`ConnectApi.MarkupType`	Type of rich text markup. • `Bold`—Bold tag. • `Code`—Code tag. • `Hyperlink`—Hyperlink anchor tag. • `Italic`—Italic tag. • `ListItem`—List item tag. • `OrderedList`—Ordered list tag. • `Paragraph`—Paragraph tag. • `Strikethrough`—Strikethrough tag. • `Underline`—Underline tag. • `UnorderedList`—Unordered list tag.
`ConnectApi.MCSFolderShareStatus`	Status of sharing a managed content space folder. • `PendingShare` • `PendingUnshare`

Enum	Description
	• `Shared`
`ConnectApi.MentionCompletionType`	Type of mention completion. • `All`—All mention completions, regardless of the type of record to which the mention refers. • `Group`—Mention completions for groups. • `User`—Mention completions for users.
`ConnectApi.MentionValidationStatus`	Type of validation error for a proposed mention, if any. • `Disallowed`—The proposed mention is invalid and is rejected because the context user is trying to mention something that is not allowed. For example, a user who is not a member of a private group is trying to mention the private group. • `Inaccessible`—The proposed mention is allowed, but the user or record being mentioned isn't notified. They don't have access to the parent record that's being discussed. • `Ok`—There is no validation error for this proposed mention.
`ConnectApi.MessageSegmentType`	Type of message segment, such as text, link, field change name, or field change value. • `EntityLink` • `FieldChange` • `FieldChangeName` • `FieldChangeValue` • `Hashtag` • `InlineImage` • `Link` • `MarkupBegin` • `MarkupEnd` • `Mention` • `MoreChanges` • `ResourceLink` • `Text`
`ConnectApi.NamedCredentialParameterType`	Type of named credential parameter. • `AllowedManagedPackageNamespaces` • `ClientCertificate`
`ConnectApi.NamedCredentialType`	Type of named credential. • `PrivateEndpoint` • `SecuredEndpoint`
`ConnectApi.NavigationMenuItemActionType`	Event, URL type, or modal navigation menu item. • `Event`—Event-based navigation.

Enum	Description
	Note: `Event` is internal only and can't be used in custom components. • `ExternalLink`—URL outside of your Experience Cloud site. • `InternalLink`—Relative URL inside your Experience Cloud site. • `Modal`—Modal, such as Account Switcher.
`ConnectApi.NavigationMenuItemOpenTarget`	Target for the navigation menu item. • `CurrentWindow`—Navigation menu item opens in the current window. • `NewWindow`—Navigation menu item opens in a new window.
`ConnectApi.NavigationMenuItemType`	Type of navigation menu item. • `DataSourceDriven`—Menu items dynamically added from a data source. • `Event`—Event, such as logging in, logging out, or switching accounts. • `ExternalLink`—URL outside of your site. • `GlobalAction`—Lets users create records that aren't related to other records. • `InternalLink`—Relative URL inside your site. • `MenuLabel`—Menu label. • `Modal`—Modal, such as Account Switcher. • `NavigationalTopic`—Dropdown list with links to the navigational topics in your site. • `SalesforceObject`—Objects such as accounts, cases, contacts, and custom objects. • `SystemLink`—System link, such as a link to Builder, Workspaces, or Setup.
`ConnectApi.NBAActionType`	Type of action. • `Flow`—Automated process tool with multiple subtypes.
`ConnectApi.NBAFlowType`	Type of recommended flow. • `AutoLaunchedFlow`—Autolaunched flow that runs in the background. • `Flow`—Screen flow that accepts user inputs.
`ConnectApi.NBATargetType`	Type of target. • `Recommendation`
`ConnectApi.OperationType`	Operation to carry out on the file. • `Add`—Adds the file to the feed element. • `Remove`—Removes the file from the feed element.
`ConnectApi.OrchestrationInstanceStatus`	Status of the orchestration instance. • `Canceled` • `Completed` • `Discontinued` • `Error`

Enum	Description
	• `InProgress` • `NotStarted` • `Suspended`
`ConnectApi.OrchestrationStepType`	Type of orchestration step. • `AsynchronousBackgroundStep` • `ApprovalStep` • `BackgroundStep` • `InteractiveStep` • `ManagedContentRoleInteractiveStep` • `ManagedContentVariantAutoPublishBackgroundStep` • `ManagedContentVariantAutoUnpublishBackgroundStep` • `ManagedContentVariantSetLockBackgroundStep` • `ManagedContentVariantSetReadyBackgroundStep` • `MuleSoftStep`
`ConnectApi.OrchestrationWorkItemStatus`	Status of the orchestration work item. • `Assigned` • `Completed`
`ConnectApi.OrderDeliveryGroupSummarySort`	Sort order for order delivery group summaries. • `IdAsc`—Sorts by ID in ascending alphanumeric order (A–Z, 0–9). • `IdDesc`—Sorts by ID in descending alphanumeric order (Z–A, 9–0).
`ConnectApi.OrderDirection`	Order direction. • `Ascending` • `Descending`
`ConnectApi.OrderItemSummarySort`	Sort order for order item summaries. • `IdAsc`—Sorts by ID in ascending alphanumeric order (A–Z, 0–9). • `IdDesc`—Sorts by ID in descending alphanumeric order (Z–A, 9–0).
`ConnectApi.OrderNulls`	Null value order. • `Firsts`—Null values are sorted first. • `Lasts`—Null values are sorted last.
`ConnectApi.OrderShipmentItemSort`	Sort order for order shipment items. • `IdAsc`—Sorts by ID in ascending alphanumeric order (A–Z, 0–9). • `IdDesc`—Sorts by ID in descending alphanumeric order (Z–A, 9–0).

Enum	Description
`ConnectApi.OrderShipmentSort`	Sort order for order shipments. • `ExpectedDeliveryDateAsc`—Sorts by the oldest expected delivery date. • `ExpectedDeliveryDateDesc`—Sorts by the most recent expected delivery date. • `ShipmentNumberAsc`—Sorts by shipment number in ascending order (0–9). • `ShipmentNumberDesc`—Sorts by shipment number in descending order (9–0).
`ConnectApi.OrderSummaryAdjustmentAggregatesStatus`	Order summary adjustment aggregate job status. • `Failed`—The adjustment aggregate data job for the order summary failed. • `InProgress`—The adjustment aggregate data job for the order summary is in progress. • `NotInitiated`—The adjustment aggregate data job for the order summary is not initiated. • `Submitted`—The adjustment aggregate data job for the order summary is submitted.
`ConnectApi.OrderSummaryAdjustmentTargetType`	Type of price adjustment in promotions. • `SplitLine`—Price adjustment on an order item. • `Header`—Price adjustment on the entire order.
`ConnectApi.OrderSummarySortOrder`	Sort order for order summaries. • `CreatedDateAsc`—Sorts by the oldest created date. • `CreatedDateDesc`—Sorts by the most recent created date. • `OrderedDateAsc`—Sorts by the oldest ordered date. • `OrderedDateDesc`—Sorts by the most recent ordered date.
`ConnectApi.PeriodType`	Time period used for forecasting. • `Month` • `Quarter` • `Week` • `Year`
`ConnectApi.PlatformActionGroupCategory`	Location of an action link group on an associated feed element. • `Primary`—The action link group is displayed in the body of the feed element. • `Overflow`—The action link group is displayed in the overflow menu of the feed element.
`ConnectApi.PlatformActionStatus`	Status of the action. • `FailedStatus`—The action link execution failed. • `NewStatus`—The action link is ready to be executed. Available for `Download` and `Ui` action links only. • `PendingStatus`—The action link is executing. Choosing this value triggers the API call for `Api` and `ApiAsync` action links. • `SuccessfulStatus`—The action link executed successfully.

Enum	Description
`ConnectApi.PlatformActionType`	Type of platform action. • `ActionLink`—An indicator on a feed element that targets an API, a web page, or a file, represented by a button in the Salesforce UI. • `CustomButton`—When clicked, opens a URL or a Visualforce page in a window or executes JavaScript. • `ProductivityAction`—Productivity actions are predefined and attached to a limited set of objects. Productivity actions include Send Email, Call, Map, View Website, and Read News. Except for the Call action, you can't edit productivity actions. • `QuickAction`—A global or object-specific action. • `StandardButton`—A predefined Salesforce button such as New, Edit, or Delete.
`ConnectApi.PriceAdjustmentTierType`	Type of price adjustment for the tier. • `AmountBasedAdjustment`—Price is adjusted by a specified amount. • `PercentageBasedAdjustment`—Price is adjusted by a specified percentage.
`ConnectApi.PricingTermUnit`	Unit of time used to define a pricing term. • `Months`—Product is priced on a monthly basis. • `Annual`—Product is priced on an annual basis.
`ConnectApi.ProductAttributeViewType`	View type for product attributes. • `ColorSwatch` • `Dropdown` • `Pill`
`ConnectApi.ProductClass`	Class of product. • `Bundle` • `Set` • `Simple` • `Variation` • `VariationParent`
`ConnectApi.ProductMediaType`	Type of product media. • `Document` • `Image` • `Video`
`ConnectApi.ProductMediaUsageType`	Usage type of a product media item within a media group. • `Attachment`—Product media group with product documents as attachments. • `Banner`—Product category media group with banner images of the product. • `Listing`—Product media group with listing images of the product. • `Standard`—Product media group with standard images and videos of the product.

Enum	Description
	• `Tile`—Product category media group with tile images of the product.
`ConnectApi.PublishSchedule`	Publish refresh schedule. • `One`—Refreshes every hour. Used to rapidly publish UI and DBT-based segments. • `Four`—Refreshes every four hours. Used to rapidly publish UI and DBT-based segments. • `Twelve`—Refreshes every twelve hours. • `TwentyFour`—Refreshes every twenty-four hours.
`ConnectApi.PublishStatus`	Publish status of a personalization audience, target, or navigation menu item. • `Draft` • `Live`
`ConnectApi.RecommendationActionType`	Action to take on a recommendation. • `follow`—Follow a file, record, topic, or user. • `join`—Join a group. • `view`—View a file, group, article, record, user, custom, or static recommendation.
`ConnectApi.RecommendationAudienceCriteriaType`	Custom recommendation audience criteria type. • `CustomList`—A custom list of users makes up the audience. • `MaxDaysInCommunity`—New members make up the audience.
`ConnectApi.RecommendationAudienceMemberOperationType`	Operation to carry out on the custom recommendation audience members. • `Add`—Adds specified members to the audience. • `Remove`—Removes specified members from the audience.
`ConnectApi.RecommendationChannel`	A way to tie custom recommendations together. For example, display recommendations in specific places in the UI or show recommendations based on time of day or geographic locations. • `CustomChannel1`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels. For example, community managers can use Experience Builder to determine where recommendations appear. • `CustomChannel2`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels. • `CustomChannel3`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels. • `CustomChannel4`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels. • `CustomChannel5`—Custom recommendation channel. Not used by default. Work with your community manager to define custom channels. • `DefaultChannel`—Default recommendation channel. Recommendations appear by default on the Home and Question Detail pages of Customer Service and Partner Central Experience Builder templates. They also appear in the feed in the Salesforce mobile web and anywhere community managers add recommendations using Experience Builder.

Enum	Description
`ConnectApi.RecommendationExplanationType`	Reason for a Chatter recommendation. • `ArticleHasRelatedContent`—Articles with related content to a context article. • `ArticleViewedTogether`—Articles often viewed together with the article that the context user just viewed. • `ArticleViewedTogetherWithViewers`—Articles often viewed together with other records that the context user views. • `Custom`—Custom recommendations. • `FilePopular`—Files with many followers or views. • `FileViewedTogether`—Files often viewed at the same time as other files that the context user views. • `FollowedTogetherWithFollowees`—Users often followed together with other records that the context user follows. • `GroupMembersFollowed`—Groups with members that the context user follows. • `GroupNew`—Recently created groups. • `GroupPopular`—Groups with many active members. • `ItemViewedTogether`—Records often viewed at the same time as other records that the context user views. • `PopularApp`—Applications that are popular. • `RecordOwned`—Records that the context user owns. • `RecordParentOfFollowed`—Parent records of records that the context user follows. • `RecordViewed`—Records that the context user recently viewed. • `TopicFollowedTogether`—Topics often followed together with the record that the context user just followed. • `TopicFollowedTogetherWithFollowees`—Topics often followed together with other records that the context user follows. • `TopicPopularFollowed`—Topics with many followers. • `TopicPopularLiked`—Topics on posts that have many likes. • `UserDirectReport`—Users who report to the context user. • `UserFollowedTogether`—Users often followed together with the record that the context user followed . • `UserFollowsSameUsers`—Users who follow the same users as the context user. • `UserManager`—The context user's manager. • `UserNew`—Recently created users. • `UserPeer`—Users who report to the same manager as the context user. • `UserPopular`—Users with many followers. • `UserViewingSameRecords`—Users who view the same records as the context user.
`ConnectApi.RecommendationReactionType`	Type of reaction to a recommendation. • `Accepted` • `Rejected`

Enum	Description
`ConnectApi.RecommendationType`	Type of record being recommended. • `apps` • `articles` • `files` • `groups` • `records` • `topics` • `users`
`ConnectApi.RecommendedObjectType`	Type of object being recommended. • `Today`—Static recommendations that don't have an ID, for example, the Today app recommendation.
`ConnectApi.RecordColumnOrder`	Order in which fields are rendered in a grid. • `LeftRight`—Fields are rendered from left to right. • `TopDown`—Fields are rendered from the top down.
`ConnectApi.RecordFieldType`	Data type of a record field. • `Address` • `Blank` • `Boolean` • `Compound` • `CreatedBy` • `Date` • `DateTime` • `Email` • `LastModifiedBy` • `Location` • `Name` • `Number` • `Percent` • `Phone` • `Picklist` • `Reference` • `Text` • `Time`
`ConnectApi.RelatedFeedPostType`	Type of related feed post. • `Answered`—Related questions that have at least one answer. • `BestAnswer`—Related questions that have a best answer.

Enum	Description
	• `Generic`—All types of related questions, including answered, with a best answer, and unanswered. • `Unanswered`—Related questions that don't have answers.
`ConnectApi.SearchBoostBuryRuleAction`	Action of the boost and bury rule. • `Boost`—Boost rule. Increases search result rankings for targeted products. • `Bury`—Bury rule. Decreases search result rankings for targeted products.
`ConnectApi.SearchBoostBuryRuleOperation`	Operation for the conditions of the target expression in the boost and bury rule. • `AllOf`—All-of operation. • `AnyOf`—Any-of operation.
`ConnectApi.SegmentType`	Type of segment. • `Dbt`—Data build tool
`ConnectApi.SellingModelType`	Type of product selling model. • `Evergreen`—A subscription without an end date. An evergreen subscription continues until the shopper affirmatively cancels it. • `OneTime`—A product that isn't sold as a subscription. • `TermDefined`—A subscription with a defined end date. The subscription continues for a specified time period. When the term ends, the subscription ends.
`ConnectApi.SitesPageType`	Type of site search result item. • `ContentPage` • `SitePage`
`ConnectApi.SocialNetworkProvider`	Social network provider. • `Facebook` • `GooglePlus` • `Instagram` • `InstagramBusiness` • `KakaoTalk` • `Kik` • `Line` • `LinkedIn` • `Messenger` • `Other` • `Pinterest` • `QQ` • `Rypple` • `SinaWeibo`

Enum	Description
	• SMS • Snapchat • Telegram • Twitter • VKontakte • WeChat • WhatsApp • YouTube
ConnectApi.SocialPostMessageType	Message type of the social post. • Comment • Direct • Post • PrivateMessage • Reply • Retweet • Tweet
ConnectApi.SocialPostStatusType	State of the social post. • ApprovalPending • ApprovalRecalled • ApprovalRejected • Deleted • Failed • Hidden • Pending • Sent • Unknown
ConnectApi.SortOrder	Order for sorting. • Ascending—Items are in ascending alphabetical order (A-Z). • Descending—Items are in descending alphabetical order (Z-A). • MostRecentlyViewed—Items are in descending chronological order by view. This sort order is valid only for Chatter feed streams.
ConnectApi.SurveyEmailStatusEnum	Status of a survey invitation email. • Failed—The survey invitation email wasn't sent. • Queued—The survey invitation email is queued for sending.

Enum	Description
`ConnectApi.SvcApptModeEnum`	Mode of the service appointment. • `Group`— Service appointment mode is Group. • `Regular`— Default mode of service appointment.
`ConnectApi.TaxTransactionStatus`	Status of a tax transaction. • `Committed`—Tax has been committed to the transaction. • `Uncommitted`—Tax hasn't been committed to the transaction.
`ConnectApi.TaxTransactionType`	Type of tax transaction. • `Credit`—Transaction is a credit transaction. • `Debit`—Transaction is a debit transaction.
`ConnectApi.TopicSort`	Order returned by the sort. • `popularDesc`—Sorts topics by popularity with the most popular first. This value is the default. • `alphaAsc`—Sorts topics alphabetically.
`ConnectApi.UpDownVoteValue`	Type of vote for a feed element or comment. • `Down` • `None` • `Up`
`ConnectApi.UserActivityType`	Type of user activity. • `Bookmark`—User bookmarked a post. • `ChatterActivity`—Total counts of posts and comments made and likes and comments received for a user. • `ChatterLike`—User liked a post or comment. • `Comment`—User commented on a post. • `CompanyVerify`—User verified comment. • `DownVote`—User downvoted a post or comment. • `FeedEntityRead`—User read a post. • `FeedRead`—User read a feed. • `Mute`—User muted a post. • `Post`—User made a post. • `TopicEndorsement`—User endorsed another user on a topic or received endorsement on a topic. • `UpVote`—User upvoted a post or comment.
`ConnectApi.UserMissionActivityType`	Type of mission activity for a user. • `FeedItemAnswerAQuestion`—User answered a question. • `FeedItemLikeSomething`—User liked a post or comment.

Enum	Description
	• `FeedItemMarkAnswerAsBest`—User marked an answer as the best answer. • `FeedItemPostQuestion`—User posted a question. • `FeedItemReceiveAComment`—User received a comment on a post. • `FeedItemReceiveALike`—User received a like on a post or comment. • `FeedItemReceiveAnAnswer`—User received an answer to a question. • `FeedItemWriteAComment`—User commented on a post. • `FeedItemWriteAPost`—User made a post. • `FeedItemYourAnswerMarkedBest`—User's answer was marked as the best answer.
`ConnectApi.UserProfileTabType`	Type of user profile tab. • `CustomVisualForce`—Tab that displays data from a Visualforce page. • `CustomWeb`—Tab that displays data from any external web-based application or web page. • `Element`—Tab that displays generic content inline. • `Feed`—Tab that displays the Chatter feed. • `Overview`—Tab that displays user details.
`ConnectApi.UserType`	Type of user. • `ChatterGuest`—User is an external user in a private group. • `ChatterOnly`—User is a Chatter Free customer. • `Guest`—User is unauthenticated. • `Internal`—User is a standard org member. • `Portal`—User is an external user in an Experience Cloud site. • `System`—User is Chatter Expert or a system user. • `Undefined`—User is a user type that is a custom object.
`ConnectApi.WishlistItemSortOrder`	Sort order for wishlist items. • `CreatedDateAsc`—Sorts by oldest creation date. • `CreatedDateDesc`—Sorts by most recent creation date.
`ConnectApi.WorkflowProcessStatus`	Status of a workflow process. • `Approved` • `Fault` • `Held` • `NoResponse` • `Pending` • `Reassigned` • `Rejected` • `Removed` • `Started`

Enum	Description
`ConnectApi.ZoneSearchResultType`	Zone search result type. • `Article`—Search results contain only articles. • `Question`—Search results contain only questions.
`ConnectApi.ZoneShowIn`	Zone search result location. • `Community`—Available in an Experience Cloud site. • `Internal`—Available internally only. • `Portal`—Available in a portal.

`ConnectApi` Exceptions

The `ConnectApi` namespace contains exception classes.

All exceptions classes support built-in methods for returning the error message and exception type. See Exception Class and Built-In Exceptions on page 3499.

The `ConnectApi` namespace contains these exceptions:

Exception	Description
`ConnectApi.ConnectApiException`	Any logic error in the way your application is utilizing `ConnectApi` code. This is equivalent to receiving a 400 error from Connect REST API.
`ConnectApi.NotFoundException`	Any issues with the specified resource being found. This is equivalent to receiving a 404 error from Connect REST API.
`ConnectApi.RateLimitException`	When you exceed the rate limit. This is equivalent to receiving a 503 Service Unavailable error from Connect REST API.

`ConnectApi` Utilities

The `ConnectApi` namespace contains a utility class.

Utility	Description
`ConnectApi.ConnectUtilities.unwrapApexWrapper()`	Unwraps obfuscated, Apex-wrapped objects into known types such as `Map<String, Object>`. Example from Apex Debug log: `core.connect.apex.ApexMapWrapper@7270879d`

Example

This example calls `getManagedContentForSite(siteId, contentKeyOrId, showAbsoluteUrl)` to get a custom content type with an image reference and uses the `ConnectApi.ConnectUtilities.unwrapApexWrapper()` utility.

```
ConnectApi.ManagedContentDeliveryDocument res =
    ConnectApi.ManagedContentDelivery.getManagedContentForSite
('0DMXXXXXXXXXXXXXXX','MCLXXXXXXXXXXXXXXXXXXXXXXXXXX',true);

//before contentBody field ApexWrapper is unwrapped
system.debug(res.contentBody);

//unwrap contentBody field in res
Map<String,Object> contentBody =
(Map<String,Object>)ConnectApi.ConnectUtilities.unwrapApexWrapper(res.contentBody);

//after contentBody field ApexWrapper is unwrapped, but image field still wrapped
system.debug(contentBody);

//before image field ApexWrapper is unwrapped
system.debug(contentBody.get('Image'));

//unwrap Image field in contentBody
Map<String,Object> Image =
(Map<String,Object>)ConnectApi.ConnectUtilities.unwrapApexWrapper(contentBody.get('Image'));

//after image field ApexWrapper is unwrapped
system.debug(Image);

//replace wrapped primary_image in contentBody with unwrapped version
contentBody.put('Image', Image);

//after contentBody field ApexWrapper is unwrapped, with image field also unwrapped
system.debug(contentBody);
```

ConnectApi Release Notes

Use the Salesforce Release Notes to learn about the most recent updates and changes to the ConnectApi namespace in Apex.

For updates and changes that impact Apex, including ConnectApi, see the Apex Release Notes.

For new and changed ConnectApi classes and enums, see ConnectApi (Connect in Apex): New and Changed Classes and Enums in the Salesforce Release Notes.

Context Namespace

The `Context` namespace provides classes and methods to manage the sharing and consumption of business application data by using Context Service.

The `Context` namespace includes the IndustriesContext class.

Database Namespace

The `Database` namespace provides classes used with DML operations.

The following are the classes in the `Database` namespace.

IN THIS SECTION:

Batchable Interface
The class that implements this interface can be executed as a batch Apex job.

BatchableContext Interface
Represents the parameter type of a batch job method and contains the batch job ID. This interface is implemented internally by Apex.

Cursor Class (Beta)
Contains methods to fetch records and to get the number of cursor rows returned from a SOQL query.

DeletedRecord Class
Contains information about a deleted record.

DeleteFilter Enum
Determines if deleted rows are included in Apex cursor rows.

DeleteResult Class
Represents the result of a delete DML operation returned by the `Database.delete` method.

DMLOptions Class
Enables you to set options related to DML operations.

DmlOptions.AssignmentRuleHeader Class
Enables setting assignment rule options.

DMLOptions.DuplicateRuleHeader Class
Determines options for using duplicate rules to detect duplicate records. Duplicate rules are part of the Duplicate Management feature.

DmlOptions.EmailHeader Class
Enables setting email options.

DuplicateError Class
Contains information about an error that occurred when an attempt was made to save a duplicate record. Use if your organization has set up duplicate rules, which are part of the Duplicate Management feature.

EmptyRecycleBinResult Class
The result of the emptyRecycleBin DML operation returned by the `Database.emptyRecycleBin` method.

Error Class
Represents information about an error that occurred during a DML operation when using a Database method.

GetDeletedResult Class
Contains the deleted records retrieved for a specific sObject type and time window.

GetUpdatedResult Class
Contains the result for the `Database.getUpdated` method call.

LeadConvert Class
Contains information used for lead conversion.

LeadConvertResult Class
The result of a lead conversion.

MergeResult Class
Contains the result of a merge Database method operation.

QueryLocator Class
Represents the record set returned by `Database.getQueryLocator` and used with Batch Apex.

QueryLocatorIterator Class
Represents an iterator over a query locator record set.

SaveResult Class
The result of an insert or update DML operation returned by a Database method.

UndeleteResult Class
The result of an undelete DML operation returned by the `Database.undelete` method.

UpsertResult Class
The result of an upsert DML operation returned by the `Database.upsert` method.

Batchable Interface

The class that implements this interface can be executed as a batch Apex job.

Namespace

Database

SEE ALSO:

Apex Developer Guide: Using Batch

Batchable Methods

The following are methods for `Batchable`.

IN THIS SECTION:

execute(jobId, recordList)
Gets invoked when the batch job executes and operates on one batch of records. Contains or calls the main execution logic for the batch job.

finish(jobId)
Gets invoked when the batch job finishes. Place any clean up code in this method.

start(jobId)
Gets invoked when the batch job starts. Returns the record set as an iterable that will be batched for execution.

start(jobId)
Gets invoked when the batch job starts. Returns the record set as a QueryLocator object that will be batched for execution.

execute(jobId, recordList)

Gets invoked when the batch job executes and operates on one batch of records. Contains or calls the main execution logic for the batch job.

Signature

```
public Void execute(Database.BatchableContext jobId, List<sObject> recordList)
```

Parameters

jobId
Type: Database.BatchableContext

Contains the job ID.

recordList
Type: List<sObject>

Contains the batch of records to process.

Return Value

Type: Void

finish(jobId)

Gets invoked when the batch job finishes. Place any clean up code in this method.

Signature

```
public Void finish(Database.BatchableContext jobId)
```

Parameters

jobId
Type: Database.BatchableContext

Contains the job ID.

Return Value

Type: Void

start(jobId)

Gets invoked when the batch job starts. Returns the record set as an iterable that will be batched for execution.

Signature

```
public System.Iterable start(Database.BatchableContext jobId)
```

Parameters

jobId
Type: Database.BatchableContext

Contains the job ID.

Return Value

Type: System.Iterable

`start(jobId)`

Gets invoked when the batch job starts. Returns the record set as a QueryLocator object that will be batched for execution.

Signature

```
public Database.QueryLocator start(Database.BatchableContext jobId)
```

Parameters

jobId
Type: Database.BatchableContext

Contains the job ID.

Return Value

Type: Database.QueryLocator

BatchableContext Interface

Represents the parameter type of a batch job method and contains the batch job ID. This interface is implemented internally by Apex.

Namespace

Database

SEE ALSO:

Batchable Interface

BatchableContext Methods

The following are methods for `BatchableContext`.

IN THIS SECTION:

getChildJobId()
Returns the ID of the current batch job chunk that is being processed.

getJobId()
Returns the batch job ID.

`getChildJobId()`

Returns the ID of the current batch job chunk that is being processed.

Signature

```
public Id getChildJobId()
```

Return Value

Type: ID

`getJobId()`

Returns the batch job ID.

Signature

```
public Id getJobId()
```

Return Value

Type: ID

Cursor Class (Beta)

Contains methods to fetch records and to get the number of cursor rows returned from a SOQL query.

Namespace

Database

Usage

Note: This feature is a Beta Service. Customer may opt to try such Beta Service in its sole discretion. Any use of the Beta Service is subject to the applicable Beta Services Terms provided at Agreements and Terms. You can provide feedback and suggestions for the feature in the Trailblazer Community.

A cursor is created when a SOQL query is executed on a `Database.getCursor()` or a `Database.getCursorWithBinds()` call. When the SOQL query is invoked, the corresponding rows are returned from the cursor. The maximum number of rows per cursor is 50 million, regardless of the operation being synchronous or asynchronous.

Example

```
public class QueryChunkingQueuable implements Queueable {
    private Database.Cursor locator;
```

```
    private integer position;

    public QueryChunkingQueuable() {
        locator = Database.getCursor
                  ('SELECT Id FROM Contact WHERE LastActivityDate = LAST_N_DAYS:400');
        position = 0;
    }

    public void execute(QueueableContext ctx) {
        List<Contact> scope = locator.fetch(position, 200);
        position += scope.size();
        // do something, like archive or delete the scope list records
        if(position < locator.getNumRecords() ) {
            // process the next chunk
            System.enqueueJob(this);
        }
    }
}
```

IN THIS SECTION:

Cursor Methods

Cursor Methods

The following are methods for `Cursor`.

IN THIS SECTION:

fetch(position, count)

Fetches cursor rows that correspond to the offset position and the specified record count. The maximum number of rows per cursor is 50 million, regardless of the operation being synchronous or asynchronous.

getNumRecords()

Gets the number of rows returned in an Apex cursor from a `Cursor.fetch(position, count)` operation.

`fetch(position, count)`

Fetches cursor rows that correspond to the offset position and the specified record count. The maximum number of rows per cursor is 50 million, regardless of the operation being synchronous or asynchronous.

Signature

```
public static List<SObject> fetch(Integer position, Integer count)
```

Parameters

`position`

Type: Integer

The offset position from which records are fetched.

count
Type: Integer

The number of sObjects to fetch from the cursor, up to a maximum of 2,000.

Return Value

Type: List on page 3603<sObject>

The list of sObjects from the SOQL query, starting from the specified position.

`getNumRecords()`

Gets the number of rows returned in an Apex cursor from a `Cursor.fetch(position, count)` operation.

Signature

```
public static Integer getNumRecords()
```

Return Value

Type: Integer

DeletedRecord Class

Contains information about a deleted record.

Namespace

Database

Usage

The `getDeletedRecords` method of the `Database.GetDeletedResult` class returns a list of `Database.DeletedRecord` objects. Use the methods in the `Database.DeletedRecord` class to retrieve details about each deleted record.

DeletedRecord Methods

The following are methods for `DeletedRecord`. All are instance methods.

IN THIS SECTION:

getDeletedDate()
Returns the deleted date of this record.

getId()
Returns the ID of a record deleted within the time window specified in the `Database.getDeleted` method.

`getDeletedDate()`

Returns the deleted date of this record.

Signature

```
public Date getDeletedDate()
```

Return Value

Type: Date

`getId()`

Returns the ID of a record deleted within the time window specified in the `Database.getDeleted` method.

Signature

```
public Id getId()
```

Return Value

Type: ID

`DeleteFilter Enum`

Determines if deleted rows are included in Apex cursor rows.

Enum Values

The following are the values of the `Database.DeleteFilter` enum.

Value	Description
`DELETED_ROWS_ONLY`	Includes only deleted rows.
`NO_DELETED_ROWS`	Excludes deleted rows.
`NO_DELETED_SHARING_ROWS`	Excludes deleted rows as per sharing rules.
`NO_FILTER`	Includes all rows, including deleted rows.

DeleteResult Class

Represents the result of a delete DML operation returned by the `Database.delete` method.

Namespace

Database

Usage

An array of `Database.DeleteResult` objects is returned with the `delete` database method. Each element in the DeleteResult array corresponds to the sObject array passed as the *`sObject[]`* parameter in the `delete` Database method; that is, the first element in the DeleteResult array matches the first element passed in the sObject array, the second element corresponds with the second element, and so on. If only one sObject is passed in, the DeleteResult array contains a single element.

Example

The following example shows how to obtain and iterate through the returned `Database.DeleteResult` objects. It deletes some queried accounts using `Database.delete` with a false second parameter to allow partial processing of records on failure. Next, it iterates through the results to determine whether the operation was successful or not for each record. It writes the ID of every record that was processed successfully to the debug log, or error messages and fields of the failed records.

```
// Query the accounts to delete
Account[] accts = [SELECT Id from Account WHERE Name LIKE 'Acme%'];
// Delete the accounts
Database.DeleteResult[] drList = Database.delete(accts, false);

// Iterate through each returned result
for(Database.DeleteResult dr : drList) {
    if (dr.isSuccess()) {
        // Operation was successful, so get the ID of the record that was processed
        System.debug('Successfully deleted account with ID: ' + dr.getId());
    }
    else {
        // Operation failed, so get all errors
        for(Database.Error err : dr.getErrors()) {
            System.debug('The following error has occurred.');
            System.debug(err.getStatusCode() + ': ' + err.getMessage());
            System.debug('Account fields that affected this error: ' + err.getFields());
        }
    }
}
```

DeleteResult Methods

The following are methods for `DeleteResult`. All are instance methods.

IN THIS SECTION:

getErrors()
If an error occurred, returns an array of one or more database error objects providing the error code and description. If no error occurred, returns an empty set.

getId()
Returns the ID of the sObject you were trying to delete.

isSuccess()
A Boolean value that is set to `true` if the DML operation was successful for this object, `false` otherwise.

`getErrors()`

If an error occurred, returns an array of one or more database error objects providing the error code and description. If no error occurred, returns an empty set.

Signature

```
public Database.Error[] getErrors()
```

Return Value

Type: Database.Error[]

`getId()`

Returns the ID of the sObject you were trying to delete.

Signature

```
public ID getId()
```

Return Value

Type: ID

`isSuccess()`

A Boolean value that is set to `true` if the DML operation was successful for this object, `false` otherwise.

Signature

```
public Boolean isSuccess()
```

Return Value

Type: Boolean

DMLOptions Class

Enables you to set options related to DML operations.

Namespace

Database

Usage

`Database.DMLOptions` is only available for Apex saved against API versions 15.0 and higher. DMLOptions settings take effect only for record operations performed using Apex DML and not through the Salesforce user interface. The DMLOptions class has three child options.

DML Child Options

DmlOptions.AssignmentRuleHeader—Enables setting assignment rule options.

DmlOptions.DuplicateRuleHeader—Determines options for using duplicate rules to detect duplicate records. Duplicate rules are part of the Duplicate Management feature.

DmlOptions.EmailHeader—Enables setting email options.

DmlOptions Properties

The following are properties for `DmlOptions`.

IN THIS SECTION:

allowFieldTruncation
Specifies the truncation behavior of large strings.

assignmentRuleHeader
Specifies the assignment rule to be used when creating a case or lead.

emailHeader
Specifies additional information regarding the automatic email that gets sent when an events occurs.

localeOptions
Specifies the language of any labels that are returned by Apex.

optAllOrNone
Specifies whether the operation allows for partial success.

`allowFieldTruncation`

Specifies the truncation behavior of large strings.

Signature

```
public Boolean allowFieldTruncation {get; set;}
```

Property Value

Type: Boolean

Usage

In Apex saved against API versions previous to 15.0, if you specify a value for a string and that value is too large, the value is truncated. For API version 15.0 and later, if a value is specified that is too large, the operation fails and an error message is returned. The `allowFieldTruncation` property allows you to specify that the previous behavior, truncation, be used instead of the new behavior in Apex saved against API versions 15.0 and later.

`assignmentRuleHeader`

Specifies the assignment rule to be used when creating a case or lead.

Signature

```
public Database.DmlOptions.Assignmentruleheader assignmentRuleHeader {get; set;}
```

Property Value

Type: Database.DMLOptions.AssignmentRuleHeader

Usage

DMLOption.AssignmentRuleHeader.useDefaultRule affects only the default assignment rule and does not disable other existing assignment rules on the object.

Note: The Database.DMLOptions object supports assignment rules for cases and leads, but not for accounts.

emailHeader

Specifies additional information regarding the automatic email that gets sent when an events occurs.

Signature

```
public Database.DmlOptions.EmailHeader emailHeader {get; set;}
```

Property Value

Type: Database.DMLOptions.EmailHeader

Usage

The Salesforce user interface allows you to specify whether or not to send an email when the following events occur.

- Creation of a new case or task
- Conversion of a case email to a contact
- New user email notification
- Lead queue email notification
- Password reset

In Apex saved against API version 15.0 or later, the Database.DMLOptions `emailHeader` property enables you to specify additional information regarding the email that gets sent when one of the events occurs because of the code's execution.

localeOptions

Specifies the language of any labels that are returned by Apex.

Signature

```
public Database.DmlOptions.LocaleOptions localeOptions {get; set;}
```

Property Value

Type: Database.DMLOptions.LocaleOptions

Usage

The value must be a valid user locale (language and country), such as de_DE or en_GB. The value is a String, 2-5 characters long. The first two characters are always an ISO language code, for example 'fr' or 'en.' If the value is further qualified by a country, then the string also has an underscore (_) and another ISO country code, for example 'US' or 'UK'. For example, the string for the United States is 'en_US', and the string for French Canadian is 'fr_CA'.

`optAllOrNone`

Specifies whether the operation allows for partial success.

Signature

```
public Boolean optAllOrNone {get; set;}
```

Property Value

Type: Boolean

Usage

If `optAllOrNone` is set to `true`, all changes are rolled back if any record causes errors. The default for this property is `false` and successfully processed records are committed while records with errors aren't. If `optAllOrNone` is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed.

This property is available in Apex saved against Salesforce API version 20.0 and later.

DmlOptions.AssignmentRuleHeader Class

Enables setting assignment rule options.

Namespace

Database

Example

The following example uses the `useDefaultRule` option:

```
Database.DMLOptions dmo = new Database.DMLOptions();
dmo.assignmentRuleHeader.useDefaultRule= true;

Lead l = new Lead(company='ABC', lastname='Smith');
l.setOptions(dmo);
insert l;
```

The following example uses the `assignmentRuleID` option:

```
Database.DMLOptions dmo = new Database.DMLOptions();
dmo.assignmentRuleHeader.assignmentRuleId= '01QD0000000EqAn';
```

```
Lead l = new Lead(company='ABC', lastname='Smith');
l.setOptions(dmo);
insert l;
```

DmlOptions.AssignmentRuleHeader Properties

The following are properties for `DmlOptions.AssignmentRuleHeader`.

IN THIS SECTION:

assignmentRuleID

Specifies the ID of a specific assignment rule to run for the case or lead. The assignment rule can be active or inactive.

useDefaultRule

If specified as `true` for a case or lead, the system uses the default (active) assignment rule for the case or lead. If specified, do not specify an `assignmentRuleId`.

assignmentRuleID

Specifies the ID of a specific assignment rule to run for the case or lead. The assignment rule can be active or inactive.

Signature

```
public Id assignmentRuleID {get; set;}
```

Property Value

Type: ID

Usage

The ID can be retrieved by querying the AssignmentRule sObject. If specified, do not specify `useDefaultRule`.

If the value is not in the correct ID format (15-character or 18-character Salesforce ID), the call fails and an exception is returned.

useDefaultRule

If specified as `true` for a case or lead, the system uses the default (active) assignment rule for the case or lead. If specified, do not specify an `assignmentRuleId`.

Signature

```
public Boolean useDefaultRule {get; set;}
```

Property Value

Type: Boolean

Usage

If there are no assignment rules in the organization, in API version 29.0 and earlier, creating a case or lead with `useDefaultRule` set to `true` results in the case or lead being assigned to the predefined default owner. In API version 30.0 and later, the case or lead is unassigned and doesn't get assigned to the default owner.

DMLOptions.DuplicateRuleHeader Class

Determines options for using duplicate rules to detect duplicate records. Duplicate rules are part of the Duplicate Management feature.

Namespace

Database

Example

The following example shows how to save an account record that's been identified as a duplicate. To learn how to iterate through duplicate errors, see DuplicateError Class

```
Database.DMLOptions dml = new Database.DMLOptions();
dml.DuplicateRuleHeader.allowSave = true;
dml.DuplicateRuleHeader.runAsCurrentUser = true;
Account duplicateAccount = new Account(Name='dupe');
Database.SaveResult sr = Database.insert(duplicateAccount, dml);
if (sr.isSuccess()) {
 System.debug('Duplicate account has been inserted in Salesforce!');
}
```

IN THIS SECTION:

DMLOptions.DuplicateRuleHeader Properties

DMLOptions.DuplicateRuleHeader Properties

The following are properties for `DMLOptions.DuplicateRuleHeader`.

IN THIS SECTION:

allowSave

For a duplicate rule, when the Alert option is enabled, bypass alerts and save duplicate records by setting this property to `true`. Prevent duplicate records from being saved by setting this property to `false`.

runAsCurrentUser

Make sure that sharing rules for the current user are enforced when duplicate rules run by setting this property to `true`. Use the sharing rules specified in the class for the request by setting this property to `false`. If no sharing rules are specified, Apex code runs in system context and sharing rules for the current user are not enforced.

`allowSave`

For a duplicate rule, when the Alert option is enabled, bypass alerts and save duplicate records by setting this property to `true`. Prevent duplicate records from being saved by setting this property to `false`.

Signature

```
public Boolean allowSave {get; set;}
```

Property Value

Type: Boolean

Example

This example shows how to save an account record that's been identified as a duplicate. `dml.DuplicateRuleHeader.allowSave = true` means the user should be allowed to save the duplicate. To learn how to iterate through duplicate errors, see DuplicateError Class.

```
Database.DMLOptions dml = new Database.DMLOptions();
dml.DuplicateRuleHeader.allowSave = true;
dml.DuplicateRuleHeader.runAsCurrentUser = true;
Account duplicateAccount = new Account(Name='dupe');
Database.SaveResult sr = Database.insert(duplicateAccount, dml);
if (sr.isSuccess()) {
 System.debug('Duplicate account has been inserted in Salesforce!');
}
```

runAsCurrentUser

Make sure that sharing rules for the current user are enforced when duplicate rules run by setting this property to `true`. Use the sharing rules specified in the class for the request by setting this property to `false`. If no sharing rules are specified, Apex code runs in system context and sharing rules for the current user are not enforced.

Signature

```
public Boolean runAsCurrentUser {get; set;}
```

Property Value

Type: Boolean

Usage

If specified as `true`, duplicate rules run for the current user, which ensures users can't view duplicate records that aren't available to them.

Use `runAsCurrentUser = true` to detect duplicates when converting leads to contacts. Typically, lead conversion Apex code runs in a system context and does not enforce sharing rules for the current user.

Example

This example shows how to set options so that duplicate rules run for the current user when saving a new account.

```
Database.DMLOptions dml = new Database.DMLOptions();
dml.DuplicateRuleHeader.allowSave = true;
dml.DuplicateRuleHeader.runAsCurrentUser = true;
Account duplicateAccount = new Account(Name='dupe');
```

```
Database.SaveResult sr = Database.insert(duplicateAccount, dml);
if (sr.isSuccess()) {
 System.debug('Duplicate account has been inserted in Salesforce!');
}
```

DmlOptions.EmailHeader Class

Enables setting email options.

Namespace

Database

Usage

Even though auto-sent emails can be triggered by actions in the Salesforce user interface, the DMLOptions settings for `emailHeader` take effect only for DML operations carried out in Apex code.

Example

In the following example, the `triggerAutoResponseEmail` option is specified:

```
Account a = new Account(name='Acme Plumbing');

    insert a;

    Contact c = new Contact(email='jplumber@salesforce.com',
firstname='Joe',lastname='Plumber', accountid=a.id);

    insert c;

    Database.DMLOptions dlo = new Database.DMLOptions();

    dlo.EmailHeader.triggerAutoResponseEmail = true;

    Case ca = new Case(subject='Plumbing Problems', contactid=c.id);

    database.insert(ca, dlo);
```

Suppose that you use an after-insert or after-update trigger to change ownership of leads, contacts, or opportunities. If you use the API to change record ownership, or if a Lightning Experience user changes a record's owner, no email notification is sent. To send email notifications to a record's new owner, set the `triggerUserEmail` property to `true`.

DmlOptions.EmailHeader Properties

The following are properties for `DmlOptions.EmailHeader`.

IN THIS SECTION:

triggerAutoResponseEmail
Indicates whether to trigger auto-response rules (`true`) or not (`false`), for leads and cases.

triggerOtherEmail
Indicates whether to trigger email outside the organization (`true`) or not (`false`).

triggerUserEmail
Indicates whether to trigger email that is sent to users in the organization (`true`) or not (`false`).

`triggerAutoResponseEmail`

Indicates whether to trigger auto-response rules (`true`) or not (`false`), for leads and cases.

Signature

```
public Boolean triggerAutoResponseEmail {get; set;}
```

Property Value

Type: Boolean

Usage

This email can be automatically triggered by a number of events, for example creating a case or resetting a user password. If this value is set to `true`, when a case is created, if there is an email address for the contact specified in `ContactID`, the email is sent to that address. If not, the email is sent to the address specified in `SuppliedEmail`

`triggerOtherEmail`

Indicates whether to trigger email outside the organization (`true`) or not (`false`).

Signature

```
public Boolean triggerOtherEmail {get; set;}
```

Property Value

Type: Boolean

Usage

This email can be automatically triggered by creating, editing, or deleting a contact for a case.

Note: Email sent through Apex because of a group event includes additional behaviors. A *group event* is an event for which `IsGroupEvent` is true. The `EventAttendee` object tracks the users, leads, or contacts that are invited to a group event. Note the following behaviors for group event email sent through Apex:

- Sending a group event invitation to a lead or contact respects the `triggerOtherEmail` option
- Email sent when updating or deleting a group event also respects the `triggerUserEmail` and `triggerOtherEmail` options, as appropriate

`triggerUserEmail`

Indicates whether to trigger email that is sent to users in the organization (`true`) or not (`false`).

Signature

```
public Boolean triggerUserEmail {get; set;}
```

Property Value

Type: Boolean

Usage

This email can be automatically triggered by a number of events; resetting a password, creating a new user, or creating or modifying a task.

Note: Adding comments to a case in Apex doesn't trigger email to users in the organization even if `triggerUserEmail` is set to `true`.

Note: Email sent through Apex because of a group event includes additional behaviors. A *group event* is an event for which `IsGroupEvent` is true. The EventAttendee object tracks the users, leads, or contacts that are invited to a group event. Note the following behaviors for group event email sent through Apex:

- Sending a group event invitation to a user respects the `triggerUserEmail` option
- Email sent when updating or deleting a group event also respects the `triggerUserEmail` and `triggerOtherEmail` options, as appropriate

DuplicateError Class

Contains information about an error that occurred when an attempt was made to save a duplicate record. Use if your organization has set up duplicate rules, which are part of the Duplicate Management feature.

Namespace

Database

Example

When you try to save a record that's identified as a duplicate record by a duplicate rule, you'll receive a duplicate error. If the duplicate rule contains the Allow action, an attempt will be made to bypass the error.

```
// Try to save a duplicate account
Account duplicateAccount = new Account(Name='Acme',  BillingCity='San Francisco');
Database.SaveResult sr = Database.insert(duplicateAccount, false);
if (!sr.isSuccess()) {

 // Insertion failed due to duplicate detected
 for(Database.Error duplicateError : sr.getErrors()){
  Datacloud.DuplicateResult duplicateResult =
                 ((Database.DuplicateError)duplicateError).getDuplicateResult();
  System.debug('Duplicate records have been detected by ' +
```

```
                    duplicateResult.getDuplicateRule());
   System.debug(duplicateResult.getErrorMessage());
  }

  // If the duplicate rule is an alert rule, we can try to bypass it
  Database.DMLOptions dml = new Database.DMLOptions();
  dml.DuplicateRuleHeader.AllowSave = true;
  Database.SaveResult sr2 = Database.insert(duplicateAccount, dml);
  if (sr2.isSuccess()) {
   System.debug('Duplicate account has been inserted in Salesforce!');
  }
 }
```

IN THIS SECTION:

DuplicateError Methods

SEE ALSO:

SaveResult Class

DuplicateResult Class

Error Class

DuplicateError Methods

The following are methods for `DuplicateError`.

IN THIS SECTION:

getDuplicateResult()

Returns the details of a duplicate rule and duplicate records found by the duplicate rule.

getFields()

Returns an array of one or more field names. Identifies which fields in the object, if any, affected the error condition.

getMessage()

Returns the error message text.

getStatusCode()

Returns a code that characterizes the error.

`getDuplicateResult()`

Returns the details of a duplicate rule and duplicate records found by the duplicate rule.

Signature

```
public Datacloud.DuplicateResult getDuplicateResult()
```

Return Value

Type: Datacloud.DuplicateResult

Example

This example shows the code used to get the possible duplicates and related match information after saving a new contact. This code is part of a custom application that implements duplicate management when users add a contact. See DuplicateResult Class on page 2435 to check out the entire sample applicaton.

```
Datacloud.DuplicateResult duplicateResult =
                            duplicateError.getDuplicateResult();
```

`getFields()`

Returns an array of one or more field names. Identifies which fields in the object, if any, affected the error condition.

Signature

```
public List<String> getFields()
```

Return Value

Type: List<String>

`getMessage()`

Returns the error message text.

Signature

```
public String getMessage()
```

Return Value

Type: String

`getStatusCode()`

Returns a code that characterizes the error.

Signature

```
public StatusCode getStatusCode()
```

Return Value

Type: StatusCode

EmptyRecycleBinResult Class

The result of the emptyRecycleBin DML operation returned by the `Database.emptyRecycleBin` method.

Namespace

Database

Usage

A list of `Database.EmptyRecycleBinResult` objects is returned by the `Database.emptyRecycleBin` method. Each object in the list corresponds to either a record ID or an sObject passed as the parameter in the `Database.emptyRecycleBin` method. The first index in the EmptyRecycleBinResult list matches the first record or sObject specified in the list, the second with the second, and so on.

EmptyRecycleBinResult Methods

The following are methods for `EmptyRecycleBinResult`. All are instance methods.

IN THIS SECTION:

getErrors()
If an error occurred during the delete for this record or sObject, returns a list of one or more Database.Error objects. If no errors occurred, the returned list is empty.

getId()
Returns the ID of the record or sObject you attempted to delete.

isSuccess()
Returns `true` if the record or sObject was successfully removed from the Recycle Bin; otherwise `false`.

`getErrors()`

If an error occurred during the delete for this record or sObject, returns a list of one or more Database.Error objects. If no errors occurred, the returned list is empty.

Signature

```
public Database.Errors[] getErrors()
```

Return Value

Type: Database.Errors []

`getId()`

Returns the ID of the record or sObject you attempted to delete.

Signature

```
public ID getId()
```

Return Value

Type: ID

isSuccess()

Returns `true` if the record or sObject was successfully removed from the Recycle Bin; otherwise `false`.

Signature

```
public Boolean isSuccess()
```

Return Value

Type: Boolean

Error Class

Represents information about an error that occurred during a DML operation when using a Database method.

Namespace

Database

Usage

`Error` class is part of `SaveResult`, which is generated when a user attempts to save a Salesforce record.

SEE ALSO:

- SaveResult Class
- DuplicateError Class

Error Methods

The following are methods for `Error`. All are instance methods.

IN THIS SECTION:

- getFields()
  Returns an array of one or more field names. Identifies which fields in the object, if any, affected the error condition.
- getMessage()
  Returns the error message text.
- getStatusCode()
  Returns a code that characterizes the error.

getFields()

Returns an array of one or more field names. Identifies which fields in the object, if any, affected the error condition.

Signature

```
public String[] getFields()
```

Return Value

Type: String[]

`getMessage()`

Returns the error message text.

Signature

```
public String getMessage()
```

Return Value

Type: String

`getStatusCode()`

Returns a code that characterizes the error.

Signature

```
public StatusCode getStatusCode()
```

Return Value

Type: StatusCode

Usage

The full list of status codes is available in the WSDL file for your organization (see *Downloading Salesforce WSDLs and Client Authentication Certificates* in the Salesforce online help.)

GetDeletedResult Class

Contains the deleted records retrieved for a specific sObject type and time window.

Namespace

Database

Usage

The `Database.getDeleted` method returns the deleted record information as a `Database.GetDeletedResult` object.

GetDeletedResult Methods

The following are methods for `GetDeletedResult`. All are instance methods.

IN THIS SECTION:

getDeletedRecords()
Returns a list of deleted records for the time window specified in the `Database.getDeleted` method call.

getEarliestDateAvailable()
Returns the date in Coordinated Universal Time (UTC) of the earliest physically deleted object for the sObject type specified in `Database.getDeleted`.

getLatestDateCovered()
Returns the date in Coordinated Universal Time (UTC) of the last date covered in the `Database.getDeleted` call.

`getDeletedRecords()`

Returns a list of deleted records for the time window specified in the `Database.getDeleted` method call.

Signature

```
public List<Database.DeletedRecord> getDeletedRecords()
```

Return Value

Type: List<Database.DeletedRecord>

`getEarliestDateAvailable()`

Returns the date in Coordinated Universal Time (UTC) of the earliest physically deleted object for the sObject type specified in `Database.getDeleted`.

Signature

```
public Date getEarliestDateAvailable()
```

Return Value

Type: Date

`getLatestDateCovered()`

Returns the date in Coordinated Universal Time (UTC) of the last date covered in the `Database.getDeleted` call.

Signature

```
public Date getLatestDateCovered()
```

Return Value

Type: Date

Usage

If there is a value, it is less than or equal to the *endDate* argument of `Database.getDeleted`. A value here indicates that, for safety, you should use this value for the *startDate* of your next call to capture the changes that started after this date but didn't complete before *endDate* and were, therefore, not returned in the previous call.

GetUpdatedResult Class

Contains the result for the `Database.getUpdated` method call.

Namespace

Database

Usage

Use the methods in this class to obtain detailed information about the updated records returned by `Database.getUpdated` for a specific time window.

GetUpdatedResult Methods

The following are methods for `GetUpdatedResult`. All are instance methods.

IN THIS SECTION:

getIds()
Returns the IDs of records updated within the time window specified in the `Database.getUpdated` method.

getLatestDateCovered()
Returns the date in Coordinated Universal Time (UTC) of the last date covered in the `Database.getUpdated` call.

`getIds()`

Returns the IDs of records updated within the time window specified in the `Database.getUpdated` method.

Signature

```
public List<Id> getIds()
```

Return Value

Type: List<ID>

`getLatestDateCovered()`

Returns the date in Coordinated Universal Time (UTC) of the last date covered in the `Database.getUpdated` call.

Signature

```
public Date getLatestDateCovered()
```

Return Value

Type: Date

LeadConvert Class

Contains information used for lead conversion.

Namespace

Database

Usage

The `convertLead` Database method converts a lead into an account and contact or an account and person account, as well as (optionally) an opportunity. The `convertLead` takes an instance of the `Database.LeadConvert` class as a parameter. Create an instance of this class and set the information required for conversion, such as setting the lead, and destination account and contact.

Note: The `Database.convertLead()` method can take one `LeadConvert` object or a list of `LeadConvert` objects.

Example

This example shows how to use the `Database.convertLead` method to convert a lead. It inserts a new lead, creates a `LeadConvert` object, sets its status to `converted`, then passes it to the `Database.convertLead` method. Finally, it verifies that the conversion was successful.

```
Lead myLead = new Lead(LastName = 'Fry', Company='Fry And Sons');
insert myLead;

Database.LeadConvert lc = new Database.LeadConvert();
lc.setLeadId(myLead.id);

LeadStatus convertStatus = [SELECT Id, ApiName FROM LeadStatus WHERE IsConverted=true LIMIT
 1];
lc.setConvertedStatus(convertStatus.ApiName);

Database.LeadConvertResult lcr = Database.convertLead(lc);
System.assert(lcr.isSuccess());
```

IN THIS SECTION:

LeadConvert Constructors

LeadConvert Methods

LeadConvert Constructors

The following are constructors for `LeadConvert`.

IN THIS SECTION:

LeadConvert()
Creates a new instance of the `Database.LeadConvert` class.

`LeadConvert()`

Creates a new instance of the `Database.LeadConvert` class.

Signature

```
public LeadConvert()
```

LeadConvert Methods

The following are methods for `LeadConvert`. All are instance methods.

IN THIS SECTION:

getAccountId()
Gets the ID of the account into which the lead will be merged.

getAccountRecord()
This method is for internal use only.

getBypassAccountDedupeCheck()
This method is for internal use only.

getBypassContactDedupeCheck()
This method is for internal use only.

getContactId()
Gets the ID of the contact into which the lead will be merged.

getContactRecord()
This method is for internal use only.

getConvertedStatus()
Gets the lead status value for a converted lead.

getLeadID()
Gets the ID of the lead to convert.

getOpportunityId()
Gets the ID of the existing opportunity that will be related to the resulting contact.

getOpportunityName()
Gets the name of the opportunity to create.

getOpportunityRecord()
This method is for internal use only.

getOwnerID()
Gets the ID of the person to own any newly created account, contact, and opportunity.

getRelatedPersonAccountId()

Gets the ID of the existing person account into which the lead will be converted.

getRelatedPersonAccountRecord()

Gets the entity record of the new person account into which the lead will be converted.

isDoNotCreateOpportunity()

Indicates whether an Opportunity is created during lead conversion (`false`, the default) or not (`true`).

isOverWriteLeadSource()

Indicates whether the `LeadSource` field on the target Contact object is overwritten with the contents of the `LeadSource` field in the source Lead object (`true`), or not (`false`, the default).

isSendNotificationEmail()

Indicates whether a notification email is sent to the owner specified by `setOwnerId` (`true`) or not (`false`, the default).

setAccountId(accountId)

Sets the ID of the account into which the lead is merged. This value is required only when updating an existing account, including person accounts.

setAccountRecord(accountRecord)

This method is for internal use only.

setBypassAccountDedupeCheck(bypassAccountDedupeCheck)

This method is for internal use only.

setBypassContactDedupeCheck(bypassContactDedupeCheck)

This method is for internal use only.

setContactId(contactId)

Sets the ID of the contact into which the lead will be merged (this contact must be associated with the account specified with `setAccountId`, and `setAccountId` must be specified). This value is required only when updating an existing contact.

setContactRecord(contactRecord)

This method is for internal use only.

setConvertedStatus(status)

Sets the lead status value for a converted lead. This field is required.

setDoNotCreateOpportunity(createOpportunity)

Specifies whether to create an opportunity during lead conversion. The default value is `false`: opportunities are created by default. Set this flag to `true` only if you do not want to create an opportunity from the lead.

setLeadId(leadId)

Sets the ID of the lead to convert. This field is required.

setOpportunityId(opportunityId)

Sets the ID of the opportunity into which the lead is merged. This value is required only when updating an existing opportunity.

setOpportunityName(opportunityName)

Sets the name of the opportunity to create. If no name is specified, this value defaults to the company name of the lead.

setOpportunityRecord(opportunityRecord)

This method is for internal use only.

setOverwriteLeadSource(overwriteLeadSource)

Specifies whether to overwrite the `LeadSource` field on the target contact object with the contents of the `LeadSource` field in the source lead object. The default value is `false`, to not overwrite the field. If you specify this as `true`, you must also specify `setContactId` for the target contact.

setOwnerId(ownerId)

Specifies the ID of the person to own any newly created account, contact, and opportunity. If the application does not specify this value, the owner of the new object will be the owner of the lead.

setRelatedPersonAccountId(relatedPersonAccountId)

Sets the ID of the existing person account into which to convert the lead. This value is required only when updating an existing person account.

setSendNotificationEmail(sendEmail)

Specifies whether to send a notification email to the owner specified by `setOwnerId`. The default value is `false`, that is, to not send email.

getAccountId()

Gets the ID of the account into which the lead will be merged.

Signature

```
public ID getAccountId()
```

Return Value

Type: ID

getAccountRecord()

This method is for internal use only.

getBypassAccountDedupeCheck()

This method is for internal use only.

getBypassContactDedupeCheck()

This method is for internal use only.

getContactId()

Gets the ID of the contact into which the lead will be merged.

Signature

```
public ID getContactId()
```

Return Value

Type: ID

getContactRecord()

This method is for internal use only.

getConvertedStatus()

Gets the lead status value for a converted lead.

Signature

```
public String getConvertedStatus()
```

Return Value

Type: String

getLeadID()

Gets the ID of the lead to convert.

Signature

```
public ID getLeadID()
```

Return Value

Type: ID

getOpportunityId()

Gets the ID of the existing opportunity that will be related to the resulting contact.

Signature

```
public ID getOpportunityId()
```

Return Value

Type: ID

getOpportunityName()

Gets the name of the opportunity to create.

Signature

```
public String getOpportunityName()
```

Return Value

Type: String

getOpportunityRecord()

This method is for internal use only.

getOwnerID()

Gets the ID of the person to own any newly created account, contact, and opportunity.

Signature

```
public ID getOwnerID()
```

Return Value

Type: ID

getRelatedPersonAccountId()

Gets the ID of the existing person account into which the lead will be converted.

Signature

```
public ID getRelatedPersonAccountId()
```

Return Value

Type: ID

getRelatedPersonAccountRecord()

Gets the entity record of the new person account into which the lead will be converted.

Signature

```
public ID getRelatedPersonAccountRecord()
```

Return Value

Type: ID

isDoNotCreateOpportunity()

Indicates whether an Opportunity is created during lead conversion (`false`, the default) or not (`true`).

Signature

```
public Boolean isDoNotCreateOpportunity()
```

Return Value

Type: Boolean

isOverWriteLeadSource()

Indicates whether the `LeadSource` field on the target Contact object is overwritten with the contents of the `LeadSource` field in the source Lead object (`true`), or not (`false`, the default).

Signature

```
public Boolean isOverWriteLeadSource()
```

Return Value

Type: Boolean

isSendNotificationEmail()

Indicates whether a notification email is sent to the owner specified by `setOwnerId` (`true`) or not (`false`, the default).

Signature

```
public Boolean isSendNotificationEmail()
```

Return Value

Type: Boolean

setAccountId(accountId)

Sets the ID of the account into which the lead is merged. This value is required only when updating an existing account, including person accounts.

Signature

```
public Void setAccountId(ID accountId)
```

Parameters

accountId
Type: ID

Return Value

Type: Void

setAccountRecord(accountRecord)

This method is for internal use only.

setBypassAccountDedupeCheck(bypassAccountDedupeCheck)

This method is for internal use only.

setBypassContactDedupeCheck(bypassContactDedupeCheck)

This method is for internal use only.

setContactId(contactId)

Sets the ID of the contact into which the lead will be merged (this contact must be associated with the account specified with `setAccountId`, and `setAccountId` must be specified). This value is required only when updating an existing contact.

Signature

```
public Void setContactId(ID contactId)
```

Parameters

contactId
Type: ID

Return Value

Type: Void

Usage

If `setContactId` is specified, then the application creates a new contact that is implicitly associated with the account. The contact name and other existing data are not overwritten (unless `setOverwriteLeadSource` is set to true, in which case only the `LeadSource` field is overwritten).

Important: If you are converting a lead into a person account, do not specify `setContactId` or an error will result. Specify only `setAccountId` of the person account.

setContactRecord(contactRecord)

This method is for internal use only.

setConvertedStatus(status)

Sets the lead status value for a converted lead. This field is required.

Signature

```
public Void setConvertedStatus(String status)
```

Parameters

status
Type: String

Return Value

Type: Void

setDoNotCreateOpportunity(createOpportunity)

Specifies whether to create an opportunity during lead conversion. The default value is `false`: opportunities are created by default. Set this flag to `true` only if you do not want to create an opportunity from the lead.

Signature

```
public Void setDoNotCreateOpportunity(Boolean createOpportunity)
```

Parameters

createOpportunity
Type: Boolean

Return Value

Type: Void

setLeadId(leadId)

Sets the ID of the lead to convert. This field is required.

Signature

```
public Void setLeadId(ID leadId)
```

Parameters

leadId
Type: ID

Return Value

Type: Void

setOpportunityId(opportunityId)

Sets the ID of the opportunity into which the lead is merged. This value is required only when updating an existing opportunity.

Signature

```
public Void setOpportunityId(ID opportunityId)
```

Parameters

opportunityId
Type: ID

Return Value

Type: Void

setOpportunityName(opportunityName)

Sets the name of the opportunity to create. If no name is specified, this value defaults to the company name of the lead.

Signature

```
public Void setOpportunityName(String opportunityName)
```

Parameters

opportunityName
Type: String

Return Value

Type: Void

Usage

The maximum length of this field is 80 characters.

If `setDoNotCreateOpportunity` is true, no Opportunity is created and this field must be left blank; otherwise, an error is returned.

setOpportunityRecord(opportunityRecord)

This method is for internal use only.

setOverwriteLeadSource(overwriteLeadSource)

Specifies whether to overwrite the `LeadSource` field on the target contact object with the contents of the `LeadSource` field in the source lead object. The default value is `false`, to not overwrite the field. If you specify this as `true`, you must also specify `setContactId` for the target contact.

Signature

```
public Void setOverwriteLeadSource(Boolean overwriteLeadSource)
```

Parameters

overwriteLeadSource
Type: Boolean

Return Value

Type: Void

setOwnerId(ownerId)

Specifies the ID of the person to own any newly created account, contact, and opportunity. If the application does not specify this value, the owner of the new object will be the owner of the lead.

Signature

```
public Void setOwnerId(ID ownerId)
```

Parameters

ownerId
Type: ID

Return Value

Type: Void

Usage

This method is not applicable when merging with existing objects—if `setOwnerId` is specified, the `ownerId` field is not overwritten in an existing account or contact.

setRelatedPersonAccountId(relatedPersonAccountId)

Sets the ID of the existing person account into which to convert the lead. This value is required only when updating an existing person account.

Signature

```
public Void setRelatedPersonAccountId(ID relatedPersonAccountId)
```

Parameters

relatedPersonAccountId
Type: ID

Return Value

Type: Void

setSendNotificationEmail(sendEmail)

Specifies whether to send a notification email to the owner specified by `setOwnerId`. The default value is `false`, that is, to not send email.

Signature

```
public Void setSendNotificationEmail(Boolean sendEmail)
```

Parameters

sendEmail
Type: Boolean

Return Value

Type: Void

LeadConvertResult Class

The result of a lead conversion.

Namespace

Database

Usage

An array of LeadConvertResult objects is returned with the `convertLead` Database method. Each element in the LeadConvertResult array corresponds to the sObject array passed as the *`SObject[]`* parameter in the `convertLead` Database method, that is, the first element in the LeadConvertResult array matches the first element passed in the SObject array, the second element corresponds to the second element, and so on. If only one sObject is passed in, the LeadConvertResult array contains a single element.

LeadConvertResult Methods

The following are methods for `LeadConvertResult`. All are instance methods.

IN THIS SECTION:

getAccountId()
The ID of the new account (if a new account was specified) or the ID of the account specified when `convertLead` was invoked.

getContactId()
The ID of the new contact (if a new contact was specified) or the ID of the contact specified when `convertLead` was invoked.

getErrors()
If an error occurred, an array of one or more database error objects providing the error code and description.

getLeadId()
The ID of the converted lead.

getOpportunityId()
The ID of the new opportunity, if one was created when `convertLead` was invoked.

getRelatedPersonAccountId()
The ID of the new or existing person account specified when `convertLead` was invoked.

isSuccess()
A Boolean value that is set to `true` if the DML operation was successful for this object, `false` otherwise

`getAccountId()`

The ID of the new account (if a new account was specified) or the ID of the account specified when `convertLead` was invoked.

Signature

```
public ID getAccountId()
```

Return Value

Type: ID

getContactId()

The ID of the new contact (if a new contact was specified) or the ID of the contact specified when `convertLead` was invoked.

Signature

```
public ID getContactId()
```

Return Value

Type: ID

getErrors()

If an error occurred, an array of one or more database error objects providing the error code and description.

Signature

```
public Database.Error[] getErrors()
```

Return Value

Type: Database.Error[]

getLeadId()

The ID of the converted lead.

Signature

```
public ID getLeadId()
```

Return Value

Type: ID

getOpportunityId()

The ID of the new opportunity, if one was created when `convertLead` was invoked.

Signature

```
public ID getOpportunityId()
```

Return Value

Type: ID

`getRelatedPersonAccountId()`

The ID of the new or existing person account specified when `convertLead` was invoked.

Signature

```
public ID getRelatedPersonAccountId()
```

Return Value

Type: ID

`isSuccess()`

A Boolean value that is set to `true` if the DML operation was successful for this object, `false` otherwise

Signature

```
public Boolean isSuccess()
```

Return Value

Type: Boolean

MergeResult Class

Contains the result of a merge Database method operation.

Namespace

Database

Usage

The `Database.merge` method returns a `Database.MergeResult` object for each merged record.

MergeResult Methods

The following are methods for `MergeResult`. All are instance methods.

IN THIS SECTION:

getErrors()
Returns a list of `Database.Error` objects representing the errors encountered, if any, during a merge operation using the `Database.merge` method. If no error occurred, returns null.

getId()
Returns the ID of the master record into which other records were merged.

getMergedRecordIds()
Returns the IDs of the records merged into the master record.

getUpdatedRelatedIds()
Returns the IDs of all related records that were reparented as a result of the merge that are viewable by the user sending the merge call.

isSuccess()
Indicates whether the merge was successful (`true`) or not (`false`).

getErrors()

Returns a list of `Database.Error` objects representing the errors encountered, if any, during a merge operation using the `Database.merge` method. If no error occurred, returns null.

Signature

```
public List<Database.Error> getErrors()
```

Return Value

Type: List<Database.Error>

getId()

Returns the ID of the master record into which other records were merged.

Signature

```
public Id getId()
```

Return Value

Type: ID

getMergedRecordIds()

Returns the IDs of the records merged into the master record.

Signature

```
public List<String> getMergedRecordIds()
```

Return Value

Type: List<String>

getUpdatedRelatedIds()

Returns the IDs of all related records that were reparented as a result of the merge that are viewable by the user sending the merge call.

Signature

```
public List<String> getUpdatedRelatedIds()
```

Return Value

Type: List<String>

`isSuccess()`

Indicates whether the merge was successful (`true`) or not (`false`).

Signature

```
public Boolean isSuccess()
```

Return Value

Type: Boolean

QueryLocator Class

Represents the record set returned by `Database.getQueryLocator` and used with Batch Apex.

Namespace

Database

QueryLocator Methods

The following are methods for `QueryLocator`. All are instance methods.

IN THIS SECTION:

getQuery()
Returns the query used to instantiate the `Database.QueryLocator` object. This is useful when testing the `start` method.

iterator()
Returns a new instance of a query locator iterator.

`getQuery()`

Returns the query used to instantiate the `Database.QueryLocator` object. This is useful when testing the `start` method.

Signature

```
public String getQuery()
```

Return Value

Type: String

Usage

You can't use the `FOR UPDATE` keywords with a getQueryLocator query to lock a set of records. The set of records in the batch is determined when the `start` method is run.

Example

```
System.assertEquals(QLReturnedFromStart.
getQuery(),
Database.getQueryLocator([SELECT Id
   FROM Account]).getQuery() );
```

`iterator()`

Returns a new instance of a query locator iterator.

Signature

```
public Database.QueryLocatorIterator iterator()
```

Return Value

Type: Database.QueryLocatorIterator

Usage

Warning: To iterate over a query locator, save the iterator instance that this method returns in a variable and then use this variable to iterate over the collection. Calling `iterator` every time you want to perform an iteration can result in incorrect behavior because each call returns a new iterator instance.

For an example, see QueryLocatorIterator Class.

QueryLocatorIterator Class

Represents an iterator over a query locator record set.

Namespace

Database

Example

This sample shows how to obtain an iterator for a query locator, which contains five accounts. This sample calls `hasNext` and `next` to get each record in the collection.

```
// Get a query locator
Database.QueryLocator q = Database.getQueryLocator(
    [SELECT Name FROM Account LIMIT 5]);
// Get an iterator
Database.QueryLocatorIterator it =  q.iterator();
```

```
// Iterate over the records
while (it.hasNext())
{
    Account a = (Account)it.next();
    System.debug(a);
}
```

QueryLocatorIterator Methods

The following are methods for `QueryLocatorIterator`. All are instance methods.

IN THIS SECTION:

hasNext()
Returns `true` if there are one or more records remaining in the collection; otherwise, returns `false`.

next()
Advances the iterator to the next sObject record and returns the sObject.

`hasNext()`

Returns `true` if there are one or more records remaining in the collection; otherwise, returns `false`.

Signature

```
public Boolean hasNext()
```

Return Value

Type: Boolean

`next()`

Advances the iterator to the next sObject record and returns the sObject.

Signature

```
public sObject next()
```

Return Value

Type: sObject

Usage

Because the return value is the generic sObject type, you must cast it if using a more specific type. For example:

```
Account a = (Account)myIterator.next();
```

Example

```
Account a = (Account)myIterator.next();
```

SaveResult Class

The result of an insert or update DML operation returned by a Database method.

Namespace

Database

Usage

An array of SaveResult objects is returned with the `insert` and `update` database methods. Each element in the SaveResult array corresponds to the sObject array passed as the *`sObject[]`* parameter in the Database method, that is, the first element in the SaveResult array matches the first element passed in the sObject array, the second element corresponds with the second element, and so on. If only one sObject is passed in, the SaveResult array contains a single element.

A SaveResult object is generated when a new or existing Salesforce record is saved.

Example

The following example shows how to obtain and iterate through the returned `Database.SaveResult` objects. It inserts two accounts using `Database.insert` with a false second parameter to allow partial processing of records on failure. One of the accounts is missing the Name required field, which causes a failure. Next, it iterates through the results to determine whether the operation was successful or not for each record. It writes the ID of every record that was processed successfully to the debug log, or error messages and fields of the failed records. This example generates one successful operation and one failure.

```
// Create two accounts, one of which is missing a required field
Account[] accts = new List<Account>{
    new Account(Name='Account1'),
    new Account()};
Database.SaveResult[] srList = Database.insert(accts, false);

// Iterate through each returned result
for (Database.SaveResult sr : srList) {
    if (sr.isSuccess()) {
        // Operation was successful, so get the ID of the record that was processed
        System.debug('Successfully inserted account. Account ID: ' + sr.getId());
    }
    else {
        // Operation failed, so get all errors
        for(Database.Error err : sr.getErrors()) {
            System.debug('The following error has occurred.');
            System.debug(err.getStatusCode() + ': ' + err.getMessage());
            System.debug('Account fields that affected this error: ' + err.getFields());
        }
```

```
        }
}
```

SEE ALSO:

Error Class

DuplicateError Class

SaveResult Methods

The following are methods for `SaveResult`. All are instance methods.

IN THIS SECTION:

getErrors()

If an error occurred, returns an array of one or more database error objects providing the error code and description. If no error occurred, returns an empty set.

getId()

Returns the ID of the sObject you were trying to insert or update.

isSuccess()

Returns a Boolean that is set to `true` if the DML operation was successful for this object, `false` otherwise.

`getErrors()`

If an error occurred, returns an array of one or more database error objects providing the error code and description. If no error occurred, returns an empty set.

Signature

```
public Database.Error[] getErrors()
```

Return Value

Type: Database.Error[]

`getId()`

Returns the ID of the sObject you were trying to insert or update.

Signature

```
public ID getId()
```

Return Value

Type: ID

Versioned Behavior Changes

In API version 53.0 and later, the method returns the sObject ID. However, if record locking fails during the update operation, the method returns a null value.

In API version 52.0 and earlier, the method returned a null value if the record wasn't updated successfully.

`isSuccess()`

Returns a Boolean that is set to `true` if the DML operation was successful for this object, `false` otherwise.

Signature

```
public Boolean isSuccess()
```

Return Value

Type: Boolean

Example

This example shows the code used to process duplicate records, which are detected when there is an unsuccessful save due to an error. This code is part of a custom application that implements duplicate management when users add a contact. See DuplicateResult Class on page 2435 to check out the entire sample applicaton.

```
if (!saveResult.isSuccess()) { ... }
```

UndeleteResult Class

The result of an undelete DML operation returned by the `Database.undelete` method.

Namespace

Database

Usage

An array of Database.UndeleteResult objects is returned with the `undelete` database method. Each element in the UndeleteResult array corresponds to the sObject array passed as the *`sObject[]`* parameter in the `undelete` Database method; that is, the first element in the UndeleteResult array matches the first element passed in the sObject array, the second element corresponds with the second element, and so on. If only one sObject is passed in, the UndeleteResults array contains a single element.

UndeleteResult Methods

The following are methods for `UndeleteResult`. All are instance methods.

IN THIS SECTION:

getErrors()
If an error occurred, returns an array of one or more database error objects providing the error code and description. If no error occurred, returns null.

getId()
Returns the ID of the sObject you were trying to undelete.

isSuccess()
Returns a Boolean value that is set to `true` if the DML operation was successful for this object, `false` otherwise.

`getErrors()`

If an error occurred, returns an array of one or more database error objects providing the error code and description. If no error occurred, returns null.

Signature

```
public Database.Error[] getErrors()
```

Return Value

Type: Database.Error[]

`getId()`

Returns the ID of the sObject you were trying to undelete.

Signature

```
public ID getId()
```

Return Value

Type: ID

Usage

If this field contains a value, the object was successfully undeleted. If this field is empty, the operation was not successful for that object.

`isSuccess()`

Returns a Boolean value that is set to `true` if the DML operation was successful for this object, `false` otherwise.

Signature

```
public Boolean isSuccess()
```

Return Value

Type: Boolean

UpsertResult Class

The result of an upsert DML operation returned by the `Database.upsert` method.

Namespace

Database

Usage

An array of Database.UpsertResult objects is returned with the `upsert` database method. Each element in the UpsertResult array corresponds to the sObject array passed as the *`sObject[]`* parameter in the `upsert` Database method; that is, the first element in the UpsertResult array matches the first element passed in the sObject array, the second element corresponds with the second element, and so on. If only one sObject is passed in, the UpsertResults array contains a single element.

UpsertResult Methods

The following are methods for `UpsertResult`. All are instance methods.

IN THIS SECTION:

getErrors()
If an error occurred, returns an array of one or more database error objects providing the error code and description. If no error occurred, returns an empty set.

getId()
Returns the ID of the sObject you were trying to update or insert.

isCreated()
A Boolean value that is set to `true` if the record was created, `false` if the record was updated.

isSuccess()
Returns a Boolean value that is set to `true` if the DML operation was successful for this object, `false` otherwise.

`getErrors()`

If an error occurred, returns an array of one or more database error objects providing the error code and description. If no error occurred, returns an empty set.

Signature

```
public Database.Error[] getErrors()
```

Return Value

Type: Database.Error []

`getId()`

Returns the ID of the sObject you were trying to update or insert.

Signature

```
public ID getId()
```

Return Value

Type: ID

Versioned Behavior Changes

In API version 53.0 and later, the method returns the sObject ID. However, if record locking fails during the update operation, the method returns a null value.

In API version 52.0 and earlier, the method returned a null value if the record wasn't updated successfully.

`isCreated()`

A Boolean value that is set to `true` if the record was created, `false` if the record was updated.

Signature

```
public Boolean isCreated()
```

Return Value

Type: Boolean

`isSuccess()`

Returns a Boolean value that is set to `true` if the DML operation was successful for this object, `false` otherwise.

Signature

```
public Boolean isSuccess()
```

Return Value

Type: Boolean

Datacloud Namespace

The `Datacloud` namespace provides classes and methods for retrieving information about duplicate rules. Duplicate rules let you control whether and when users can save duplicate records within Salesforce.

The `Datacloud` namespace is related to the Duplicate Management feature. For more information, see Manage Duplicate Records in *Salesforce Help* and Duplicate Management in Trailhead. The `Datacloud` namespace isn't related to the Salesforce Data Cloud product. See Data Cloud.

The following are the classes in the `Datacloud` namespace.

IN THIS SECTION:

AdditionalInformationMap Class
Represents other information, if any, about matched records.

DuplicateResult Class

Represents the details of a duplicate rule that detected duplicate records and information about those duplicate records.

FieldDiff Class

Represents the name of a matching rule field and how the values of the field compare for the duplicate and its matching record.

FindDuplicates Class

Performs rule-based searches for duplicate records. The input is an array of sObjects. Each sObject represents a record you want to find duplicates of. The output identifies the detected duplicates for each input sObject based on active duplicate rules for the given object.

FindDuplicatesByIds Class

Performs rule-based searches for duplicate records. The input is an array of IDs. Each ID specifies records to search for duplicates among. The duplicates are detected based on the active duplicate rules applicable to the object type corresponding to the input IDs.

FindDuplicatesResult Class

Output for rule-based searches for duplicate records. `FindDuplicatesResult` contains results of detecting duplicates using instances of `FindDuplicates` or `FindDuplicatesByIds` classes.

MatchRecord Class

Represents a duplicate record detected by a matching rule.

MatchResult Class

Represents the duplicate results for a matching rule.

AdditionalInformationMap Class

Represents other information, if any, about matched records.

Namespace

Datacloud

IN THIS SECTION:

AdditionalInformationMap Methods

AdditionalInformationMap Methods

The following are methods for `AdditionalInformationMap`.

IN THIS SECTION:

getName()

Returns the element name.

getValue()

Returns the value of the element.

`getName()`

Returns the element name.

Signature

```
public String getName()
```

Return Value

Type: String

`getValue()`

Returns the value of the element.

Signature

```
public String getValue()
```

Return Value

Type: String

DuplicateResult Class

Represents the details of a duplicate rule that detected duplicate records and information about those duplicate records.

Namespace

Datacloud

Usage

The `DuplicateResult` class and its methods are available to organizations that use duplicate rules.

`DuplicateResult` is contained within `DuplicateError`, which is part of `SaveResult`. `SaveResult` is generated when a user attempts to save a record in Salesforce.

Example

This example shows a custom application that lets users add a contact. When a contact is saved, an alert displays if there are duplicate records.

The sample application consists of a Visualforce page and an Apex controller. The Visualforce page is listed first so that you can see how the page makes use of the Apex controller. Save the Apex class first before saving the Visualforce page.

```
<apex:page controller="ContactDedupeController">
    <apex:form >
        <apex:pageBlock title="Duplicate Records" rendered="{!hasDuplicateResult}">
            <apex:pageMessages />
            <apex:pageBlockTable value="{!duplicateRecords}" var="item">
                <apex:column >
                    <apex:facet name="header">Name</apex:facet>
                 <apex:outputLink value="/{!item['Id']}">{!item['Name']}</apex:outputLink>
```

```
                </apex:column>
                <apex:column >
                    <apex:facet name="header">Owner</apex:facet>
                    <apex:outputField value="{!item['OwnerId']}"/>
                </apex:column>
                <apex:column >
                    <apex:facet name="header">Last Modified Date</apex:facet>
                    <apex:outputField value="{!item['LastModifiedDate']}"/>
                </apex:column>
            </apex:pageBlockTable>
        </apex:pageBlock>

        <apex:pageBlock title="Contact" mode="edit">
            <apex:pageBlockButtons >
                <apex:commandButton value="Save" action="{!save}"/>
            </apex:pageBlockButtons>

            <apex:pageBlockSection >
                <apex:inputField value="{!Contact.FirstName}"/>
                <apex:inputField value="{!Contact.LastName}"/>
                <apex:inputField value="{!Contact.Email}"/>
                <apex:inputField value="{!Contact.Phone}"/>
                <apex:inputField value="{!Contact.AccountId}"/>
            </apex:pageBlockSection>
        </apex:pageBlock>
    </apex:form>
</apex:page>
```

This sample is the Apex controller for the page. This controller contains the action method for the Save button. The `save` method inserts the new contact. If errors are returned, this method iterates through each error, checks if it's a duplicate error, adds the error message to the page, and returns information about the duplicate records to be displayed on the page.

```
public class ContactDedupeController {

    // Initialize a variable to hold the contact record you're processing
    private final Contact contact;

    // Initialize a list to hold any duplicate records
    private List<sObject> duplicateRecords;

    // Define variable that's true if there are duplicate records
    public boolean hasDuplicateResult{get;set;}

    // Define the constructor
    public ContactDedupeController() {

        // Define the values for the contact you're processing based on its ID
        Id id = ApexPages.currentPage().getParameters().get('id');
        this.contact = (id == null) ? new Contact() :
            [SELECT Id, FirstName, LastName, Email, Phone, AccountId
             FROM Contact WHERE Id = :id];

        // Initialize empty list of potential duplicate records
        this.duplicateRecords = new List<sObject>();
        this.hasDuplicateResult = false;
```

```
}

// Return contact and its values to the Visualforce page for display
public Contact getContact() {
    return this.contact;
}

// Return duplicate records to the Visualforce page for display
public List<sObject> getDuplicateRecords() {
    return this.duplicateRecords;
}

// Process the saved record and handle any duplicates
public PageReference save() {

    // Optionally, set DML options here, use "DML" instead of "false"
    //   in the insert()
    // Database.DMLOptions dml = new Database.DMLOptions();
    // dml.DuplicateRuleHeader.allowSave = true;
    // dml.DuplicateRuleHeader.runAsCurrentUser = true;
    Database.SaveResult saveResult = Database.insert(contact, false);

    if (!saveResult.isSuccess()) {
        for (Database.Error error : saveResult.getErrors()) {
            // If there are duplicates, an error occurs
            // Process only duplicates and not other errors
            //   (e.g., validation errors)
            if (error instanceof Database.DuplicateError) {
                // Handle the duplicate error by first casting it as a
                //   DuplicateError class
                // This lets you use methods of that class
                //  (e.g., getDuplicateResult())
                Database.DuplicateError duplicateError =
                        (Database.DuplicateError)error;
                Datacloud.DuplicateResult duplicateResult =
                        duplicateError.getDuplicateResult();

                // Display duplicate error message as defined in the duplicate rule
                ApexPages.Message errorMessage = new ApexPages.Message(
                        ApexPages.Severity.ERROR, 'Duplicate Error: ' +
                        duplicateResult.getErrorMessage());
                ApexPages.addMessage(errorMessage);

                // Get duplicate records
                this.duplicateRecords = new List<sObject>();

                // Return only match results of matching rules that
                //  find duplicate records
                Datacloud.MatchResult[] matchResults =
                        duplicateResult.getMatchResults();

                // Just grab first match result (which contains the
                //   duplicate record found and other match info)
                Datacloud.MatchResult matchResult = matchResults[0];
```

```
                    Datacloud.MatchRecord[] matchRecords = matchResult.getMatchRecords();


                    // Add matched record to the duplicate records variable
                    for (Datacloud.MatchRecord matchRecord : matchRecords) {
                        System.debug('MatchRecord: ' + matchRecord.getRecord());
                        this.duplicateRecords.add(matchRecord.getRecord());
                    }
                    this.hasDuplicateResult = !this.duplicateRecords.isEmpty();
                }
            }

            //If there's a duplicate record, stay on the page
            return null;
        }

        //  After save, navigate to the view page:
        return (new ApexPages.StandardController(contact)).view();
    }

}
```

IN THIS SECTION:

DuplicateResult Methods

SEE ALSO:

SaveResult Class

DuplicateError Class

DuplicateResult Methods

The following are methods for `DuplicateResult`.

IN THIS SECTION:

getDuplicateRule()

Returns the developer name of the executed duplicate rule that returned duplicate records.

getErrorMessage()

Returns the error message configured by the administrator to warn users they may be creating duplicate records. This message is associated with a duplicate rule.

getMatchResults()

Returns the duplicate records and match information.

isAllowSave()

Indicates whether the duplicate rule will allow a record that's identified as a duplicate to be saved. Set to `true` if duplicate rule should allow save; otherwise, `false`.

getDuplicateRule()

Returns the developer name of the executed duplicate rule that returned duplicate records.

Signature

```
public String getDuplicateRule()
```

Return Value

Type: String

getErrorMessage()

Returns the error message configured by the administrator to warn users they may be creating duplicate records. This message is associated with a duplicate rule.

Signature

```
public String getErrorMessage()
```

Return Value

Type: String

Example

This example shows the code used to display the error message when duplicates are found while saving a new contact. This code is part of a custom application that lets users add a contact. When a contact is saved, an alert displays if there are duplicate records. Review DuplicateResult Class on page 2435 to check out the entire sample applicaton.

```
ApexPages.Message errorMessage = new ApexPages.Message(
                            ApexPages.Severity.ERROR, 'Duplicate Error: ' +
                            duplicateResult.getErrorMessage());
                    ApexPages.addMessage(errorMessage);
```

getMatchResults()

Returns the duplicate records and match information.

Signature

```
public List<Datacloud.MatchResult> getMatchResults()
```

Return Value

Type: List<Datacloud.MatchResult>

Example

This example shows the code used to return duplicate record and match information and assign it to the `matchResults` variable. This code is part of a custom application that implements duplicate management when users add a contact. See DuplicateResult Class on page 2435 to check out the entire sample applicaton.

```
Datacloud.MatchResult[] matchResults =
                            duplicateResult.getMatchResults();
```

`isAllowSave()`

Indicates whether the duplicate rule will allow a record that's identified as a duplicate to be saved. Set to `true` if duplicate rule should allow save; otherwise, `false`.

Signature

```
public Boolean isAllowSave()
```

Return Value

Type: Boolean

FieldDiff Class

Represents the name of a matching rule field and how the values of the field compare for the duplicate and its matching record.

Namespace

Datacloud

IN THIS SECTION:

FieldDiff Methods

FieldDiff Methods

The following are methods for `FieldDiff`.

IN THIS SECTION:

getDifference()
Returns how the field values compare for the duplicate and its matching record.

getName()
Returns the name of a field on a matching rule that detected duplicates.

`getDifference()`

Returns how the field values compare for the duplicate and its matching record.

Signature

```
public String getDifference()
```

Return Value

Type: String

Possible values include:

- `SAME`: Indicates the field values match exactly.
- `DIFFERENT`: Indicates that the field values do not match.
- `NULL`: Indicates that the field values are a match because both values are blank.

`getName()`

Returns the name of a field on a matching rule that detected duplicates.

Signature

```
public String getName()
```

Return Value

Type: String

FindDuplicates Class

Performs rule-based searches for duplicate records. The input is an array of sObjects. Each sObject represents a record you want to find duplicates of. The output identifies the detected duplicates for each input sObject based on active duplicate rules for the given object.

Namespace

Datacloud

IN THIS SECTION:

FindDuplicates Methods

FindDuplicates Methods

The following are methods for `FindDuplicates`.

IN THIS SECTION:

findDuplicates(sObjects)
Identifies duplicates for sObjects provided and returns a list of `FindDuplicatesResult` objects.

`findDuplicates(sObjects)`

Identifies duplicates for sObjects provided and returns a list of `FindDuplicatesResult` objects.

Usage

Use `FindDuplicates` to apply active duplicate rules associated with an object to records represented by input sObjects.

`FindDuplicates` uses the duplicate rules for the object that has the same type as the input sObjects.

Input

- All sObjects in the input array must be of the same object type, and that type must correspond to an object type that supports duplicate rules.
- The input array is limited to 50 elements. If you exceed this limit, an exception is thrown with the following message: `Configuration error: The number of records to check is greater than the permitted batch size.`

Output

- The output of `FindDuplicates` is an array of objects with the same number of elements as the input array, and in the same order. The output objects encapsulate record IDs for duplicate records. The output objects also contain values from the duplicate records.
- Each element contains an array of `DuplicateResult` objects. If `FindDuplicates` doesn't find any duplicates, the `duplicateRule` field in `DuplicateResult` contains the name of the duplicate rule that `FindDuplicates` applied, but the `matchResults` array is empty.

Example

```
Account acct = new Account();
acct.Name = 'Acme';
acct.BillingStreet = '123 Fake St';
acct.BillingCity = 'Springfield';
acct.BillingState = 'VT';
acct.BillingCountry = 'US';

List<Account> acctList = new List<Account>();
acctList.add(acct);

if (Datacloud.FindDuplicates.findDuplicates(acctList).size() == 0) {
// If the new account doesn't have duplicates, insert it.
    insert(acct);
}
```

Signature

```
public static List<Datacloud.FindDuplicatesResult> findDuplicates(List<SObject> sObjects)
```

Parameters

sObjects

Type: List<SObject>

An array of sObjects for which you want to find duplicates.

Return Value

Type: List<FindDuplicatesResult>

FindDuplicatesByIds Class

Performs rule-based searches for duplicate records. The input is an array of IDs. Each ID specifies records to search for duplicates among. The duplicates are detected based on the active duplicate rules applicable to the object type corresponding to the input IDs.

Namespace

Datacloud

IN THIS SECTION:

FindDuplicatesByIds Methods

FindDuplicatesByIds Methods

The following are methods for `FindDuplicatesByIds`.

IN THIS SECTION:

findDuplicatesByIds(ids)
Identifies duplicates of sObjects provided and returns a list of `FindDuplicatesResult` objects.

`findDuplicatesByIds(ids)`

Identifies duplicates of sObjects provided and returns a list of `FindDuplicatesResult` objects.

Usage

Use `FindDuplicatesByIds` to apply active duplicate rules associated with an object to records represented by the record IDs.

`FindDuplicatesByIds` uses the duplicate rules for the object that has the same type as the input record IDs. For example, if the record ID represents an Account, `FindDuplicatesByIds` uses the duplicate rules associated with the Account object.

Input

- All record IDs in the input array must be of the same object type, and that type must correspond to an object type that supports duplicate rules.
- The input array is limited to 50 elements. If you exceed this limit, an exception is thrown with the following message: `Configuration error: The number of records to check is greater than the permitted batch size.`

Output

- The output of `FindDuplicatesByIds` is an array of objects with the same number of elements as the input array, and in the same order. The output objects encapsulate record IDs for duplicate records. The output objects also contain values from the duplicate records.
- Each element contains an array of `DuplicateResult` objects. If `FindDuplicatesByIds` doesn't find any duplicates, the `duplicateRule` field in `DuplicateResult` contains the name of the duplicate rule that `FindDuplicatesByIds` applied, but the `matchResults` array is empty.

Example

```
Account acct = new Account(name='Salesforce');
insert acct;

List<Id> idList = new List<Id>();
idList.add(acct.id);

if (Datacloud.FindDuplicatesByIds.findDuplicatesByIds(idList).size() > 0) {
System.debug('Found duplicates');
}
```

Signature

```
public static List<Datacloud.FindDuplicatesResult> findDuplicatesByIds(List<Id> ids)
```

Parameters

ids

Type: ListID>

A list of IDs for which you want to find duplicates.

Return Value

Type: List<FindDuplicatesResult>

FindDuplicatesResult Class

Output for rule-based searches for duplicate records. `FindDuplicatesResult` contains results of detecting duplicates using instances of `FindDuplicates` or `FindDuplicatesByIds` classes.

Namespace

Datacloud

IN THIS SECTION:

FindDuplicatesResult Properties

FindDuplicatesResult Methods

FindDuplicatesResult Properties

The following are properties for `FindDuplicatesResult`.

IN THIS SECTION:

duplicateresults

A list of `DuplicateResult` objects representing the results of calling `FindDuplicates.findDuplicates(sObjects)` or `FindDuplicatesByIds.findDuplicatesByIds(ids)`. Elements in the list correspond to sObjects or IDs in the input list.

errors

A list of `Database.Error` objects holding errors resulting from calling `FindDuplicates.findDuplicates(sObjects)` or `FindDuplicatesByIds.findDuplicatesByIds(ids)`.

success

Boolean signifying whether the call to `FindDuplicates.findDuplicates(sObjects)` or `FindDuplicatesByIds.findDuplicatesByIds(ids)` was successful.

duplicateresults

A list of `DuplicateResult` objects representing the results of calling `FindDuplicates.findDuplicates(sObjects)` or `FindDuplicatesByIds.findDuplicatesByIds(ids)`. Elements in the list correspond to sObjects or IDs in the input list.

Signature

```
public List<Datacloud.DuplicateResult> duplicateresults
```

Property Value

Type: List<DuplicateResult>

errors

A list of `Database.Error` objects holding errors resulting from calling `FindDuplicates.findDuplicates(sObjects)` or `FindDuplicatesByIds.findDuplicatesByIds(ids)`.

Signature

```
public List<Database.Error> errors {get; set;}
```

Property Value

Type: List<Database.Error>

success

Boolean signifying whether the call to `FindDuplicates.findDuplicates(sObjects)` or `FindDuplicatesByIds.findDuplicatesByIds(ids)` was successful.

Signature

```
public Boolean success {get; set;}
```

Property Value

Type: Boolean

FindDuplicatesResult Methods

The following are methods for `FindDuplicatesResult`.

IN THIS SECTION:

getDuplicateResults()
Returns a list of `DuplicateResult` objects representing the results of calling `FindDuplicates.findDuplicates(sObjects)` or `FindDuplicatesByIds.findDuplicatesByIds(ids)`. Elements in the list correspond to sObjects or IDs in the input list.

getErrors()
Returns a list of `DatabaseError` objects containing errors resulting from calling `FindDuplicates.findDuplicates(sObjects)` or `FindDuplicatesByIds.findDuplicatesByIds(ids)`, if errors were encountered.

isSuccess()
Returns a Boolean signifying whether the call to `FindDuplicates.findDuplicates(sObjects)` or `FindDuplicatesByIds.findDuplicatesByIds(ids)` was successful.

getDuplicateResults()

Returns a list of `DuplicateResult` objects representing the results of calling `FindDuplicates.findDuplicates(sObjects)` or `FindDuplicatesByIds.findDuplicatesByIds(ids)`. Elements in the list correspond to sObjects or IDs in the input list.

Example

```
Account acct = new Account(name='Salesforce');
List<Account> acctList = new List<Account>();
acctList.add(acct);

Datacloud.FindDuplicatesResult[] results = Datacloud.FindDuplicates.findDuplicates(acctList);
for (Datacloud.FindDuplicatesResult findDupeResult : results) {
  for (Datacloud.DuplicateResult dupeResult : findDupeResult.getDuplicateResults()) {
    for (Datacloud.MatchResult matchResult : dupeResult.getMatchResults()) {
      for (Datacloud.MatchRecord matchRecord : matchResult.getMatchRecords()) {
          System.debug('Duplicate Record: ' + matchRecord.getRecord());
      }
    }
  }
}
```

Signature

```
public List<Datacloud.DuplicateResult> getDuplicateResults()
```

Return Value

Type: List<DuplicateResult>

getErrors()

Returns a list of `DatabaseError` objects containing errors resulting from calling `FindDuplicates.findDuplicates(sObjects)` or `FindDuplicatesByIds.findDuplicatesByIds(ids)`, if errors were encountered.

Signature

```
public List<Database.Error> getErrors()
```

Return Value

Type: List<Database.Error>

`isSuccess()`

Returns a Boolean signifying whether the call to `FindDuplicates.findDuplicates(sObjects)` or `FindDuplicatesByIds.findDuplicatesByIds(ids)` was successful.

Signature

```
public Boolean isSuccess()
```

Return Value

Type: Boolean

MatchRecord Class

Represents a duplicate record detected by a matching rule.

Namespace

Datacloud

IN THIS SECTION:

MatchRecord Methods

MatchRecord Methods

The following are methods for `MatchRecord`.

IN THIS SECTION:

getAdditionalInformation()
Returns other information about a matched record. For example, a `matchGrade` represents the quality of the data for the D&B fields in the matched record.

getFieldDiffs()
Returns all matching rule fields and how each field value compares for the duplicate and its matching record.

getMatchConfidence()
Returns the ranking of how similar a matched record's data is to the data in your request. Must be equal to or greater than the value of the `minMatchConfidence` specified in your request. Returns -1 if unused.

getRecord()
Returns the fields and field values for the duplicate.

getAdditionalInformation()

Returns other information about a matched record. For example, a `matchGrade` represents the quality of the data for the D&B fields in the matched record.

Signature

```
public List<Datacloud.AdditionalInformationMap> getAdditionalInformation()
```

Return Value

Type: List<Datacloud.AdditionalInformationMap>

getFieldDiffs()

Returns all matching rule fields and how each field value compares for the duplicate and its matching record.

Signature

```
public List<Datacloud.FieldDiff> getFieldDiffs()
```

Return Value

Type: List<Datacloud.FieldDiff>

getMatchConfidence()

Returns the ranking of how similar a matched record's data is to the data in your request. Must be equal to or greater than the value of the `minMatchConfidence` specified in your request. Returns -1 if unused.

Signature

```
public Double getMatchConfidence()
```

Return Value

Type: Double

getRecord()

Returns the fields and field values for the duplicate.

Signature

```
public SObject getRecord()
```

Return Value

Type: SObject

MatchResult Class

Represents the duplicate results for a matching rule.

Namespace

Datacloud

IN THIS SECTION:

MatchResult Methods

MatchResult Methods

The following are methods for `MatchResult`.

IN THIS SECTION:

getEntityType()
Returns the entity type of the matching rule.

getErrors()
Returns errors that occurred during matching for the matching rule.

getMatchEngine()
Returns the match engine for the matching rule.

getMatchRecords()
Returns information about the duplicates for the matching rule.

getRule()
Returns the developer name of the matching rule.

getSize()
Returns the number of duplicates detected by the matching rule.

isSuccess()
Returns `false` if there's an error with the matching rule, and `true` if the matching rule successfully ran.

`getEntityType()`

Returns the entity type of the matching rule.

Signature

```
public String getEntityType()
```

Return Value

Type: String

`getErrors()`

Returns errors that occurred during matching for the matching rule.

Signature

```
public List<Database.Error> getErrors()
```

Return Value

Type: List<Database.Error>

`getMatchEngine()`

Returns the match engine for the matching rule.

Signature

```
public String getMatchEngine()
```

Return Value

Type: String

`getMatchRecords()`

Returns information about the duplicates for the matching rule.

Signature

```
public List<Datacloud.MatchRecord> getMatchRecords()
```

Return Value

Type: List<Datacloud.MatchRecord>

`getRule()`

Returns the developer name of the matching rule.

Signature

```
public String getRule()
```

Return Value

Type: String

`getSize()`

Returns the number of duplicates detected by the matching rule.

Signature

```
public Integer getSize()
```

Return Value

Type: Integer

`isSuccess()`

Returns `false` if there's an error with the matching rule, and `true` if the matching rule successfully ran.

Signature

```
public Boolean isSuccess()
```

Return Value

Type: Boolean

DataRetrieval Namespace

The `DataRetrieval` namespace provides classes and methods to record details of customer-agent engagements, as well as transcripts of their conversations.

The following are the classes in the `DataRetrieval` namespace.

Engagement Class

Engagements Class

EngagementRecordDetails Class

EngagementRecordDetailsList Class

FieldDetailsRepresentation Class

ObjectDetailsRepresentation Class

RecordDetailsRepresentation Class

RecordTranscripts Class

RecordTranscriptsList Class

Transcript Class

DataSource Namespace

The `DataSource` namespace provides the classes for the Apex Connector Framework. Use the Apex Connector Framework to develop a custom adapter for Salesforce Connect. Then connect your Salesforce organization to any data anywhere via the Salesforce Connect custom adapter.

The following are the classes in the `DataSource` namespace.

IN THIS SECTION:

AsyncDeleteCallback Class

A callback class that the `Database.deleteAsync` method references. Salesforce calls this class after the remote `deleteAsync` operation is completed. This class provides the compensating transaction in the completion context of the delete operation. Extend this class to define the actions to execute after the remote delete operation finishes execution.

AsyncSaveCallback Class

A callback class that the `Database.insertAsync` or `Database.updateAsync` method references. Salesforce calls this class after the remote operation is completed. This class provides the compensating transaction in the completion context of the insert or update operation. Extend this class to define the actions to execute after the remote insert or update operation finishes execution.

AuthenticationCapability Enum

Specifies the types of authentication that can be used to access the external system.

AuthenticationProtocol Enum

Determines what type of credentials are used to authenticate to the external system.

Capability Enum

Declares which functional operations the external system supports. Also specifies required endpoint settings for the external data source definition.

Column Class

Describes a column on a `DataSource.Table`. This class extends the `DataSourceUtil` class and inherits its methods.

ColumnSelection Class

Identifies the list of columns to return during a query or search.

Connection Class

Extend this class to enable your Salesforce org to sync the external system's schema and to handle queries, searches, and write operations (upsert and delete) of the external data. This class extends the `DataSourceUtil` class and inherits its methods.

ConnectionParams Class

Contains the credentials for authenticating to the external system.

DataSourceUtil Class

Parent class for the `DataSource.Provider`, `DataSource.Connection`, `DataSource.Table`, and `DataSource.Column` classes.

DataType Enum

Specifies the data types that are supported by the Apex Connector Framework.

DeleteContext Class

An instance of `DeleteContext` is passed to the `deleteRows()` method on your `Database.Connection` class. The class provides context information about the delete request to the implementor of `deleteRows()`.

DeleteResult Class

Represents the result of a delete operation on an sObject record. The result is returned by the `DataSource.deleteRows` method of the `DataSource.Connection` class.

Filter Class

Represents a `WHERE` clause in a SOSL or SOQL query.

FilterType Enum

Referenced by the `type` property on a `DataSource.Filter`.

IdentityType Enum

Determines which set of credentials is used to authenticate to the external system.

Order Class

Contains details about how to sort the rows in the result set. Equivalent to an `ORDER BY` statement in a SOQL query.

OrderDirection Enum

Specifies the direction for sorting rows based on column values.

Provider Class

Extend this base class to create a custom adapter for Salesforce Connect. The class informs Salesforce of the functional and authentication capabilities that are supported by or required to connect to the external system. This class extends the `DataSourceUtil` class and inherits its methods.

QueryAggregation Enum

Specifies how to aggregate a column in a query.

QueryContext Class

An instance of `QueryContext` is provided to the `query` method on your `DataSource.Connection` class. The instance corresponds to a SOQL request.

QueryUtils Class

Contains helper methods to locally filter, sort, and apply limit and offset clauses to data rows. This helper class is provided for your convenience during early development and tests, but it isn't supported for use in production environments.

ReadContext Class

Abstract base class for the `QueryContext` and `SearchContext` classes.

SearchContext Class

An instance of `SearchContext` is provided to the search method on your `DataSource.Connection` class. The instance corresponds to a search or SOSL request.

SearchUtils Class

Helper class for implementing search on a custom adapter for Salesforce Connect.

Table Class

Describes a table on an external system that the Salesforce Connect custom adapter connects to. This class extends the `DataSourceUtil` class and inherits its methods.

TableResult Class

Contains the results of a search or query.

TableSelection Class

Contains a breakdown of the SOQL or SOSL query. Its properties represent the FROM, ORDER BY, SELECT, and WHERE clauses in the query.

UpsertContext Class

An instance of `UpsertContext` is passed to the `upsertRows()` method on your `Datasource.Connection` class. This class provides context information about the upsert request to the implementor of `upsertRows()`.

UpsertResult Class

Represents the result of an upsert operation on an external object record. The result is returned by the `upsertRows` method of the `DataSource.Connection` class.

DataSource Exceptions

The `DataSource` namespace contains exception classes.

AsyncDeleteCallback Class

A callback class that the `Database.deleteAsync` method references. Salesforce calls this class after the remote `deleteAsync` operation is completed. This class provides the compensating transaction in the completion context of the delete operation. Extend this class to define the actions to execute after the remote delete operation finishes execution.

Namespace

DataSource

IN THIS SECTION:

AsyncDeleteCallback Methods

AsyncDeleteCallback Methods

The following are methods for `AsyncDeleteCallback`.

IN THIS SECTION:

processDelete(deleteResult)
Override this method to define actions that Salesforce executes after a remote `Database.deleteAsync` operation is completed. For example, based on the results of the remote operation, you can update custom object data or other data that's stored in the Salesforce org..

`processDelete(deleteResult)`

Override this method to define actions that Salesforce executes after a remote `Database.deleteAsync` operation is completed. For example, based on the results of the remote operation, you can update custom object data or other data that's stored in the Salesforce org..

Signature

```
public void processDelete(Database.DeleteResult deleteResult)
```

Parameters

`deleteResult`
Type: Database.DeleteResult

The result of the asynchronous delete operation.

Return Value

Type: void

AsyncSaveCallback Class

A callback class that the `Database.insertAsync` or `Database.updateAsync` method references. Salesforce calls this class after the remote operation is completed. This class provides the compensating transaction in the completion context of the insert or update operation. Extend this class to define the actions to execute after the remote insert or update operation finishes execution.

Namespace

DataSource

IN THIS SECTION:

AsyncSaveCallback Methods

AsyncSaveCallback Methods

The following are methods for `AsyncSaveCallback`.

IN THIS SECTION:

processSave(saveResult)
Override this method to define actions that Salesforce executes after the remote `Database.insertAsync` or `Database.updateAsync` operation is completed. For example, based on the results of the remote operation, you can update custom object data or other data that's stored in the Salesforce org.

`processSave(saveResult)`

Override this method to define actions that Salesforce executes after the remote `Database.insertAsync` or `Database.updateAsync` operation is completed. For example, based on the results of the remote operation, you can update custom object data or other data that's stored in the Salesforce org.

Signature

```
public void processSave(Database.SaveResult saveResult)
```

Parameters

`saveResult`
Type: Database.SaveResult

The result of the asynchronous insert or update operation.

Return Value

Type: void

AuthenticationCapability Enum

Specifies the types of authentication that can be used to access the external system.

Usage

The `DataSource.Provider` class returns `DataSource.AuthenticationCapability` enum values. The returned values determine which authentication settings are available on the external data source definition in Salesforce.

If you set up callouts in your `DataSource.Connection` class, you can specify the callout endpoints as named credentials instead of URLs. If you do so for all callouts, return `ANONYMOUS` as the sole entry in the list of data source authentication capabilities. That way, the external data source definition doesn't require authentication settings. Salesforce manages all authentication for Apex callouts that specify a named credential as the callout endpoint so that your code doesn't have to.

Enum Values

The following are the values of the `DataSource.AuthenticationCapability` enum.

Value	Description
`ANONYMOUS`	No credentials are required to authenticate to the external system.
`BASIC`	A username and password can be used to authenticate to the external system.
`CERTIFICATE`	A security certificate can be supplied when establishing each connection to the external system.
`OAUTH`	OAuth can be used to authenticate to the external system.

AuthenticationProtocol Enum

Determines what type of credentials are used to authenticate to the external system.

Enum Values

The following are the values of the `DataSource.AuthenticationProtocol` enum.

Value	Description
`NONE`	No credentials are used to authenticate to the external system.
`OAUTH`	OAuth 2.0 is used to authenticate to the external system.
`PASSWORD`	A username and password are used to authenticate to the external system.

Capability Enum

Declares which functional operations the external system supports. Also specifies required endpoint settings for the external data source definition.

Usage

The `DataSource.Provider` class returns `DataSource.Capability` enum values, which:

- Specify the functional capabilities of the external system.

- Determine which endpoint settings are available on the external data source definition in Salesforce.

Enum Values

The following are the values of the `DataSource.Capability` enum.

Value	Description
`MULTI_PICKLIST`	The external system supports multi-picklist fields.
`PICKLIST`	The external system supports picklist fields.
`QUERY_PAGINATION_SERVER_DRIVEN`	With server-driven paging, the external system determines the page sizes and batch boundaries. The external system's paging settings can optimize the external system's performance and improve the load times for external objects in your org. Also, the external data set can change while your users or the Lightning Platform are paging through the result set. Typically, server-driven paging adjusts batch boundaries to accommodate changing data sets more effectively than client-driven paging. If you enable server-driven paging on an external data source, Salesforce ignores the requested page sizes, including the default `queryMore()` batch size of 500 rows. The pages returned by the external system determine the batches, but each page can't exceed 2,000 rows. Also, the Apex code must generate a query token and use it to determine and fetch the next batch of results.
`QUERY_TOTAL_SIZE`	The external system can provide the total number of rows that meet the query criteria, even when requested to return a smaller batch size. This capability enables you to simplify how you paginate results by using `queryMore()`.
`REQUIRE_ENDPOINT`	Requires the administrator to specify the endpoint in the URL field in the external data source definition.
`REQUIRE_HTTPS`	Requires the endpoint URL to use secure HTTP. If `REQUIRE_ENDPOINT` isn't declared, `REQUIRE_HTTPS` is ignored.
`ROW_CREATE`	Allows creating of external data.
`ROW_DELETE`	Allows deleting external data.
`ROW_QUERY`	Allows API and SOQL queries of the external data. Also allows reports on the external objects.
`ROW_UPDATE`	Allows updating external data.
`SEARCH`	Allows SOSL and Salesforce searches of the external data. When the custom adapter declares the `SEARCH` capability, you can control which external objects are searchable by selecting or deselecting **Allow Search** on each external object. However, syncing always overwrites the external object's search status to match the search status of the external data source.

Value	Description
	Only text, text area, and long text area fields on external objects can be searched. If an external object has no searchable fields, searches on that object return no records.

SEE ALSO:

Salesforce Help: Validate and Sync an External Data Source

Column Class

Describes a column on a `DataSource.Table`. This class extends the `DataSourceUtil` class and inherits its methods.

Namespace

DataSource

Usage

A list of column metadata is provided by the `DataSource.Connection` class when the `sync()` method is invoked. Each column can become a field on an external object.

The metadata is stored in Salesforce. Updating the Apex code to return new or updated values for the column metadata doesn't automatically update the stored metadata in Salesforce.

IN THIS SECTION:

Column Properties

Column Methods

Column Properties

The following are properties for `Column`.

IN THIS SECTION:

decimalPlaces
If the data type is numeric, the number of decimal places to the right of the decimal point.

description
Description of what the column represents.

filterable
Whether a result set can be filtered based on the values of the column.

isPicklistAlphabeticallySorted
Returns `true` if the picklist is sorted alphabetically, `false` otherwise.

isPicklistRestricted
Returns `true` if the picklist is restricted, `false` otherwise.

label

User-friendly name for the column that appears in the Salesforce user interface.

length

If the column is a string data type, the number of characters in the column. If the column is a numeric data type, the total number of digits on both sides of the decimal point, but excluding the decimal point.

name

Name of the column in the external system.

picklistValues

If the data type is a picklist, the picklist values.

referenceTargetField

API name of the custom field on the parent object whose values are compared against this column's values. Matching values identify related records in an indirect lookup relationship. Applies only when the column's data type is `INDIRECT_LOOKUP_TYPE`. For other data types, this value is ignored.

referenceTo

API name of the parent object in the relationship that's represented by this column. Applies only when the column's data type is `LOOKUP_TYPE`, `EXTERNAL_LOOKUP_TYPE`, or `INDIRECT_LOOKUP_TYPE`. For other data types, this value is ignored.

sortable

Whether a result set can be sorted based on the values of the column via an `ORDER BY` clause.

type

Data type of the column.

decimalPlaces

If the data type is numeric, the number of decimal places to the right of the decimal point.

Signature

```
public Integer decimalPlaces {get; set;}
```

Property Value

Type: Integer

description

Description of what the column represents.

Signature

```
public String description {get; set;}
```

Property Value

Type: String

filterable

Whether a result set can be filtered based on the values of the column.

Signature

```
public Boolean filterable {get; set;}
```

Property Value

Type: Boolean

isPicklistAlphabeticallySorted

Returns `true` if the picklist is sorted alphabetically, `false` otherwise.

Signature

```
public Boolean isPicklistAlphabeticallySorted {get; set;}
```

Property Value

Type: Boolean

isPicklistRestricted

Returns `true` if the picklist is restricted, `false` otherwise.

Signature

```
public Boolean isPicklistRestricted {get; set;}
```

Property Value

Type: Boolean

label

User-friendly name for the column that appears in the Salesforce user interface.

Signature

```
public String label {get; set;}
```

Property Value

Type: String

`length`

If the column is a string data type, the number of characters in the column. If the column is a numeric data type, the total number of digits on both sides of the decimal point, but excluding the decimal point.

Signature

```
public Integer length {get; set;}
```

Property Value

Type: Integer

`name`

Name of the column in the external system.

Signature

```
public String name {get; set;}
```

Property Value

Type: String

`picklistValues`

If the data type is a picklist, the picklist values.

Signature

```
public List<Map<String,String>> picklistValues {get; set;}
```

Property Value

Type: List<Map<String,String>>

`referenceTargetField`

API name of the custom field on the parent object whose values are compared against this column's values. Matching values identify related records in an indirect lookup relationship. Applies only when the column's data type is `INDIRECT_LOOKUP_TYPE`. For other data types, this value is ignored.

Signature

```
public String referenceTargetField {get; set;}
```

Property Value

Type: String

referenceTo

API name of the parent object in the relationship that's represented by this column. Applies only when the column's data type is `LOOKUP_TYPE`, `EXTERNAL_LOOKUP_TYPE`, or `INDIRECT_LOOKUP_TYPE`. For other data types, this value is ignored.

Signature

```
public String referenceTo {get; set;}
```

Property Value

Type: String

sortable

Whether a result set can be sorted based on the values of the column via an `ORDER BY` clause.

Signature

```
public Boolean sortable {get; set;}
```

Property Value

Type: Boolean

type

Data type of the column.

Signature

```
public DataSource.DataType type {get; set;}
```

Property Value

Type: DataSource.DataType

Column Methods

The following are methods for `Column`.

IN THIS SECTION:

boolean(name)
Returns a new column of data type `BOOLEAN_TYPE`.

currency(name, length, decimalPlaces)
Returns a new column of data type `CURRENCY_TYPE`.

date(name)
Returns a new column of data type `DATE_TYPE`.

datetime(name)

Returns a new column of data type `DATETIME_TYPE`.

email(name)

Returns a new column of data type `EMAIL_TYPE`.

externalLookup(name, domain)

Returns a new column of data type `EXTERNAL_LOOKUP_TYPE`.

get(name, label, description, isSortable, isFilterable, type, length, decimalPlaces, referenceTo, referenceTargetField, picklistValuesObj, isPicklistAlphabeticallySorted, isPicklistRestricted)

Returns a new column with the 13 specified `Column` property values.

get(name, label, description, isSortable, isFilterable, type, length, decimalPlaces, referenceTo, referenceTargetField)

Returns a new column with the ten specified `Column` property values.

get(name, label, description, isSortable, isFilterable, type, length, decimalPlaces)

Returns a new column with the eight specified `Column` property values.

get(name, label, description, isSortable, isFilterable, type, length)

Returns a new column with the seven specified `Column` property values.

indirectLookup(name, domain, targetField)

Returns a new column of data type `INDIRECT_LOOKUP_TYPE`.

integer(name, length)

Returns a new numeric column with no decimal places using the specified name and length.

lookup(name, domain)

Returns a new column of data type `LOOKUP_TYPE`.

multipicklist(name, picklistValues, isPicklistAlphabeticallySorted, isPicklistRestricted)

Returns a new column of data type `PICKLIST_MULTISELECT_TYPE` with the specified name and picklist values. You can also specify whether the picklist is sorted alphabetically or if the picklist is restricted.

multipicklist(name, picklistValues)

Returns a new column of data type `PICKLIST_MULTISELECT_TYPE` with the specified name and picklist values.

number(name, length, decimalPlaces)

Returns a new column of data type `NUMBER_TYPE`.

percent(name, length, decimalPlaces)

Returns a new column of data type `PERCENT_TYPE`.

phone(name)

Returns a new column of data type `PHONE_TYPE`.

picklist(name, picklistValues, isPicklistAlphabeticallySorted, isPicklistRestricted)

Returns a new column of data type `PICKLIST_TYPE` with the specified name and picklist values. You can also specify whether the picklist is sorted alphabetically or if the picklist is restricted.

picklist(name, picklistValues)

Returns a new column of data type `PICKLIST_TYPE` with the specified name and picklist values.

text(name, label, length)

Returns a new column of data type `STRING_SHORT_TYPE` or `STRING_LONG_TYPE`, with the specified name, label, and length.

text(name, length)
Returns a new column of data type `STRING_SHORT_TYPE` or `STRING_LONG_TYPE`, with the specified name and length.

text(name)
Returns a new column of data type `STRING_SHORT_TYPE` with the specified name and the length of 255 characters.

textarea(name)
Returns a new column of data type `STRING_LONG_TYPE` with the specified name and the length of 32,000 characters.

time(name)
Returns a new column of data type `Time` with the specified name.

url(name, length)
Returns a new column of data type `URL_TYPE` with the specified name and length.

url(name)
Returns a new column of data type `URL_TYPE` with the specified name and the length of 1,000 characters.

boolean(name)

Returns a new column of data type `BOOLEAN_TYPE`.

Signature

```
public static DataSource.Column boolean(String name)
```

Parameters

name
Type: String
Name of the column.

Return Value

Type: DataSource.Column

currency(name, length, decimalPlaces)

Returns a new column of data type `CURRENCY_TYPE`.

Signature

```
public static DataSource.Column currency(String name, Integer length, Integer
decimalPlaces)
```

Parameters

name
Type: String
Name of the column.

length
Type: Integer

Number of characters allowed in the column.

decimalPlaces
Type: Integer

Number of decimal places to the right of the decimal point.

Return Value

Type: DataSource.Column

date(name)

Returns a new column of data type DATE_TYPE.

Signature

```
public static DataSource.Column date(String name)
```

Parameters

name
Type: String

Name of the column.

Return Value

Type: DataSource.Column

datetime(name)

Returns a new column of data type DATETIME_TYPE.

Signature

```
public static DataSource.Column datetime(String name)
```

Parameters

name
Type: String

Name of the column.

Return Value

Type: DataSource.Column

email(name)

Returns a new column of data type EMAIL_TYPE.

Signature

```
public static DataSource.Column email(String name)
```

Parameters

name
Type: String

Name of the column.

Return Value

Type: DataSource.Column

externalLookup(name, domain)

Returns a new column of data type `EXTERNAL_LOOKUP_TYPE`.

Signature

```
public static DataSource.Column externalLookup(String name, String domain)
```

Parameters

name
Type: String

Name of the column.

domain
Type: String

API name of the parent object in the external lookup relationship.

Return Value

Type: DataSource.Column

The returned column has these property values.

Property	Value
name	*name*
label	*name*
description	*name*
isSortable	true
isFilterable	true
type	DataSource.DataType.EXTERNAL_LOOKUP_TYPE
length	255

Property	Value
decimalPlaces	0
referenceTo	*domain*
referenceTargetField	null

get(name, label, description, isSortable, isFilterable, type, length, decimalPlaces, referenceTo, referenceTargetField, picklistValuesObj, isPicklistAlphabeticallySorted, isPicklistRestricted)

Returns a new column with the 13 specified `Column` property values.

Signature

```
public static DataSource.Column get(String name, String label, String description,
Boolean isSortable, Boolean isFilterable, DataSource.DataType type, Integer length,
Integer decimalPlaces, String referenceTo, String referenceTargetField, Object
picklistValuesObj, Boolean isPicklistAlphabeticallySorted, Boolean isPicklistRestricted)
```

Parameters

See Column Properties on page 2458 for information about each parameter.

name
Type: String

label
Type: String

description
Type: String

isSortable
Type: Boolean

isFilterable
Type: Boolean

type
Type: DataSource.DataType

length
Type: Integer

decimalPlaces
Type: Integer

referenceTo
Type: String

referenceTargetField
Type: String

picklistValuesObj
Type: Object

isPicklistAlphabeticallySorted
Type: Boolean

isPicklistRestricted
Type: Boolean

Return Value

Type: DataSource.Column

get(name, label, description, isSortable, isFilterable, type, length, decimalPlaces, referenceTo, referenceTargetField)

Returns a new column with the ten specified `Column` property values.

Signature

```
public static DataSource.Column get(String name, String label, String description,
Boolean isSortable, Boolean isFilterable, DataSource.DataType type, Integer length,
Integer decimalPlaces, String referenceTo, String referenceTargetField)
```

Parameters

See Column Properties on page 2458 for information about each parameter.

name
Type: String

label
Type: String

description
Type: String

isSortable
Type: Boolean

isFilterable
Type: Boolean

type
Type: DataSource.DataType

length
Type: Integer

decimalPlaces
Type: Integer

referenceTo
Type: String

referenceTargetField
Type: String

Return Value

Type: DataSource.Column

get(name, label, description, isSortable, isFilterable, type, length, decimalPlaces)

Returns a new column with the eight specified `Column` property values.

Signature

```
public static DataSource.Column get(String name, String label, String description,
Boolean isSortable, Boolean isFilterable, DataSource.DataType type, Integer length,
Integer decimalPlaces)
```

Parameters

See Column Properties on page 2458 for information about each parameter.

name
Type: String

label
Type: String

description
Type: String

isSortable
Type: Boolean

isFilterable
Type: Boolean

type
Type: DataSource.DataType

length
Type: Integer

decimalPlaces
Type: Integer

Return Value

Type: DataSource.Column

get(name, label, description, isSortable, isFilterable, type, length)

Returns a new column with the seven specified `Column` property values.

Signature

```
public static DataSource.Column get(String name, String label, String description,
Boolean isSortable, Boolean isFilterable, DataSource.DataType type, Integer length)
```

Parameters

See Column Properties on page 2458 for information about each parameter.

name
Type: String

label
Type: String

description
Type: String

isSortable
Type: Boolean

isFilterable
Type: Boolean

type
Type: DataSource.DataType

length
Type: Integer

Return Value

Type: DataSource.Column

indirectLookup(name, domain, targetField)

Returns a new column of data type `INDIRECT_LOOKUP_TYPE`.

Signature

```
public static DataSource.Column indirectLookup(String name, String domain, String targetField)
```

Parameters

name
Type: String

Name of the column.

domain
Type: String

API name of the parent object in the indirect lookup relationship.

targetField
Type: String

API name of the custom field on the parent object whose values are compared against this column's values. Matching values identify related records in an indirect lookup relationship.

Return Value

Type: DataSource.Column

The returned column has these property values.

Property	Value
name	*name*
label	*name*
description	*name*
isSortable	true
isFilterable	true
type	DataSource.DataType.INDIRECT_LOOKUP_TYPE
length	255
decimalPlaces	0
referenceTo	*domain*
referenceTargetField	*targetField*

integer(name, length)

Returns a new numeric column with no decimal places using the specified name and length.

Signature

```
public static DataSource.Column integer(String name, Integer length)
```

Parameters

name
Type: String

The column name.

length
Type: Integer

The column length.

Return Value

Type: DataSource.Column

lookup(name, domain)

Returns a new column of data type `LOOKUP_TYPE`.

Signature

```
public static DataSource.Column lookup(String name, String domain)
```

Parameters

name
Type: String

Name of the column.

domain
Type: String

API name of the parent object in the lookup relationship.

Return Value

Type: DataSource.Column

The returned column has these property values.

Property	Value
name	*name*
label	*name*
description	*name*
isSortable	true
isFilterable	true
type	DataSource.DataType.LOOKUP_TYPE
length	255
decimalPlaces	0
referenceTo	*domain*
referenceTargetField	null

multipicklist(name, picklistValues, isPicklistAlphabeticallySorted, isPicklistRestricted)

Returns a new column of data type `PICKLIST_MULTISELECT_TYPE` with the specified name and picklist values. You can also specify whether the picklist is sorted alphabetically or if the picklist is restricted.

Signature

```
public static DataSource.Column multipicklist(String name, List<Map<String,String>>
picklistValues, Boolean isPicklistAlphabeticallySorted, Boolean isPicklistRestricted)
```

Parameters

name
Type: String

Name of the column.

picklistValues
Type: List<Map<String,String>>

isPicklistAlphabeticallySorted
Indicates whether the picklist is sorted alphabetically.

isPicklistRestricted
Type: Boolean

Indicates whether the picklist is restricted.

Return Value

Type: DataSource.Column

multipicklist(name, picklistValues)

Returns a new column of data type `PICKLIST_MULTISELECT_TYPE` with the specified name and picklist values.

Signature

```
public static DataSource.Column multipicklist(String name, List<Map<String,String>>
picklistValues)
```

Parameters

name
Type: String

Name of the column.

picklistValues
Type: List<Map<String,String>>

List of picklist values.

Return Value

Type: DataSource.Column

number(name, length, decimalPlaces)

Returns a new column of data type `NUMBER_TYPE`.

Signature

```
public static DataSource.Column number(String name, Integer length, Integer
decimalPlaces)
```

Parameters

See Column Properties on page 2458 for information about each parameter.

name
Type: String

length
Type: Integer

decimalPlaces
Type: Integer

Return Value

Type: DataSource.Column

The returned column has these property values.

Property	Value
name	*name*
label	*name*
description	*name*
isSortable	true
isFilterable	true
type	DataSource.DataType.NUMBER_TYPE
length	*length*
decimalPlaces	*decimalPlaces*

percent(name, length, decimalPlaces)

Returns a new column of data type `PERCENT_TYPE`.

Signature

```
public static DataSource.Column percent(String name, Integer length, Integer
decimalPlaces)
```

Parameters

name
Type: String

Name of the column.

length
Type: Integer

Number of characters allowed in the column.

decimalPlaces
Type: Integer

Number of decimal places to the right of the decimal point.

Return Value

Type: DataSource.Column

phone(name)

Returns a new column of data type `PHONE_TYPE`.

Signature

```
public static DataSource.Column phone(String name)
```

Parameters

name
Type: String

Name of the column.

Return Value

Type: DataSource.Column

picklist(name, picklistValues, isPicklistAlphabeticallySorted, isPicklistRestricted)

Returns a new column of data type `PICKLIST_TYPE` with the specified name and picklist values. You can also specify whether the picklist is sorted alphabetically or if the picklist is restricted.

Signature

```
public static DataSource.Column picklist(String name, List<Map<String,String>>
picklistValues, Boolean isPicklistAlphabeticallySorted, Boolean isPicklistRestricted)
```

Parameters

name
Type: String

Name of the column.

picklistValues
Type: List<Map<String,String>>

List of picklist values.

isPicklistAlphabeticallySorted
Indicates whether the picklist is sorted alphabetically.

isPicklistRestricted
Type: Boolean

Indicates whether the picklist is restricted.

Return Value

Type: DataSource.Column

picklist(name, picklistValues)

Returns a new column of data type `PICKLIST_TYPE` with the specified name and picklist values.

Signature

```
public static DataSource.Column picklist(String name, List<Map<String,String>>
picklistValues)
```

Parameters

name
Type: String

Name of the column.

picklistValues
Type: List<Map<String,String>>

List of picklist values.

Return Value

Type: DataSource.Column

text(name, label, length)

Returns a new column of data type `STRING_SHORT_TYPE` or `STRING_LONG_TYPE`, with the specified name, label, and length.

Signature

```
public static DataSource.Column text(String name, String label, Integer length)
```

Parameters

name
Type: String

Name of the column.

label
Type: String

User-friendly name for the column that appears in the Salesforce user interface.

length
Type: Integer

Number of characters allowed in the column.

Return Value

Type: DataSource.Column

The returned column has these property values.

Property	Value
name	*name*
label	*label*
description	*label*
isSortable	true
isFilterable	true
type	DataSource.DataType.STRING_SHORT_TYPE if *length* is 255 or less DataSource.DataType.STRING_LONG_TYPE if *length* is greater than 255
length	*length*
decimalPlaces	0

text(name, length)

Returns a new column of data type `STRING_SHORT_TYPE` or `STRING_LONG_TYPE`, with the specified name and length.

Signature

```
public static DataSource.Column text(String name, Integer length)
```

Parameters

name
Type: String

Name of the column.

length
Type: Integer

Number of characters allowed in the column.

Return Value

Type: DataSource.Column

The returned column has these property values.

Property	Value
name	*name*
label	*name*
description	*name*
isSortable	true
isFilterable	true
type	DataSource.DataType.STRING_SHORT_TYPE if *length* is 255 or less DataSource.DataType.STRING_LONG_TYPE if *length* is greater than 255
length	*length*
decimalPlaces	0

`text(name)`

Returns a new column of data type `STRING_SHORT_TYPE` with the specified name and the length of 255 characters.

Signature

```
public static DataSource.Column text(String name)
```

Parameters

name
Type: String

Name of the column.

Return Value

Type: DataSource.Column

The returned column has these property values.

Property	Value
name	*name*
label	*name*
description	*name*
isSortable	true
isFilterable	true
type	DataSource.DataType.STRING_SHORT_TYPE

Property	Value
length	255
decimalPlaces	0

`textarea(name)`

Returns a new column of data type `STRING_LONG_TYPE` with the specified name and the length of 32,000 characters.

Signature

```
public static DataSource.Column textarea(String name)
```

Parameters

`name`
Type: String

Name of the column.

Return Value

Type: DataSource.Column

The returned column has these property values.

Property	Value
name	*`name`*
label	*`name`*
description	*`name`*
isSortable	true
isFilterable	true
type	DataSource.DataType.STRING_LONG_TYPE
length	32000
decimalPlaces	0

`time(name)`

Returns a new column of data type `Time` with the specified name.

Signature

```
public static DataSource.Column time(String name)
```

Parameters

name
Type: String

Name of the column.

Return Value

Type: DataSource.Column

url(name, length)

Returns a new column of data type `URL_TYPE` with the specified name and length.

Signature

```
public static DataSource.Column url(String name, Integer length)
```

Parameters

name
Type: String

Name of the column.

length
Type: Integer

Number of characters allowed in the column.

Return Value

Type: DataSource.Column

The returned column has these property values.

Property	Value
name	*name*
label	*name*
description	*name*
isSortable	true
isFilterable	true
type	DataSource.DataType.URL_TYPE
length	*length*
decimalPlaces	0

`url(name)`

Returns a new column of data type `URL_TYPE` with the specified name and the length of 1,000 characters.

Signature

```
public static DataSource.Column url(String name)
```

Parameters

`name`
Type: String

Name of the column.

Return Value

Type: DataSource.Column

The returned column has these property values.

Property	Value
name	*`name`*
label	*`name`*
description	*`name`*
isSortable	true
isFilterable	true
type	DataSource.DataType.URL_TYPE
length	1000
decimalPlaces	0

ColumnSelection Class

Identifies the list of columns to return during a query or search.

Namespace

DataSource Namespace

Usage

This class is associated with the `SELECT` clause for a SOQL query, or the `RETURNING` clause for a SOSL query.

IN THIS SECTION:

ColumnSelection Properties

ColumnSelection Properties

The following are properties for `ColumnSelection`.

IN THIS SECTION:

aggregation
How to aggregate the column's data.

columnName
Name of the selected column.

tableName
Name of the column's table.

aggregation

How to aggregate the column's data.

Signature

```
public DataSource.QueryAggregation aggregation {get; set;}
```

Property Value

Type: DataSource.QueryAggregation

columnName

Name of the selected column.

Signature

```
public String columnName {get; set;}
```

Property Value

Type: String

tableName

Name of the column's table.

Signature

```
public String tableName {get; set;}
```

Property Value

Type: String

Connection Class

Extend this class to enable your Salesforce org to sync the external system's schema and to handle queries, searches, and write operations (upsert and delete) of the external data. This class extends the `DataSourceUtil` class and inherits its methods.

Namespace

DataSource

Usage

Your `DataSource.Connection` and `DataSource.Provider` classes compose a custom adapter for Salesforce Connect.

Changing the `sync` method on the `DataSource.Connection` class doesn't automatically resync any external objects.

Example

```
global class SampleDataSourceConnection extends DataSource.Connection {
    global SampleDataSourceConnection(DataSource.ConnectionParams connectionParams) {
    }

    override global List<DataSource.Table> sync() {
        List<DataSource.Table> tables = new List<DataSource.Table>();
        List<DataSource.Column> columns;
        columns = new List<DataSource.Column>();
        columns.add(DataSource.Column.text('Name', 255));
        columns.add(DataSource.Column.text('ExternalId', 255));
        columns.add(DataSource.Column.url('DisplayUrl'));
        tables.add(DataSource.Table.get('Sample', 'Title', columns));
        return tables;
    }

    override global DataSource.TableResult query(DataSource.QueryContext c) {
       return DataSource.TableResult.get(c, DataSource.QueryUtils.process(c, getRows()));

    }

    override global List<DataSource.TableResult> search(DataSource.SearchContext c) {

        List<DataSource.TableResult> results = new List<DataSource.TableResult>();
        for (DataSource.TableSelection tableSelection : c.tableSelections) {
            results.add(DataSource.TableResult.get(tableSelection, getRows()));
        }
        return results;
    }

    // Helper method to get record values from the external system for the Sample table.
    private List<Map<String, Object>> getRows () {
       // Get row field values for the Sample table from the external system via a callout.

        HttpResponse response = makeGetCallout();
        // Parse the JSON response and populate the rows.
        Map<String, Object> m = (Map<String, Object>)JSON.deserializeUntyped(
```

```
            response.getBody());
    Map<String, Object> error = (Map<String, Object>)m.get('error');
    if (error != null) {
        throwException(string.valueOf(error.get('message')));
    }
    List<Map<String,Object>> rows = new List<Map<String,Object>>();
    List<Object> jsonRows = (List<Object>)m.get('value');
    if (jsonRows == null) {
        rows.add(foundRow(m));
    } else {
        for (Object jsonRow : jsonRows) {
            Map<String,Object> row = (Map<String,Object>)jsonRow;
            rows.add(foundRow(row));
        }
    }
    return rows;
}

global override List<DataSource.UpsertResult> upsertRows(DataSource.UpsertContext
        context) {
   if (context.tableSelected == 'Sample') {
       List<DataSource.UpsertResult> results = new List<DataSource.UpsertResult>();
       List<Map<String, Object>> rows = context.rows;

       for (Map<String, Object> row : rows){
          // Make a callout to insert or update records in the external system.
          HttpResponse response;
          // Determine whether to insert or update a record.
          if (row.get('ExternalId') == null){
             // Send a POST HTTP request to insert new external record.
             // Make an Apex callout and get HttpResponse.
             response = makePostCallout(
                 '{"name":"' + row.get('Name') + '","ExternalId":"' +
                 row.get('ExternalId') + '"');
          }
          else {
             // Send a PUT HTTP request to update an existing external record.
             // Make an Apex callout and get HttpResponse.
             response = makePutCallout(
                 '{"name":"' + row.get('Name') + '","ExternalId":"' +
                 row.get('ExternalId') + '"',
                 String.valueOf(row.get('ExternalId')));
          }

          // Check the returned response.
          // First, deserialize it.
          Map<String, Object> m = (Map<String, Object>)JSON.deserializeUntyped(
                  response.getBody());
          if (response.getStatusCode() == 200){
              results.add(DataSource.UpsertResult.success(
                      String.valueOf(m.get('id'))));
          }
          else {
             results.add(DataSource.UpsertResult.failure(
```

```
                                String.valueOf(m.get('id')),
                    'The callout resulted in an error: ' +
                    response.getStatusCode()));
            }
        }
        return results;
    }
    return null;
}

global override List<DataSource.DeleteResult> deleteRows(DataSource.DeleteContext
        context) {
    if (context.tableSelected == 'Sample'){
        List<DataSource.DeleteResult> results = new List<DataSource.DeleteResult>();
        for (String externalId : context.externalIds){
            HttpResponse response = makeDeleteCallout(externalId);
            if (response.getStatusCode() == 200){
                results.add(DataSource.DeleteResult.success(externalId));
            }
            else {
                results.add(DataSource.DeleteResult.failure(externalId,
                            'Callout delete error:'
                            + response.getBody()));
            }
        }
        return results;
    }
    return null;
 }

// Helper methods

// Make a GET callout
 private static HttpResponse makeGetCallout() {
     HttpResponse response;
     // Make callout
     // ...
     return response;
 }

 // Populate a row based on values from the external system.
 private Map<String,Object> foundRow(Map<String,Object> foundRow) {
    Map<String,Object> row = new Map<String,Object>();
    row.put('ExternalId', string.valueOf(foundRow.get('Id')));
    row.put('DisplayUrl', string.valueOf(foundRow.get('DisplayUrl')));
    row.put('Name', string.valueOf(foundRow.get('Name')));
    return row;
}

 // Make a POST callout
 private static HttpResponse makePostCallout(String jsonBody) {
     HttpResponse response;
     // Make callout
     // ...
```

```
        return response;
    }

    // Make a PUT callout
    private static HttpResponse makePutCallout(String jsonBody, String externalID) {
        HttpResponse response;
        // Make callout
        // ...
        return response;
    }

    // Make a DELETE callout
    private static HttpResponse makeDeleteCallout(String externalID) {
        HttpResponse response;
        // Make callout
        // ...
        return response;
    }
}
```

IN THIS SECTION:

Connection Methods

Connection Methods

The following are methods for `Connection`.

IN THIS SECTION:

deleteRows(deleteContext)
Invoked when external object records are deleted via the Salesforce user interface, APIs, or Apex.

query(queryContext)
Invoked by a SOQL query of an external object. A SOQL query is generated and executed when a user visits an external object's list view or record detail page in Salesforce. Returns the results of the query.

search(searchContext)
Invoked by a SOSL query of an external object or when a user performs a Salesforce global search that also searches external objects. Returns the results of the query.

sync()
Invoked when an administrator clicks **Validate and Sync** on the external data source detail page. Returns a list of tables that describe the external system's schema.

upsertRows(upsertContext)
Invoked when external object records are created or updated via the Salesforce user interface, APIs, or Apex.

`deleteRows(deleteContext)`

Invoked when external object records are deleted via the Salesforce user interface, APIs, or Apex.

Signature

```
public List<DataSource.DeleteResult> deleteRows(DataSource.DeleteContext deleteContext)
```

Parameters

deleteContext
Type: DataSource.DeleteContext

Contains context information about the delete request.

Return Value

Type: List<DataSource.DeleteResult>

The results of the delete operation.

query(queryContext)

Invoked by a SOQL query of an external object. A SOQL query is generated and executed when a user visits an external object's list view or record detail page in Salesforce. Returns the results of the query.

Signature

```
public DataSource.TableResult query(DataSource.QueryContext queryContext)
```

Parameters

queryContext
Type: DataSource.QueryContext

Represents the query to run against a data table.

Return Value

Type: DataSource.TableResult

search(searchContext)

Invoked by a SOSL query of an external object or when a user performs a Salesforce global search that also searches external objects. Returns the results of the query.

Signature

```
public List<DataSource.TableResult> search(DataSource.SearchContext searchContext)
```

Parameters

searchContext
Type: DataSource.SearchContext

Represents the query to run against an external data table.

Return Value

Type: List<DataSource.TableResult>

`sync()`

Invoked when an administrator clicks **Validate and Sync** on the external data source detail page. Returns a list of tables that describe the external system's schema.

Signature

```
public List<DataSource.Table> sync()
```

Return Value

Type: List<DataSource.Table>

Each returned table can be used to create an external object in Salesforce. On the Validate External Data Source page, the administrator views the list of returned tables and selects which tables to sync. When the administrator clicks **Sync**, an external object is created for each selected table. Each column within the selected tables also becomes a field in the external object.

`upsertRows(upsertContext)`

Invoked when external object records are created or updated via the Salesforce user interface, APIs, or Apex.

Signature

```
public List<DataSource.UpsertResult> upsertRows(DataSource.UpsertContext upsertContext)
```

Parameters

`upsertContext`
Type: DataSource.UpsertContext

Contains context information about the upsert request.

Return Value

Type: List<DataSource.UpsertResult>

The results of the upsert operation.

ConnectionParams Class

Contains the credentials for authenticating to the external system.

Namespace

DataSource

Usage

If your extension of the `DataSource.Provider` class returns `DataSource.AuthenticationCapability` values that indicate support for authentication, the `DataSource.Connection` class is instantiated with a `DataSource.ConnectionParams` instance in the constructor.

The authentication credentials in the `DataSource.ConnectionParams` instance depend on the `Identity Type` field of the external data source definition in Salesforce.

- If `Identity Type` is set to `Named Principal`, the credentials come from the external data source definition.
- If `Identity Type` is set to `Per User`:
  - For queries and searches, the credentials are specific to the current user who invokes the query or search. The credentials come from the user's authentication settings for the external system.
  - For administrative connections, such as syncing the external system's schema, the credentials come from the external data source definition.

The values in this class can appear in debug logs and can be accessed by users who have the "Author Apex" permission. If you require better security, we recommend that you specify named credentials instead of URLs as your Apex callout endpoints. Salesforce manages all authentication for Apex callouts that specify a named credential as the callout endpoint so that your code doesn't have to.

IN THIS SECTION:

ConnectionParams Properties

ConnectionParams Properties

The following are properties for `ConnectionParams`.

IN THIS SECTION:

certificateName
The name of the certificate for establishing each connection to the external system.

endpoint
The URL of the external system.

oauthToken
The OAuth token that's issued by the external system.

password
The password for authenticating to the external system.

principalType
An instance of DataSource.IdentityType, which determines which set of credentials to use to access the external system.

protocol
The type of protocol that's used to authenticate to the external system.

repository
Reserved for future use.

username
The username for authenticating to the external system.

certificateName

The name of the certificate for establishing each connection to the external system.

Signature

```
public String certificateName {get; set;}
```

Property Value

Type: String

The value comes from the external data source definition in Salesforce.

endpoint

The URL of the external system.

Signature

```
public String endpoint {get; set;}
```

Property Value

Type: String

The value comes from the external data source definition in Salesforce.

oauthToken

The OAuth token that's issued by the external system.

Signature

```
public String oauthToken {get; set;}
```

Property Value

Type: String

password

The password for authenticating to the external system.

Signature

```
public String password {get; set;}
```

Property Value

Type: String

The value depends on the `Identity Type` field of the external data source definition in Salesforce.

- If `Identity Type` is set to `Named Principal`, the credentials come from the external data source definition.
- If `Identity Type` is set to `Per User`:
  - For queries and searches, the credentials are specific to the current user who invokes the query or search. The credentials come from the user's authentication settings for the external system.
  - For administrative connections, such as syncing the external system's schema, the credentials come from the external data source definition.

`principalType`

An instance of DataSource.IdentityType, which determines which set of credentials to use to access the external system.

Signature

```
public DataSource.IdentityType principalType {get; set;}
```

Property Value

Type: DataSource.IdentityType

`protocol`

The type of protocol that's used to authenticate to the external system.

Signature

```
public DataSource.AuthenticationProtocol protocol {get; set;}
```

Property Value

Type: DataSource.AuthenticationProtocol

`repository`

Reserved for future use.

Signature

```
public String repository {get; set;}
```

Property Value

Type: String

Reserved for future use.

`username`

The username for authenticating to the external system.

Signature

```
public String username {get; set;}
```

Property Value

Type: String

The value depends on the `Identity Type` field of the external data source definition in Salesforce.

- If `Identity Type` is set to `Named Principal`, the credentials come from the external data source definition.
- If `Identity Type` is set to `Per User`:
  - For queries and searches, the credentials are specific to the current user who invokes the query or search. The credentials come from the user's authentication settings for the external system.
  - For administrative connections, such as syncing the external system's schema, the credentials come from the external data source definition.

DataSourceUtil Class

Parent class for the `DataSource.Provider`, `DataSource.Connection`, `DataSource.Table`, and `DataSource.Column` classes.

Namespace

DataSource

IN THIS SECTION:

DataSourceUtil Methods

DataSourceUtil Methods

The following are methods for `DataSourceUtil`.

IN THIS SECTION:

logWarning(message)
Logs the error message in the debug log.

throwException(message)
Throws a `DataSourceException` and displays the provided message to the user.

`logWarning(message)`

Logs the error message in the debug log.

Signature

```
public void logWarning(String message)
```

Parameters

message
Type: String

The error message.

Return Value

Type: void

`throwException(message)`

Throws a `DataSourceException` and displays the provided message to the user.

Signature

```
public void throwException(String message)
```

Parameters

message
Type: String

Error message to display to the user.

Return Value

Type: void

DataType Enum

Specifies the data types that are supported by the Apex Connector Framework.

Usage

The `DataSource.DataType` enum is referenced by the `type` property on the `DataSource.Column` class.

Enum Values

The following are the values of the `DataSource.DataType` enum.

Value	Description
`BOOLEAN_TYPE`	Boolean
`CURRENCY_TYPE`	Currency
`DATE_TYPE`	Date
`DATETIME_TYPE`	Date/Time
`EMAIL_TYPE`	Email

Value	Description
EXTERNAL_LOOKUP_TYPE	External lookup relationship
INDIRECT_LOOKUP_TYPE	Indirect lookup relationship
LOOKUP_TYPE	Lookup relationship
NUMBER_TYPE	Number
PERCENT_TYPE	Percent
PHONE_TYPE	Phone
PICKLIST_MULTISELECT_TYPE	Multi-select picklist
PICKLIST_TYPE	Picklist
STRING_LONG_TYPE	Long text area
STRING_SHORT_TYPE	Text area
TIME_TYPE	Time
URL_TYPE	URL

DeleteContext Class

An instance of `DeleteContext` is passed to the `deleteRows()` method on your `Database.Connection` class. The class provides context information about the delete request to the implementor of `deleteRows()`.

Namespace

DataSource

Usage

The Apex Connector Framework creates context for operations. Context is comprised of parameters about the operations, which other methods can use. An instance of the `DeleteContext` class packages these parameters into an object that can be used when a `deleteRows()` operation is initiated.

IN THIS SECTION:

DeleteContext Properties

DeleteContext Properties

The following are properties for `DeleteContext`.

IN THIS SECTION:

externalIds
The external IDs of the rows representing external object records to delete.

tableSelected
The name of the table to delete rows from.

externalIds

The external IDs of the rows representing external object records to delete.

Signature

```
public List<String> externalIds {get; set;}
```

Property Value

Type: List<String>

tableSelected

The name of the table to delete rows from.

Signature

```
public String tableSelected {get; set;}
```

Property Value

Type: String

DeleteResult Class

Represents the result of a delete operation on an sObject record. The result is returned by the `DataSource.deleteRows` method of the `DataSource.Connection` class.

Namespace

DataSource

Usage

A delete operation on external object records generates an array of objects of type `DataSource.DeleteResult`. Its methods create result records that indicate whether the delete operation succeeded or failed.

IN THIS SECTION:

DeleteResult Properties
DeleteResult Methods

DeleteResult Properties

The following are properties for `DeleteResult`.

IN THIS SECTION:

errorMessage
The error message that's generated by a failed delete operation. Recorded with a result of type `DataSource.DeleteResult`.

externalId
The unique identifier of a row that represents an external object record to delete.

success
Indicates whether a delete operation succeeded or failed.

errorMessage

The error message that's generated by a failed delete operation. Recorded with a result of type `DataSource.DeleteResult`.

Signature

```
public String errorMessage {get; set;}
```

Property Value

Type: String

externalId

The unique identifier of a row that represents an external object record to delete.

Signature

```
public String externalId {get; set;}
```

Property Value

Type: String

success

Indicates whether a delete operation succeeded or failed.

Signature

```
public Boolean success {get; set;}
```

Property Value

Type: Boolean

DeleteResult Methods

The following are methods for `DeleteResult`.

IN THIS SECTION:

equals(obj)
Maintains the integrity of lists of type `DeleteResult` by determining the equality of external objects in a list. This method is dynamic and is based on the `equals` method in Java.

failure(externalId, errorMessage)
Creates a delete result indicating the failure of a delete request for a given external ID.

hashCode()
Maintains the integrity of lists of type `DeleteResult` by determining the uniqueness of the external object records in a list.

success(externalId)
Creates a delete result indicating the successful completion of a delete request for a given external ID.

equals(obj)

Maintains the integrity of lists of type `DeleteResult` by determining the equality of external objects in a list. This method is dynamic and is based on the `equals` method in Java.

Signature

```
public Boolean equals(Object obj)
```

Parameters

obj
Type: Object

External object whose key is to be validated.

For information about the `equals` method, see Using Custom Types in Map Keys and Sets.

Return Value

Type: Boolean

failure(externalId, errorMessage)

Creates a delete result indicating the failure of a delete request for a given external ID.

Signature

```
public static DataSource.DeleteResult failure(String externalId, String errorMessage)
```

Parameters

externalId
Type: String

The unique identifier of the sObject record to delete.

errorMessage
Type: String

The reason the delete operation failed.

Return Value

Type: DataSource.DeleteResult

Status result of the delete operation.

`hashCode()`

Maintains the integrity of lists of type `DeleteResult` by determining the uniqueness of the external object records in a list.

Signature

```
public Integer hashCode()
```

Return Value

Type: Integer

`success(externalId)`

Creates a delete result indicating the successful completion of a delete request for a given external ID.

Signature

```
public static DataSource.DeleteResult success(String externalId)
```

Parameters

`externalId`
Type: String

The unique identifier of the sObject record to delete.

Return Value

Type: DataSource.DeleteResult

Status result of the delete operation for the sObject with the given external ID.

Filter Class

Represents a `WHERE` clause in a SOSL or SOQL query.

Namespace

DataSource

Usage

Compound types require child filters. Specifically, the `subfilters` property can't be null if the `type` property is `NOT_`, `AND_`, or `OR_`.

IN THIS SECTION:

Filter Properties

Filter Properties

The following are properties for `Filter`.

IN THIS SECTION:

columnName
Name of the column that's being evaluated in a simple comparative type of filter.

columnValue
Value that the filter compares records against in a simple comparative type of filter.

subfilters
List of subfilters for compound filter types, such as `NOT_`, `AND_`, and `OR_`.

tableName
Name of the table whose column is being evaluated in a simple comparative type of filter.

type
Type of filter operation that limits the returned data.

`columnName`

Name of the column that's being evaluated in a simple comparative type of filter.

Signature

```
public String columnName {get; set;}
```

Property Value

Type: String

`columnValue`

Value that the filter compares records against in a simple comparative type of filter.

Signature

```
public Object columnValue {get; set;}
```

Property Value

Type: Object

`subfilters`

List of subfilters for compound filter types, such as `NOT_`, `AND_`, and `OR_`.

Signature

```
public List<DataSource.Filter> subfilters {get; set;}
```

Property Value

Type: List<DataSource.Filter>

`tableName`

Name of the table whose column is being evaluated in a simple comparative type of filter.

Signature

```
public String tableName {get; set;}
```

Property Value

Type: String

`type`

Type of filter operation that limits the returned data.

Signature

```
public DataSource.FilterType type {get; set;}
```

Property Value

Type: DataSource.FilterType

FilterType Enum

Referenced by the `type` property on a `DataSource.Filter`.

Usage

Determines how to limit the returned data.

Enum Values

The following are the values of the `DataSource.FilterType` enum.

Value	Description
`AND_`	This compound filter type returns all rows that match all the subfilters.
`CONTAINS`	Simple comparative filter type.
`ENDS_WITH`	Simple comparative filter type.

Value	Description
EQUALS	Simple comparative filter type.
GREATER_THAN	Simple comparative filter type.
GREATER_THAN_OR_EQUAL_TO	Simple comparative filter type.
LESS_THAN	Simple comparative filter type.
LESS_THAN_OR_EQUAL_TO	Simple comparative filter type.
LIKE_	Simple comparative filter type.
NOT_	This compound filter type returns the rows that don't match the subfilter.
NOT_EQUALS	Simple comparative filter type.
OR_	This compound filter type returns all rows that match any of the subfilters.
STARTS_WITH	Simple comparative filter type.

IdentityType Enum

Determines which set of credentials is used to authenticate to the external system.

Usage

The relevant credentials are passed to your `DataSource.Connection` class.

Enum Values

The following are the values of the `DataSource.IdentityType` enum.

Value	Description
ANONYMOUS	No credentials are used to authenticate to the external system.
NAMED_USER	The credentials in the external data source definition are used to authenticate to the external system, regardless of which user is accessing the external data from your organization.
PER_USER	For queries and searches, the credentials are specific to the current user who invokes the query or search. The credentials come from the user's authentication settings for the external system. For administrative connections, such as syncing the external system's schema, the credentials come from the external data source definition.

Order Class

Contains details about how to sort the rows in the result set. Equivalent to an `ORDER BY` statement in a SOQL query.

Namespace

DataSource

Usage

Used in the `order` property on the `DataSource.TableSelection` class.

IN THIS SECTION:

Order Properties

Order Methods

Order Properties

The following are properties for `Order`.

IN THIS SECTION:

columnName
Name of the column whose values are used to sort the rows in the result set.

direction
Direction for sorting rows based on column values.

tableName
Name of the table whose column values are used to sort the rows in the result set.

`columnName`

Name of the column whose values are used to sort the rows in the result set.

Signature

```
public String columnName {get; set;}
```

Property Value

Type: String

`direction`

Direction for sorting rows based on column values.

Signature

```
public DataSource.OrderDirection direction {get; set;}
```

Property Value

Type: DataSource.OrderDirection

tableName

Name of the table whose column values are used to sort the rows in the result set.

Signature

```
public String tableName {get; set;}
```

Property Value

Type: String

Order Methods

The following are methods for `Order`.

IN THIS SECTION:

get(tableName, columnName, direction)
Creates an instance of the DataSource.Order class.

get(tableName, columnName, direction)

Creates an instance of the DataSource.Order class.

Signature

```
public static DataSource.Order get(String tableName, String columnName,
DataSource.OrderDirection direction)
```

Parameters

tableName
Type: String

Name of the table whose column values are used to sort the rows in the result set.

columnName
Type: String

Name of the column whose values are used to sort the rows in the result set.

direction
Type: DataSource.OrderDirection

Direction for sorting rows based on column values.

Return Value

Type: DataSource.Order

OrderDirection Enum

Specifies the direction for sorting rows based on column values.

Usage

Used by the direction property on the DataSource.Order class.

Enum Values

The following are the values of the `DataSource.OrderDirection` enum.

Value	Description
`ASCENDING`	Sort rows in ascending order (A–Z).
`DESCENDING`	Sort rows in descending order (Z–A).

Provider Class

Extend this base class to create a custom adapter for Salesforce Connect. The class informs Salesforce of the functional and authentication capabilities that are supported by or required to connect to the external system. This class extends the `DataSourceUtil` class and inherits its methods.

Namespace

DataSource

Usage

Create an Apex class that extends `DataSource.Provider` to specify the following.

- The types of authentication that can be used to access the external system
- The features that are supported for the connection to the external system
- The Apex class that extends `DataSource.Connection` to sync the external system's schema and to handle the queries and searches of the external data

The values that are returned by the `DataSource.Provider` class determine which settings are available in the external data source definition in Salesforce. To access the external data source definition from Setup, enter *`External Data Sources`* in the `Quick Find` box, then select **External Data Sources**.

IN THIS SECTION:

Provider Methods

Provider Methods

The following are methods for `Provider`.

IN THIS SECTION:

getAuthenticationCapabilities()
Returns the types of authentication that can be used to access the external system.

getCapabilities()
Returns the functional operations that the external system supports and the required endpoint settings for the external data source definition in Salesforce.

getConnection(connectionParams)
Returns a connection that points to an instance of the external data source.

`getAuthenticationCapabilities()`

Returns the types of authentication that can be used to access the external system.

Signature

```
public List<DataSource.AuthenticationCapability> getAuthenticationCapabilities()
```

Return Value

Type: List<DataSource.AuthenticationCapability>

`getCapabilities()`

Returns the functional operations that the external system supports and the required endpoint settings for the external data source definition in Salesforce.

Signature

```
public List<DataSource.Capability> getCapabilities()
```

Return Value

Type: List<DataSource.Capability>

`getConnection(connectionParams)`

Returns a connection that points to an instance of the external data source.

Signature

```
public DataSource.Connection getConnection(DataSource.ConnectionParams connectionParams)
```

Parameters

`connectionParams`
Type: DataSource.ConnectionParams

Credentials for authenticating to the external system.

Return Value

Type: DataSource.Connection

QueryAggregation Enum

Specifies how to aggregate a column in a query.

Usage

Used by the aggregation property on the DataSource.ColumnSelection class.

Enum Values

The following are the values of the `DataSource.QueryAggregation` enum.

Value	Description
`AVG`	Reserved for future use.
`COUNT`	Returns the number of rows that meet the query criteria.
`MAX`	Reserved for future use.
`MIN`	Reserved for future use.
`NONE`	No aggregation.
`SUM`	Reserved for future use.

QueryContext Class

An instance of `QueryContext` is provided to the `query` method on your `DataSource.Connection` class. The instance corresponds to a SOQL request.

Namespace

DataSource

IN THIS SECTION:

QueryContext Properties

QueryContext Methods

QueryContext Properties

The following are properties for `QueryContext`.

IN THIS SECTION:

queryMoreToken
Query token that's used for server-driven paging to determine and fetch the subsequent batch of results.

tableSelection
Query details that represent the `FROM`, `ORDER BY`, `SELECT`, and `WHERE` clauses in a SOQL or SOSL query.

queryMoreToken

Query token that's used for server-driven paging to determine and fetch the subsequent batch of results.

Signature

```
public String queryMoreToken {get; set;}
```

Property Value

Type: String

tableSelection

Query details that represent the `FROM`, `ORDER BY`, `SELECT`, and `WHERE` clauses in a SOQL or SOSL query.

Signature

```
public DataSource.TableSelection tableSelection {get; set;}
```

Property Value

Type: DataSource.TableSelection

QueryContext Methods

The following are methods for `QueryContext`.

IN THIS SECTION:

get(metadata, offset, maxResults, tableSelection)
Creates an instance of the `QueryContext` class.

get(metadata, offset, maxResults, tableSelection)

Creates an instance of the `QueryContext` class.

Signature

```
public static DataSource.QueryContext get(List<DataSource.Table> metadata, Integer
offset, Integer maxResults, DataSource.TableSelection tableSelection)
```

Parameters

metadata
Type: List<DataSource.Table>

List of table metadata that describes the external system's tables to query.

offset
Type: Integer

Used for client-driven paging. Specifies the starting row offset into the query's result set.

maxResults
Type: Integer

Used for client-driven paging. Specifies the maximum number of rows to return in each batch.

tableSelection
Type: DataSource.TableSelection

Query details that represent the `FROM`, `ORDER BY`, `SELECT`, and `WHERE` clauses in a SOQL or SOSL query.

Return Value

Type: `DataSource.QueryContext`

QueryUtils Class

Contains helper methods to locally filter, sort, and apply limit and offset clauses to data rows. This helper class is provided for your convenience during early development and tests, but it isn't supported for use in production environments.

Namespace

DataSource

Usage

The `DataSource.QueryUtils` class and its helper methods can process query results locally within your Salesforce org. This class is provided for your convenience to simplify the development of your Salesforce Connect custom adapter for initial tests. However, the `DataSource.QueryUtils` class and its methods aren't supported for use in production environments that use callouts to retrieve data from external systems. Complete the filtering and sorting on the external system before sending the query results to Salesforce. When possible, use server-driven paging or another technique to have the external system determine the appropriate data subsets according to the limit and offset clauses in the query.

IN THIS SECTION:

QueryUtils Methods

QueryUtils Methods

The following are methods for `QueryUtils`.

IN THIS SECTION:

applyLimitAndOffset(queryContext, rows)
Returns a subset of data rows after locally applying limit and offset clauses from the query. This helper method is provided for your convenience during early development and tests, but it isn't supported for use in production environments.

filter(queryContext, rows)
Returns a subset of data rows after locally ordering and applying filters from the query. This helper method is provided for your convenience during early development and tests, but it isn't supported for use in production environments.

process(queryContext, rows)
Returns data rows after locally filtering, sorting, ordering, and applying limit and offset clauses from the query. This helper method is provided for your convenience during early development and tests, but it isn't supported for use in production environments.

sort(queryContext, rows)
Returns data rows after locally sorting and applying the order from the query. This helper method is provided for your convenience during early development and tests, but it isn't supported for use in production environments.

`applyLimitAndOffset(queryContext, rows)`

Returns a subset of data rows after locally applying limit and offset clauses from the query. This helper method is provided for your convenience during early development and tests, but it isn't supported for use in production environments.

Signature

```
public static List<Map<String,Object>> applyLimitAndOffset(DataSource.QueryContext
queryContext, List<Map<String,Object>> rows)
```

Parameters

queryContext
Type: DataSource.QueryContext

Represents the query to run against a data table.

rows
Type: List<Map<String, Object>>

Rows of data.

Return Value

Type: List<Map<String, Object>>

`filter(queryContext, rows)`

Returns a subset of data rows after locally ordering and applying filters from the query. This helper method is provided for your convenience during early development and tests, but it isn't supported for use in production environments.

Signature

```
public static List<Map<String,object>> filter(DataSource.QueryContext queryContext,
List<Map<String,Object>> rows)
```

Parameters

queryContext
Type: DataSource.QueryContext

queryContext

rows
Type: List<Map<String, Object>>

Rows of data.

Return Value

Type: List<Map<String, Object>>

process(queryContext, rows)

Returns data rows after locally filtering, sorting, ordering, and applying limit and offset clauses from the query. This helper method is provided for your convenience during early development and tests, but it isn't supported for use in production environments.

Signature

```
public static List<Map<String,object>> process(DataSource.QueryContext queryContext,
List<Map<String,Object>> rows)
```

Parameters

queryContext
Type: DataSource.QueryContext

Represents the query to run against a data table.

rows
Type: List<Map<String, Object>>

Rows of data.

Return Value

Type: List<Map<String, Object>>

sort(queryContext, rows)

Returns data rows after locally sorting and applying the order from the query. This helper method is provided for your convenience during early development and tests, but it isn't supported for use in production environments.

Signature

```
public static List<Map<String,ject>> sort(DataSource.QueryContext queryContext,
List<Map<String,object>> rows)
```

Parameters

queryContext
Type: DataSource.QueryContext

Represents the query to run against a data table.

rows
Type: List<Map<String, Object>>

Rows of data.

Return Value

Type: List<Map<String, Object>>

ReadContext Class

Abstract base class for the `QueryContext` and `SearchContext` classes.

Namespace

DataSource

IN THIS SECTION:

ReadContext Properties

ReadContext Properties

The following are properties for `ReadContext`.

IN THIS SECTION:

maxResults
Maximum number of rows that the query can return.

metadata
Describes the external system's tables to query.

offset
The starting row offset into the query's result set. Used for client-driven paging.

`maxResults`

Maximum number of rows that the query can return.

Signature

```
public Integer maxResults {get; set;}
```

Property Value

Type: Integer

`metadata`

Describes the external system's tables to query.

Signature

```
public List<DataSource.Table> metadata {get; set;}
```

Property Value

Type: List<DataSource.Table>

`offset`

The starting row offset into the query's result set. Used for client-driven paging.

Signature

```
public Integer offset {get; set;}
```

Property Value

Type: Integer

SearchContext Class

An instance of `SearchContext` is provided to the search method on your `DataSource.Connection` class. The instance corresponds to a search or SOSL request.

Namespace

DataSource

IN THIS SECTION:

SearchContext Constructors

SearchContext Properties

SearchContext Constructors

The following are constructors for `SearchContext`.

IN THIS SECTION:

SearchContext(metadata, offset, maxResults, tableSelections, searchPhrase)

Creates an instance of the `SearchContext` class with the specified parameter values.

SearchContext()

Creates an instance of the `SearchContext` class.

SearchContext(metadata, offset, maxResults, tableSelections, searchPhrase)

Creates an instance of the `SearchContext` class with the specified parameter values.

Signature

```
public SearchContext(List<DataSource.Table> metadata, Integer offset, Integer maxResults,
List<DataSource.TableSelection> tableSelections, String searchPhrase)
```

Parameters

metadata

Type: List<DataSource.Table>

List of table metadata that describes the external system's tables to query.

offset

Type: Integer

Specifies the starting row offset into the query's result set.

maxResults

Type: Integer

Specifies the maximum number of rows to return in each batch.

tableSelections

Type: List<DataSource.TableSelection>

List of queries and their details. The details represent the `FROM`, `ORDER BY`, `SELECT`, and `WHERE` clauses in each SOQL or SOSL query.

searchPhrase

Type: String

The user-entered search string as a case-sensitive single phrase, with all non-alphanumeric characters removed.

SearchContext()

Creates an instance of the `SearchContext` class.

Signature

```
public SearchContext()
```

SearchContext Properties

The following are properties for `SearchContext`.

IN THIS SECTION:

searchPhrase
The user-entered search string as a case-sensitive single phrase, with all non-alphanumeric characters removed.

tableSelections
List of queries and their details. The details represent the FROM, ORDER BY, SELECT, and WHERE clauses in each SOQL or SOSL query.

`searchPhrase`

The user-entered search string as a case-sensitive single phrase, with all non-alphanumeric characters removed.

Signature

```
public String searchPhrase {get; set;}
```

Property Value

Type: String

`tableSelections`

List of queries and their details. The details represent the FROM, ORDER BY, SELECT, and WHERE clauses in each SOQL or SOSL query.

Signature

```
public List<DataSource.TableSelection> tableSelections {get; set;}
```

Property Value

Type: List<DataSource.TableSelection>

SearchUtils Class

Helper class for implementing search on a custom adapter for Salesforce Connect.

Namespace

DataSource

Usage

We recommend that you develop your own search implementation that can search columns in addition to the designated name field.

IN THIS SECTION:

SearchUtils Methods

SearchUtils Methods

The following are methods for `SearchUtils`.

IN THIS SECTION:

searchByName(searchDetails, connection)
Queries all the tables and returns each row whose designated name field contains the search phrase.

`searchByName(searchDetails, connection)`

Queries all the tables and returns each row whose designated name field contains the search phrase.

Signature

```
public static List<DataSource.TableResult> searchByName(DataSource.SearchContext
searchDetails, DataSource.Connection connection)
```

Parameters

`searchDetails`
Type: DataSource.SearchContext

The `SearchContext` class that specifies which data to search and what to search for.

`connection`
Type: DataSource.Connection

The `DataSource.Connection` class that connects to the external system.

Return Value

Type: List<DataSource.TableResult>

Table Class

Describes a table on an external system that the Salesforce Connect custom adapter connects to. This class extends the `DataSourceUtil` class and inherits its methods.

Namespace

DataSource

Usage

A list of table metadata is provided by the `DataSource.Connection` class when the `sync()` method is invoked. Each table can become an external object in Salesforce.

The metadata is stored in Salesforce. Updating the Apex code to return new or updated values for the table metadata doesn't automatically update the stored metadata in Salesforce.

IN THIS SECTION:

Table Properties
Table Methods

Table Properties

The following are properties for `Table`.

IN THIS SECTION:

columns
List of table columns.

description
Description of what the table represents.

labelPlural
Plural form of the user-friendly name for the table. The `labelPlural` becomes the object's plural label in the Salesforce user interface.

labelSingular
Singular form of the user-friendly name for the table. The `labelSingular` becomes the object label in the Salesforce user interface. We recommend that you make object labels unique across all standard, custom, and external objects in the org.

name
Name of the table on the external system.

nameColumn
Name of the table column that becomes the name field of the external object when the administrator syncs the table.

`columns`

List of table columns.

Signature

```
public List<DataSource.Column> columns {get; set;}
```

Property Value

Type: List<DataSource.Column>

`description`

Description of what the table represents.

Signature

```
public String description {get; set;}
```

Property Value

Type: String

`labelPlural`

Plural form of the user-friendly name for the table. The `labelPlural` becomes the object's plural label in the Salesforce user interface.

Signature

```
public String labelPlural {get; set;}
```

```
DataSource.Table, labelPlural
```

Property Value

Type: String

`labelSingular`

Singular form of the user-friendly name for the table. The `labelSingular` becomes the object label in the Salesforce user interface. We recommend that you make object labels unique across all standard, custom, and external objects in the org.

Signature

```
public String labelSingular {get; set;}
```

Property Value

Type: String

`name`

Name of the table on the external system.

Signature

```
public String name {get; set;}
```

Property Value

Type: String

`nameColumn`

Name of the table column that becomes the name field of the external object when the administrator syncs the table.

Signature

```
public String nameColumn {get; set;}
```

Property Value

Type: String

Table Methods

The following are methods for `Table`.

IN THIS SECTION:

get(name, labelSingular, labelPlural, description, nameColumn, columns)
Returns the table metadata with the specified parameter values.

get(name, nameColumn, columns)
Returns the table metadata with the specified parameter values, using the name for the labels and description.

get(name, labelSingular, labelPlural, description, nameColumn, columns)

Returns the table metadata with the specified parameter values.

Signature

```
public static DataSource.Table get(String name, String labelSingular, String labelPlural,
String description, String nameColumn, List<DataSource.Column> columns)
```

Parameters

name
Type: String

Name of the external table.

labelSingular
Type: String

Singular form of the user-friendly name for the table. The `labelSingular` becomes the object label in the Salesforce user interface.

labelPlural
Type: String

Plural form of the user-friendly name for the table. The `labelPlural` becomes the object's plural label in the Salesforce user interface.

description
Type: String

Description of the external table.

nameColumn
Type: String

Name of the table column that becomes the name field of the external object when the administrator syncs the table.

columns
Type: List<DataSource.Column>

List of table columns.

Return Value

Type: DataSource.Table

get(name, nameColumn, columns)

Returns the table metadata with the specified parameter values, using the name for the labels and description.

Signature

```
public static DataSource.Table get(String name, String nameColumn,
List<DataSource.Column> columns)
```

```
DataSource.Table, get, [String, String, List<DataSource.Column>], DataSource.Table
```

Parameters

name
Type: String

Name of the external table.

nameColumn
Type: String

Name of the table column that becomes the name field of the external object when the administrator syncs the table.

columns
Type: List<DataSource.Column>

List of table columns.

Return Value

Type: DataSource.Table

The returned table metadata has these property values.

Property	Value
name	*name*
labelSingular	*name*
labelPlural	*name*
description	*name*
nameColumn	*nameColumn*
columns	*columns*

TableResult Class

Contains the results of a search or query.

Namespace

DataSource

IN THIS SECTION:

TableResult Properties

TableResult Methods

TableResult Properties

The following are properties for `TableResult`.

IN THIS SECTION:

errorMessage
errorMessage

queryMoreToken
Query token that's used for server-driven paging to determine and fetch the subsequent batch of results. This token is passed back to the Apex data source on subsequent queries in the `queryMoreToken` property on the `QueryContext`.

rows
Rows of data.

success
Whether the search or query was successful.

tableName
Name of the table that was queried.

totalSize
The total number of rows that meet the query criteria, even when the external system is requested to return a smaller batch size.

errorMessage

errorMessage

Signature

```
public String errorMessage {get; set;}
```

Property Value

Type: String

queryMoreToken

Query token that's used for server-driven paging to determine and fetch the subsequent batch of results. This token is passed back to the Apex data source on subsequent queries in the `queryMoreToken` property on the `QueryContext`.

Signature

```
public String queryMoreToken {get; set;}
```

Property Value

Type: String

rows

Rows of data.

Signature

```
public List<Map<String,Object>> rows {get; set;}
```

Property Value

Type: List<Map<String, Object>>

`success`

Whether the search or query was successful.

Signature

```
public Boolean success {get; set;}
```

Property Value

Type: Boolean

`tableName`

Name of the table that was queried.

Signature

```
public String tableName {get; set;}
```

Property Value

Type: String

`totalSize`

The total number of rows that meet the query criteria, even when the external system is requested to return a smaller batch size.

Signature

```
public Integer totalSize {get; set;}
```

Property Value

Type: Integer

TableResult Methods

The following are methods for `TableResult`.

IN THIS SECTION:

error(errorMessage)
Returns failed search or query results with the provided error message.

get(success, errorMessage, tableName, rows, totalSize)
Returns a subset of data rows in a `TableResult` with the provided property values and the number of rows in the table.

get(success, errorMessage, tableName, rows)
Returns a subset of data rows in a `TableResult` with the provided property values.

get(queryContext, rows)
Returns the subset of data rows that meet the query criteria, and the number of rows in the table, in a `TableResult`.

get(tableSelection, rows)
Returns the subset of data rows that meet the query criteria, and the number of rows in the table, in a `TableResult`.

error(errorMessage)

Returns failed search or query results with the provided error message.

Signature

```
public static DataSource.TableResult error(String errorMessage)
```

Parameters

errorMessage
Type: String

errorMessage

Return Value

Type: DataSource.TableResult

The returned `TableResult` has these property values.

Property	Value
success	false
errorMessage	*errorMessage*
tableName	null
rows	null
rows.size()	0

get(success, errorMessage, tableName, rows, totalSize)

Returns a subset of data rows in a `TableResult` with the provided property values and the number of rows in the table.

Signature

```
public static DataSource.TableResult get(Boolean success, String errorMessage, String
tableName, List<Map<String,Object>> rows, Integer totalSize)
```

Parameters

success
Type: Boolean

Whether the search or query was successful.

errorMessage
Type: String

errorMessage

tableName
Type: String

Name of the table that was queried.

rows
Type: List<Map<String, Object>>

Rows of data.

totalSize
Type: Integer

The total number of rows that meet the query criteria, even when the external system is requested to return a smaller batch size.

Return Value

Type: DataSource.TableResult

get(success, errorMessage, tableName, rows)

Returns a subset of data rows in a `TableResult` with the provided property values.

Signature

```
public static DataSource.TableResult get(Boolean success, String errorMessage, String
tableName, List<Map<String,Object>> rows)
```

Parameters

success
Type: Boolean

Whether the search or query was successful.

errorMessage
Type: String

errorMessage

tableName
Type: String

Name of the table that was queried.

rows
Type: List<Map<String, Object>>

Rows of data.

Return Value

Type: DataSource.TableResult

get(queryContext, rows)

Returns the subset of data rows that meet the query criteria, and the number of rows in the table, in a `TableResult`.

Signature

```
public static DataSource.TableResult get(DataSource.QueryContext queryContext,
List<Map<String,Object>> rows)
```

Parameters

queryContext
Type: DataSource.QueryContext

Represents the query to run against a data table.

rows
Type: List<Map<String, Object>>

Rows of data.

Return Value

Type: DataSource.TableResult

get(tableSelection, rows)

Returns the subset of data rows that meet the query criteria, and the number of rows in the table, in a `TableResult`.

Signature

```
public static DataSource.TableResult get(DataSource.TableSelection tableSelection,
List<Map<String,Object>> rows)
```

Parameters

tableSelection
Type: DataSource.TableSelection

Query details that represent the `FROM`, `ORDER BY`, `SELECT`, and `WHERE` clauses in a SOQL or SOSL query.

rows
Type: List<Map<String, Object>>

Rows of data.

Return Value

Type: DataSource.TableResult

TableSelection Class

Contains a breakdown of the SOQL or SOSL query. Its properties represent the FROM, ORDER BY, SELECT, and WHERE clauses in the query.

Namespace

DataSource

IN THIS SECTION:

TableSelection Properties

TableSelection Properties

The following are properties for `TableSelection`.

IN THIS SECTION:

columnsSelected
List of columns to query. Corresponds to the `SELECT` clause in a SOQL or SOSL query.

filter
Identifies the query filter, which can be a compound filter that has a list of subfilters. The filter corresponds to the `WHERE` clause in a SOQL or SOSL query.

order
Identifies the order for sorting the query results. Corresponds to the `ORDER BY` clause in a SOQL or SOSL query.

tableSelected
Name of the table to query. Corresponds to the `FROM` clause in a SOQL or SOSL query.

`columnsSelected`

List of columns to query. Corresponds to the `SELECT` clause in a SOQL or SOSL query.

Signature

```
public List<DataSource.ColumnSelection> columnsSelected {get; set;}
```

Property Value

Type: List<DataSource.ColumnSelection>

`filter`

Identifies the query filter, which can be a compound filter that has a list of subfilters. The filter corresponds to the `WHERE` clause in a SOQL or SOSL query.

Signature

```
public DataSource.Filter filter {get; set;}
```

Property Value

Type: DataSource.Filter

`order`

Identifies the order for sorting the query results. Corresponds to the `ORDER BY` clause in a SOQL or SOSL query.

Signature

```
public List<DataSource.Order> order {get; set;}
```

Property Value

Type: List<DataSource.Order>

`tableSelected`

Name of the table to query. Corresponds to the `FROM` clause in a SOQL or SOSL query.

Signature

```
public String tableSelected {get; set;}
```

Property Value

Type: String

UpsertContext Class

An instance of `UpsertContext` is passed to the `upsertRows()` method on your `Datasource.Connection` class. This class provides context information about the upsert request to the implementor of `upsertRows()`.

Namespace

DataSource

Usage

The Apex Connector Framework creates the contet for operations. Context is comprised of parameters about the operations, which other methods can use. An instance of the `UpsertContext` class packages these parameters into an object that can be used when an `upsertRows()` operation is initiated.

IN THIS SECTION:

UpsertContext Properties

UpsertContext Properties

The following are properties for `UpsertContext`.

IN THIS SECTION:

rows
List of rows corresponding to the external object records to upsert.
tableSelected
The name of the table to upsert rows in.

rows

List of rows corresponding to the external object records to upsert.

Signature

```
public List<Map<String,ANY>> rows {get; set;}
```

Property Value

Type: List<Map<String,Object>>

tableSelected

The name of the table to upsert rows in.

Signature

```
public String tableSelected {get; set;}
```

Property Value

Type: String

UpsertResult Class

Represents the result of an upsert operation on an external object record. The result is returned by the `upsertRows` method of the `DataSource.Connection` class.

Namespace

DataSource

Usage

An upsert operation on external object records generates an array of objects of type `DataSource.UpsertResult`. Its methods create result records that indicate whether the upsert operation succeeded or failed.

IN THIS SECTION:

UpsertResult Properties

UpsertResult Methods

UpsertResult Properties

The following are properties for `UpsertResult`.

IN THIS SECTION:

errorMessage
The error message that's generated by a failed upsert operation.

externalId
The unique identifier of a row that represents an external object record to upsert.

success
Indicates whether a delete operation succeeded or failed.

errorMessage

The error message that's generated by a failed upsert operation.

Signature

```
public String errorMessage {get; set;}
```

Property Value

Type: String

externalId

The unique identifier of a row that represents an external object record to upsert.

Signature

```
public String externalId {get; set;}
```

Property Value

Type: String

success

Indicates whether a delete operation succeeded or failed.

Signature

```
public Boolean success {get; set;}
```

Property Value

Type: Boolean

UpsertResult Methods

The following are methods for `UpsertResult`.

IN THIS SECTION:

equals(obj)
Maintains the integrity of lists of type `UpsertResult` by determining the equality of external object records in a list. This method is dynamic and is based on the `equals` method in Java.

failure(externalId, errorMessage)
Creates an upsert result that indicates the failure of a delete request for a given external ID.

hashCode()
Maintains the integrity of lists of type `UpsertResult` by determining the uniqueness of the external object records in a list.

success(externalId)
Creates a delete result that indicates the successful completion of an upsert request for a given external ID.

`equals(obj)`

Maintains the integrity of lists of type `UpsertResult` by determining the equality of external object records in a list. This method is dynamic and is based on the `equals` method in Java.

Signature

```
public Boolean equals(Object obj)
```

Parameters

obj
Type: Object

External object whose key is to be validated.

Return Value

Type: Boolean

`failure(externalId, errorMessage)`

Creates an upsert result that indicates the failure of a delete request for a given external ID.

Signature

```
public static DataSource.UpsertResult failure(String externalId, String errorMessage)
```

Parameters

externalId
Type: String

The unique identifier of the external object record to upsert.

errorMessage
Type: String

The reason the upsert operation failed.

Return Value

Type: DataSource.UpsertResult

Status result for the upsert operation.

hashCode()

Maintains the integrity of lists of type `UpsertResult` by determining the uniqueness of the external object records in a list.

Signature

```
public Integer hashCode()
```

Return Value

Type: Integer

success(externalId)

Creates a delete result that indicates the successful completion of an upsert request for a given external ID.

Signature

```
public static DataSource.UpsertResult success(String externalId)
```

Parameters

externalId
Type: String

The unique identifier of the external object record to upsert.

Return Value

Type: DataSource.UpsertResult

Status result of the upsert operation for the external object record with the given external ID.

DataSource Exceptions

The `DataSource` namespace contains exception classes.

All exception classes support built-in methods for returning the error message and exception type. See Exception Class and Built-In Exceptions.

The `DataSource` namespace contains these exceptions.

Exception	Description	Methods
`DataSource.DataSourceException`	Throw this exception to indicate that an error occurred while communicating with an external data source.	To get the error message and write it to debug log, use the `String getMessage()`.
`DataSource.OAuthTokenExpiredException`	Throw this exception to indicate that an OAuth token has expired. The system then attempts to refresh the token automatically and restart the query, search, or sync operation.	To get the error message and write it to debug log, use the `String getMessage()`.

DataWeave Namespace

The DataWeave namespace provides classes and methods to support the invocation of DataWeave scripts from Apex.

DataWeave is the MuleSoft expression language for accessing, parsing, and transforming data that travels through a Mule application. For detailed information, see DataWeave Language.

These are the classes in the `DataWeave` namespace.

IN THIS SECTION:

Result Class
Contains methods to retrieve data that was transformed using Script class methods.

Script Class
Contains the `createScript()` method to load DataWeave scripts and the `execute()` method to obtain script output in a `DataWeave.Result` object.

SEE ALSO:

DataWeave in Apex

Result Class

Contains methods to retrieve data that was transformed using Script class methods.

Namespace

DataWeave

Example

See Script Class for an example to run a DataWeave script from Apex and retrieve the resulting script output.

IN THIS SECTION:

Result Methods

Result Methods

The following are methods for `Result`.

IN THIS SECTION:

getValue()
Returns the result of a DataWeave script execution as an object.

getValueAsString()
Returns the result of a DataWeave script execution as a string value.

`getValue()`

Returns the result of a DataWeave script execution as an object.

Signature

```
public Object getValue()
```

Return Value

Type: Object

`getValueAsString()`

Returns the result of a DataWeave script execution as a string value.

Signature

```
public String getValueAsString()
```

Return Value

Type: String

Script Class

Contains the `createScript()` method to load DataWeave scripts and the `execute()` method to obtain script output in a `DataWeave.Result` object.

Namespace

DataWeave

This example runs a DataWeave script from Apex and retrieves the resulting script output. First deploy the script to the org as `ContactsToJson.dwl`.

```
%dw 2.0
input records application/java
output application/json
---
{
  users: records map(record) -> {
    firstName: record.FirstName,
    lastName: record.LastName
  }
```

Then, execute the script from Apex.

```
List<Contact> data = [SELECT FirstName, LastName FROM Contact WHERE LastName LIMIT 5];
Map<String, Object> args = new Map<String, Object>{ 'records' => data };
DataWeave.Script script = DataWeave.Script.createScript('ContactsToJson');

DataWeave.Result result = script.execute(args);
string jsonOutput = result.getValueAsString();
```

IN THIS SECTION:

Script Methods

Script Methods

The following are methods for `Script`.

IN THIS SECTION:

createScript(scriptName)
Loads a DataWeave 2.0 script from the `.dwl` metadata file that is deployed in an org. The script can then be run using the `Script.execute` method.

createScript(namespace, scriptName)
Loads a DataWeave 2.0 script from a specified namespace. The script can then be run using the `Script.execute` method.

execute(parameters)
Executes the DataWeave script that is loaded using the `createScript()` method and returns the script output.

toString()
Returns the name of the script.

`createScript(scriptName)`

Loads a DataWeave 2.0 script from the `.dwl` metadata file that is deployed in an org. The script can then be run using the `Script.execute` method.

Signature

```
public static createScript(String scriptName)
```

Parameters

scriptName
Type: String

The name of the deployed metadata `.dwl` script (not including the file extension).

Return Value

Type: DataWeave.Script

DataWeave script that is used as a parameter in the `Script.execute()` method.

createScript(namespace, scriptName)

Loads a DataWeave 2.0 script from a specified namespace. The script can then be run using the `Script.execute` method.

Signature

```
public static dataweave.Script createScript(String namespace, String scriptName)
```

Parameters

namespace
Type: String

The namespace name for the deployed script. If the namespace name is null, the caller namespace is used. If the namespace name is empty, the org namespace is used.

scriptName
Type: String

The name of the deployed metadata `.dwl` script (not including the file extension).

Return Value

Type: DataWeave.Script

DataWeave script that is used as a parameter in the `Script.execute()` method.

execute(parameters)

Executes the DataWeave script that is loaded using the `createScript()` method and returns the script output.

Signature

```
public execute(Map<String,Object> parameters)
```

Parameters

parameters
Type: Map<String,Object>

Input to the DataWeave script. The keys correspond to the input directive names defined in the DataWeave header.

See Input Directive and DataWeave Header.

Return Value

Type: DataWeave.Result

The `DataWeave.Result` object contains the script output.

`toString()`

Returns the name of the script.

Signature

```
public String toString()
```

Return Value

Type: String

Dom Namespace

The `Dom` namespace provides classes and methods for parsing and creating XML content.

The following are the classes in the `Dom` namespace.

IN THIS SECTION:

Document Class
Use the `Document` class to process XML content. You can parse nested XML content that's up to 50 nodes deep.

XmlNode Class
Use the `XmlNode` class to work with a node in an XML document.

XmlNodeType Enum
Specifies the node type in an XML document.

Document Class

Use the `Document` class to process XML content. You can parse nested XML content that's up to 50 nodes deep.

Namespace

Dom

Usage

One common application is to use it to create the body of a request for HttpRequest or to parse a response accessed by HttpResponse.

IN THIS SECTION:

Document Constructors

Document Methods

SEE ALSO:

Reading and Writing XML Using the DOM

Document Constructors

The following are constructors for `Document`.

IN THIS SECTION:

Document()
Creates a new instance of the `Dom.Document` class.

`Document()`

Creates a new instance of the `Dom.Document` class.

Signature

```
public Document()
```

Document Methods

The following are methods for `Document`. All are instance methods.

IN THIS SECTION:

createRootElement(name, namespace, prefix)
Creates the top-level root element for a document.

getRootElement()
Returns the top-level root element node in the document. If this method returns `null`, the root element has not been created yet.

load(xml)
Parse the XML representation of the document specified in the *`xml`* argument and load it into a document.

toXmlString()
Returns the XML representation of the document as a String.

`createRootElement(name, namespace, prefix)`

Creates the top-level root element for a document.

Signature

```
public Dom.XmlNode createRootElement(String name, String namespace, String prefix)
```

Parameters

name
Type: String

namespace
Type: String

prefix
Type: String

Return Value

Type: Dom.XmlNode

Usage

For more information about namespaces, see Reading and Writing XML Using the DOM.

Calling this method more than once on a document generates an error as a document can have only one root element.

getRootElement()

Returns the top-level root element node in the document. If this method returns `null`, the root element has not been created yet.

Signature

```
public Dom.XmlNode getRootElement()
```

Return Value

Type: Dom.XmlNode

load(xml)

Parse the XML representation of the document specified in the *xml* argument and load it into a document.

Signature

```
public Void load(String xml)
```

Parameters

xml
Type: String

Return Value

Type: Void

Example

```
Dom.Document doc = new Dom.Document();
doc.load(xml);
```

`toXmlString()`

Returns the XML representation of the document as a String.

Signature

```
public String toXmlString()
```

Return Value

Type: String

XmlNode Class

Use the `XmlNode` class to work with a node in an XML document.

Namespace

Dom

XmlNode Methods

The following are methods for `XmlNode`. All are instance methods.

IN THIS SECTION:

addChildElement(name, namespace, prefix)
Creates a child element node for this node.

addCommentNode(text)
Creates a child comment node for this node.

addTextNode(text)
Creates a child text node for this node.

getAttribute(key, keyNamespace)
Returns *`namespacePrefix:attributeValue`* for the given key and key namespace.

getAttributeCount()
Returns the number of attributes for this node.

getAttributeKeyAt(index)
Returns the attribute key for the given index. Index values start at 0.

getAttributeKeyNsAt(index)
Returns the attribute key namespace for the given index.

getAttributeValue(key, keyNamespace)
Returns the attribute value for the given key and key namespace.

getAttributeValueNs(key, keyNamespace)
Returns the attribute value namespace for the given key and key namespace.

getChildElement(name, namespace)
Returns the child element node for the node with the given name and namespace.

getChildElements()
Returns the child element nodes for this node. This doesn't include child text or comment nodes.

getChildren()
Returns the child nodes for this node. This includes all node types.

getName()
Returns the element name.

getNamespace()
Returns the namespace of the element.

getNamespaceFor(prefix)
Returns the namespace of the element for the given prefix.

getNodeType()
Returns the node type.

getParent()
Returns the parent of this element.

getPrefixFor(namespace)
Returns the prefix of the given namespace.

getText()
Returns the text for this node.

insertBefore(newChild, refChild)
Inserts a new child node before the specified node.

removeAttribute(key, keyNamespace)
Removes the attribute with the given key and key namespace. Returns `true` if successful, `false` otherwise.

removeChild(childNode)
Removes the given child node.

setAttribute(key, value)
Sets the key attribute value.

setAttributeNs(key, value, keyNamespace, valueNamespace)
Sets the key attribute value.

setNamespace(prefix, namespace)
Sets the namespace for the given prefix.

addChildElement(name, namespace, prefix)

Creates a child element node for this node.

Signature

```
public Dom.XmlNode addChildElement(String name, String namespace, String prefix)
```

Parameters

name
Type: String

The *name* argument can't have a `null` value.

namespace
Type: String

prefix
Type: String

Return Value

Type: Dom.XmlNode

Usage

- If the *namespace* argument has a non-`null` value and the *prefix* argument is `null`, the namespace is set as the default namespace.
- If the *prefix* argument is `null`, Salesforce automatically assigns a prefix for the element. The format of the automatic prefix is `ns`***i***, where *i* is a number.If the *prefix* argument is `''`, the namespace is set as the default namespace.

addCommentNode(text)

Creates a child comment node for this node.

Signature

```
public Dom.XmlNode addCommentNode(String text)
```

Parameters

text
Type: String

The *text* argument can't have a `null` value.

Return Value

Type: Dom.XmlNode

addTextNode(text)

Creates a child text node for this node.

Signature

```
public Dom.XmlNode addTextNode(String text)
```

Parameters

text
Type: String

The *text* argument can't have a `null` value.

Return Value

Type: Dom.XmlNode

getAttribute(key, keyNamespace)

Returns *namespacePrefix:attributeValue* for the given key and key namespace.

Signature

```
public String getAttribute(String key, String keyNamespace)
```

Parameters

key
Type: String

keyNamespace
Type: String

Return Value

Type: String

Example

For example, for the `<xyz a:b="c:d" />` element:

- `getAttribute` returns `c:d`
- `getAttributeValue` returns `d`

getAttributeCount()

Returns the number of attributes for this node.

Signature

```
public Integer getAttributeCount()
```

Return Value

Type: Integer

getAttributeKeyAt(index)

Returns the attribute key for the given index. Index values start at 0.

Signature

```
public String getAttributeKeyAt(Integer index)
```

Parameters

index
Type: Integer

Return Value

Type: String

getAttributeKeyNsAt(index)

Returns the attribute key namespace for the given index.

Signature

```
public String getAttributeKeyNsAt(Integer index)
```

Parameters

index
Type: Integer

Return Value

Type: String

getAttributeValue(key, keyNamespace)

Returns the attribute value for the given key and key namespace.

Signature

```
public String getAttributeValue(String key, String keyNamespace)
```

Parameters

key
Type: String

keyNamespace
Type: String

Return Value

Type: String

Example

For example, for the `<xyz a:b="c:d" />` element:

- `getAttribute` returns `c:d`
- `getAttributeValue` returns `d`

getAttributeValueNs(key, keyNamespace)

Returns the attribute value namespace for the given key and key namespace.

Signature

```
public String getAttributeValueNs(String key, String keyNamespace)
```

Parameters

key
Type: String

keyNamespace
Type: String

Return Value

Type: String

getChildElement(name, namespace)

Returns the child element node for the node with the given name and namespace.

Signature

```
public Dom.XmlNode getChildElement(String name, String namespace)
```

Parameters

name
Type: String

namespace
Type: String

Return Value

Type: Dom.XmlNode

getChildElements()

Returns the child element nodes for this node. This doesn't include child text or comment nodes.

Signature

```
public Dom.XmlNode[] getChildElements()
```

Return Value

Type: Dom.XmlNode[]

getChildren()

Returns the child nodes for this node. This includes all node types.

Signature

```
public Dom.XmlNode[] getChildren()
```

Return Value

Type: Dom.XmlNode[]

getName()

Returns the element name.

Signature

```
public String getName()
```

Return Value

Type: String

getNamespace()

Returns the namespace of the element.

Signature

```
public String getNamespace()
```

Return Value

Type: String

getNamespaceFor(prefix)

Returns the namespace of the element for the given prefix.

Signature

```
public String getNamespaceFor(String prefix)
```

Parameters

prefix
Type: String

Return Value

Type: String

getNodeType()

Returns the node type.

Signature

```
public Dom.XmlNodeType getNodeType()
```

Return Value

Type: Dom.XmlNodeType

Uses `XmlNodeType` enum to return *COMMENT*, *ELEMENT*, or *TEXT* as the node type.

getParent()

Returns the parent of this element.

Signature

```
public Dom.XmlNode getParent()
```

Return Value

Type: Dom.XmlNode

getPrefixFor(namespace)

Returns the prefix of the given namespace.

Signature

```
public String getPrefixFor(String namespace)
```

Parameters

namespace
Type: String

The *namespace* argument can't have a `null` value.

Return Value

Type: String

getText()

Returns the text for this node.

Signature

```
public String getText()
```

Return Value

Type: String

insertBefore(newChild, refChild)

Inserts a new child node before the specified node.

Signature

```
public Dom.XmlNode insertBefore(Dom.XmlNode newChild, Dom.XmlNode refChild)
```

Parameters

newChild
Type: Dom.XmlNode

The node to insert.

refChild
Type: Dom.XmlNode

The node before the new node.

Return Value

Type: Dom.XmlNode

Usage

- If *refChild* is `null`, *newChild* is inserted at the end of the list.
- If *refChild* doesn't exist, an exception is thrown.

removeAttribute(key, keyNamespace)

Removes the attribute with the given key and key namespace. Returns `true` if successful, `false` otherwise.

Signature

```
public Boolean removeAttribute(String key, String keyNamespace)
```

Parameters

key
Type: String

keyNamespace
Type: String

Return Value

Type: Boolean

removeChild(childNode)

Removes the given child node.

Signature

```
public Boolean removeChild(Dom.XmlNode childNode)
```

Parameters

childNode
Type: Dom.XmlNode

Return Value

Type: Boolean

setAttribute(key, value)

Sets the key attribute value.

Signature

```
public Void setAttribute(String key, String value)
```

Parameters

key
Type: String

value
Type: String

Return Value

Type: Void

setAttributeNs(key, value, keyNamespace, valueNamespace)

Sets the key attribute value.

Signature

```
public Void setAttributeNs(String key, String value, String keyNamespace, String valueNamespace)
```

Parameters

key
Type: String

value
Type: String

keyNamespace
Type: String

valueNamespace
Type: String

Return Value

Type: Void

setNamespace(prefix, namespace)

Sets the namespace for the given prefix.

Signature

```
public Void setNamespace(String prefix, String namespace)
```

Parameters

prefix
Type: String

namespace
Type: String

Return Value

Type: Void

XmlNodeType Enum

Specifies the node type in an XML document.

Usage

Use `XMLNodeType` enum with the `getNodeType()` method in the `XmlNode` class.

Enum Values

The following are the values of the `Dom.XMLNodeType` enum.

Value	Description
`COMMENT`	Dom node of type comment.
`ELEMENT`	Dom node of type element.
`TEXT`	Dom node of type text.

EventBus Namespace

The `EventBus` namespace provides classes and methods for platform events and Change Data Capture events.

The following are the classes in the `EventBus` namespace.

IN THIS SECTION:

ChangeEventHeader Class

Contains header fields of Change Data Capture events.

EventPublishFailureCallback Interface

Implement this interface to track platform event messages that failed to publish. The `onFailure()` method in this interface is called when the final result of the asynchronous publish operation becomes available.

EventPublishSuccessCallback Interface

Implement this interface to track platform event messages that were published successfully. The `onSuccess()` method in this interface is called when the final result of the asynchronous publish operation becomes available.

FailureResult Interface

Contains the result of an Apex publish callback when the event publishing failed. This interface is used as a parameter in the `onFailure` method of the `EventPublishFailureCallback` interface.

SuccessResult Interface

Contains the result of an Apex publish callback when the event publishing succeeded. This interface is used as a parameter in the `onSuccess` method of the `EventPublishSuccessCallback` interface.

TestBroker Class

Contains methods that simulate the successful delivery or failed publishing of platform event or change event messages in an Apex test.

TriggerContext Class

Provides information about the platform event or change event trigger that's currently executing, such as how many times the trigger was retried due to the `EventBus.RetryableException`. Also, provides a method to resume trigger executions.

SEE ALSO:

Platform Events Developer Guide

ChangeEventHeader Class

Contains header fields of Change Data Capture events.

Namespace

EventBus

IN THIS SECTION:

ChangeEventHeader Properties

SEE ALSO:

Change Data Capture Developer Guide

ChangeEventHeader Properties

The following are properties for `ChangeEventHeader`.

IN THIS SECTION:

changedfields

A list of the fields that were changed in an update operation, including the `LastModifiedDate` system field. This field is empty for other operations, including record creation. This property is available in Apex saved using API version 47.0 or later.

changeorigin

Only populated for changes done by API apps or from Lightning Experience; empty otherwise. The Salesforce API and the API client ID that initiated the change, if set by the client. Use this field to detect whether your app initiated the change to not process the change again and potentially avoid a deep cycle of changes.

changetype

The operation that caused the change.

commitnumber

The system change number (SCN) of a committed transaction, which increases sequentially. This field is provided for diagnostic purposes. The field value is not guaranteed to be unique in Salesforce—it is unique only in a single database instance. If your Salesforce org migrates to another database instance, the commit number might not be unique or sequential.

committimestamp

The date and time when the change occurred, represented as the number of milliseconds since January 1, 1970 00:00:00 GMT.

commituser

The ID of the user that ran the change operation.

difffields

Contains the names of fields whose values are sent as a unified diff because they contain large text values.

entityname

The API name of the standard or custom object that the change pertains to. For example, Account or MyObject__c.

nulledfields

Contains the names of fields whose values were changed to null in an update operation. Use this field in Apex change event messages to determine if a field was changed to null in an update and isn't an unchanged field.

recordids

One or more record IDs for the changed records. Typically, this field contains one record ID. If in one transaction the same change occurred in multiple records of the same object type during one second, Salesforce merges the change notifications. In this case, Salesforce sends one change event for all affected records and the `recordIds` field contains the IDs for all records that have the same change.

sequencenumber

The sequence of the change within a transaction. The sequence number starts from 1.

transactionkey

A string that uniquely identifies each Salesforce transaction. You can use this key to identify and group all changes that were made in the same transaction.

changedfields

A list of the fields that were changed in an update operation, including the `LastModifiedDate` system field. This field is empty for other operations, including record creation. This property is available in Apex saved using API version 47.0 or later.

Signature

```
public List<String> changedfields {get; set;}
```

Property Value

Type: List<String>

changeorigin

Only populated for changes done by API apps or from Lightning Experience; empty otherwise. The Salesforce API and the API client ID that initiated the change, if set by the client. Use this field to detect whether your app initiated the change to not process the change again and potentially avoid a deep cycle of changes.

Signature

```
public String changeorigin {get; set;}
```

Property Value

Type: String

The format of the `changeOrigin` field value is:

```
com/salesforce/api/<API_Name>/<API_Version>;client=<Client_ID>
```

- `<API_Name>` is the name of the Salesforce API used to make the data change. It can take one of these values: soap, rest, bulkapi, xmlrpc, oldsoap, toolingsoap, toolingrest, apex, apexdebuggerrest.
- `<API_Version>` is the version of the API call that made the change and is in the format *XX.X*.
- `<Client_ID>` is a string that contains the client ID of the app that initiated the change. If the client ID is not set in the API call, `client=<Client_ID>` is not appended to the `changeOrigin` field.

Example:

```
com/salesforce/api/soap/49.0;client=Astro
```

The client ID is set in the Call Options header of an API call. For an example on how to set the Call Options header, see:

- REST API: Sforce-Call-Options Header. (Bulk API also uses the Sforce-Call-Options header.)
- SOAP API: CallOptions Header. (Apex API also uses the CallOptions element.)

changetype

The operation that caused the change.

Signature

```
public String changetype {get; set;}
```

Property Value

Type: String

Can be one of the following values:

- CREATE
- UPDATE
- DELETE
- UNDELETE
- SNAPSHOT (reserved for future use)

For gap events, the change type starts with the GAP_ prefix.

- GAP_CREATE
- GAP_UPDATE
- GAP_DELETE
- GAP_UNDELETE

For overflow events, the change type is GAP_OVERFLOW.

commitnumber

The system change number (SCN) of a committed transaction, which increases sequentially. This field is provided for diagnostic purposes. The field value is not guaranteed to be unique in Salesforce—it is unique only in a single database instance. If your Salesforce org migrates to another database instance, the commit number might not be unique or sequential.

Signature

```
public Long commitnumber {get; set;}
```

Property Value

Type: Long

committimestamp

The date and time when the change occurred, represented as the number of milliseconds since January 1, 1970 00:00:00 GMT.

Signature

```
public Long committimestamp {get; set;}
```

Property Value

Type: Long

commituser

The ID of the user that ran the change operation.

Signature

```
public String commituser {get; set;}
```

Property Value

Type: String

difffields

Contains the names of fields whose values are sent as a unified diff because they contain large text values.

Signature

```
public List<String> difffields {get; set;}
```

Property Value

Type: List<String>

SEE ALSO:

Change Data Capture Developer Guide: Sending Data Differences for Fields of Updated Records

entityname

The API name of the standard or custom object that the change pertains to. For example, Account or MyObject__c.

Signature

```
public String entityname {get; set;}
```

Property Value

Type: String

nulledfields

Contains the names of fields whose values were changed to null in an update operation. Use this field in Apex change event messages to determine if a field was changed to null in an update and isn't an unchanged field.

Signature

```
public List<String> nulledfields {get; set;}
```

Property Value

Type: List<String>

`recordids`

One or more record IDs for the changed records. Typically, this field contains one record ID. If in one transaction the same change occurred in multiple records of the same object type during one second, Salesforce merges the change notifications. In this case, Salesforce sends one change event for all affected records and the `recordIds` field contains the IDs for all records that have the same change.

Signature

```
public List<String> recordids {get; set;}
```

Property Value

Type: List<String>

Examples of operations with same changes are:

- Update of fieldA to valueA in Account records.
- Deletion of Account records.
- Renaming or replacing a picklist value that results in updating the field value in all affected records.

The `recordIds` field can contain a wildcard value when a change event message is generated for custom field type conversions that cause data loss. In this case, the `recordIds` value is the three-character prefix of the object, followed by the wildcard character `*`. For example, for accounts, the value is `001*`.

`sequencenumber`

The sequence of the change within a transaction. The sequence number starts from 1.

Signature

```
public Integer sequencenumber {get; set;}
```

Property Value

Type: Integer

A lead conversion is an example of a transaction that can have multiple changes. A lead conversion results in the following sequence of changes, all within the same transaction.

1. Create an account
2. Create a contact
3. Create an opportunity
4. Update a lead

`transactionkey`

A string that uniquely identifies each Salesforce transaction. You can use this key to identify and group all changes that were made in the same transaction.

Signature

```
public String transactionkey {get; set;}
```

Property Value

Type: String

EventPublishFailureCallback Interface

Implement this interface to track platform event messages that failed to publish. The `onFailure()` method in this interface is called when the final result of the asynchronous publish operation becomes available.

Namespace

EventBus

Usage

For more information, see Get the Result of Asynchronous Platform Event Publishing with Apex Publish Callbacks in the *Platform Events Developer Guide*.

IN THIS SECTION:

EventPublishFailureCallback Methods

EventPublishFailureCallback Example Implementation

EventPublishFailureCallback Methods

The following are methods for `EventPublishFailureCallback`.

IN THIS SECTION:

onFailure(result)
The system invokes this method when the final result of `EventBus.publish` is available and the publishing of the platform event message failed.

`onFailure(result)`

The system invokes this method when the final result of `EventBus.publish` is available and the publishing of the platform event message failed.

Signature

```
public void onFailure(eventbus.FailureResult result)
```

Parameters

result
Type: EventBus.FailureResult

The final result of `EventBus.publish`.

Return Value

Type: void

EventPublishFailureCallback Example Implementation

For an example implementation and a test class, see Get the Result of Asynchronous Platform Event Publishing with Apex Publish Callbacks in the *Platform Events Developer Guide*.

EventPublishSuccessCallback Interface

Implement this interface to track platform event messages that were published successfully. The `onSuccess()` method in this interface is called when the final result of the asynchronous publish operation becomes available.

Namespace

EventBus

Usage

For more information, see Get the Result of Asynchronous Platform Event Publishing with Apex Publish Callbacks in the *Platform Events Developer Guide*.

IN THIS SECTION:

EventPublishSuccessCallback Methods

EventPublishSuccessCallback Example Implementation

EventPublishSuccessCallback Methods

The following are methods for `EventPublishSuccessCallback`.

IN THIS SECTION:

onSuccess(result)
The system invokes this method when the final result of `EventBus.publish` is available and the publishing of the platform event message succeeded.

`onSuccess(result)`

The system invokes this method when the final result of `EventBus.publish` is available and the publishing of the platform event message succeeded.

Signature

```
public void onSuccess(eventbus.SuccessResult result)
```

Parameters

`result`
Type: EventBus.SuccessResult

The final result of `EventBus.publish`.

Return Value

Type: void

EventPublishSuccessCallback Example Implementation

For an example implementation and a test class, see Get the Result of Asynchronous Platform Event Publishing with Apex Publish Callbacks in the *Platform Events Developer Guide*.

FailureResult Interface

Contains the result of an Apex publish callback when the event publishing failed. This interface is used as a parameter in the `onFailure` method of the `EventPublishFailureCallback` interface.

Namespace

EventBus

IN THIS SECTION:

FailureResult Methods

FailureResult Methods

The following are methods for `FailureResult`.

IN THIS SECTION:

getEventUuids()
Returns a list of `EventUuid` field values of each platform event that is included in `EventBus.EventPublishFailureCallback`.

`getEventUuids()`

Returns a list of `EventUuid` field values of each platform event that is included in `EventBus.EventPublishFailureCallback`.

Signature

```
public List<String> getEventUuids()
```

Return Value

Type: List<String>

SuccessResult Interface

Contains the result of an Apex publish callback when the event publishing succeeded. This interface is used as a parameter in the `onSuccess` method of the `EventPublishSuccessCallback` interface.

Namespace

EventBus

IN THIS SECTION:

SuccessResult Methods

SuccessResult Methods

The following are methods for `SuccessResult`.

IN THIS SECTION:

getEventUuids()

Returns a list of `EventUuid` field values of each platform event that is included in the `EventBus.EventPublishSuccessCallback`.

`getEventUuids()`

Returns a list of `EventUuid` field values of each platform event that is included in the `EventBus.EventPublishSuccessCallback`.

Signature

```
public List<String> getEventUuids()
```

Return Value

Type: List<String>

TestBroker Class

Contains methods that simulate the successful delivery or failed publishing of platform event or change event messages in an Apex test.

Namespace

EventBus

IN THIS SECTION:

TestBroker Methods

TestBroker Methods

The following are methods for `TestBroker`.

IN THIS SECTION:

deliver()
Delivers platform event messages to the test event bus. Use this method to deliver test event messages multiple times and verify that event subscribers have processed the test events each step of the way.

fail()
Causes the publishing of platform event messages to fail in the test event bus. Use this method to test Apex publish callbacks.

`deliver()`

Delivers platform event messages to the test event bus. Use this method to deliver test event messages multiple times and verify that event subscribers have processed the test events each step of the way.

Signature

```
public void deliver()
```

Return Value

Type: void

Usage

Enclose `Test.getEventBus().deliver()` within the `Test.startTest()` and `Test.stopTest()` statement block.

```
Test.startTest();
// Create test events
// ...
// Publish test events with EventBus.publish()
// ...
// Deliver test events
Test.getEventBus().deliver();
// Perform validation
```

```
// ...
Test.stopTest();
```

SEE ALSO:

Platform Events Developer Guide

`fail()`

Causes the publishing of platform event messages to fail in the test event bus. Use this method to test Apex publish callbacks.

Signature

```
public void fail()
```

Return Value

Type: void

Usage

```
// Create test events
// ...
// Publish test events with EventBus.publish()
// ...
// Fail publishing of test events
Test.getEventBus().fail();
// Perform validation
// ...
```

For more information, see <link>Get the Result of Asynchronous Platform Event Publishing with Apex Publish Callbacks<link/> in the *Platform Events Developer Guide*.

TriggerContext Class

Provides information about the platform event or change event trigger that's currently executing, such as how many times the trigger was retried due to the `EventBus.RetryableException`. Also, provides a method to resume trigger executions.

Namespace

EventBus

IN THIS SECTION:

TriggerContext Properties

TriggerContext Methods

TriggerContext Properties

The following are properties for `TriggerContext`.

IN THIS SECTION:

lastError
Read-only. The error message that the last thrown `EventBus.RetryableException` contains.

retries
Read-only. The number of times the trigger was retried due to throwing the `EventBus.RetryableException`.

`lastError`

Read-only. The error message that the last thrown `EventBus.RetryableException` contains.

Signature

```
public String lastError {get;}
```

Property Value

Type: String

Usage

The error message that this property returns is the message that was passed in when creating the `EventBus.RetryableException` exception, as follows.

```
throw new EventBus.RetryableException(
             'Condition is not met, so retrying the trigger again.');
```

`retries`

Read-only. The number of times the trigger was retried due to throwing the `EventBus.RetryableException`.

Signature

```
public Integer retries {get;}
```

Property Value

Type: Integer

TriggerContext Methods

The following are methods for `TriggerContext`.

IN THIS SECTION:

currentContext()
Returns an instance of the `EventBus.TriggerContext` class containing information about the currently executing trigger.

getResumeCheckpoint()
Returns the replay ID that was set by `setResumeCheckpoint()`. The returned value is the replay ID of the event message after which trigger processing resumes in a new trigger invocation.

setResumeCheckpoint(resumeReplayId)
Sets a checkpoint in the event stream where the platform event trigger resumes execution in a new invocation. Use this method to recover from limit and uncaught exceptions, or to control the number of events processed in one trigger execution. When calling this method, pass in the replay ID of the last successfully processed event message. When the trigger stops execution before all events in `Trigger.New` are processed, either because of an uncaught exception or intentionally, the trigger is invoked again. The new execution starts with the event message in the stream after the one with the checkpointed Replay ID.

`currentContext()`

Returns an instance of the `EventBus.TriggerContext` class containing information about the currently executing trigger.

Signature

```
public static eventbus.TriggerContext currentContext()
```

Return Value

Type: EventBus.TriggerContext

Information about the currently executing trigger.

`getResumeCheckpoint()`

Returns the replay ID that was set by `setResumeCheckpoint()`. The returned value is the replay ID of the event message after which trigger processing resumes in a new trigger invocation.

Signature

```
public String getResumeCheckpoint()
```

Return Value

Type: String

`setResumeCheckpoint(resumeReplayId)`

Sets a checkpoint in the event stream where the platform event trigger resumes execution in a new invocation. Use this method to recover from limit and uncaught exceptions, or to control the number of events processed in one trigger execution. When calling this method, pass in the replay ID of the last successfully processed event message. When the trigger stops execution before all events in `Trigger.New` are processed, either because of an uncaught exception or intentionally, the trigger is invoked again. The new execution starts with the event message in the stream after the one with the checkpointed Replay ID.

Signature

```
public void setResumeCheckpoint(String resumeReplayId)
```

Parameters

resumeReplayId
Type: String

The replay ID of the last successfully processed platform event message, after which to resume processing in a new trigger execution context.

Return Value

Type: void

Usage

The method throws an `EventBus.InvalidReplayIdException` if the supplied Replay ID is not valid—the replay ID is not in the current trigger batch of events, in the `Trigger.new` list.

Example

This snippet shows how to call the method and pass in the replayId property of an event instance.

```
EventBus.TriggerContext.currentContext().setResumeCheckpoint(event.replayId);
```

ExternalService Namespace

The `ExternalService` namespace provides dynamically generated Apex service interfaces and Apex classes for complex object data types.

The `ExternalService` namespace doesn't define a fixed set of classes. The namespace reflects OpenAPI-compatible external service registrations with active operations for type-safe outbound calls. The object schema, in the API specification that is associated with the registered external service, maps to Apex types.

SEE ALSO:

Salesforce Help: Invoke External Service Callouts Using Apex

Flow Namespace

The `Flow` namespace provides a class for advanced access to flows from Apex such as from Visualforce controllers and asynchronous Apex.

The following is the class in the `Flow` namespace.

IN THIS SECTION:

Interview Class

The `Flow.Interview` class provides advanced controller access to flows and the ability to start a flow.

Interview Class

The `Flow.Interview` class provides advanced controller access to flows and the ability to start a flow.

Namespace

Flow

Usage

SOQL and DML limits apply during flow execution. See Per-Transaction Flow Limits in Salesforce Help.

To create an Interview object, you have two options.

Note: We recommend only using `createInterview()` if you must reuse your method or class. Using `createInterview()` has these drawbacks.

- If you package a class that uses `createInterview()`, you have to add the associated flow manually.
- If you delete a flow, Salesforce doesn't check if it's referenced with `createInterview()`.

- Create the object directly in your class by using:
  - No namespace: `Flow.Interview.flowName`
  - Namespace: `Flow.Interview.namespace.flowName`
- Create the object dynamically by using `createInterview()`

To enforce sharing rules, run the flow or Apex on API version 62.0 or later. The Apex class must be declared using the `with sharing` keyword. The flow runs more securely in the default context when an Apex class that's declared using the `with sharing` keyword launches an autolaunched flow. The flow enforces the sharing rules of the user that executes the Apex class. Data access is restricted to the sharing rules of the user that executed the Apex class. For example, a query can return fewer rows than it did in system context without sharing. An operation can fail because the user doesn't have the correct permissions.

Examples: Starting Flow Interviews

These examples are all sample controllers that start an interview for the flow from the Build a Discount Calculator project on Trailhead. Each shows a different permutation, based on:

- Whether the interview is created statically, with `Flow.Interview.myFlow`, or dynamically, with `createInterview()`.
- Whether the flow is managed or local.

Interview Created Statically for a Local Flow

```
{
  Map<String, Object> inputs = new Map<String, Object>();
  inputs.put('AccountID', myAccount);
  inputs.put('OpportunityID', myOppty);

  Flow.Interview.Calculate_discounts myFlow =
    new Flow.Interview.Calculate_discounts(inputs);
  myFlow.start();
}
```

Interview Created Dynamically for a Local Flow

```
public void callFlow(String flowName, Map <String, Object> inputs) {
  Flow.Interview myFlow = Flow.Interview.createInterview(flowName, inputs);
  myFlow.start();
}
```

Interview Created Statically for a Managed Flow

```
{
  Map<String, Object> inputs = new Map<String, Object>();
  inputs.put('AccountID', myAccount);
  inputs.put('OpportunityID', myOppty);

  Flow.Interview.myNamespace.Calculate_discounts myFlow =
    new Flow.Interview.myNamespace.Calculate_discounts(inputs);
  myFlow.start();
}
```

Interview Created Dynamically for a Managed Flow

```
public void callFlow(String namespace, String flowName, Map <String, Object> inputs) {
  Flow.Interview myFlow = Flow.Interview.createInterview(namespace, flowName, inputs);
  myFlow.start();
}
```

Example: Getting Variable Values

This sample uses the `getVariableValue` method to obtain breadcrumb (navigation) information from a flow. If that flow contains subflow elements, and each of the referenced flows also contains a *`vaBreadCrumb`* variable, you can provide users with breadcrumbs regardless of which flow the interview is running.

```
public class SampleController {

   //Instance of the flow
   public Flow.Interview.Flow_Template_Gallery myFlow {get; set;}

   public String getBreadCrumb() {
      String aBreadCrumb;
      if (myFlow==null) { return 'Home';}
      else aBreadCrumb = (String) myFlow.getVariableValue('vaBreadCrumb');

      return(aBreadCrumb==null ? 'Home': aBreadCrumb);

   }
}
```

SEE ALSO:

Tooling API Objects: FlowTestCoverage

Apex Developer Guide: Add a Test Class

Salesforce Help: Launch a Flow from Apex

Apex Developer Guide: Launch a Flow from Apex

Interview Methods

The following are instance methods for `Interview`.

`createInterview(namespace, flowName, inputVariables)`

Creates an interview for a namespaced flow.

Signature

```
public static Flow.Interview createInterview(String namespace, String flowName,
Map<String,ANY> inputVariables)
```

Parameters

`namespace`
Type: String

The flow's namespace.

`flowName`
Type: String

The flow's API name.

`inputVariables`
Type: Map<String,Object>

Initial values for the flow's input variables.

Return Value

Type: Flow.Interview

Usage

Use this method to dynamically create a Flow.Interview object for the `start()` method.

How you get output variable values from an interview depends on the type of the Apex variable where you're storing the interview.

- If the variable is cast to a specific flow, you can use *myFlow.myVar* to access a variable, where *myVar* is the name of the variable.

```
system.debug('My Output Variable: ' + myFlow.varName);
```

- If the variable is of type Flow.Interview but not cast to a specific flow, you must use getVariableValue() to access the flow's variables.

```
system.debug('My Output Variable: ' + myFlow.getVariableValue('varName'));
```

If the flow doesn't exist in the current org, a TypeException is thrown.

`createInterview(flowName, inputVariables)`

Creates an interview for a flow.

Signature

```
public static Flow.Interview createInterview(String flowName, Map<String,Object>
inputVariables)
```

Parameters

flowName
Type: String

The flow's API name.

inputVariables
Type: Map<String,Object>

Initial values for the flow's input variables.

Return Value

Type: Flow.Interview

Usage

Use this method to dynamically create a Flow.Interview object for the `start()` method.

How you get output variable values from an interview depends on the type of the Apex variable where you're storing the interview.

- If the variable is cast to a specific flow, you can use *myFlow.myVar* to access a variable, where *myVar* is the name of the variable.

```
system.debug('My Output Variable: ' + myFlow.varName);
```

- If the variable is of type Flow.Interview but not cast to a specific flow, you must use getVariableValue() to access the flow's variables.

```
system.debug('My Output Variable: ' + myFlow.getVariableValue('varName'));
```

If the flow doesn't exist in the current org, a TypeException is thrown.

`getVariableValue(variableName)`

Returns the value of the specified flow variable. The flow variable can be in the flow embedded in the Visualforce page, or in a separate flow that is called by a subflow element.

Signature

```
public Object getVariableValue(String variableName)
```

Parameters

variableName
Type: String

Specifies the unique name of the flow variable.

Return Value

Type: Object

Usage

The returned variable value comes from whichever flow the interview is running. If the specified variable can't be found in that flow, the method returns `null`.

This method checks for the existence of the variable at run time only, not at compile time.

`start()`

Starts an instance (interview) for an autolaunched or user provisioning flow.

Signature

```
public Void start()
```

Return Value

Type: Void

Usage

This method can be used only with flows that have one of these types.

- Autolaunched Flow
- User Provisioning Flow

For details, see "Flow Types" in Salesforce Help.

When a flow user invokes an autolaunched flow, the active flow version runs. If there's no active version, the latest version runs. When a flow admin invokes a flow, the latest version always runs.

FormulaEval Namespace

The FormulaEval namespace provides classes and methods to evaluate dynamic formulas for SObjects and Apex objects. Use the methods to avoid unnecessary DML statements to recalculate formula field values or evaluate dynamic formula expressions.

When using a formula against an SObject or Apex object as the context object, the class methods or properties referenced by the formula must be global.

```
Account myAcc = new Account(Name='123');
       FormulaEval.FormulaInstance ff = Formula.builder()
           .withType(Schema.Account.class)
           .withReturnType(FormulaEval.FormulaReturnType.STRING)
           .withFormula('name & " (" & website & ")"')
           .build();

//Use the list of field names returned by the getReferenced method to generate dynamic
soql
       String fieldNameList = String.join(ff.getReferencedFields(),',');
       String queryStr = 'select ' + fieldNameList + ' from Account LIMIT 1'; //select
name, website from Account
       Account s = Database.query(queryStr);
       system.debug(ff.evaluate(s));
```

The following are the classes and enums in the `FormulaEval` namespace.

IN THIS SECTION:

FormulaBuilder Class
Contains methods to build and validate user-defined formulas.

FormulaGlobal Enum
Specifies a global variable that references data in your organization in the `withGlobalVariables(formulaGlobals)` method.

FormulaInstance Class
Contains a method to evaluate the formula instance.

FormulaReturnType Enum
Specifies the return type for the `withReturnType(returnType)` method.

FormulaBuilder Class

Contains methods to build and validate user-defined formulas.

Namespace

FormulaEval

Usage

The context type that corresponds to the Apex class used in the builder `withType()` method must be a global, user-defined Apex class. Any fields or properties that the formula references must also be global.

IN THIS SECTION:

FormulaBuilder Methods

FormulaBuilder Methods

The following are methods for `FormulaBuilder`.

IN THIS SECTION:

build()
Validates and returns the formula instance created using the `FormulaBuilder` methods.

treatNumericNullAsZero(isNumericNullZero)
Optional. Indicates whether a null for a numeric data type is treated as zero while evaluating the formula with the `build()` method.

withFormula(formulaText)
Required. Sets the formula expression that the `build()` method uses to create the formula instance.

withGlobalVariables(formulaGlobals)
Optional. Sets the list of global variables that can be referenced in the formula expression created with the `build()` method.

withReturnType(returnType)
Required. Sets the formula output data type for the formula instance created with the `build()` method.

withType(contextType)
Sets the Apex type that corresponds to the Apex class used with the `build()` method.

withType(contextType)
Sets the Apex type that corresponds to the Apex class used with the `build()` method to SObject type.

build()

Validates and returns the formula instance created using the `FormulaBuilder` methods.

Signature

```
public FormulaEval.FormulaInstance build()
```

Return Value

Type: FormulaEval.FormulaInstance

Returns an instance of the `FormulaInstance` object. If the formula validation such as field references, functions, or syntax, fails, the method throws a `FormulaValidationException` exception.

treatNumericNullAsZero(isNumericNullZero)

Optional. Indicates whether a null for a numeric data type is treated as zero while evaluating the formula with the `build()` method.

Signature

```
public FormulaEval.FormulaBuilder treatNumericNullAsZero(Boolean isNumericNullZero)
```

Parameters

isNumericNullZero
Type: Boolean

If `true`, null for numeric is treated as zero. The default value is `false`.

Return Value

Type: FormulaEval.FormulaBuilder

withFormula(formulaText)

Required. Sets the formula expression that the `build()` method uses to create the formula instance.

Signature

```
public FormulaEval.FormulaBuilder withFormula(String formulaText)
```

Parameters

formulaText
Type: String

Return Value

Type: FormulaEval.FormulaBuilder

`withGlobalVariables(formulaGlobals)`

Optional. Sets the list of global variables that can be referenced in the formula expression created with the `build()` method.

Signature

```
public FormulaEval.FormulaBuilder withGlobalVariables(List<formulaeval.FormulaGlobal>
formulaGlobals)
```

Parameters

`formulaGlobals`
Type: List<FormulaEval.FormulaGlobal>

Uses values from the `FormulaGlobal` enum.

Return Value

Type: FormulaEval.FormulaBuilder

`withReturnType(returnType)`

Required. Sets the formula output data type for the formula instance created with the `build()` method.

Signature

```
public FormulaEval.FormulaBuilder withReturnType(formulaeval.FormulaReturnType
returnType)
```

Parameters

`returnType`
Type: FormulaEval.FormulaReturnType

Uses values from the `FormulaReturnType` enum.

Return Value

Type: FormulaEval.FormulaBuilder

`withType(contextType)`

Sets the Apex type that corresponds to the Apex class used with the `build()` method.

Signature

```
public formulaeval.FormulaBuilder withType(System.Type contextType)
```

Parameters

contextType
Type: System.Type

An instance of the Apex class type.

Return Value

Type: FormulaEval.FormulaBuilder

withType(contextType)

Sets the Apex type that corresponds to the Apex class used with the `build()` method to SObject type.

Signature

```
public formulaeval.FormulaBuilder withType(Schema.SObjectType contextSObjectType)
```

Parameters

contextSObjectType
Type: Schema.SObjectType

An instance of the SObject type.

Return Value

Type: FormulaEval.FormulaBuilder

Example

This example uses an SObject type as an input in the `withType()` method to build and evaluate a formula.

```
FormulaEval.FormulaInstance ff = Formula.builder()
    .withReturnType(FormulaEval.FormulaReturnType.Boolean)
    .withType(Account.SObjectType)
    .withFormula('ISBLANK(Site)')
    .build();

Boolean siteIsBlank = (Boolean)ff.evaluate(new Account(Site='Test'));
Assert.isFalse(siteIsBlank);
```

FormulaGlobal Enum

Specifies a global variable that references data in your organization in the `withGlobalVariables(formulaGlobals)` method.

Enum Values

The following are the values of the `FormulaEval.FormulaGlobal` enum.

Value	Description
`CUSTOMMETADATA`	A custom metadata record.
`LABEL`	A global variable to use when referencing a custom label.
`ORGANIZATION`	A global variable to use when referencing information about your company profile, such as organization's city, fax, ID, or other details.
`PERMISSION`	A global variable to use when referencing information about the current user's custom permission access.
`PROFILE`	A global variable to use when referencing information about the current user's profile, such as license type or name.
`SETUP`	A global variable to use when referencing a custom setting of type `hierarchy`.
`SYSTEM`	A global variable that exposes *`OriginDateTime`* and represents the literal value of 1900-01-01 00:00:00. Use this global variable when performing date/time offset calculations, or to assign a literal value to a date/time field.
`USER`	A global variable to use when referencing information about the current user, such as alias, title, and ID.
`USERROLE`	A global variable to use when referencing information about the current user's role, such as role name, description, and ID.

FormulaInstance Class

Contains a method to evaluate the formula instance.

Namespace

FormulaEval

Example

```
global class MotorYacht {
   global Integer lengthInYards;
   global Integer numOfGuestCabins;
   global String name;
   global Account owner;
}
```

```
MotorYacht aBoat = new MotorYacht();
aBoat.lengthInYards = 52;
aBoat.numOfGuestCabins = 4;
aBoat.name = 'RV Foo';
FormulaEval.FormulaInstance isItSuper = Formula.builder()
                            .withReturnType(FormulaEval.FormulaReturnType.STRING)
                            .withType(MotorYacht.class)
                            .withFormula('IF(lengthInYards < 100, "Not Super", "Super")')
```

```
                            .build();
isItSuper.evaluate(aBoat); //=> "Not Super"

aBoat.owner = new Account(Name='Acme Watercraft', Site='New York');
FormulaEval.FormulaInstance ownerDetails = Formula.builder()
                            .withReturnType(FormulaEval.FormulaReturnType.STRING)
                            .withType(MotorYacht.class)
                            .withFormula('owner.Name & " (" & owner.Site & ")"')
                            .build();
ownerDetails.evaluate(aBoat); //=> "Acme Watercraft (New York)"
```

Usage

The context type in the `withType` method must be a global, user-defined Apex class. Any fields or properties that the formula references must also be global.

IN THIS SECTION:

FormulaInstance Methods

FormulaInstance Methods

The following are methods for `FormulaInstance`.

IN THIS SECTION:

evaluate(contextObject)

Calculates the formula expression and returns the formula output.

getReferencedFields()

Returns a set of strings that denote the field names referenced in a formula.

`evaluate(contextObject)`

Calculates the formula expression and returns the formula output.

Signature

```
public Object evaluate(Object contextObject)
```

Parameters

`contextObject`

Type: Object

An instance of the Apex class as generated with the `FormulaBuilder.builder()` method.

Return Value

Type: Object

Apex type that corresponds to the Apex class as configured by the `withType()` method in the `FormulaBuilder` class.

`getReferencedFields()`

Returns a set of strings that denote the field names referenced in a formula.

Signature

```
public Set<String> getReferencedFields()
```

Return Value

Type: Set<String>

Usage

A formula is built and evaluated in the context of the current namespace of the subscriber org. If you package a formula that references fields, the fields must be fully qualified with the namespace name.

Example

```
FormulaEval.FormulaInstance ff = Formula.builder()
                                 .withType(Schema.Account.class)
                                 .withReturnType(FormulaEval.FormulaReturnType.STRING)
                                 .withFormula('name & website')
                                 .build();

// Returns the field names 'name', and 'website' required to process the formula
Set<String> fieldNames = ff.getReferencedFields();

// Use the list of field names to generate dynamic soql
String queryStr = 'select ' + string.join(fieldNames, ', ') + ' from Account limit 1';
List<sObject> accounts = Database.query(queryStr);
string formulaOutput = (string)ff.evaluate(accounts[0]);
System.debug(formulaOutput);
FormulaEval.FormulaInstance ff = Formula.builder()
```

FormulaReturnType Enum

Specifies the return type for the `withReturnType(returnType)` method.

Enum Values

The following are the values of the `FormulaEval.FormulaReturnType` enum.

Value	Description
`BOOLEAN`	A value that can only be assigned `true`, `false`, or `null`.
`DATE`	A value that indicates a particular day.
`DATETIME`	A value that indicates a particular day and time, such as a timestamp.

Value	Description
`DECIMAL`	A number that includes a decimal point. Decimal is an arbitrary precision number.
`DOUBLE`	A 64-bit number that includes a decimal point.
`ID`	Any valid 18-character Lightning Platform record identifier.
`INTEGER`	A 32-bit number that doesn't include a decimal point.
`LONG`	A 64-bit number that doesn't include a decimal point.
`STRING`	Any set of characters surrounded by single quotes.
`TIME`	A value that indicates a particular time.

fsccashflow Namespace

The `fsccashflow` namespace provides classes used in the FSCCashFlow Flexcards and its child Flexcards.

The `fsccashflow` namespace has these classes.

IN THIS SECTION:

FSCCashFlowUtil Class

Use the callable FSCCashFlowUtil class to manage and validate data for party income and expense entities by passing in the action and the corresponding arguments. This class provides utility methods used in FSCCashFlow Flexcard and its child Flexcards.

FSCCashFlowUtil Class

Use the callable FSCCashFlowUtil class to manage and validate data for party income and expense entities by passing in the action and the corresponding arguments. This class provides utility methods used in FSCCashFlow Flexcard and its child Flexcards.

Namespace

fsccashflow Namespace

Usage

The Financial Goals FlexCards use Integration Procedures that call the FSCHouseholdService class. These FlexCards display information about Financial Goals.

IN THIS SECTION:

FSCCashFlowUtil Methods

FSCCashFlowUtil Methods

The `FSCCashFlowUtil` has these methods.

IN THIS SECTION:

GetPartyIncomeFrequencyLabel
Returns the picklist values for the party income frequency field on the party income entity.

GetPartyIncomeTypeLabel
Returns the picklist values for the party income type field on the party income entity.

GetPartyIncomeStatusLabel
Returns the picklist values for the party income status field on the party income entity.

CalculateIncomeExpenseSummary
Calculates the monthly income, total income, average monthly income, monthly expense, total expense, average monthly expense from a list of income and expenses provided.

GetPartyExpenseFrequencyLabel
Returns the picklist values for the party expense frequency field on the party expense entity.

GetPartyExpenseTypeLabel
Returns the picklist values for the party expense type field on the party expense entity.

GetPartyExpenseStatusLabel
Returns the picklist values for the party expense status field on the party expense entity.

PerformIncomeValidation
Performs validations on Party Income records. Ensure that the start date is not earlier than the end date.

PerformExpenseValidation
Performs validations on Party Income records.

GetDurationDateRange
Returns the start and end date given a duration.For example, if you input the number 3 on the date 10/29/2024, it will return a start date of 7/1/2024 and an end date of 10/1/2024.

HandleUpsertError
Helper method that constructs the error response for upsert of a partyIncome or partyExpense record.

CheckReadAccess
Checks for read access on the partyIncome and partyExpense entities.

CheckCrudOnIncome
Checks create, update and delete access on partyIncome entity.

CheckCrudOnExpense
Checks create, update and delete access on partyExpense entity.

`GetPartyIncomeFrequencyLabel`

Returns the picklist values for the party income frequency field on the party income entity.

Signature

```
call(String action, Map<String, Object> args
```

Return Value

Returns a list of picklist labels for Party Income frequency.

GetPartyIncomeTypeLabel

Returns the picklist values for the party income type field on the party income entity.

Signature

```
call(String action, Map<String, Object> args
```

Return Value

Returns a list of picklist labels for Party Income type.

GetPartyIncomeStatusLabel

Returns the picklist values for the party income status field on the party income entity.

Signature

```
call(String action, Map<String, Object> args
```

Return Value

Returns a list of picklist labels for Party Income status.

CalculateIncomeExpenseSummary

Calculates the monthly income, total income, average monthly income, monthly expense, total expense, average monthly expense from a list of income and expenses provided.

Signature

```
call(String action, Map<String, Object> args
```

Return Value

Returns income and expense details.

Examples

Input and output JSON example of the actions are as follows.

Input format:

```
[
    {
        "Duration": "12",
        "PartyExpenseList": [
            {
                "Name": "PE-0000000004",
                "UsageType": "CashFlow",
                "RecurrenceInterval": "Monthly",
                "Type": "Child Care",
                "Id": "2n3SG000007dkzpYAA",
```

```
                "TotalAmount": 999.99,
                "PartyId": "001SG000004TCczYAG",
                "Status": "Active",
                "StartDate": "2024-01-29T08:00:00.000Z"
            }
        ],
        "PartyIncomeList": [
            {
                "Name": "PI-0000000003",
                "UsageType": "CashFlow",
                "IncomeFrequency": "Monthly",
                "IncomeType": "Salary",
                "Id": "2m3SG000007dkzpYAA",
                "IncomeAmount": 999.99,
                "PartyId": "001SG000004TCczYAG",
                "IncomeStatus": "Active",
                "StartDate": "2024-01-29T08:00:00.000Z"
            }
        ]
    }
]
```

Output format:

```
[
    {
        "MonthlyIncome": {
            "Nov 2023": 0,
            "Aug 2024": 999.99,
            "Oct 2023": 0,
            "Jan 2024": 96.7732258064516,
            "Mar 2024": 999.99,
            "Jul 2024": 999.99,
            "Apr 2024": 999.99,
            "Dec 2023": 0,
            "Jun 2024": 999.99,
            "Sep 2024": 999.99,
            "Feb 2024": 999.99,
            "May 2024": 999.99
        },
        "MonthlyExpense": {
            "Nov 2023": 0,
            "Aug 2024": 999.99,
            "Oct 2023": 0,
            "Jan 2024": 96.7732258064516,
            "Mar 2024": 999.99,
            "Jul 2024": 999.99,
            "Apr 2024": 999.99,
            "Dec 2023": 0,
            "Jun 2024": 999.99,
            "Sep 2024": 999.99,
            "Feb 2024": 999.99,
            "May 2024": 999.99
        },
        "AvgMonthlyExpense": 674.72,
```

```
        "TotalIncome": 8096.69,
        "TotalSurplus": 0,
        "AvgMonthlyIncome": 674.72,
        "AvgMonthlySurplus": 0,
        "TotalExpense": 8096.69
    }
]
```

GetPartyExpenseFrequencyLabel

Returns the picklist values for the party expense frequency field on the party expense entity.

Signature

```
call(String action, Map<String, Object> args
```

Return Value

Returns a list of picklist labels for Party Expense frequency.

GetPartyExpenseTypeLabel

Returns the picklist values for the party expense type field on the party expense entity.

Signature

```
call(String action, Map<String, Object> args
```

Return Value

Returns a list of picklist labels for Party Expense type.

GetPartyExpenseStatusLabel

Returns the picklist values for the party expense status field on the party expense entity.

Signature

```
call(String action, Map<String, Object> args
```

Return Value

Returns a list of picklist labels for Party Expense status.

PerformIncomeValidation

Performs validations on Party Income records. Ensure that the start date is not earlier than the end date.

Signature

```
call(String action, Map<String, Object> args
```

Return Value

Returns a list of picklist labels for Party Income type.

Examples

Input and output JSON example of the actions are as follows.

Input format:

```
[
    {
        "IncomeFrequency": "Weekly",
        "IncomeFrequencyLabelObject": {
            "value": "Weekly",
            "label": "Weekly"
        },
        "MemberOptionsList": [
            {
                "value": "001OG00000xx6gAYAQ",
                "label": "Okee PA"
            },
            {
                "value": "id2",
                "label": "Name2"
            }
        ],
        "IsHousehold": true,
        "IncomeAmount": 100,
        "IncomeStatusOptions": [
            {
                "value": "Active",
                "label": "Active"
            },
            {
                "value": "Inactive",
                "label": "Inactive"
            }
        ],
        "PartyId": "001OG00000xx6gAYAQ",
        "IncomeStatus": "Active",
        "Party": {
            "Name": "Okee PA",
            "Id": "001OG00000xx6gAYAQ"
        },
        "IncomeTypeOptions": [
            {
                "value": "Salary",
                "label": "Salary"
            },
            {
                "value": "Commission",
                "label": "Commission"
            },
            {
```

```
                "value": "Fees",
                "label": "Fees"
            },
            {
                "value": "Rent",
                "label": "Rent"
            }
        ],
        "StartDate": "2024-02-02T00:00:00.000Z",
        "Name": "PI-0000000009",
        "FrequencyOptions": [
            {
                "value": "Weekly",
                "label": "Weekly"
            },
            {
                "value": "Monthly",
                "label": "Monthly"
            },
            {
                "value": "Yearly",
                "label": "Yearly"
            }
        ],
        "UsageType": "CashFlow",
        "IncomeId": "2m3OG000000009xxAQ",
        "IsPersonAccount": false,
        "IncomeTypeLabelObject": {
            "value": "Salary",
            "label": "Salary"
        }
    }
]
```

Output format:

```
[
    {
        "dateErrorMessage": null,
        "IncomeFrequency": "Weekly",
        "IncomeFrequencyLabelObject": {
            "value": "Weekly",
            "label": "Weekly"
        },
        "MemberOptionsList": [
            {
                "value": "001OG000003f6gAYAQ",
                "label": "Okee PA"
            },
            {
                "value": "id2",
                "label": "Name2"
            }
        ],
        "requiredFieldErrorMessage": "Required fields:Type",
```

```
        "IsHousehold": true,
        "IncomeAmount": 100,
        "PartyId": "001OG000003f6gAYAQ",
        "IncomeStatusOptions": [
            {
                "value": "Active",
                "label": "Active"
            },
            {
                "value": "Inactive",
                "label": "Inactive"
            }
        ],
        "Party": {
            "Name": "Okee PA",
            "Id": "001OG000003f6gAYAQ"
        },
        "IncomeStatus": "Active",
        "hasUpsertError": false,
        "IncomeTypeOptions": [
            {
                "value": "Salary",
                "label": "Salary"
            },
            {
                "value": "Commission",
                "label": "Commission"
            },
            {
                "value": "Fees",
                "label": "Fees"
            },
            {
                "value": "Rent",
                "label": "Rent"
            }
        ],
        "StartDate": "2024-02-02T00:00:00.000Z",
        "Name": "PI-0000000009",
        "FrequencyOptions": [
            {
                "value": "Weekly",
                "label": "Weekly"
            },
            {
                "value": "Monthly",
                "label": "Monthly"
            },
            {
                "value": "Yearly",
                "label": "Yearly"
            }
        ],
        "UsageType": "CashFlow",
```

```
        "IncomeId": "2m3OG000000009IYAQ",
        "IsPersonAccount": false,
        "IncomeTypeLabelObject": {
            "value": "Salary",
            "label": "Salary"
        }
    }
]
```

`PerformExpenseValidation`

Performs validations on Party Income records.

Signature

```
call(String action, Map<String, Object> args
```

Return Value

Returns a list of picklist labels for Party Income frequency.

Examples

Output JSON example of the actions are as follows.

Output format:

```
{
    "Required fields": "Expense Type, Member, Amount, Start Date, Frequency"
}
```

`GetDurationDateRange`

Returns the start and end date given a duration.For example, if you input the number 3 on the date 10/29/2024, it will return a start date of 7/1/2024 and an end date of 10/1/2024.

Signature

```
call(String action, Map<String, Object> args
```

Return Value

Returns the start and end date for a specified duration.

Examples

Output JSON example of the actions are as follows.

Output format:

```
{
          "DurationStartDate": "2024-02-02T00:00:00.000Z",
```

```
          "DurationEndDate": "2024-05-02T00:00:00.000Z"
        }
```

HandleUpsertError

Helper method that constructs the error response for upsert of a partyIncome or partyExpense record.

Signature

```
call(String action, Map<String, Object> args
```

Return Value

Returns a list of errors encountered while upserting the record in the database.

Examples

Input and output JSON example of the action are as follows.

Input format:

```
[
    {
        "Name": "PI-0000000003",
        "UsageType": "CashFlow",
        "IncomeFrequency": "Monthly",
        "IncomeType": "Salary",
        "Id": "2m3SG000007dkxxYAA",
        "IncomeAmount": 999.99,
        "PartyId": "001SG000004TCxxYAG",
        "IncomeStatus": "Active",
        "StartDate": "2024-01-29T08:00:00.000Z"
    }
]
```

Output format:

```
[ { "UpsertError‟: "Invalid Id‟ } ]
```

CheckReadAccess

Checks for read access on the partyIncome and partyExpense entities.

Signature

```
call(String action, Map<String, Object> args
```

Return Value

Returns True or False based on whether read access is granted or not.

Examples

Output JSON example of the action are as follows.

Output format:

```
{ "isAccessible" : "true" }
```

`CheckCrudOnIncome`

Checks create, update and delete access on partyIncome entity.

Signature

```
call(String action, Map<String, Object> args
```

Return Value

Returns True or False based on whether create, update and delete access on the partyIncome entity is given.

Examples

Output JSON example of the action are as follows.

Output format:

```
{ "isCreatable" : "true", "isUpdateable" : "true", "isDeletable": "true" }
```

`CheckCrudOnExpense`

Checks create, update and delete access on partyExpense entity.

Signature

```
call(String action, Map<String, Object> args
```

Return Value

Returns True or False based on whether create, update and delete access on the partyExpense entity is given.

Examples

Output JSON example of the action are as follows.

Output format:

```
{ "isCreatable" : "true", "isUpdateable" : "true", "isDeletable": "true" }
```

Functions Namespace

The Functions namespace provides classes and methods used to invoke and manage Salesforce Functions.

Salesforce Functions is your code, run on demand, in the Salesforce Functions trusted elastic compute cloud. Upload your complex business logic code, written using your preferred languages and frameworks, and Salesforce Functions takes care of everything else necessary to invoke your code in a secure, multi-tenant aware, and self-scaling environment. For more details on Salesforce Functions, see Salesforce Functions.

The following are the classes in the `functions` namespace.

IN THIS SECTION:

Function Class
Use the Function class to access deployed Salesforce Functions, and invoke them synchronously or asynchronously.

FunctionCallback Interface
Represents the callback Salesforce calls when an asynchronous, queued Function invocation has completed.

FunctionErrorType Enum
Represents the error type of FunctionInvocationError.

FunctionInvocation Interface
Use FunctionInvocation to get the status and results of a synchronous or asynchronous Function invocation.

FunctionInvocationError Interface
Use FunctionInvocationError to get detailed error information about a failed Function invocation.

FunctionInvocationStatus Enum
Represents the status of a Function invocation.

FunctionInvokeMock Interface
Use the `FunctionInvokeMock` interface to mock Salesforce Functions responses during testing.

MockFunctionInvocationFactory Class
Use the `MockFunctionInvocationFactory` methods to generate appropriate mock responses for testing Salesforce Functions.

Function Class

Use the Function class to access deployed Salesforce Functions, and invoke them synchronously or asynchronously.

Namespace

functions

Usage

The Function class represents an instance of a deployed Function you can invoke from your org. You can invoke Functions synchronously, or asynchronously using asynchronous Apex.

If your Function takes longer than 2 minutes to return, the request times out. To avoid timing out, consider using asynchronous invocation. Invoking a Function asynchronously doesn't count against asynchronous Apex limits, such as Apex Queueable limits.

Before synchronously invoking a Function, commit any pending data operations in Apex, otherwise you get a CalloutException. For asynchronous invocations, the queued invocation doesn't happen if the Apex transaction isn't committed. Any data operations that happen in the Function itself aren't considered part of the Apex transaction.

Functions can't be invoked in an Apex test. A "Function invocations from Apex tests are not supported" CalloutException is thrown if Apex determines that a Function is being invoked during a test. If you must run tests against code that invokes Functions, mock your Function invocations during the tests. See FunctionInvocation Example Implementation for an example of a mocked FunctionInvocation that you can use in testing.

Example

The following example synchronously invokes a deployed "accountfunction" Function:

```
functions.Function accountFunction = functions.Function.get('MyProject.accountfunction');
functions.FunctionInvocation invocation = accountFunction.invoke('{ "accountName" : "Acct",
 "contactName" : "MyContact", "opportunityName" : "Oppty" }');
String jsonResponse = invocation.getResponse();
```

The following example asynchronously invokes a deployed "AccountFunction" Function, using the provided callback:

```
functions.Function accountFunction = functions.Function.get('MyProject.accountfunction');
accountFunction.invoke('{ "accountName" : "Acct", "contactName" : "MyContact",
"opportunityName" : "Oppty" }', new MyCallback());

public class MyCallback
  implements functions.FunctionCallback {
    public void handleResponse(functions.FunctionInvocation result) {
      // Handle result of function invocation
      // ...
    }
}
```

IN THIS SECTION:

Function Methods

Function Methods

The following are methods for `Function`.

IN THIS SECTION:

get(functionName)

Returns the Function instance for the named Function and Project. The Function must be properly deployed and have appropriate permissions to work with the org running your Apex code.

get(namespace, functionName)

Returns the Function instance for the named Function, Project, and Namespace. The Function must be properly deployed and have appropriate permissions to work with the org running your Apex code.

invoke(payload, callback)

Invokes the Function asynchronously.

invoke(payload)

Invokes the Function synchronously.

`get(functionName)`

Returns the Function instance for the named Function and Project. The Function must be properly deployed and have appropriate permissions to work with the org running your Apex code.

Signature

```
public static functions.Function get(String functionName)
```

Parameters

`functionName`
Type: String

The name of the Salesforce Function and the Functions Project that the Function is part of. The format of the parameter string is "*`project name.function name`*". For example, to retrieve the `generatepdf` Function in the `Onboarding` Function Project, use `Onboarding.generatepdf`. The Function and Project must be deployed to a compute environment connected to the org.

Return Value

Type: functions.Function

Returns a Function instance that you can invoke.

Usage

The `Function.get()` method can throw the following exceptions:

- `InvalidParameterValueException` — The *`functionName`* parameter doesn't have the correct *`project name.function name`* format.
- `NoDataFoundException` — The project or Function name provided in the *`functionName`* parameter couldn't be found. Make sure the project and Function name are spelled correctly and that the project and Function have been properly deployed.

get(namespace, functionName)

Returns the Function instance for the named Function, Project, and Namespace. The Function must be properly deployed and have appropriate permissions to work with the org running your Apex code.

Signature

```
public static functions.Function get(String namespace, String functionName)
```

Parameters

`namespace`
Type: String

The name of the Namespace that both the Salesforce Function and the Functions Project are part of. The org the Function is in must be `global` to access across namespaces. Default value is the same org where the method is being called.

`functionName`
Type: String

The name of the Salesforce Function and the Functions Project that the Function is part of. The format of the parameter string is "*project name*.*function name*". For example, to retrieve the `generatepdf` Function in the `Onboarding` Function Project, use `Onboarding.generatepdf`. The Function and Project must be deployed to a compute environment connected to the org.

Return Value

Type: functions.Function

Returns a Function instance that you can invoke.

Usage

The `Function.get()` method can throw the following exceptions:

- `InvalidParameterValueException` — The *`functionName`* parameter doesn't have the correct *`project name.function name`* format.
- `NoDataFoundException` — The project or Function name provided in the *`functionName`* parameter couldn't be found. Make sure the project and Function name are spelled correctly and that the project and Function have been properly deployed.
- `RuntimeException` — The function is `public` yet references a function across namespaces. Make sure to retrieve references across namespaces only in a `global` org.

`invoke(payload, callback)`

Invokes the Function asynchronously.

Signature

```
public functions.FunctionInvocation invoke(String payload, functions.FunctionCallback callback)
```

Parameters

`payload`
Type: String

The payload data that gets passed to the Function. Specify your payload data in a JSON-format string.

`callback`
Type: functions.FunctionCallback

A FunctionCallback implementation that gets called when your Function is invoked asynchronously.

Return Value

Type: functions.FunctionInvocation

Returns a FunctionInvocation that contains information about the results of the invocation, such as the Function response, or error results.

Usage

The `Function.invoke(payload, callback)` method can throw the following exceptions:

- `CalloutException` — One of the following conditions causes this exception to be thrown:
  - Salesforce Functions isn't enabled on the current org. For more details on enabling Functions, see Configure Orgs for Functions in the Functions Developer Guide.
  - The Function is being invoked in an Apex test. Functions can't be invoked in tests.
  - The "Functions" permission set is missing or has incorrect permissions for `FunctionInvocationRequest`. For more details on the correct permissions for `FunctionInvocationRequest` see Function Permissions in the Functions Developer Guide.
  - The provided payload isn't valid JSON.
  - The Function hasn't completed deployment to a compute environment or invocation request returns a 404 HTTP error.
- `InvalidParameterValueException` — The *`callback`* parameter is null or references a class that doesn't implement `functions.FunctionCallback`.
- `NoDataFoundException` — A reference for the Function couldn't be found in the current org. Make sure the project and Function have been properly deployed.

`invoke(payload)`

Invokes the Function synchronously.

Signature

```
public functions.FunctionInvocation invoke(String payload)
```

Parameters

`payload`
Type: String

The payload data that gets passed to the Function. Specify your payload data in a JSON-format string.

Return Value

Type: functions.FunctionInvocation

Returns a FunctionInvocation that contains information about the results of the invocation, such as the Function response, or error results.

Usage

The `Function.invoke(payload)` method can throw the following exceptions:

- `CalloutException` — One of the following conditions causes this exception to be thrown:
  - Salesforce Functions isn't enabled on the current org. For more details on enabling Functions, see Configure Orgs for Functions in the Functions Developer Guide.
  - The Function is being invoked in an Apex test. Functions can't be invoked in tests.
  - The provided payload isn't valid JSON.
  - There are pending DML operations.
  - The Function is being synchronously invoked from an Apex trigger.
  - The Function hasn't completed deployment to a compute environment or invocation request returns a 404 HTTP error.

  - The Function request returns a 5xx HTTP error.
  - The Function invocation has exceeded the time limit for synchronous invocations. For details on the time limit and work-arounds, see Limits in the Functions Developer Guide.
- `NoDataFoundException` — A reference for the Function couldn't be found in the current org. Make sure the project and Function have been properly deployed.

FunctionCallback Interface

Represents the callback Salesforce calls when an asynchronous, queued Function invocation has completed.

Namespace

functions

Usage

When invoking Functions asynchronously via `Function.invoke(payload, callback)`, you provide your own class that implements FunctionCallback.

IN THIS SECTION:

FunctionCallback Methods

FunctionCallback Example Implementation

FunctionCallback Methods

The following are methods for `FunctionCallback`.

IN THIS SECTION:

handleResponse(var1)
Called when an asynchronous Function invocation has completed.

`handleResponse(var1)`

Called when an asynchronous Function invocation has completed.

Signature

```
public void handleResponse(functions.FunctionInvocation var1)
```

Parameters

`var1`

Type: functions.FunctionInvocation

The result parameter contains JSON response information and error information.

Return Value

Type: void

FunctionCallback Example Implementation

This is an example implementation of the `functions.FunctionCallback` interface.

```
public class MyCallback
  implements functions.FunctionCallback {
    public void handleResponse(functions.FunctionInvocation result) {
      // Handle result of function invocation
      String jsonResponse = result.getResponse();
      System.debug('Got response ' + jsonResponse);
      JSONParser parser = JSON.createParser(jsonResponse);
      // Process JSON using your own data class...
    }
}
```

The following example uses this implementation when invoking a Function asynchronously:

```
myFunction.invoke('{ "accountName" : "Acct", "contactName" : "MyContact", "opportunityName"
 : "Oppty" }', new MyCallback());
```

FunctionErrorType Enum

Represents the error type of FunctionInvocationError.

Enum Values

These are the values of the `functions.FunctionErrorType` enum.

Value	Description
`FUNCTION_EXCEPTION`	A known exception resulting from the Function logic itself. Examples include an exception thrown from the Function code, or an exception thrown from a library or framework the Function uses.
`RUNTIME_EXCEPTION`	A known exception resulting from the Salesforce Functions runtime. For example, a malformed payload passed to the Function when invoked results in this error type.
`UNEXPECTED_FUNCTION_EXCEPTION`	An unknown exception. For example, a network or system-level issue within the Salesforce Functions infrastructure results in this error type.

FunctionInvocation Interface

Use FunctionInvocation to get the status and results of a synchronous or asynchronous Function invocation.

Namespace

functions

Usage

The results of a Function invocation are passed back via FunctionInvocation. Use this instance to determine if the invocation was successful, and any results from the Function invocation.

You can also implement your own FunctionInvocation interface if you run Apex tests with your Function invocation code. Your test code can create and use your own FunctionInvocation instance in place of using the results from a call to `Function.invoke()`.

IN THIS SECTION:

FunctionInvocation Methods

FunctionInvocation Example Implementation

FunctionInvocation Methods

The following are methods for `FunctionInvocation`.

IN THIS SECTION:

getError()
Returns error information for a Function invocation.

getInvocationId()
Returns the invocation ID of the Function invocation.

getResponse()
Returns the response string of the Function invocation.

getStatus()
Returns the status of the Function invocation.

`getError()`

Returns error information for a Function invocation.

Signature

```
public functions.FunctionInvocationError getError()
```

Return Value

Type: functions.FunctionInvocationError

Contains a `FunctionInvocationError` instance that you can use to get information about any invocation errors. If the Function was invoked successfully, the returned instance is null.

`getInvocationId()`

Returns the invocation ID of the Function invocation.

Signature

```
public String getInvocationId()
```

Return Value

Type: String

This ID is available after a call to either the synchronous or asynchronous `Function.invoke()` methods. For asynchronous invocations, this ID can be used to check the status of the queued invocation.

`getResponse()`

Returns the response string of the Function invocation.

Signature

```
public String getResponse()
```

Return Value

Type: String

The response string is the raw request JSON response, which can be parsed using the JSONParser Class.

`getStatus()`

Returns the status of the Function invocation.

Signature

```
public functions.FunctionInvocationStatus getStatus()
```

Return Value

Type: functions.FunctionInvocationStatus

The result of the invocation, such as `FunctionInvocationStatus.SUCCESS` or `FunctionInvocationStatus.ERROR`.

FunctionInvocation Example Implementation

This is an example implementation of the `functions.FunctionInvocation` interface.

```
public class MyFunctionInvocationError
  implements functions.FunctionInvocationError {
    public String getMessage() {
      return 'Mock error message for testing';
    }
    public functions.FunctionErrorType getType() {
      return functions.FunctionErrorType.FUNCTION_EXCEPTION;
    }
}

public class MyFunctionInvocation
  implements functions.FunctionInvocation {
    public functions.FunctionInvocationStatus getStatus() {
      return functions.FunctionInvocationStatus.ERROR;
    }
```

```
    public String getResponse() {
      return 'Mock response for testing';
    }
    public String getInvocationId() {
      return 'MOCKTESTID';
    }
    public functions.FunctionInvocationError getError() {
      functions.FunctionInvocationError testError = new MyFunctionInvocationError();
      return testError;
    }
}
```

The following example tests the implementation:

```
functions.FunctionInvocation testInvocation = new MyFunctionInvocation();
if (testInvocation.getStatus() == functions.FunctionInvocationStatus.ERROR) {
    System.debug('Error: ' + (testInvocation.getError() != null ?
testInvocation.getError().getMessage() : 'UNKNOWN'));
    return;
}
```

FunctionInvocationError Interface

Use FunctionInvocationError to get detailed error information about a failed Function invocation.

Namespace

functions

Usage

FunctionInvocationError contains various error information such as the error message at the time of the error.

IN THIS SECTION:

FunctionInvocationError Methods

FunctionInvocationError Example Implementation

FunctionInvocationError Methods

The following are methods for `FunctionInvocationError`.

IN THIS SECTION:

getMessage()

Returns the error message from a Function invocation error.

getType()

Returns the error type for FunctionInvocationError.

`getMessage()`

Returns the error message from a Function invocation error.

Signature

```
public String getMessage()
```

Return Value

Type: String

`getType()`

Returns the error type for FunctionInvocationError.

Signature

```
public functions.FunctionErrorType getType()
```

Return Value

Type: functions.FunctionErrorType

FunctionInvocationError Example Implementation

This is an example implementation of the `functions.FunctionInvocationError` interface.

```
public class MyFunctionInvocationError
  implements functions.FunctionInvocationError {
      public String getMessage() {
          return 'Mock error message for testing';
      }
      public functions.FunctionErrorType getType() {
          return functions.FunctionErrorType.FUNCTION_EXCEPTION;
      }
}
```

This example tests the implementation.

```
functions.FunctionInvocationError testError = new MyFunctionInvocationError();
System.debug('Error: ' + testError.getMessage() + ' Type: ' + testError.getType());
```

FunctionInvocationStatus Enum

Represents the status of a Function invocation.

Enum Values

The following are the values of the `functions.FunctionInvocationStatus` enum.

Value	Description
`ERROR`	The invocation failed. Check the FunctionInvocation and FunctionInvocationError returned by the invoke call to debug the issue.
`PENDING`	The invocation is pending. If the Function is being invoked asynchronously, the invocation is still in the asynch queue.
`SUCCESS`	The invocation succeeded. Use `FunctionInvocation.getResponse()` with the FunctionInvocation instance returned by the invoke call to get any response returned by the Function.

FunctionInvokeMock Interface

Use the `FunctionInvokeMock` interface to mock Salesforce Functions responses during testing.

Namespace

functions

Usage

To mock Salesforce Functions testing, implement an appropriate mock response in the `respond(functionName,payload)` method of the `FunctionInvokeMock` interface. During mock testing of a Salesforce Functions, Apex runtime sends the response specified in the `respond()` method, rather than invoking the function itself. Appropriate success and error messages can be configured with the `createSuccessResponse(invocationId,message)` and `createErrorResponse(invocationId,functionsErrorType,errorMsg)` methods in `Functions.MockFunctionInvocationFactory`.

IN THIS SECTION:

FunctionInvokeMock Methods

FunctionInvokeMock Example Implementation

FunctionInvokeMock Methods

The following are methods for `FunctionInvokeMock`.

IN THIS SECTION:

respond(functionName, payload)

The mock response implemented in the `Functions.FunctionInvokeMock` interface. The response is sent by Apex runtime when the `Test.setMock()` method is called.

`respond(functionName, payload)`

The mock response implemented in the `Functions.FunctionInvokeMock` interface. The response is sent by Apex runtime when the `Test.setMock()` method is called.

Signature

```
public functions.FunctionInvocation respond(String functionName, String payload)
```

Parameters

functionName
Type: String

The name of the Salesforce Function and the Functions Project that the Function is part of. The format of the parameter string is "*project name.function name*".

payload
Type: String

The JSON-format payload data that is passed to the Function.

Return Value

Type: FunctionInvocation Interface

The result of the mock call to Salesforce Functions. Appropriate responses can be generated by using the `createSuccessResponse()` and `createErrorResponse()` methods in the `Functions.MockFunctionInvocationFactory` class.

FunctionInvokeMock Example Implementation

This is sample implementation of the `functions.FunctionInvokeMock` interface.

```
@isTest
public class FunctionsInvokeMockImpl implements functions.FunctionInvokeMock {
    public functions.FunctionInvocation respond(String functionName, String payload) {
       // return mock success response
       String invocationId = '000000000000000';
       String response = 'mockResponse';
       return functions.MockFunctionInvocationFactory.createSuccessResponse(invocationId,
 response);
    }
}
```

This example shows the minimal setup required for testing synchronous and asynchronous functions and is simplified to include both function invocations and the `FunctionCallback` class.

```
@isTest
public class FunctionTest {
    @isTest
    static void testSyncFunctionCall() {
            // Set mock class to respond to function invocations
        Test.setMock( functions.FunctionInvokeMock.class, new FunctionsInvokeMockInner());


          functions.Function mockedFunction = functions.Function.get('example.function');


            Test.startTest();
            // Synchronous function call
```

```
            functions.FunctionInvocation invokeResult = mockedFunction.invoke('{}');
            Test.stopTest();

            // Verify that the received response contains expected mock values
            System.assertEquals(functions.FunctionInvocationStatus.SUCCESS,
invokeResult.getStatus());
            System.assertEquals('mockResponse', invokeResult.getResponse());
            System.assertEquals('000000000000000', invokeResult.getInvocationId());


        }

        @isTest
        static void testAsyncFunctionCall() {
            // Set mock class to respond to function invocations
            Test.setMock( functions.FunctionInvokeMock.class, new
FunctionsInvokeMockInner());

          functions.Function mockedFunction = functions.Function.get('example.function2');


            Test.startTest();
            //Asynchronous function invocation with callback
            mockedFunction.invoke('{}', new DemoCallback());
            Test.stopTest();
            // Include assertions here about the expected callback processing
        }

         public class DemoCallback implements functions.FunctionCallback {
            public void handleResponse(functions.FunctionInvocation invokeResult) {
                // Handle result of function invocation
                // The callback is included in the example here for convenience
                // It would normally be defined in the classes being tested

                // Verify that the received response contains expected mock values
                System.assertEquals(invokeResult.getStatus(),
functions.FunctionInvocationStatus.ERROR);
                functions.FunctionInvocationError resultError = invokeResult.getError();
             System.assertEquals('bang', invokeResult.getError().getMessage());
             System.assertEquals('000000000000000', invokeResult.getInvocationId());

            }
        }

        public class FunctionsInvokeMockInner implements functions.FunctionInvokeMock {
          public functions.FunctionInvocation respond(String functionName, String payload)
 {
                // return mock success response
                String invocationId = '000000000000000';

                if(functionName == 'example.function2') {
                    return functions.MockFunctionInvocationFactory.createErrorResponse(
                        invocationId,
                        functions.FunctionErrorType.FUNCTION_EXCEPTION,
```

```
                        'bang');
                }

                String response = 'mockResponse';
                return
functions.MockFunctionInvocationFactory.createSuccessResponse(invocationId, response);

            }
        }


    }
```

MockFunctionInvocationFactory Class

Use the `MockFunctionInvocationFactory` methods to generate appropriate mock responses for testing Salesforce Functions.

Namespace

functions

Usage

To mock Salesforce Functions testing, implement an appropriate mock response in the `respond(functionName,payload)` method of the `FunctionInvokeMock` interface. During mock testing of a Salesforce Functions, the Apex runtime sends the response specified in the `respond()` method, rather than invoking the function itself. Appropriate success and error messages can be configured with the `createSuccessResponse(invocationId,message)` and `createErrorResponse(invocationId,functionsErrorType,errorMsg)` methods.

See FunctionInvokeMock Example Implementation.

IN THIS SECTION:

MockFunctionInvocationFactory Methods

MockFunctionInvocationFactory Methods

The following are methods for `MockFunctionInvocationFactory`.

IN THIS SECTION:

createErrorResponse(invocationId, functionsErrorType, errMsg)
Generate a response for an error condition during mock testing of Salesforce Functions.

createSuccessResponse(invocationId, response)
Generate a response for a successful mock test of Salesforce Functions.

`createErrorResponse(invocationId, functionsErrorType, errMsg)`

Generate a response for an error condition during mock testing of Salesforce Functions.

Signature

```
public static functions.FunctionInvocation createErrorResponse(String invocationId,
functions.FunctionErrorType functionsErrorType, String errMsg)
```

Parameters

invocationId
Type: String

The ID associated with a call to either the synchronous or asynchronous `Function.invoke()` method.

functionsErrorType
Type: FunctionErrorType Enum

The error type of `FunctionInvocationError`.

errMsg
Type: String

The error message.

Return Value

Type: FunctionInvocation Interface

createSuccessResponse(invocationId, response)

Generate a response for a successful mock test of Salesforce Functions.

Signature

```
public static functions.FunctionInvocation createSuccessResponse(String invocationId,
String response)
```

Parameters

invocationId
Type: String

The ID associated with a call to either the synchronous or asynchronous `Function.invoke()` method.

response
Type: String

The message indicating success.

Return Value

Type: FunctionInvocation Interface

industriesNlpSvc

Stores the objects used in Industries Einstein Natural Language Processing (NLP) services.

The industriesNlpSvc namespace contains these classes that are the outputs for the transformNlpActionResult Invocable action.

- `NlpResponse` — Stores the NLP Summarization result performed for an NLP Operation involving summarization use cases such as SurveyLongSummarization and SurveyShortSummarization.
- `NlpSummarizationResult` — Provides the summary obtained as result of NLP Operation.

IN THIS SECTION:

NlpResponse Class
Stores the result for an NLP Operation. NLP operation can be SurveyLongSummarization and SurveyShortSummarization.

NlpSummarizationResult Class
Provides the summary obtained as result of NLP Operation.

NlpResponse Class

Stores the result for an NLP Operation. NLP operation can be SurveyLongSummarization and SurveyShortSummarization.

Namespace

industriesNlpSvc

IN THIS SECTION:

NlpResponse Properties

NlpResponse Properties

The following are properties for `NlpResponse`.

IN THIS SECTION:

summarizationResult
Represents the property that stores the NLP Summarization result performed for an NLP Operation. NLP operation can be SurveyLongSummarization and SurveyShortSummarization.

errors
Represents the property to store errors that occurred as a result of the NLP Operation.

`summarizationResult`

Represents the property that stores the NLP Summarization result performed for an NLP Operation. NLP operation can be SurveyLongSummarization and SurveyShortSummarization.

Signature

```
public industriesNlpSvc.NlpSummarizationResult summarizationResult {get; set;}
```

Property Value

Type: List<industriesNlpSvc.NlpSummarizationResult on page 2604>

errors

Represents the property to store errors that occurred as a result of the NLP Operation.

Signature

```
public List<String> errors {get; set;}
```

Property Value

Type: List<String>

NlpSummarizationResult Class

Provides the summary obtained as result of NLP Operation.

Namespace

industriesNlpSvc

IN THIS SECTION:

NlpSummarizationResult Properties

NlpSummarizationResult Properties

The following are properties for `NlpSummarizationResult`:

IN THIS SECTION:

summary
Represents the field that captures the summary obtained as result of NLP Operation.

summary

Represents the field that captures the summary obtained as result of NLP Operation.

Signature

```
public String summary {get; set;}
```

Property Value

Type: List<String>

IndustriesDigitalLending Namespace

The `industriesDigitalLending` namespace provides classes used in the Digital Lending OmniScripts and Integration Procedures.

The industriesDigitalLending namespace contains these classes:

- `DigitalLendingIntakeRecordsWrapper` — Use the callable DigitalLendingIntakeRecordsWrapper class to call utility methods from OmniScripts used in Digital Lending application intake process.
- `DigitalLendingPostIntakeRecordsWrapper` — Use the callable DigitalLendingPostIntakeRecordsWrapper class to call utility methods from integration procedures used in Digital Lending post intake in FlexCards.
- `DigitalLendingProductsApi` — Use the callable DigitalLendingProductsApi class to call utility methods from integration procedures used in Digital Lending FlexCards.
- `DigitalLendingUtils` — Use the callable DigitalLendingUtils class to call utility methods from integration procedures used in Digital Lending PostIntake FlexCards.
- `PricingExecutionWrapper` — Use the callable PricingExecutionWrapper class to call utility methods from integration procedures used in Digital Lending FlexCards.

See industriesDigitalLending namespace for more information about the available classes and methods.

Invocable Namespace

The `Invocable` namespace provides classes for calling invocable actions from Apex.

These classes are in the `Invocable` namespace.

IN THIS SECTION:

Action Class
Contains methods to create, update, and retrieve information about invocable actions.

Action.Error Class
Contains methods to retrieve errors returned by invocable actions.

Action.Result Class
Contains methods to retrieve results from invocable actions called from Apex code.

Action Class

Contains methods to create, update, and retrieve information about invocable actions.

Namespace

Invocable

IN THIS SECTION:

Action Methods

SEE ALSO:

Apex Developer Guide: InvocableMethod Annotation

Salesforce Help: Launch a Flow from Apex

Action Methods

These methods are for `Action`.

IN THIS SECTION:

addInvocation()
Creates an empty invocation in preparation for calling an invocable action. After you create the invocation, you can add parameters to the invocation.

clearInvocations()
Clears the existing invocations from the action.

clone()
Creates a copy of the `Invocable.Action`.

createCustomAction(type, namespace, name)
Creates a wrapper for a custom invocable action in a specified package namespace.

createCustomAction(type, name)
Creates a wrapper for a custom invocable action.

createStandardAction(type)
Creates a wrapper for a standard invocable action.

getName()
Gets the name of an invocable action.

getNamespace()
Gets the namespace of a custom invocable action.

getType()
Gets the type of an invocable action.

invoke()
Invokes an invocable action from Apex code.

isStandard()
Determines whether an invocable action is a standard invocable action.

setInvocationParameter(parameterName, parameterValue)
Sets a value for an invocable action parameter.

setInvocations(invocations)
Initializes the invocations for an action from a pre-existing list of invocations.

`addInvocation()`

Creates an empty invocation in preparation for calling an invocable action. After you create the invocation, you can add parameters to the invocation.

Signature

```
public Invocable.Action addInvocation()
```

Return Value

Type: Invocable.Action on page 2605

clearInvocations()

Clears the existing invocations from the action.

Signature

```
public Invocable.Action clearInvocations()
```

Return Value

Type: Invocable.Action on page 2605

clone()

Creates a copy of the `Invocable.Action`.

Signature

```
public Object clone()
```

Return Value

Type: Object

createCustomAction(type, namespace, name)

Creates a wrapper for a custom invocable action in a specified package namespace.

Signature

```
public static Invocable.Action createCustomAction(String type, String namespace, String
name)
```

Parameters

type
Type: String

Type of invocable action.

namespace
Type: String

Namespace where the invocable action is located.

name
Type: String

Name for the custom invocable action.

Return Value

Type: Invocable.Action

createCustomAction(type, name)

Creates a wrapper for a custom invocable action.

Signature

```
public static Invocable.Action createCustomAction(String type, String name)
```

Parameters

type
Type: String

Type of invocable action.

name
Type: String

Name for the custom invocable action.

Return Value

Type: Invocable.Action

createStandardAction(type)

Creates a wrapper for a standard invocable action.

Signature

```
public static Invocable.Action createStandardAction(String type)
```

Parameters

type
Type: String

Type of invocable action.

Return Value

Type: Invocable.Action

getName()

Gets the name of an invocable action.

Signature

```
public String getName()
```

Return Value

Type: String

Name of the invocable action.

getNamespace()

Gets the namespace of a custom invocable action.

Signature

```
public String getNamespace()
```

Return Value

Type: String

Namespace of the custom invocable action.

getType()

Gets the type of an invocable action.

Signature

```
public String getType()
```

Return Value

Type: String

Type of invocable action.

invoke()

Invokes an invocable action from Apex code.

Signature

```
public List<Invocable.Action.Result> invoke()
```

Return Value

Type: List<Invocable.Action.Result>

isStandard()

Determines whether an invocable action is a standard invocable action.

Signature

```
public Boolean isStandard()
```

Return Value

Type: Boolean

This method returns `true` if the invocable action is a standard invocable action.

`setInvocationParameter(parameterName, parameterValue)`

Sets a value for an invocable action parameter.

Signature

```
public Invocable.Action setInvocationParameter(String parameterName, Object
parameterValue)
```

Parameters

parameterName
Type: String

Name of the invocable action parameter to set.

parameterValue
Type: Object

Value to set the invocable action parameter to.

Return Value

Type: Invocable.Action on page 2605

`setInvocations(invocations)`

Initializes the invocations for an action from a pre-existing list of invocations.

Signature

```
public Invocable.Action setInvocations(List<Map<String,ANY>> invocations)
```

Parameters

invocations
Type: List on page 3603<Map on page 3624<String on page 3833,ANY>>

List of invocations for the invocable action.

Return Value

Type: Invocable.Action on page 2605

Action.Error Class

Contains methods to retrieve errors returned by invocable actions.

Namespace

Invocable

IN THIS SECTION:

Action.Error Methods

Action.Error Methods

These methods are for `Action.Error`.

IN THIS SECTION:

clone()
Creates a copy of the `Invocable.Action.Error`.
getCode()
Gets the error code returned by an invocable action.
getMessage()
Gets the error message returned by an invocable action.

`clone()`

Creates a copy of the `Invocable.Action.Error`.

Signature

```
public Object clone()
```

Return Value

Type: Object

`getCode()`

Gets the error code returned by an invocable action.

Signature

```
public String getCode()
```

Return Value

Type: String

`getMessage()`

Gets the error message returned by an invocable action.

Signature

```
public String getMessage()
```

Return Value

Type: String

Action.Result Class

Contains methods to retrieve results from invocable actions called from Apex code.

Namespace

Invocable

IN THIS SECTION:

Action.Result Methods

Action.Result Methods

The methods are for `Action.Result`.

IN THIS SECTION:

clone()
Creates a copy of the `Invocable.Action.Result`.

getAction()
Gets the invocable action that was invoked and caused a result to be returned.

getErrors()
Gets a list of errors that were returned by an invocable action.

getInvocationParameters()
Gets a list of the parameter values set for an invocable action. This method returns a list that contains the input parameter values for each invocation of an action. Each map in the list contains a key for the name of each input parameter.

getOutputParameters()
Gets a list of the parameter values returned by an invocable action. This method returns a list that contains the result for each invocation of an action. Each map in the list contains a key for the name of each output parameter.

isSuccess()
Determines if an invocable action ran without errors.

`clone()`

Creates a copy of the `Invocable.Action.Result`.

Signature

```
public Object clone()
```

Return Value

Type: Object

`getAction()`

Gets the invocable action that was invoked and caused a result to be returned.

Signature

```
public Invocable.Action getAction()
```

Return Value

Type: Invocable.Action on page 2605

`getErrors()`

Gets a list of errors that were returned by an invocable action.

Signature

```
public List on page 3833<Invocable.Action.Error on page 2610> getErrors()
```

Return Value

Type: List on page 3833<Invocable.Action.Error on page 2610>

`getInvocationParameters()`

Gets a list of the parameter values set for an invocable action. This method returns a list that contains the input parameter values for each invocation of an action. Each map in the list contains a key for the name of each input parameter.

Signature

```
public Map<String,Object> getInvocationParameters()
```

Return Value

Type: Map on page 3624<String on page 3833,Object>

`getOutputParameters()`

Gets a list of the parameter values returned by an invocable action. This method returns a list that contains the result for each invocation of an action. Each map in the list contains a key for the name of each output parameter.

Signature

```
public Map<String,Object> getOutputParameters()
```

Return Value

Type: Map on page 3624<String on page 3833,Object>

`isSuccess()`

Determines if an invocable action ran without errors.

Signature

```
public Boolean isSuccess()
```

Return Value

Type: Boolean

This method returns `true` if the invocable action ran successfully.

IsvPartners Namespace

The `IsvPartners` namespace provides a class associated with Salesforce ISV partner use cases, such as optimizing code, providing great customer trial experiences, and driving feature adoption.

These are the classes in the `IsvPartners` namespace.

IN THIS SECTION:

AppAnalytics Class
Contains methods to help with AppExchange App Analytics use cases, such as minimizing subscriber attrition and obtaining product insights.

AppAnalytics Class

Contains methods to help with AppExchange App Analytics use cases, such as minimizing subscriber attrition and obtaining product insights.

Namespace

IsvPartners

Usage

Use `AppAnalytics` and its methods to log App Analytics custom interactions.

Example

```
public void submitClicked() {
        Id jobId = System.enqueueJob(new MyQueueable(colorValue));
        IsvPartners.AppAnalytics.logCustomInteraction(
             MyPageInteractions.SUBMIT_CLICKED, jobId);
```

IN THIS SECTION:

AppAnalytics Methods

AppAnalytics Methods

These are methods for `AppAnalytics`.

IN THIS SECTION:

logCustomInteraction(interactionLabel, interactionId)
Logs the custom interaction using a label that you provide as an enum value and an interaction ID.

logCustomInteraction(interactionLabel, interactionUuid)
Logs the custom interaction using a label that you provide as an enum value and an interaction ID that you provide as an Apex UUID.

logCustomInteraction(interactionLabel)
Logs the custom interaction using a label that you provide as an enum value.

`logCustomInteraction(interactionLabel, interactionId)`

Logs the custom interaction using a label that you provide as an enum value and an interaction ID.

Signature

```
public static void logCustomInteraction(Object interactionLabel, Id interactionId)
```

Parameters

`interactionLabel`
Type: Object

A value used to label the custom interaction. The value of *`interactionLabel`* must be an enum with the same namespace as the code that calls the `logCustomInteraction` method.

`interactionId`
Type: Id

An Apex ID that is associated with the custom interaction. The `interactionId` that you provide is hashed and tokenized before it's included in AppExchange App Analytics package usage logs.

Return Value

Type: Void

`logCustomInteraction(interactionLabel, interactionUuid)`

Logs the custom interaction using a label that you provide as an enum value and an interaction ID that you provide as an Apex UUID.

Signature

```
public static void logCustomInteraction(Object interactionLabel, System.UUID
interactionUuid)
```

Parameters

interactionLabel
Type: Object

A value used to label the custom interaction. The value of *interactionLabel* must be an enum with the same namespace as the code that calls the `logCustomInteraction` method.

interactionUuid
Type: System.UUID

An Apex UUID that is associated with the custom interaction. The `interactionId` that you provide is hashed and tokenized before being included in AppExchange App Analytics package usage logs.

Return Value

Type: Void

`logCustomInteraction(interactionLabel)`

Logs the custom interaction using a label that you provide as an enum value.

Signature

```
public static void logCustomInteraction(Object interactionLabel)
```

Parameters

interactionLabel
Type: Object

A value used to label the custom interaction. The value of *interactionLabel* must be an enum with the same namespace as the code that calls the `logCustomInteraction` method.

Return Value

Type: Void

KbManagement Namespace

The `KbManagement` namespace provides a class for managing knowledge articles.

The following is the class in the `KbManagement` namespace.

IN THIS SECTION:

PublishingService Class
Use the methods in the `KbManagement.PublishingService` class to manage the lifecycle of an article and its translations.

PublishingService Class

Use the methods in the `KbManagement.PublishingService` class to manage the lifecycle of an article and its translations.

Namespace

KbManagement

Usage

Use the methods in the `KbManagement.PublishingService` class to manage the following parts of the lifecycle of an article and its translations:

- Publishing
- Updating
- Retrieving
- Deleting
- Submitting for translation
- Setting a translation to complete or incomplete status
- Archiving
- Assigning review tasks for draft articles or translations

Note: Date values are based on GMT.

To use the methods in this class, you must enable Salesforce Knowledge. See Salesforce Knowledge Implementation Guide for more information on setting up Salesforce Knowledge.

PublishingService Methods

The following are methods for `PublishingService`. All methods are static.

IN THIS SECTION:

archiveOnlineArticle(articleId, scheduledDate)

Archives an online version of an article. If the specified scheduledDate is null, the article is archived immediately. Otherwise, it archives the article on the scheduled date.

assignDraftArticleTask(articleId, assigneeId, instructions, dueDate, sendEmailNotification)

Assigns a review task related to a draft article.

assignDraftTranslationTask(articleVersionId, assigneeId, instructions, dueDate, sendEmailNotification)

Assigns a review task related to a draft translation.

cancelScheduledArchivingOfArticle(articleId)

Cancels the scheduled archiving of an online article.

cancelScheduledPublicationOfArticle(articleId)

Cancels the scheduled publication of a draft article.

completeTranslation(articleVersionId)

Puts a translation in a completed state that is ready to publish.

deleteArchivedArticle(articleId)

Deletes an archived article.

deleteArchivedArticleVersion(articleId, versionNumber)

Deletes a specific archived version of a published article.

deleteDraftArticle(articleId)

Deletes a draft article.

deleteDraftTranslation(articleVersionId)

Deletes a draft translation.

editArchivedArticle(articleId)

Creates a draft article from the archived primary version and returns the new draft primary version ID of the article.

editOnlineArticle(articleId, unpublish)

Creates a draft article from the online version and returns the new draft primary version ID of the article. Also, unpublishes the online article, if *unpublish* is set to `true`.

editPublishedTranslation(articleId, language, unpublish)

Creates a draft version of the online translation for a specific language and returns the new draft primary version ID of the article. Also, unpublishes the article, if set to `true`.

publishArticle(articleId, flagAsNew)

Publishes an article. If *flagAsNew* is set to `true`, the article is published as a major version.

restoreOldVersion(articleId, versionNumber)

Creates a draft article from an existing online article based on the specified archived version of the article and returns the article version ID.

scheduleForPublication(articleId, scheduledDate)

Schedules the article for publication as a major version. If the specified date is null, the article is published immediately.

setTranslationToIncomplete(articleVersionId)

Sets a draft translation that is ready for publication back to "in progress" status.

submitForTranslation(articleId, language, assigneeId, dueDate)

Submits an article for translation to the specified language. Also assigns the specified user and due date to the submittal and returns new ID of the draft translation.

archiveOnlineArticle(articleId, scheduledDate)

Archives an online version of an article. If the specified scheduledDate is null, the article is archived immediately. Otherwise, it archives the article on the scheduled date.

Signature

```
public static Void archiveOnlineArticle(String articleId, Datetime scheduledDate)
```

Parameters

articleId
Type: String

scheduledDate
Type: Datetime

Return Value

Type: Void

Example

```
String articleId = 'Insert article ID';
Datetime scheduledDate = Datetime.newInstanceGmt(2012, 12,1,13,30,0);
KbManagement.PublishingService.archiveOnlineArticle(articleId, scheduledDate);
```

assignDraftArticleTask(articleId, assigneeId, instructions, dueDate, sendEmailNotification)

Assigns a review task related to a draft article.

Signature

```
public static Void assignDraftArticleTask(String articleId, String assigneeId, String
instructions, Datetime dueDate, Boolean sendEmailNotification)
```

Parameters

articleId
Type: String

assigneeId
Type: String

instructions
Type: String

dueDate
Type: Datetime

sendEmailNotification
Type: Boolean

Return Value

Type: Void

Example

```
String articleId = 'Insert article ID';
String assigneeId = '';
String instructions = 'Please review this draft.';
Datetime dueDate = Datetime.newInstanceGmt(2012, 12, 1);
KbManagement.PublishingService.assignDraftArticleTask(articleId, assigneeId, instructions,
 dueDate, true);
```

assignDraftTranslationTask(articleVersionId, assigneeId, instructions, dueDate, sendEmailNotification)

Assigns a review task related to a draft translation.

Signature

```
public static Void assignDraftTranslationTask(String articleVersionId, String assigneeId,
String instructions, Datetime dueDate, Boolean sendEmailNotification)
```

Parameters

articleVersionId
Type: String

assigneeId
Type: String

instructions
Type: String

dueDate
Type: Datetime

sendEmailNotification
Type: Boolean

Return Value

Type: Void

Example

```
String articleId = 'Insert article ID';
String assigneeId = 'Insert assignee ID';
String instructions = 'Please review this draft.';
Datetime dueDate = Datetime.newInstanceGmt(2012, 12, 1);
KbManagement.PublishingService.assignDraftTranslationTask(articleId, assigneeId,
instructions, dueDate, true);
```

cancelScheduledArchivingOfArticle(articleId)

Cancels the scheduled archiving of an online article.

Signature

```
public static Void cancelScheduledArchivingOfArticle(String articleId)
```

Parameters

articleId
Type: String

Return Value

Type: Void

Example

```
String articleId = 'Insert article ID';
KbManagement.PublishingService.cancelScheduledArchivingOfArticle (articleId);
```

cancelScheduledPublicationOfArticle(articleId)

Cancels the scheduled publication of a draft article.

Signature

```
public static Void cancelScheduledPublicationOfArticle(String articleId)
```

Parameters

articleId
Type: String

Return Value

Type: Void

Example

```
String articleId = 'Insert article ID';
KbManagement.PublishingService.cancelScheduledPublicationOfArticle (articleId);
```

completeTranslation(articleVersionId)

Puts a translation in a completed state that is ready to publish.

Signature

```
public static Void completeTranslation(String articleVersionId)
```

Parameters

articleVersionId
Type: String

Return Value

Type: Void

Example

```
String articleVersionId = 'Insert article ID';
KbManagement.PublishingService.completeTranslation(articleVersionId);
```

deleteArchivedArticle(articleId)

Deletes an archived article.

Signature

```
public static Void deleteArchivedArticle(String articleId)
```

Parameters

articleId
Type: String

Return Value

Type: Void

Example

```
String articleId = 'Insert article ID';
KbManagement.PublishingService.deleteArchivedArticle(articleId);
```

deleteArchivedArticleVersion(articleId, versionNumber)

Deletes a specific archived version of a published article.

Signature

```
public static Void deleteArchivedArticleVersion(String articleId, Integer versionNumber)
```

Parameters

articleId
Type: String

versionNumber
Type: Integer

Return Value

Type: Void

Example

```
String articleId = 'Insert article ID';
Integer versionNumber = 1;
KbManagement.PublishingService.deleteArchivedArticleVersion(articleId, versionNumber);
```

deleteDraftArticle(articleId)

Deletes a draft article.

Signature

```
public static Void deleteDraftArticle(String articleId)
```

Parameters

articleId
Type: String

Return Value

Type: Void

Example

```
String articleId = 'Insert article ID';
KbManagement.PublishingService.deleteDraftArticle(articleId);
```

deleteDraftTranslation(articleVersionId)

Deletes a draft translation.

Signature

```
public static Void deleteDraftTranslation(String articleVersionId)
```

Parameters

articleVersionId
Type: String

Return Value

Type: Void

Example

```
String articleVersionId = 'Insert article ID';
KbManagement.PublishingService.deleteDraftTranslation (articleVersionId);
```

editArchivedArticle(articleId)

Creates a draft article from the archived primary version and returns the new draft primary version ID of the article.

Signature

```
public static String editArchivedArticle(String articleId)
```

Parameters

articleId
Type: String

Return Value

Type: String

Example

```
String articleId = 'Insert article ID';
String id = KbManagement.PublishingService.editArchivedArticle(articleId);
```

editOnlineArticle(articleId, unpublish)

Creates a draft article from the online version and returns the new draft primary version ID of the article. Also, unpublishes the online article, if *unpublish* is set to `true`.

Signature

```
public static String editOnlineArticle(String articleId, Boolean unpublish)
```

Parameters

articleId
Type: String

unpublish
Type: Boolean

Return Value

Type: String

Example

```
String articleId = 'Insert article ID';
String id = KbManagement.PublishingService.editOnlineArticle (articleId, true);
```

editPublishedTranslation(articleId, language, unpublish)

Creates a draft version of the online translation for a specific language and returns the new draft primary version ID of the article. Also, unpublishes the article, if set to `true`.

Signature

```
public static String editPublishedTranslation(String articleId, String language, Boolean
unpublish)
```

Parameters

articleId
Type: String

language
Type: String

unpublish
Type: Boolean

Return Value

Type: String

Example

```
String articleId = 'Insert article ID';
String language = 'fr';
String id = KbManagement.PublishingService.editPublishedTranslation(articleId, language,
true);
```

publishArticle(articleId, flagAsNew)

Publishes an article. If *flagAsNew* is set to `true`, the article is published as a major version.

Signature

```
public static Void publishArticle(String articleId, Boolean flagAsNew)
```

Parameters

articleId
Type: String

flagAsNew
Type: Boolean

Return Value

Type: Void

Example

```
String articleId = 'Insert article ID';
KbManagement.PublishingService.publishArticle(articleId, true);
```

restoreOldVersion(articleId, versionNumber)

Creates a draft article from an existing online article based on the specified archived version of the article and returns the article version ID.

Signature

```
public static String restoreOldVersion(String articleId, Integer versionNumber)
```

Parameters

articleId
Type: String

versionNumber
Type: Integer

Return Value

Type: String

Example

```
String articleId = 'Insert article ID';
String id = KbManagement.PublishingService.restoreOldVersion (articleId, 1);
```

scheduleForPublication(articleId, scheduledDate)

Schedules the article for publication as a major version. If the specified date is null, the article is published immediately.

Signature

```
public static Void scheduleForPublication(String articleId, Datetime scheduledDate)
```

Parameters

articleId
Type: String

scheduledDate
Type: Datetime

Return Value

Type: Void

Example

```
String articleId = 'Insert article ID';
Datetime scheduledDate = Datetime.newInstanceGmt(2012, 12,1,13,30,0);
KbManagement.PublishingService.scheduleForPublication(articleId, scheduledDate);
```

setTranslationToIncomplete(articleVersionId)

Sets a draft translation that is ready for publication back to “in progress” status.

Signature

```
public static Void setTranslationToIncomplete(String articleVersionId)
```

Parameters

articleVersionId
Type: String

Return Value

Type: Void

Example

```
String articleVersionId = 'Insert article ID';
KbManagement.PublishingService.setTranslationToIncomplete(articleVersionId);
```

submitForTranslation(articleId, language, assigneeId, dueDate)

Submits an article for translation to the specified language. Also assigns the specified user and due date to the submittal and returns new ID of the draft translation.

Signature

```
public static String submitForTranslation(String articleId, String language, String
assigneeId, Datetime dueDate)
```

Parameters

articleId
Type: String

language
Type: String

assigneeId
Type: String

dueDate
Type: Datetime

Return Value

Type: String

Example

```
String articleId = 'Insert article ID';
String language = 'fr';
String assigneeId = 'Insert assignee ID';
Datetime dueDate = Datetime.newInstanceGmt(2012, 12,1);
```

```
String id = KbManagement.PublishingService.submitForTranslation(articleId, language,
assigneeId, dueDate);
```

LxScheduler Namespace

The `LxScheduler` namespace provides an interface and classes for integrating Salesforce Scheduler with external calendars.

The following are the classes and the interface in the `LxScheduler` namespace.

IN THIS SECTION:

GetAppointmentCandidatesInput Class
Contains information about the available service resources (appointment candidates) based on work type group and service territories.

GetAppointmentCandidatesInputBuilder Class
Contains methods to build an instance of the `lxscheduler.GetAppointmentCandidatesInput` class.

GetAppointmentSlotsInput Class
Contains information about the available appointment time slots for a resource based on given work type group and territories.

GetAppointmentSlotsInputBuilder Class
Contains methods to build an instance of the `lxscheduler.GetAppointmentSlotsInput` class.

SchedulerResources Class
Contains methods that holds the business logic to get resources availability.

SkillRequirement Class
Contains information about the set of skills that are required to complete a particular task for a work type.

SkillRequirementBuilder Class
Contains methods to build an instance of the `lxscheduler.SkillRequirement` class.

WorkType Class
Contains information about the type of work to be performed.

WorkTypeBuilder Class
Contains methods to build an instance of the `lxscheduler.WorkType` class.

ServiceResourceScheduleHandler Interface
Allows an implementing class to check external calendar events to find already booked time slots for the requested service resources. This interface is part of Salesforce Scheduler.

ServiceAppointmentRequestInfo Class
Represents the list of parameters that are passed to the ServiceResourceScheduleHandler interface. This class is implemented internally by Apex.

ServiceResourceInfo Class
Contains information about a service resource.

ServiceResourceSchedule Class
Use this class to pass results from your implemented Apex class to the ServiceResourceScheduleHandler interface methods.

UnavailableTimeslot Class

Use this class to pass the unavailable time slots to the lxscheduler.ServiceResourceSchedule class. Timezones that differ across operating hours are handled and results are always returned in UTC.

SEE ALSO:

Apex Interface Implementation Limitations and Error Codes

GetAppointmentCandidatesInput Class

Contains information about the available service resources (appointment candidates) based on work type group and service territories.

Set up Salesforce Scheduler before making requests. This setup includes creating or configuring Service Resources, Service Territory Members, Work Type Groups, Work Types, Work Type Group Members, and Service Territory Work Types. See Set Up Salesforce Scheduler for more information.

The appointment time slots are determined based on multiple factors, such as field values, scheduled appointments, absences, Scheduler Settings, and Scheduling Policies to determine available time slots. See How Salesforce Scheduler Determines Available Time Slots for more information.

The following factors are considered for returning start time and end time of resources.

Resource Availability
Determined using service territory member, service territory, work type, and account operating hours fields.

Resource Unavailability
Determined by resource absences, existing appointments that the resource is assigned to. The resource must be marked as a required resource for the appointment with a status that isn't in closed, canceled, or completed.

Appointment Start Time Interval in the Scheduling Policy
Appointment start time interval field in the Scheduling Policy is used to determine when the appointment can start. This interval can be 5, 10, 15, 20, 30, or 60. By default, it's set to 15.

Work Type Duration
The end time is calculated as start time + duration of the work type.

Note: If asset scheduling is enabled, the response also includes asset-based candidates.

Namespace

LxScheduler

Usage

The constructor for this class can't be called directly. Create an instance of this class using the GetAppointmentCandidatesInputBuilder.build() method.

This example shows how to get a list of available appointment candidates based on `workTypeGroupId`:

```
//Build input for GetAppointmentCandidates API
   lxscheduler.GetAppointmentCandidatesInput input = new
lxscheduler.GetAppointmentCandidatesInputBuilder()
     .setWorkTypeGroupId('0VSRM0000000ABc4AM')
     .setTerritoryIds(new List<String>{'0HhRM0000000FXd0AM'})
     .setStartTime(System.now().format('yyyy-MM-dd\'T\'HH:mm:ssZ','America/New_York'))
```

```
.setEndTime(System.now().addDays(5).format('yyyy-MM-dd\'T\'HH:mm:ssZ','America/New_York'))

     .setAccountId('001RM0000053iQgYAI')
     .setSchedulingPolicyId('0VrRM00000000Bx')
     .setApiVersion(Double.valueOf('50.0'))
     .build();

   String response = lxscheduler.SchedulerResources.getAppointmentCandidates(input);
```

This example shows how to get a list of available appointment candidates based on `workType`:

```
//Build WorkType
   lxscheduler.WorkType workType = new lxscheduler.WorkTypeBuilder()
     .setId('08qRM0000000G9RYAU')
     .build();

   lxscheduler.GetAppointmentCandidatesInput input = new
lxscheduler.GetAppointmentCandidatesInputBuilder()
     .setWorkType(workType)
     .setTerritoryIds(new List<String>{'0HhRM0000000FXd0AM'})
     .setStartTime(System.now().format('yyyy-MM-dd\'T\'HH:mm:ssZ','America/New_York'))

.setEndTime(System.now().addDays(5).format('yyyy-MM-dd\'T\'HH:mm:ssZ','America/New_York'))

     .setAccountId('001RM0000053iQgYAI')
     .setSchedulingPolicyId('0VrRM00000000Bx')
     .setApiVersion(Double.valueOf('50.0'))
     .build();

   String response = lxscheduler.SchedulerResources.getAppointmentCandidates(input);
```

This example shows how to get a list of available candidate appointments based on `durationInMinutes` and without the `workTypeGroupId` or `workType` fields:

Important: If you're using shifts, you must specify the `workTypeGroupId` or `workType` field.

```
//Build SkillRequirement
   lxscheduler.SkillRequirement skillReq = new lxscheduler.SkillRequirementBuilder()
     .setSkillId('0C5RM0000004EZS0A2')
     .setSkillLevel(90)
     .build();

//Build WorkType
   lxscheduler.WorkType workType = new lxscheduler.WorkTypeBuilder()
     .setDurationInMinutes(15)
     .setBlockTimeBeforeAppointmentInMinutes(5)
     .setBlockTimeAfterAppointmentInMinutes(5)
     .setTimeFrameStartInMinutes(10080)
     .setTimeFrameEndInMinutes(40320)
     .setOperatingHoursId('0OHRM0000000FmG4AU')
     .setSkillRequirements(new List<lxscheduler.SkillRequirement>{skillReq})
     .build();

   lxscheduler.GetAppointmentCandidatesInput input = new
```

```
lxscheduler.GetAppointmentCandidatesInputBuilder()
     .setWorkType(workType)
     .setTerritoryIds(new List<String>{'0HhRM0000000FXd0AM'})
     .setSchedulingPolicyId('0VrRM00000000Bx')
     .setApiVersion(Double.valueOf('50.0'))
     .build();

   String response = lxscheduler.SchedulerResources.getAppointmentCandidates(input);
```

This example shows a sample response of a list of available candidates:

```
[
   {
      "startTime": "2021-02-16T16:15:00.000+0000",
      "endTime": "2021-02-16T16:16:00.000+0000",
      "resources": [
         "0Hnxx0000004C9BCAU"
      ],
      "territoryId": "0Hhxx0000004C92CAE"
   },
   {
      "startTime": "2021-02-16T16:30:00.000+0000",
      "endTime": "2021-02-16T16:31:00.000+0000",
      "resources": [
        "0Hnxx0000004C9BCAU"
      ],
      "territoryId": "0Hhxx0000004C92CAE"
   },
]
```

GetAppointmentCandidatesInputBuilder Class

Contains methods to build an instance of the `lxscheduler.GetAppointmentCandidatesInput` class.

A Builder object is obtained by invoking one of the `GetAppointmentCandidatesInputBuilder` methods defined by the `GetAppointmentCandidatesInput` class.

Namespace

LxScheduler

IN THIS SECTION:

GetAppointmentCandidatesInputBuilder Methods

GetAppointmentCandidatesInputBuilder Methods

The following are methods for `GetAppointmentCandidatesInputBuilder`.

IN THIS SECTION:

build()
Returns an instance of the `lxscheduler.GetAppointmentCandidatesInput` object.

setAccountId(accountId)
Sets the ID of the associated account for which you want to create the appointments.

setAllowConcurrent(allowConcurrent)
Allows the scheduling of concurrent appointments.

setApiVersion(apiVersion)
Sets the API version of the business logic for the `getAppointmentCandidates` method.

setCorrelationId(correlationId)
Sets the correlation ID.

setEndTime(endTime)
Sets the scheduling end time.

setEngagementChannelTypeIds(engagementChannelTypeIds)
Sets an engagement channel type.

setFilterByResources(filterByResources)
Enables filtering resources using a comma-separated list of service resource IDs.

setResourceLimitApptDistribution(resourceLimitApptDistribution)
Sets the number of service resources to show during appointment scheduling.

setSchedulingPolicyId(schedulingPolicyId)
Sets the ID of the AppointmentSchedulingPolicy object.

setStartTime(startTime)
Sets the scheduling start time to the specified time.

setTerritoryIds(territoryIds)
Sets the service territory IDs.

setWorkType(workType)
Sets the type of work to be performed.

setWorkTypeGroupId(workTypeGroupId)
Sets the ID of the work type group.

`build()`

Returns an instance of the `lxscheduler.GetAppointmentCandidatesInput` object.

Signature

```
public lxscheduler.GetAppointmentCandidatesInput build()
```

Return Value

Type: lxscheduler.GetAppointmentCandidatesInput

`setAccountId(accountId)`

Sets the ID of the associated account for which you want to create the appointments.

Signature

```
public lxscheduler.GetAppointmentCandidatesInputBuilder setAccountId(String accountId)
```

Parameters

accountId
Type: String

Return Value

Type: LxScheduler.GetAppointmentCandidatesInputBuilder

`setAllowConcurrent(allowConcurrent)`

Allows the scheduling of concurrent appointments.

Signature

```
public lxscheduler.GetAppointmentCandidatesInputBuilder setAllowConcurrent(Boolean
allowConcurrent)
```

Parameters

allowConcurrent
Type: Boolean

If true, allows scheduling of concurrent appointments in a time slot. The default is false.

Available in API version 47.0 and later.

Return Value

Type: LxScheduler.GetAppointmentCandidatesInputBuilder

`setApiVersion(apiVersion)`

Sets the API version of the business logic for the `getAppointmentCandidates` method.

Signature

```
public lxscheduler.GetAppointmentCandidatesInputBuilder setApiVersion(Double apiVersion)
```

Parameters

apiVersion
Type: Double

Usage

The specified parameter must use the correct API version. For example, if API version is set to 45.0 and `filterByResources` is set (which is available in API version 51.0 and later), then this field is ignored. If no API version or incorrect API version is passed in the request body, by default the latest version is used.

Note: The API is available since version 45.0.

Return Value

Type: LxScheduler.GetAppointmentCandidatesInputBuilder

`setCorrelationId(correlationId)`

Sets the correlation ID.

Signature

```
public lxscheduler.GetAppointmentCandidatesInputBuilder setCorrelationId(String
correlationId)
```

Parameters

`correlationId`
Type: String

ID to pass custom information to the `ServiceResourceScheduleHandler` Apex interface. For example, you can use the correlation ID to identify the app, website, or any other external system that calls this Apex interface implementation. If you don't pass a custom value, a randomly generated identifier is passed. Available in API version 53.0 and later.

Return Value

Type: LxScheduler.GetAppointmentCandidatesInputBuilder

`setEndTime(endTime)`

Sets the scheduling end time.

Signature

```
public lxscheduler.GetAppointmentCandidatesInputBuilder setEndTime(String endTime)
```

Parameters

`endTime`
Type: String

The latest time that a time slot can end (inclusive).

Note: If end time is not specified, it defaults to 31 days.

Usage

The specified string should use the standard date format "['yyyy-MM-dd\'T\'HH:mm:ssZ']" in the local time zone. Defaults to the user's time zone.

Return Value

Type: LxScheduler.GetAppointmentCandidatesInputBuilder

`setEngagementChannelTypeIds(engagementChannelTypeIds)`

Sets an engagement channel type.

Signature

```
public lxscheduler.GetAppointmentCandidatesInputBuilder
setEngagementChannelTypeIds(List<String> engagementChannelTypeIds)
```

Parameters

`engagementChannelTypeIds`
Type: List<String>

The ID of the engagement channel type record. The availability of service resources is filtered based on the engagement channel type selected. This field is available in API version 56.0 and later.

Note: This field supports only one engagement channel type ID.

Return Value

Type: LxScheduler.GetAppointmentCandidatesInputBuilder

Usage

You can use engagement channel types only in these cases:

- The **Schedule Appointments Using Engagement Channels** setting is enabled in Salesforce Scheduler Settings in your Salesforce org.
- Shifts are defined in the scheduling policy. For more information on setting up shifts in scheduling policy, see Define Shift Rules in Scheduling Policy.

  Note: Engagement channel types are not supported with operating-hours rules in the scheduling policy.

`setFilterByResources(filterByResources)`

Enables filtering resources using a comma-separated list of service resource IDs.

Signature

```
public lxscheduler.GetAppointmentCandidatesInputBuilder setFilterByResources(List<String>
filterByResources)
```

Parameters

filterByResources
Type: List<String>

Gets only eligible resources that are both in the list and in the selected service territory sorted by the order in which the resource IDs are passed. This field is available in API version 51.0 and later.

Return Value

Type: LxScheduler.GetAppointmentCandidatesInputBuilder

setResourceLimitApptDistribution(resourceLimitApptDistribution)

Sets the number of service resources to show during appointment scheduling.

Signature

```
public lxscheduler.GetAppointmentCandidatesInputBuilder
setResourceLimitApptDistribution(Integer resourceLimitApptDistribution)
```

Parameters

resourceLimitApptDistribution
Type: Integer

Specify the maximum number of service resources that you want to show during appointment scheduling when appointment distribution is enabled. Available in API version 53.0 and later.

Return Value

Type: LxScheduler.GetAppointmentCandidatesInputBuilder

setSchedulingPolicyId(schedulingPolicyId)

Sets the ID of the AppointmentSchedulingPolicy object.

Signature

```
public lxscheduler.GetAppointmentCandidatesInputBuilder setSchedulingPolicyId(String
schedulingPolicyId)
```

Parameters

schedulingPolicyId
Type: String

The ID of the `AppointmentSchedulingPolicy` object. If no scheduling policy is passed in the request body, the default configurations are used.

Return Value

Type: LxScheduler.GetAppointmentCandidatesInputBuilder

`setStartTime(startTime)`

Sets the scheduling start time to the specified time.

Signature

```
public lxscheduler.GetAppointmentCandidatesInputBuilder setStartTime(String startTime)
```

Parameters

`startTime`
Type: String

The earliest time that a time slot can begin (inclusive). You can also use a time from the past.

Usage

The specified string should use the standard date format "['yyyy-MM-dd\'T\'HH:mm:ssZ']" in the local time zone. Defaults to the user's time zone.

Return Value

Type: LxScheduler.GetAppointmentCandidatesInputBuilder

`setTerritoryIds(territoryIds)`

Sets the service territory IDs.

Signature

```
public lxscheduler.GetAppointmentCandidatesInputBuilder setTerritoryIds(List<String>
territoryIds)
```

Parameters

`territoryIds`
Type: List<String>

List of service territory IDs, where the work that is being requested is performed. This is a required field.

Return Value

Type: LxScheduler.GetAppointmentCandidatesInputBuilder

`setWorkType(workType)`

Sets the type of work to be performed.

Signature

```
public lxscheduler.GetAppointmentCandidatesInputBuilder setWorkType(lxscheduler.WorkType
workType)
```

Parameters

workType
Type: lxscheduler.WorkType

This method takes input as an instance of the `lxscheduler.WorkType` class. Build the instance of the input class using the `lxscheduler.WorkTypeBuilder` class.

Required if *workTypeGroupId* is not given. If id of the *workType* is given, the rest of *workType* fields are optional.

Usage

Return Value

Type: LxScheduler.GetAppointmentCandidatesInputBuilder

`setWorkTypeGroupId(workTypeGroupId)`

Sets the ID of the work type group.

Signature

```
public lxscheduler.GetAppointmentCandidatesInputBuilder setWorkTypeGroupId(String
workTypeGroupId)
```

Parameters

workTypeGroupId
Type: String

The ID of the work type group containing the work types that are being performed. Required if *workType* is not given. If *workType* is given, then you must provide either *id* or *durationInMinutes*, but not both.

Return Value

Type: LxScheduler.GetAppointmentCandidatesInputBuilder

GetAppointmentSlotsInput Class

Contains information about the available appointment time slots for a resource based on given work type group and territories.

The appointment time slots are determined based on your Salesforce Scheduler data model configurations. Here are some prerequisites that you can consider while setting up data.

- Set up Salesforce Scheduler before making your requests. The setup includes creating or configuring Service Resources, Service Territory Members, Work Type Groups, Work Types, Work Type Group Members, and Service Territory Work Types. See Manage Business Information in Salesforce Scheduler for more information.
- Configure a work type mapped for each territory in the request body via Service Territory Work Type. Map the same work type to the work type group, via work type group member.

The following factors affect how time slots are calculated and returned.

- Timezones that differ across operating hours are handled and results are always returned in UTC.
- The resource must be marked as a required resource on the assigned resource object.

- The resource is considered unavailable If the status categories of the resource assigned to service appointments are other than `Canceled`, `Cannot Complete`, and `Completed`.
- Resource Absences of all types are considered unavailable from start to end.
- The following fields of Work Type records, if configured, are used to fine-tune time slot requirements. For more information, see Create Work Types in Salesforce Scheduler.

Parameter	Description
`Timeframe Start`	Time slots sooner than `current time + Timeframe Start` aren't returned.
`Timeframe End`	Time slots later than `current time + Timeframe End` aren't returned.
`Block Time Before Appointment`	The time period before the appointment is considered as unavailable.
`Block Time After Appointment`	The time period after the appointment is considered as unavailable.
`Operating Hours`	The overlap of all operating hours from the account, work type, service territory, and service territory member are considered while determining time slots. For more information, see Set Up Operating Hours in Salesforce Scheduler.

- Only the time slots within the period of 31 days from the start date are returned.
- Salesforce Scheduler uses multiple factors, such as field values, scheduled appointments, absences, Scheduler Settings, and Scheduling Policies to determine available time slots, including the earliest and latest appointment slots. See How Does Salesforce Scheduler Determine Available Time Slots.

> Note: If asset scheduling is enabled, you can provide an asset-based service resource in `requiredResourceIds` to retrieve available timeslots for the asset resource.

Namespace

LxScheduler

Usage

The constructor for this class can't be called directly. Create an instance of this class using the GetAppointmentSlotsInputBuilder.build() method.

This example shows how to get a list of available time slots based on `workTypeGroupId`:

```
//Build input for GetAppointmentSlots API
   lxscheduler.GetAppointmentSlotsInput input = new
lxscheduler.GetAppointmentSlotsInputBuilder()
      .setWorkTypeGroupId('0VSxx0000004C92GAE')
      .setTerritoryIds(new List<String>{'0Hhxx0000004C92CAE'})
      .setStartTime(System.now().format('yyyy-MM-dd\'T\'HH:mm:ssZ'))
      .setEndTime(System.now().addDays(1).format('yyyy-MM-dd\'T\'HH:mm:ssZ'))
      .setAccountId('001xx000003GYK0AAO')
      .setRequiredResourceIds(new List<String>{'0Hnxx0000004C92CAE'})
      .setSchedulingPolicyId('0Vrxx0000004CAe')
      .setApiVersion(Double.valueOf('48.0'))
```

```
      .build();

String response = lxscheduler.SchedulerResources.getAppointmentSlots(input);
```

This example shows how to get a list of available time slots based on `workType`:

```
//Build WorkType
   lxscheduler.WorkType workType = new lxscheduler.WorkTypeBuilder()
      .setId('08qxx0000004C92AAE')
      .build();

   lxscheduler.GetAppointmentSlotsInput input = new
lxscheduler.GetAppointmentSlotsInputBuilder()
      .setWorkType(workType)
      .setTerritoryIds(new List<String>{'0Hhxx0000004C92CAE'})
      .setStartTime(System.now().format('yyyy-MM-dd\'T\'HH:mm:ssZ'))
      .setEndTime(System.now().addDays(1).format('yyyy-MM-dd\'T\'HH:mm:ssZ'))
      .setAccountId('001xx000003GYK0AAO')
      .setRequiredResourceIds(new List<String>{'0Hnxx0000004C92CAE'})
      .setSchedulingPolicyId('0Vrxx0000004CAe')
      .setApiVersion(Double.valueOf('48.0'))
      .build();

String response = lxscheduler.SchedulerResources.getAppointmentSlots(input);
```

This example shows how to get a list of available time slots based on `durationInMinutes` and without `workTypeGroupId` or `workType` fields:

```
//Build WorkType
   lxscheduler.WorkType workType = new lxscheduler.WorkTypeBuilder()
      .setDurationInMinutes(60)
      .build();

   lxscheduler.GetAppointmentSlotsInput input = new
lxscheduler.GetAppointmentSlotsInputBuilder()
      .setWorkType(workType)
      .setTerritoryIds(new List<String>{'0Hhxx0000004C92CAE'})
      .setRequiredResourceIds(new List<String>{'0Hnxx0000004C92CAE'})
      .setApiVersion(Double.valueOf('48.0'))
      .build();

   String response = lxscheduler.SchedulerResources.getAppointmentSlots(input);
```

This example shows a sample response of a list of available time slots:

```
[
   {
     "territoryId": "0Hhxx0000004C92CAE",
     "startTime": "2021-02-10T16:00:00.000+0000",
     "endTime": "2021-02-10T16:15:00.000+0000",
     "remainingAppointments": 1
   },
   {
     "territoryId": "0Hhxx0000004C92CAE",
     "startTime": "2021-02-10T16:15:00.000+0000",
     "endTime": "2021-02-10T16:30:00.000+0000",
```

```
      "remainingAppointments": 1
    },
]
```

GetAppointmentSlotsInputBuilder Class

Contains methods to build an instance of the `lxscheduler.GetAppointmentSlotsInput` class.

A Builder object is obtained by invoking one of the `GetAppointmentSlotsInputBuilder` methods defined by the `GetAppointmentSlotsInput` class.

Namespace

LxScheduler

IN THIS SECTION:

GetAppointmentSlotsInputBuilder Methods

GetAppointmentSlotsInputBuilder Methods

The following are methods for `GetAppointmentSlotsInputBuilder`.

IN THIS SECTION:

build()
Returns an instance of the `lxscheduler.GetAppointmentSlotsInput` object.

setAccountId(accountId)
Sets the ID of the associated account for which you want to create appointments.

setAllowConcurrentScheduling(allowConcurrentScheduling)
Allows the scheduling of concurrent appointments.

setApiVersion(apiVersion)
Sets the API version of the business logic for the `getAppointmentSlots` method.

setCorrelationId(correlationId)
Sets the correlation ID.

setEndTime(endTime)
Sets the scheduling end time.

setEngagementChannelTypeIds(engagementChannelTypeIds)
Sets an engagement channel type.

setPrimaryResourceId(primaryResourceId)
Sets the ID of the primary resource.

setRequiredResourceIds(requiredResourceIds)
Sets the resource IDs.

setSchedulingPolicyId(schedulingPolicyId)
Sets the ID of the `AppointmentSchedulingPolicy` object.

setStartTime(startTime)
Sets the scheduling start time.

setTerritoryIds(territoryIds)
Sets the IDs of service territories.

setWorkType(workType)
Sets the type of work to be performed.

setWorkTypeGroupId(workTypeGroupId)
Sets the ID of the work type group.

build()

Returns an instance of the `lxscheduler.GetAppointmentSlotsInput` object.

Signature

```
public lxscheduler.GetAppointmentSlotsInput build()
```

Return Value

Type: lxscheduler.GetAppointmentSlotsInput

setAccountId(accountId)

Sets the ID of the associated account for which you want to create appointments.

Signature

```
public lxscheduler.GetAppointmentSlotsInputBuilder setAccountId(String accountId)
```

Parameters

accountId
Type: String

The ID of the associated account.

Return Value

Type: lxscheduler.GetAppointmentSlotsInputBuilder

setAllowConcurrentScheduling(allowConcurrentScheduling)

Allows the scheduling of concurrent appointments.

Signature

```
public lxscheduler.GetAppointmentSlotsInputBuilder setAllowConcurrentScheduling(Boolean
allowConcurrentScheduling)
```

Parameters

allowConcurrentScheduling
Type: Boolean

If true, allows scheduling of concurrent appointments in a time slot. If false, concurrent appointments are not allowed. The default is false. Available in API version 47.0 and later.

Return Value

Type: lxscheduler.GetAppointmentSlotsInputBuilder

setApiVersion(apiVersion)

Sets the API version of the business logic for the `getAppointmentSlots` method.

Signature

```
public lxscheduler.GetAppointmentSlotsInputBuilder setApiVersion(Double apiVersion)
```

Parameters

apiVersion
Type: Double

Usage

The specified parameter must use the correct API version. For example, if API version is set to 45.0 and *primaryResourceId* is set (which is available in API version 48.0 and later), then this field is ignored. If no API version or incorrect API version is passed in the request body, by default the latest version is used.

Note: The API is available since version 45.0.

Return Value

Type: lxscheduler.GetAppointmentSlotsInputBuilder

setCorrelationId(correlationId)

Sets the correlation ID.

Signature

```
public lxscheduler.GetAppointmentSlotsInputBuilder setCorrelationId(String correlationId)
```

Parameters

correlationId
Type: String

ID to pass custom information to the `ServiceResourceScheduleHandler` Apex interface. For example, you can use the correlation ID to identify the app, website, or any other external system that calls this Apex interface implementation. If you don't pass a custom value, a randomly generated identifier is passed. Available in API version 53.0 and later.

Return Value

Type: lxscheduler.GetAppointmentSlotsInputBuilder

`setEndTime(endTime)`

Sets the scheduling end time.

Signature

```
public lxscheduler.GetAppointmentSlotsInputBuilder setEndTime(String endTime)
```

Parameters

`endTime`
Type: String

The latest time that a time slot can end (inclusive). If end time is not specified, it defaults to 31 days.

Usage

The specified string should use the standard date format "['yyyy-MM-dd\'T\'HH:mm:ssZ']" in the local time zone. Defaults to the user's time zone.

Return Value

Type: lxscheduler.GetAppointmentSlotsInputBuilder

`setEngagementChannelTypeIds(engagementChannelTypeIds)`

Sets an engagement channel type.

Signature

```
public lxscheduler.GetAppointmentSlotsInputBuilder
setEngagementChannelTypeIds(List<String> engagementChannelTypeIds)
```

Parameters

`engagementChannelTypeIds`
Type: List<String>

The ID of the engagement channel type record. The availability of time slots is filtered based on the engagement channel type selected. This field is available in API version 56.0 and later.

Note: This field supports only one engagement channel type ID.

Return Value

Type: lxscheduler.GetAppointmentSlotsInputBuilder

Usage

You can use engagement channel types only in these cases:

- The **Schedule Appointments Using Engagement Channels** setting is enabled in Salesforce Scheduler Settings in your Salesforce org.
- Shifts are defined in the scheduling policy. For more information on setting up shifts in scheduling policy, see Define Shift Rules in Scheduling Policy.

  Note: Engagement channel types are not supported with operating-hours rules in the scheduling policy.

`setPrimaryResourceId(primaryResourceId)`

Sets the ID of the primary resource.

Signature

```
public lxscheduler.GetAppointmentSlotsInputBuilder setPrimaryResourceId(String
primaryResourceId)
```

Parameters

`primaryResourceId`
Type: String

The ID of the primary resource in multi-resource scheduling. Required only when multi-resource scheduling is enabled. Available in API version 48.0 and later.

Return Value

Type: lxscheduler.GetAppointmentSlotsInputBuilder

`setRequiredResourceIds(requiredResourceIds)`

Sets the resource IDs.

Signature

```
public lxscheduler.GetAppointmentSlotsInputBuilder setRequiredResourceIds(List<String>
requiredResourceIds)
```

Parameters

`requiredResourceIds`
Type: List<String>

List of resource IDs that must be available during the time slot. This is a required field.

Return Value

Type: lxscheduler.GetAppointmentSlotsInputBuilder

`setSchedulingPolicyId(schedulingPolicyId)`

Sets the ID of the `AppointmentSchedulingPolicy` object.

Signature

```
public lxscheduler.GetAppointmentSlotsInputBuilder setSchedulingPolicyId(String schedulingPolicyId)
```

Parameters

`schedulingPolicyId`
Type: String

If no scheduling policy is passed in the request body, the default configurations are used.

Return Value

Type: lxscheduler.GetAppointmentSlotsInputBuilder

`setStartTime(startTime)`

Sets the scheduling start time.

Signature

```
public lxscheduler.GetAppointmentSlotsInputBuilder setStartTime(String startTime)
```

Parameters

`startTime`
Type: String

The earliest time that a time slot can begin (inclusive). Defaults to the current time of the request, if empty.

Usage

The specified string should use the standard date format "['yyyy-MM-dd\'T\'HH:mm:ssZ']" in the local time zone. Defaults to the user's time zone.

Return Value

Type: lxscheduler.GetAppointmentSlotsInputBuilder

`setTerritoryIds(territoryIds)`

Sets the IDs of service territories.

Signature

```
public lxscheduler.GetAppointmentSlotsInputBuilder setTerritoryIds(List<String>
territoryIds)
```

Parameters

territoryIds
Type: List<String>

List of IDs of service territories, where the work that is being requested is performed. This is a required field.

Return Value

Type: lxscheduler.GetAppointmentSlotsInputBuilder

setWorkType(workType)

Sets the type of work to be performed.

Signature

```
public lxscheduler.GetAppointmentSlotsInputBuilder setWorkType(lxscheduler.WorkType
workType)
```

Parameters

workType
Type: lxscheduler.WorkType

This method takes input as an instance of the `lxscheduler.WorkType` class. Build the instance of the input class using the `lxscheduler.WorkTypeBuilder` class.

Required if *workTypeGroupId* is not given.

Return Value

Type: lxscheduler.GetAppointmentSlotsInputBuilder

setWorkTypeGroupId(workTypeGroupId)

Sets the ID of the work type group.

Signature

```
public lxscheduler.GetAppointmentSlotsInputBuilder setWorkTypeGroupId(String
workTypeGroupId)
```

Parameters

workTypeGroupId
Type: String

The ID of the work type group containing the work types that are being performed.

Return Value

Type: lxscheduler.GetAppointmentSlotsInputBuilder

SchedulerResources Class

Contains methods that holds the business logic to get resources availability.

Namespace

LxScheduler

Implementation Considerations

Apex implementation of the methods in the `SchedulerResources` class should adhere to Apex Governor Limits. It includes synchronous heap size limit, synchronous CPU time limit, and synchronous concurrent transactions for long running transactions. To avoid governor limits, you must tune the input by reducing the time frame, limiting number of service resources, or limiting number or territories at a time. This will reduce the overall transaction time and response size of the implementation. For more information on standard Apex Governer Limits, see Salesforce Developer Limits and Allocations Quick Reference.

Example

To get list of available service resources (appointment candidates):

```
String response = lxscheduler.SchedulerResources.getAppointmentCandidates(input);
```

To get a list of available appointment time slots for a resource:

```
String response = lxscheduler.SchedulerResources.getAppointmentSlots(input);
```

IN THIS SECTION:

SchedulerResources Methods

SchedulerResources Methods

The following are methods for `SchedulerResources`.

IN THIS SECTION:

getAppointmentCandidates(getAppointmentCandidatesInput)
Returns a list of service resources based on work type group or work type and service territories.

getAppointmentSlots(getAppointmentSlotsInput)
Returns a list of available appointment time slots for a resource based on given work type group or work type and service territories.

setAppointmentCandidatesMock(expectedResponse)
Sets a mock object when running tests for the `getAppointmentCandidates` method.

setAppointmentSlotsMock(expectedResponse)

Sets a mock object when running tests for the `getAppointmentSlots` method.

`getAppointmentCandidates(getAppointmentCandidatesInput)`

Returns a list of service resources based on work type group or work type and service territories.

Set up Salesforce Scheduler before making requests. This setup includes creating or configuring Service Resources, Service Territory Members, Work Type Groups, Work Types, Work Type Group Members, and Service Territory Work Types. See Set Up Salesforce Scheduler for more information.

The appointment time slots are determined based on multiple factors, such as field values, scheduled appointments, absences, Scheduler Settings, and Scheduling Policies to determine available time slots. See How Salesforce Scheduler Determines Available Time Slots for more information.

The following factors are considered for returning start time and end time of resources.

Resource Availability

Determined using service territory member, service territory, work type, and account operating hours fields.

Resource Unavailability

Determined by resource absences, existing appointments that the resource is assigned to. The resource must be marked as a required resource for the appointment with a status that isn't in closed, canceled, or completed.

Appointment Start Time Interval in the Scheduling Policy

Appointment start time interval field in the Scheduling Policy is used to determine when the appointment can start. This interval can be 5, 10, 15, 20, 30, or 60. By default, it's set to 15.

Work Type Duration

The end time is calculated as start time + duration of the work type.

Note: If asset scheduling is enabled, the response also includes asset-based candidates.

Signature

```
public static String getAppointmentCandidates(lxscheduler.GetAppointmentCandidatesInput
getAppointmentCandidatesInput)
```

Parameters

`getAppointmentCandidatesInput`

Type: lxscheduler.GetAppointmentCandidatesInput

This method takes input as an instance of the `lxscheduler.GetAppointmentCandidatesInput` class. Build the instance of the input class using the `lxscheduler.GetAppointmentCandidatesInputBuilder` class.

Return Value

Type: String

`getAppointmentSlots(getAppointmentSlotsInput)`

Returns a list of available appointment time slots for a resource based on given work type group or work type and service territories.

The appointment time slots are determined based on your Salesforce Scheduler data model configurations. Here are some prerequisites that you can consider while setting up data.

- Set up Salesforce Scheduler before making your requests. The setup includes creating or configuring Service Resources, Service Territory Members, Work Type Groups, Work Types, Work Type Group Members, and Service Territory Work Types. See Manage Business Information in Salesforce Scheduler for more information.
- Configure a work type mapped for each territory in the request body via Service Territory Work Type. Map the same work type to the work type group, via work type group member.

The following factors affect how time slots are calculated and returned.

- Timezones that differ across operating hours are handled and results are always returned in UTC.
- The resource must be marked as a required resource on the assigned resource object.
- The resource is considered unavailable If the status categories of the resource assigned to service appointments are other than `Canceled`, `Cannot Complete`, and `Completed`.
- Resource Absences of all types are considered unavailable from start to end.
- The following fields of Work Type records, if configured, are used to fine-tune time slot requirements. For more information, see Create Work Types in Salesforce Scheduler.

Parameter	Description
`Timeframe Start`	Time slots sooner than `current time + Timeframe Start` aren't returned.
`Timeframe End`	Time slots later than `current time + Timeframe End` aren't returned.
`Block Time Before Appointment`	The time period before the appointment is considered as unavailable.
`Block Time After Appointment`	The time period after the appointment is considered as unavailable.
`Operating Hours`	The overlap of all operating hours from the account, work type, service territory, and service territory member are considered while determining time slots. For more information, see Set Up Operating Hours in Salesforce Scheduler.

- Only the time slots within the period of 31 days from the start date are returned.
- Salesforce Scheduler uses multiple factors, such as field values, scheduled appointments, absences, Scheduler Settings, and Scheduling Policies to determine available time slots, including the earliest and latest appointment slots. See How Does Salesforce Scheduler Determine Available Time Slots.

  Note: If asset scheduling is enabled, you can provide an asset-based service resource in `requiredResourceIds` to retrieve available timeslots for the asset resource.

Signature

```
public static String getAppointmentSlots(lxscheduler.GetAppointmentSlotsInput
getAppointmentSlotsInput)
```

Parameters

`getAppointmentSlotsInput`
Type: lxscheduler.GetAppointmentSlotsInput

This method takes input as an instance of the `lxscheduler.GetAppointmentSlotsInput` class. Build the instance of the input class using the `lxscheduler.GetAppointmentSlotsInputBuilder` class.

Return Value

Type: String

setAppointmentCandidatesMock(expectedResponse)

Sets a mock object when running tests for the `getAppointmentCandidates` method.

This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Signature

```
public static void setAppointmentCandidatesMock(String expectedResponse)
```

Parameters

expectedResponse
Type: String

Return Value

Type: void

This example shows a sample implementation of the `GetAppointmentCandidates` class:

```
public class AppointmentCandidateService {
   //Instance members for parsing
   public String startTime;
   public String endTime;
   public List<String> resources;
   public String territoryId;
   public static List<AppointmentCandidateService> getAppointmentCandidates(){
      //Build input for GetAppointmentCandidates API
      lxscheduler.GetAppointmentCandidatesInput input = new
lxscheduler.GetAppointmentCandidatesInputBuilder()
         .setWorkTypeGroupId('0VSRM0000000AGT4A2')
         .setTerritoryIds(new List<String>{'0HhRM0000000G8W0AU'})
        .setStartTime(System.now().format('yyyy-MM-dd\'T\'HH:mm:ssZ','America/Los_Angeles'))


.setEndTime(System.now().addDays(2).format('yyyy-MM-dd\'T\'HH:mm:ssZ','America/Los_Angeles'))

         .setSchedulingPolicyId('0VrRM00000000D0')
         .setApiVersion(Double.valueOf('50.0'))
         .build();
      List<AppointmentCandidateService> vList =
parse(lxscheduler.SchedulerResources.getAppointmentCandidates(input));
      return vList;
   }
   private static List<AppointmentCandidateService> parse(String json) {
      return (List<AppointmentCandidateService>) System.JSON.deserialize(json,
List<AppointmentCandidateService>.class);
   }
}
```

This example shows how to set a sample mock using the `setAppointmentCandidatesMock` method:

```
@isTest
private class GetAppointmentCandidatesTest {
   static testMethod void getAppCandidatesTest() {
      String expectedResponse = '[' +
                                        '  {' +
                                       '    \"startTime\": \"2021-03-18T16:00:00.000+0000\",'
 +
                                        '    \"endTime\": \"2021-03-18T17:00:00.000+0000\",'
+
                                        '    \"resources\": [' +
                                        '      \"0HnRM0000000Fxv0AE\"' +
                                        '    ],' +
                                        '    \"territoryId\": \"0HhRM0000000G8W0AU\"' +
                                        '  },' +
                                        '  {' +
                                       '    \"startTime\": \"2021-03-18T19:00:00.000+0000\",'
 +
                                        '    \"endTime\": \"2021-03-18T20:00:00.000+0000\",'
+
                                        '    \"resources\": [' +
                                        '      \"0HnRM0000000Fxv0AE\"' +
                                        '    ],' +
                                        '    \"territoryId\": \"0HhRM0000000G8W0AU\"' +
                                        '  }' +
                                        ']';
      lxscheduler.SchedulerResources.setAppointmentCandidatesMock(expectedResponse);

      Test.startTest();
         List<AppointmentCandidateService> candidateList =
AppointmentCandidateService.getAppointmentCandidates();
         System.assertEquals(2, candidateList.size(), 'Should return only 2 records!');
      Test.stopTest();
   }
}
```

`setAppointmentSlotsMock(expectedResponse)`

Sets a mock object when running tests for the `getAppointmentSlots` method.

This constructor is intended for test usage and throws an exception if used outside of the Apex test context.

Signature

```
public static void setAppointmentSlotsMock(String expectedResponse)
```

Parameters

`expectedResponse`
Type: String

Return Value

Type: void

SkillRequirement Class

Contains information about the set of skills that are required to complete a particular task for a work type.

Namespace

LxScheduler

Usage

The constructor for this class can't be called directly. Create an instance of this class using the SkillRequirementBuilder.build() method.

SkillRequirementBuilder Class

Contains methods to build an instance of the `lxscheduler.SkillRequirement` class.

A Builder object is obtained by invoking one of the `SkillRequirementBuilder` methods defined by the `SkillRequirement` class.

Namespace

LxScheduler

IN THIS SECTION:

SkillRequirementBuilder Methods

SkillRequirementBuilder Methods

The following are methods for `SkillRequirementBuilder`.

IN THIS SECTION:

build()
Returns an instance of the `lxscheduler.SkillRequirement` object.

setSkillId(skillId)
Sets the skill that is required to complete a particular task for a work type. This is a required field.

setSkillLevel(skillLevel)
Sets the level of the skill that is required to complete a particular task for a work type

`build()`

Returns an instance of the `lxscheduler.SkillRequirement` object.

Signature

```
public lxscheduler.SkillRequirement build()
```

Return Value

Type: lxscheduler.SkillRequirement

`setSkillId(skillId)`

Sets the skill that is required to complete a particular task for a work type. This is a required field.

Signature

```
public lxscheduler.SkillRequirementBuilder setSkillId(String skillId)
```

Parameters

`skillId`
Type: String

Return Value

Type: lxscheduler.SkillRequirementBuilder

`setSkillLevel(skillLevel)`

Sets the level of the skill that is required to complete a particular task for a work type

Signature

```
public lxscheduler.SkillRequirementBuilder setSkillLevel(Double skillLevel)
```

Parameters

`skillLevel`
Type: Double

The skill levels can range from zero to 99.99. Depending on your business needs, you might want the skill level to reflect years of experience, certification levels, or license classes.

Return Value

Type: lxscheduler.SkillRequirementBuilder

WorkType Class

Contains information about the type of work to be performed.

Namespace

LxScheduler

Usage

The constructor for this class can't be called directly. Create an instance of this class using the WorkTypeBuilder.build() method.

WorkTypeBuilder Class

Contains methods to build an instance of the `lxscheduler.WorkType` class.

A Builder object is obtained by invoking one of the `WorkTypeBuilder` methods defined by the `WorkType` class.

Namespace

LxScheduler

IN THIS SECTION:

WorkTypeBuilder Methods

WorkTypeBuilder Methods

The following are methods for `WorkTypeBuilder`.

IN THIS SECTION:

build()
Returns an instance of the `lxscheduler.WorkType` object.

setBlockTimeAfterAppointmentInMinutes(blockTimeAfterAppointmentInMinutes)
Sets the time period, in minutes.

setBlockTimeBeforeAppointmentInMinutes(blockTimeBeforeAppointmentInMinutes)
Sets the time period, in minutes.

setDurationInMinutes(durationInMinutes)
Sets the event length.

setId(id)
Sets the ID of the work type to the specified ID.

setOperatingHoursId(operatingHoursId)
Sets the overlap of operating hours.

setSkillRequirements(skillRequirements)
Sets the skills that are required to complete a particular task for a work type.

setTimeFrameEndInMinutes(timeFrameEndInMinutes)
Sets the end of the timeframe.

setTimeFrameStartInMinutes(timeFrameStartInMinutes)
Sets the beginning of the timeframe.

build()

Returns an instance of the `lxscheduler.WorkType` object.

Signature

```
public lxscheduler.WorkType build()
```

Return Value

Type: lxscheduler.WorkType

setBlockTimeAfterAppointmentInMinutes(blockTimeAfterAppointmentInMinutes)

Sets the time period, in minutes.

Signature

```
public lxscheduler.WorkTypeBuilder setBlockTimeAfterAppointmentInMinutes(Integer
blockTimeAfterAppointmentInMinutes)
```

Parameters

blockTimeAfterAppointmentInMinutes
Type: Integer

The time period after the appointment is considered unavailable.

Return Value

Type: lxscheduler.WorkTypeBuilder

setBlockTimeBeforeAppointmentInMinutes(blockTimeBeforeAppointmentInMinutes)

Sets the time period, in minutes.

Signature

```
public lxscheduler.WorkTypeBuilder setBlockTimeBeforeAppointmentInMinutes(Integer
blockTimeBeforeAppointmentInMinutes)
```

Parameters

blockTimeBeforeAppointmentInMinutes
Type: Integer

The time period before the appointment is considered as unavailable.

Return Value

Type: lxscheduler.WorkTypeBuilder

`setDurationInMinutes(durationInMinutes)`

Sets the event length.

Signature

```
public lxscheduler.WorkTypeBuilder setDurationInMinutes(Integer durationInMinutes)
```

Parameters

`durationInMinutes`
Type: Integer

Contains the event length, in minutes. Required if *`id`* is not given.

Return Value

Type: lxscheduler.WorkTypeBuilder

`setId(id)`

Sets the ID of the work type to the specified ID.

Signature

```
public lxscheduler.WorkTypeBuilder setId(String id)
```

Parameters

`id`
Type: String

The ID of the work type. Required if you're using shifts or if *`durationInMinutes`* is not given.

Return Value

Type: lxscheduler.WorkTypeBuilder

`setOperatingHoursId(operatingHoursId)`

Sets the overlap of operating hours.

Signature

```
public lxscheduler.WorkTypeBuilder setOperatingHoursId(String operatingHoursId)
```

Parameters

`operatingHoursId`
Type: String

The overlap of all operating hours from the account, work type, service territory, and service territory member are considered while determining time slots.

Return Value

Type: lxscheduler.WorkTypeBuilder

`setSkillRequirements(skillRequirements)`

Sets the skills that are required to complete a particular task for a work type.

Signature

```
public lxscheduler.WorkTypeBuilder
setSkillRequirements(List<lxscheduler.SkillRequirement> skillRequirements)
```

Parameters

`skillRequirements`
Type: List<lxscheduler.SkillRequirement>

This method takes input as an instance of the `lxscheduler.SkillRequirement` class. Build the instance of the input class using the `lxscheduler.SkillRequirementBuilder` class.

Return Value

Type: lxscheduler.WorkTypeBuilder

`setTimeFrameEndInMinutes(timeFrameEndInMinutes)`

Sets the end of the timeframe.

Signature

```
public lxscheduler.WorkTypeBuilder setTimeFrameEndInMinutes(Integer
timeFrameEndInMinutes)
```

Parameters

`timeFrameEndInMinutes`
Type: Integer

Return Value

Type: lxscheduler.WorkTypeBuilder

`setTimeFrameStartInMinutes(timeFrameStartInMinutes)`

Sets the beginning of the timeframe.

Signature

```
public lxscheduler.WorkTypeBuilder setTimeFrameStartInMinutes(Integer
timeFrameStartInMinutes)
```

Parameters

timeFrameStartInMinutes
Type: Integer

Return Value

Type: lxscheduler.WorkTypeBuilder

ServiceResourceScheduleHandler Interface

Allows an implementing class to check external calendar events to find already booked time slots for the requested service resources. This interface is part of Salesforce Scheduler.

Namespace

LxScheduler

Usage

The `lxscheduler.ServiceResourceScheduleHandler` interface is called by Salesforce Scheduler APIs.

To implement this interface, you must first declare a class with the `implements` keyword as follows:

```
public class ServiceResourceScheduleHandlerImpl implements
LxScheduler.ServiceResourceScheduleHandler{}
```

Next, your class must provide an implementation for the following method:

```
public static List<LxScheduler.ServiceResourceSchedule>
getUnavailableTimeslots(LxScheduler.ServiceAppointmentRequestInfo requestInfo){
      //Your code here
}
```

The implemented method must be declared as `global` or `public`.

IN THIS SECTION:

ServiceResourceScheduleHandler Methods

ServiceResourceScheduleHandler Example Implementation

ServiceResourceScheduleHandler Methods

The following are methods for `ServiceResourceScheduleHandler`.

IN THIS SECTION:

getUnavailableTimeslots(var1)

Passes the required information to get unavailable time slots from an external system. The implementation of this method returns the `lxscheduler.ServiceResourceSchedule` class.

`getUnavailableTimeslots(var1)`

Passes the required information to get unavailable time slots from an external system. The implementation of this method returns the `lxscheduler.ServiceResourceSchedule` class.

Signature

```
public List<lxscheduler.ServiceResourceSchedule>
getUnavailableTimeslots(lxscheduler.ServiceAppointmentRequestInfo var1)
```

Parameters

var1

Type: lxscheduler.ServiceAppointmentRequestInfo

Represents the list of parameters that are passed to the ServiceResourceScheduleHandler interface.

Return Value

Type: List<lxscheduler.ServiceResourceSchedule>

ServiceResourceScheduleHandler Example Implementation

This is an example implementation of the `lxscheduler.ServiceResourceScheduleHandler` interface.

```
/**
 * Implement interface lxscheduler.ServiceResourceScheduleHandler
 * This class is called when fetching service resources and time slots through Salesforce
 Scheduler API.*/
   Public class ServiceResourceScheduleHandlerImpl implements
lxscheduler.ServiceResourceScheduleHandler{

    // The main interface method.
    public static List<lxscheduler.ServiceResourceSchedule>
getUnavailableTimeslots(lxscheduler.ServiceAppointmentRequestInfo requestInfo){
        //Request info values.
        List<lxscheduler.ServiceResourceInfo>
serviceResources=requestInfo.getServiceResources();
        DateTime startDate=requestInfo.getStartDate();
        DateTime endDate=requestInfo.getEndDate();


        List<lxscheduler.ServiceResourceSchedule> resourceUnavailability = new
List<lxscheduler.ServiceResourceSchedule>();
        Set<lxscheduler.UnavailableTimeslot> unavailabilityIntervals = new
Set<lxscheduler.UnavailableTimeslot>();

        //This is a dummy response. Implement your own business logic to connect to your
internal or external systems.
        for (Integer i = 0; i < 5; i++) {
             //Set the unavailability intervals of a service resource.
            unavailabilityIntervals.add(new
lxscheduler.UnavailableTimeslot(startDate.addMinutes(15*i),startDate.addMinutes(15*(i+1))));
```

```
        }

       for (lxscheduler.ServiceResourceInfo ServiceResource:serviceResources) {
            //Set the unavailability of Service resource.
        resourceUnavailability.add(new
lxscheduler.ServiceResourceSchedule(serviceResource.getServiceResourceId(),unavailabilityIntervals));


       }

        return resourceUnavailability;
    }
}
```

This example shows how to set a sample test mock using the `lxscheduler.ServiceResourceScheduleHandler` interface.

```
@isTest
private class ServiceResourceScheduleHandlerImplTest {
   static testMethod void getUnavailableTimeslotsTest() {

      //Initializing the test execution with mock values. Change it according to the
implementation.
      //In case of non-test execution, the lxscheduler.ServiceAppointmentRequestInfo
instance will automatically initialize.

      //Mock values for lxscheduler.ServiceResourceInfo
      String userId = '005D2000000I1N6IAK';
      String userName = 'someuser@example.com';
      String email = 'someuser@example.com';
      String serviceResourceId = '0HnD20000004C9bKAE';
      List<String> territoryIds = new List<String>();
      String resourceType = 'T';
      lxscheduler.ServiceResourceInfo serviceResInfo = new
lxscheduler.ServiceResourceInfo(userId, userName, email,
                                                      serviceResourceId, territoryIds,
resourceType);

      //Mock values for lxscheduler.ServiceAppointmentRequestInfo
      DateTime startDate = System.now();
      DateTime endDate = System.now();
      List<lxscheduler.ServiceResourceInfo> serviceResources = new
List<lxscheduler.ServiceResourceInfo>();
      serviceResources.add(serviceResInfo);
      String schedulingPolicyId = '0VrD20000004C9S';
      String workTypeGroupId = '0VSD20000004C93OAE';
      String accountId = '001D2000002pkXwIAI';
      String primaryResourceId = '0HnD20000004C9bKAE';
      String workTypeId = '08qD20000004C9XIAU';
      String correlationId = 'SOME_ID';

      lxscheduler.ServiceAppointmentRequestInfo mockRequestInfo = new
lxscheduler.ServiceAppointmentRequestInfo(startDate, endDate, serviceResources,
                                                                  schedulingPolicyId,
workTypeGroupId, accountId,
```

```
                                                                    primaryResourceId,
workTypeId, correlationId);

      ServiceResourceScheduleHandlerImpl.getUnavailableTimeslots(mockRequestInfo);
   }
}
```

ServiceAppointmentRequestInfo Class

Represents the list of parameters that are passed to the ServiceResourceScheduleHandler interface. This class is implemented internally by Apex.

Namespace

LxScheduler

IN THIS SECTION:

ServiceAppointmentRequestInfo Constructors

ServiceAppointmentRequestInfo Methods

ServiceAppointmentRequestInfo Constructors

The following are constructors for `ServiceAppointmentRequestInfo`.

IN THIS SECTION:

ServiceAppointmentRequestInfo(startDate, endDate, ServiceResources, SchedulingPolicyId, workTypeGroupId, accountId, primaryResourceId, workTypeId, correlationId)

Creates a new instance of the `lxscheduler.ServiceAppointmentRequestInfo` class using the specified start date, end date, service resources, scheduling policy, work type group, account ID, primary resource, work type, and correlation.

`ServiceAppointmentRequestInfo(startDate, endDate, ServiceResources, SchedulingPolicyId, workTypeGroupId, accountId, primaryResourceId, workTypeId, correlationId)`

Creates a new instance of the `lxscheduler.ServiceAppointmentRequestInfo` class using the specified start date, end date, service resources, scheduling policy, work type group, account ID, primary resource, work type, and correlation.

Signature

```
public ServiceAppointmentRequestInfo(Datetime startDate, Datetime endDate,
List<lxscheduler.ServiceResourceInfo> ServiceResources, String SchedulingPolicyId,
String workTypeGroupId, String accountId, String primaryResourceId, String workTypeId,
String correlationId)
```

Parameters

startDate
Type: Datetime

The start date and time for which unavailable time slots are requested.

endDate
Type: Datetime

The end date and time for which unavailable time slots are requested.

ServiceResources
Type: List<lxscheduler.ServiceResourceInfo>

The list of requested service resources for the unavailable time slots.

SchedulingPolicyId
Type: String

The ID of the scheduling policy .

workTypeGroupId
Type: String

The work type group ID.

accountId
Type: String

The account ID of an existing user.

primaryResourceId
Type: String

The ID of the primary service resource.

workTypeId
Type: String

The work type ID.

correlationId
Type: String

A unique identifier for a service appointment request.

ServiceAppointmentRequestInfo Methods

The following are methods for `ServiceAppointmentRequestInfo`.

IN THIS SECTION:

getAccountId()
Returns the account ID of the customer if the API request contains one.

getCorrelationId()
Returns a unique identifier for a request.

getEndDate()
Returns the end date and time for which unavailable time slots are requested.

getPrimaryResourceId()
Returns the ID of the primary service resource.

getSchedulingPolicyId()
Returns the ID of the scheduling policy that the API request contains.

getServiceResources()
Returns the list of requested service resources for the unavailable time slots.

getStartDate()
Returns the start date and time for which unavailable time slots are requested.

getWorkTypeGroupId()
Returns the work type group ID if the API request contains one.

getWorkTypeId()
Returns the work type ID if the API request contains one.

`getAccountId()`

Returns the account ID of the customer if the API request contains one.

Signature

```
public String getAccountId()
```

Return Value

Type: String

`getCorrelationId()`

Returns a unique identifier for a request.

Signature

```
public String getCorrelationId()
```

Return Value

Type: String

`getEndDate()`

Returns the end date and time for which unavailable time slots are requested.

Signature

```
public Datetime getEndDate()
```

Return Value

Type: Datetime

getPrimaryResourceId()

Returns the ID of the primary service resource.

Signature

```
public String getPrimaryResourceId()
```

Return Value

Type: String

getSchedulingPolicyId()

Returns the ID of the scheduling policy that the API request contains.

Signature

```
public String getSchedulingPolicyId()
```

Return Value

Type: String

getServiceResources()

Returns the list of requested service resources for the unavailable time slots.

Signature

```
public List<lxscheduler.ServiceResourceInfo> getServiceResources()
```

Return Value

Type: List<lxscheduler.ServiceResourceInfo>

getStartDate()

Returns the start date and time for which unavailable time slots are requested.

Signature

```
public Datetime getStartDate()
```

Return Value

Type: Datetime

getWorkTypeGroupId()

Returns the work type group ID if the API request contains one.

Signature

```
public String getWorkTypeGroupId()
```

Return Value

Type: String

`getWorkTypeId()`

Returns the work type ID if the API request contains one.

Signature

```
public String getWorkTypeId()
```

Return Value

Type: String

ServiceResourceInfo Class

Contains information about a service resource.

Namespace

LxScheduler

IN THIS SECTION:

ServiceResourceInfo Constructors

ServiceResourceInfo Methods

ServiceResourceInfo Constructors

The following are constructors for `ServiceResourceInfo`.

IN THIS SECTION:

ServiceResourceInfo(userId, userName, email, serviceResourceId, territoryIds, resourceType)
Creates a new instance of the `lxscheduler.ServiceResourceInfo` class using the specified service resource details.

`ServiceResourceInfo(userId, userName, email, serviceResourceId, territoryIds, resourceType)`

Creates a new instance of the `lxscheduler.ServiceResourceInfo` class using the specified service resource details.

Signature

```
public ServiceResourceInfo(String userId, String userName, String email, String
serviceResourceId, List<String> territoryIds, String resourceType)
```

Parameters

userId
Type: String

The user ID of the service resource.

userName
Type: String

The user name of the service resource.

email
Type: String

The email ID of the service resource.

serviceResourceId
Type: String

The ID of the service resource.

territoryIds
Type: List<String>

A list of requested service territories for the service resource.

resourceType
Type: String

The type of the service resource such as Technician or Asset.

ServiceResourceInfo Methods

The following are methods for `ServiceResourceInfo`.

IN THIS SECTION:

getEmail()
Returns the email ID of the service resource.

getResourceType()
Returns the type of the service resource such as Technician or Asset.

getServiceResourceId()
Returns the ID of the service resource.

getTerritoryIds()
Returns a list of requested service territories for the service resource.

getUserId()
Returns the user ID of the service resource.

getUserName()
Returns the user name of the service resource.

getEmail()

Returns the email ID of the service resource.

Signature

```
public String getEmail()
```

Return Value

Type: String

getResourceType()

Returns the type of the service resource such as Technician or Asset.

Signature

```
public String getResourceType()
```

Return Value

Type: String

getServiceResourceId()

Returns the ID of the service resource.

Signature

```
public String getServiceResourceId()
```

Return Value

Type: String

getTerritoryIds()

Returns a list of requested service territories for the service resource.

Signature

```
public List<String> getTerritoryIds()
```

Return Value

Type: List<String>

getUserId()

Returns the user ID of the service resource.

Signature

```
public String getUserId()
```

Return Value

Type: String

`getUserName()`

Returns the user name of the service resource.

Signature

```
public String getUserName()
```

Return Value

Type: String

ServiceResourceSchedule Class

Use this class to pass results from your implemented Apex class to the ServiceResourceScheduleHandler interface methods.

Namespace

LxScheduler

IN THIS SECTION:

ServiceResourceSchedule Constructors

ServiceResourceSchedule Properties

ServiceResourceSchedule Constructors

The following are constructors for `ServiceResourceSchedule`.

IN THIS SECTION:

ServiceResourceSchedule(serviceResourceId, unavailableTimeslots)
Creates a new instance of lxscheduler.ServiceResourceSchedule class.

`ServiceResourceSchedule(serviceResourceId, unavailableTimeslots)`

Creates a new instance of lxscheduler.ServiceResourceSchedule class.

Signature

```
public ServiceResourceSchedule(String serviceResourceId,
Set<lxscheduler.UnavailableTimeslot> unavailableTimeslots)
```

Parameters

serviceResourceId
Type: String

Record ID of the service resource.

unavailableTimeslots
Type: Set<lxscheduler.UnavailableTimeslot>

An instance of lxscheduler.UnavailableTimeslot class.

ServiceResourceSchedule Properties

The following are properties for `ServiceResourceSchedule`.

IN THIS SECTION:

serviceResourceId
Record ID of the service resource.

unavailableTimeslots
An instance of lxscheduler.UnavailableTimeslot class.

serviceResourceId

Record ID of the service resource.

Signature

```
public String serviceResourceId {get; set;}
```

Property Value

Type: String

unavailableTimeslots

An instance of lxscheduler.UnavailableTimeslot class.

Signature

```
public Set<lxscheduler.UnavailableTimeslot> unavailableTimeslots {get; set;}
```

Property Value

Type: Set<lxscheduler.UnavailableTimeslot>

UnavailableTimeslot Class

Use this class to pass the unavailable time slots to the lxscheduler.ServiceResourceSchedule class. Timezones that differ across operating hours are handled and results are always returned in UTC.

Namespace

LxScheduler

IN THIS SECTION:

UnavailableTimeslot Constructors

UnavailableTimeslot Properties

UnavailableTimeslot Constructors

The following are constructors for `UnavailableTimeslot`.

IN THIS SECTION:

UnavailableTimeslot(timeMin, timeMax)
Creates an instance of lxscheduler.UnavailableTimeslot class.

`UnavailableTimeslot(timeMin, timeMax)`

Creates an instance of lxscheduler.UnavailableTimeslot class.

Signature

```
public UnavailableTimeslot(Datetime timeMin, Datetime timeMax)
```

Parameters

`timeMin`
Type: Datetime

Start time of an unavailable time slot.

`timeMax`
Type: Datetime

End time of an unavailable time slot.

UnavailableTimeslot Properties

The following are properties for `UnavailableTimeslot`.

IN THIS SECTION:

timeMax
End time of an unavailable time slot.

timeMin
Start time of an unavailable time slot.

timeMax

End time of an unavailable time slot.

Signature

```
public Datetime timeMax {get; set;}
```

Property Value

Type: Datetime

timeMin

Start time of an unavailable time slot.

Signature

```
public Datetime timeMin {get; set;}
```

Property Value

Type: Datetime

Messaging Namespace

The `Messaging` namespace provides classes and methods for Salesforce outbound and inbound email functionality.

The following are the classes in the `Messaging` namespace.

IN THIS SECTION:

AttachmentRetrievalOption Enum
Provides options for including attachment metadata only, attachment metadata and content, or excluding attachments.

Email Class (Base Email Methods)
Contains base email methods common to both single and mass email.

EmailFileAttachment Class
EmailFileAttachment is used in SingleEmailMessage to specify attachments passed in as part of the request, as opposed to existing documents in Salesforce.

InboundEmail Class
Represents an inbound email object.

InboundEmail.AuthenticationResult Class
Contains the authentication type and response for inbound emails.

InboundEmail.AuthenticationResultField Class
Contains field data from the authentication result response for inbound emails.

InboundEmail.BinaryAttachment Class
An InboundEmail object stores binary attachments in an InboundEmail.BinaryAttachment object.

InboundEmail.TextAttachment Class

An InboundEmail object stores text attachments in an InboundEmail.TextAttachment object.

InboundEmailResult Class

The InboundEmailResult object is used to return the result of the email service. If this object is null, the result is assumed to be successful.

InboundEnvelope Class

The InboundEnvelope object stores the envelope information associated with the inbound email, and has the following fields.

MassEmailMessage Class

Contains methods for sending mass email.

InboundEmail.Header Class

An InboundEmail object stores RFC 2822 email header information in an InboundEmail.Header object with the following properties.

PushNotification Class

`PushNotification` is used to configure push notifications and send them from an Apex trigger.

PushNotificationPayload Class

Contains methods to create the notification message payload for an Apple device.

CustomNotification Class

`CustomNotification` is used to create, configure, and send custom notifications from Apex code.

RenderEmailTemplateBodyResult Class

Contains the results for rendering email templates.

RenderEmailTemplateError Class

Represents an error that the `RenderEmailTemplateBodyResult` object can contain.

SendEmailError Class

Represents an error that the SendEmailResult object may contain.

SendEmailResult Class

Contains the result of sending an email message.

SingleEmailMessage Class

Contains methods for sending single email messages.

AttachmentRetrievalOption Enum

Provides options for including attachment metadata only, attachment metadata and content, or excluding attachments.

Namespace

Messaging

Usage

Use these enum values with the renderStoredEmailTemplate(templateId, whoId, whatId, attachmentRetrievalOption) method.

Enum Values

The following are the values of the `Messaging.AttachmentRetrievalOption` enum.

Value	Description
METADATA_ONLY	Includes only the file name, content type, and the object ID in the `fileAttachments` property of `Messaging.SingleEmailMessage`. Note: When the template is rendered from a Visualforce template (and not from a static file attached to the template), the object ID is not available.
METADATA_WITH_BODY	Includes the attachment content, in addition to the file name, content type, and the object ID in the `fileAttachments` property of `Messaging.SingleEmailMessage`.
NONE	Doesn't include any attachments in `Messaging.SingleEmailMessage`.

Email Class (Base Email Methods)

Contains base email methods common to both single and mass email.

Namespace

Messaging

Usage

Note: If templates are not being used, all email content must be in plain text, HTML, or both.Visualforce email templates cannot be used for mass email.

Email Methods

The following are methods for `Email`. All are instance methods.

IN THIS SECTION:

setBccSender(bcc)

Indicates whether the email sender receives a copy of the email that is sent. For a mass mail, the sender is only copied on the first email sent.

setReplyTo(replyAddress)

Optional. The email address that receives the message when a recipient replies.

setTemplateID(templateId)

The ID of the template to be merged to create this email. Specify a value for `setTemplateId`, `setHtmlBody`, or `setPlainTextBody`. Or, you can define both `setHtmlBody` and `setPlainTextBody`.

setSaveAsActivity(saveAsActivity)

Optional. The default value is `true`, meaning the email is saved as an activity. This argument only applies if the recipient list is based on `targetObjectId` or `targetObjectIds`. If HTML email tracking is enabled for the organization, you will be able to track open rates.

setSenderDisplayName(displayName)

Optional. The name that appears on the From line of the email. This cannot be set if the object associated with a `setOrgWideEmailAddressId` for a SingleEmailMessage has defined its `DisplayName` field.

setUseSignature(useSignature)

Indicates whether the email includes an email signature if the user has one configured. The default is `true`, meaning if the user has a signature it is included in the email unless you specify `false`.

`setBccSender(bcc)`

Indicates whether the email sender receives a copy of the email that is sent. For a mass mail, the sender is only copied on the first email sent.

Signature

```
public Void setBccSender(Boolean bcc)
```

Parameters

bcc
Type: Boolean

Return Value

Type: Void

Usage

Note: If the BCC compliance option is set at the organization level, the user cannot add BCC addresses on standard messages. The following error code is returned: `BCC_NOT_ALLOWED_IF_BCC_ COMPLIANCE_ENABLED`. Contact your Salesforce representative for information on BCC compliance.

`setReplyTo(replyAddress)`

Optional. The email address that receives the message when a recipient replies.

Signature

```
public Void setReplyTo(String replyAddress)
```

Parameters

replyAddress
Type: String

Return Value

Type: Void

setTemplateID(templateId)

The ID of the template to be merged to create this email. Specify a value for `setTemplateId`, `setHtmlBody`, or `setPlainTextBody`. Or, you can define both `setHtmlBody` and `setPlainTextBody`.

Signature

```
public Void setTemplateID(ID templateId)
```

Parameters

templateId
Type: ID

Return Value

Type: Void

Usage

Note: `setHtmlBody` and `setPlainTextBody` apply only to single email methods, not to mass email methods.

setSaveAsActivity(saveAsActivity)

Optional. The default value is `true`, meaning the email is saved as an activity. This argument only applies if the recipient list is based on `targetObjectId` or `targetObjectIds`. If HTML email tracking is enabled for the organization, you will be able to track open rates.

Signature

```
public Void setSaveAsActivity(Boolean saveAsActivity)
```

Parameters

saveAsActivity
Type: Boolean

Return Value

Type: Void

setSenderDisplayName(displayName)

Optional. The name that appears on the From line of the email. This cannot be set if the object associated with a `setOrgWideEmailAddressId` for a SingleEmailMessage has defined its `DisplayName` field.

Signature

```
public Void setSenderDisplayName(String displayName)
```

Parameters

displayName
Type: String

Return Value

Type: Void

setUseSignature(useSignature)

Indicates whether the email includes an email signature if the user has one configured. The default is `true`, meaning if the user has a signature it is included in the email unless you specify `false`.

Signature

```
public Void setUseSignature(Boolean useSignature)
```

Parameters

useSignature
Type: Boolean

Return Value

Type: Void

EmailFileAttachment Class

EmailFileAttachment is used in SingleEmailMessage to specify attachments passed in as part of the request, as opposed to existing documents in Salesforce.

Namespace

Messaging

IN THIS SECTION:

EmailFileAttachment Constructors

EmailFileAttachment Properties

EmailFileAttachment Constructors

The following are constructors for `EmailFileAttachment`.

IN THIS SECTION:

EmailFileAttachment()
Creates a new instance of the `Messaging.EmailFileAttachment` class.

`EmailFileAttachment()`

Creates a new instance of the `Messaging.EmailFileAttachment` class.

Signature

```
public EmailFileAttachment()
```

EmailFileAttachment Properties

The following are properties for `EmailFileAttachment`.

IN THIS SECTION:

body
Gets or sets the attachment itself.

contenttype
Gets or sets the attachment's Content-Type.

filename
Gets or sets the name of the file to attach.

id
Read-Only. Gets the attachment ID.

inline
Specifies a Content-Disposition of inline (`true`) or attachment (`false`).

`body`

Gets or sets the attachment itself.

Signature

```
public Blob body {get; set;}
```

Property Value

Type: Blob

`contenttype`

Gets or sets the attachment's Content-Type.

Signature

```
public String contenttype {get; set;}
```

Property Value

Type: String

filename

Gets or sets the name of the file to attach.

Signature

```
public String filename {get; set;}
```

Property Value

Type: String

id

Read-Only. Gets the attachment ID.

Signature

```
public Id id {get;}
```

Property Value

Type: Id

inline

Specifies a Content-Disposition of inline (`true`) or attachment (`false`).

Signature

```
public Boolean inline {get; set;}
```

Property Value

Type: Boolean

InboundEmail Class

Represents an inbound email object.

Namespace

Messaging

IN THIS SECTION:

InboundEmail Constructors

InboundEmail Properties

InboundEmail Constructors

The following are constructors for `InboundEmail`.

IN THIS SECTION:

InboundEmail()
Creates a new instance of the `Messaging.InboundEmail` class.

`InboundEmail()`

Creates a new instance of the `Messaging.InboundEmail` class.

Signature

```
public InboundEmail()
```

InboundEmail Properties

The following are properties for `InboundEmail`.

IN THIS SECTION:

authenticationResults
A list of authentication results received with the email, if any.

binaryAttachments
A list of binary attachments received with the email, if any.

ccAddresses
A list of carbon copy (CC) addresses, if any.

fromAddress
The email address that appears in the From field.

fromName
The name that appears in the From field, if any.

headers
A list of the RFC 2822 headers in the email.

htmlBody
The HTML version of the email, if specified by the sender.

htmlBodyIsTruncated
Indicates whether the HTML body text is truncated (`true`) or not (`false`.)

inReplyTo
The In-Reply-To field of the incoming email. Identifies the email or emails to which this one is a reply (parent emails). Contains the parent email or emails' message-IDs.

messageId
The Message-ID—the incoming email's unique identifier.

plainTextBody
The plain text version of the email, if specified by the sender.

plainTextBodyIsTruncated
Indicates whether the plain body text is truncated (`true`) or not (`false`.)

references
The References field of the incoming email. Identifies an email thread. Contains a list of the parent emails' References and message IDs, and possibly the In-Reply-To fields.

replyTo
The email address that appears in the reply-to header.

subject
The subject line of the email, if any.

textAttachments
A list of text attachments received with the email, if any.

toAddresses
The email address that appears in the `To` field.

`authenticationResults`

A list of authentication results received with the email, if any.

Signature

```
public InboundEmail.AuthenticationResult[] authenticationResults {get; set;}
```

Property Value

Type: InboundEmail.AuthenticationResult[]

Usage

Examples of authentication results include `dkim`, `dmarc`, and `spf`.

`binaryAttachments`

A list of binary attachments received with the email, if any.

Signature

```
public InboundEmail.BinaryAttachment[] binaryAttachments {get; set;}
```

Property Value

Type: InboundEmail.BinaryAttachment[]

Usage

Examples of binary attachments include image, audio, application, and video files.

ccAddresses

A list of carbon copy (CC) addresses, if any.

Signature

```
public String[] ccAddresses {get; set;}
```

Property Value

Type: String[]

fromAddress

The email address that appears in the From field.

Signature

```
public String fromAddress {get; set;}
```

Property Value

Type: String

fromName

The name that appears in the From field, if any.

Signature

```
public String fromName {get; set;}
```

Property Value

Type: String

headers

A list of the RFC 2822 headers in the email.

Signature

```
public InboundEmail.Header[] headers {get; set;}
```

Property Value

Type: InboundEmail.Header[]

Usage

The list of the RFC 2822 headers includes:

- Recieved from
- Custom headers
- Message-ID
- Date

htmlBody

The HTML version of the email, if specified by the sender.

Signature

```
public String htmlBody {get; set;}
```

Property Value

Type: String

htmlBodyIsTruncated

Indicates whether the HTML body text is truncated (`true`) or not (`false`.)

Signature

```
public Boolean htmlBodyIsTruncated {get; set;}
```

Property Value

Type: Boolean

inReplyTo

The In-Reply-To field of the incoming email. Identifies the email or emails to which this one is a reply (parent emails). Contains the parent email or emails' message-IDs.

Signature

```
public String inReplyTo {get; set;}
```

Property Value

Type: String

messageId

The Message-ID—the incoming email's unique identifier.

Signature

```
public String messageId {get; set;}
```

Property Value

Type: String

plainTextBody

The plain text version of the email, if specified by the sender.

Signature

```
public String plainTextBody {get; set;}
```

Property Value

Type: String

plainTextBodyIsTruncated

Indicates whether the plain body text is truncated (`true`) or not (`false`.)

Signature

```
public Boolean plainTextBodyIsTruncated {get; set;}
```

Property Value

Type: Boolean

references

The References field of the incoming email. Identifies an email thread. Contains a list of the parent emails' References and message IDs, and possibly the In-Reply-To fields.

Signature

```
public String[] references {get; set;}
```

Property Value

Type: String[]

replyTo

The email address that appears in the reply-to header.

Signature

```
public String replyTo {get; set;}
```

Property Value

Type: String

Usage

If there is no reply-to header, this field is identical to the `fromAddress` field.

subject

The subject line of the email, if any.

Signature

```
public String subject {get; set;}
```

Property Value

Type: String

textAttachments

A list of text attachments received with the email, if any.

Signature

```
public InboundEmail.TextAttachment[] textAttachments {get; set;}
```

Property Value

Type: InboundEmail.TextAttachment[]

Usage

The text attachments can be any of the following:

- Attachments with a Multipurpose Internet Mail Extension (MIME) type of `text`
- Attachments with a MIME type of `application/octet-stream` and a file name that ends with either a `.vcf` or `.vcs` extension. These are saved as `text/x-vcard` and `text/calendar` MIME types, respectively.

toAddresses

The email address that appears in the `To` field.

Signature

```
public String[] toAddresses {get; set;}
```

Property Value

Type: String[]

InboundEmail.AuthenticationResult Class

Contains the authentication type and response for inbound emails.

Namespace

Messaging

IN THIS SECTION:

InboundEmail.AuthenticationResult Constructors

InboundEmail.AuthenticationResult Properties

InboundEmail.AuthenticationResult Constructors

The following are constructors for `InboundEmail.AuthenticationResult`.

IN THIS SECTION:

InboundEmail.AuthenticationResult()
Creates a new instance of the `Messaging.InboundEmail.AuthenticationResult` class.

`InboundEmail.AuthenticationResult()`

Creates a new instance of the `Messaging.InboundEmail.AuthenticationResult` class.

Signature

```
public InboundEmail.AuthenticationResult()
```

InboundEmail.AuthenticationResult Properties

The following are properties for `InboundEmail.AuthenticationResult`.

IN THIS SECTION:

authenticationResultFields
Additional information in authentication result headers. Examples include: `name: smtp.mailfrom` and `value: example.com`.

method
The authentication method used for the security check. Possible values include `dkim`, `dmarc`, or `spf`.

result
The result of the authentication check. When the email service is configured to verify the legitimacy of the sending server before processing a message, possible values include `pass` or `fail`. Otherwise, the value returned is `none`.

`authenticationResultFields`

Additional information in authentication result headers. Examples include: `name: smtp.mailfrom` and `value: example.com`.

Signature

```
public InboundEmail.AuthenticationResultField[] authenticationResultFields {get; set;}
```

Property Value

Type: InboundEmail.AuthenticationResultField[]

`method`

The authentication method used for the security check. Possible values include `dkim`, `dmarc`, or `spf`.

Signature

```
public String method {get; set;}
```

Property Value

Type: String

`result`

The result of the authentication check. When the email service is configured to verify the legitimacy of the sending server before processing a message, possible values include `pass` or `fail`. Otherwise, the value returned is `none`.

Signature

```
public String result {get; set;}
```

Property Value

Type: String

InboundEmail.AuthenticationResultField Class

Contains field data from the authentication result response for inbound emails.

Namespace

Messaging

IN THIS SECTION:

InboundEmail.AuthenticationResultField Constructors

InboundEmail.AuthenticationResultField Properties

InboundEmail.AuthenticationResultField Constructors

The following are constructors for `InboundEmail.AuthenticationResultField`.

IN THIS SECTION:

InboundEmail.AuthenticationResultField()
Creates a new instance of the `Messaging.InboundEmail.AuthenticationResultField` class.

`InboundEmail.AuthenticationResultField()`

Creates a new instance of the `Messaging.InboundEmail.AuthenticationResultField` class.

Signature

```
public InboundEmail.AuthenticationResultField()
```

InboundEmail.AuthenticationResultField Properties

The following are properties for `InboundEmail.AuthenticationResultField`.

IN THIS SECTION:

name
The authentication result field name. For example: `smtp.mailfrom`.
value
The authentication result field value. For example: `example.com`.

`name`

The authentication result field name. For example: `smtp.mailfrom`.

Signature

```
public String name {get; set;}
```

Property Value

Type: String

`value`

The authentication result field value. For example: `example.com`.

Signature

```
public String value {get; set;}
```

Property Value

Type: String

InboundEmail.BinaryAttachment Class

An InboundEmail object stores binary attachments in an InboundEmail.BinaryAttachment object.

Namespace

Messaging

Usage

Examples of binary attachments include image, audio, application, and video files.

IN THIS SECTION:

InboundEmail.BinaryAttachment Constructors

InboundEmail.BinaryAttachment Properties

InboundEmail.BinaryAttachment Constructors

The following are constructors for `InboundEmail.BinaryAttachment`.

IN THIS SECTION:

InboundEmail.BinaryAttachment()
Creates a new instance of the `Messaging.InboundEmail.BinaryAttachment` class.

`InboundEmail.BinaryAttachment()`

Creates a new instance of the `Messaging.InboundEmail.BinaryAttachment` class.

Signature

```
public InboundEmail.BinaryAttachment()
```

InboundEmail.BinaryAttachment Properties

The following are properties for `InboundEmail.BinaryAttachment`.

IN THIS SECTION:

body
The body of the attachment.

fileName
The name of the attached file.

headers
Any header values associated with the attachment. Examples of header names include `Content-Type`, `Content-Transfer-Encoding`, and `Content-ID`.

mimeTypeSubType
The primary and sub MIME-type.

body

The body of the attachment.

Signature

```
public Blob body {get; set;}
```

Property Value

Type: Blob

fileName

The name of the attached file.

Signature

```
public String fileName {get; set;}
```

Property Value

Type: String

headers

Any header values associated with the attachment. Examples of header names include `Content-Type`, `Content-Transfer-Encoding`, and `Content-ID`.

Signature

```
public List<Messaging.InboundEmail.Header> headers {get; set;}
```

Property Value

Type: List<Messaging.InboundEmail.Header>

mimeTypeSubType

The primary and sub MIME-type.

Signature

```
public String mimeTypeSubType {get; set;}
```

Property Value

Type: String

InboundEmail.TextAttachment Class

An InboundEmail object stores text attachments in an InboundEmail.TextAttachment object.

Namespace

Messaging

Usage

The text attachments can be any of the following:

- Attachments with a Multipurpose Internet Mail Extension (MIME) type of `text`
- Attachments with a MIME type of `application/octet-stream` and a file name that ends with either a `.vcf` or `.vcs` extension. These are saved as `text/x-vcard` and `text/calendar` MIME types, respectively.

IN THIS SECTION:

InboundEmail.TextAttachment Constructors

InboundEmail.TextAttachment Properties

InboundEmail.TextAttachment Constructors

The following are constructors for `InboundEmail.TextAttachment`.

IN THIS SECTION:

InboundEmail.TextAttachment()

Creates a new instance of the `Messaging.InboundEmail.TextAttachment` class.

`InboundEmail.TextAttachment()`

Creates a new instance of the `Messaging.InboundEmail.TextAttachment` class.

Signature

```
public InboundEmail.TextAttachment()
```

InboundEmail.TextAttachment Properties

The following are properties for `InboundEmail.TextAttachment`.

IN THIS SECTION:

body

The body of the attachment.

bodyIsTruncated

Indicates whether the attachment body text is truncated (`true`) or not (`false`.)

charset
The original character set of the body field. The body is re-encoded as UTF-8 as input to the Apex method.

fileName
The name of the attached file.

headers
Any header values associated with the attachment. Examples of header names include `Content-Type`, `Content-Transfer-Encoding`, and `Content-ID`.

mimeTypeSubType
The primary and sub MIME-type.

body

The body of the attachment.

Signature

```
public String body {get; set;}
```

Property Value

Type: String

bodyIsTruncated

Indicates whether the attachment body text is truncated (`true`) or not (`false`.)

Signature

```
public Boolean bodyIsTruncated {get; set;}
```

Property Value

Type: Boolean

charset

The original character set of the body field. The body is re-encoded as UTF-8 as input to the Apex method.

Signature

```
public String charset {get; set;}
```

Property Value

Type: String

fileName

The name of the attached file.

Signature

```
public String fileName {get; set;}
```

Property Value

Type: String

`headers`

Any header values associated with the attachment. Examples of header names include `Content-Type`, `Content-Transfer-Encoding`, and `Content-ID`.

Signature

```
public List<Messaging.InboundEmail.Header> headers {get; set;}
```

Property Value

Type: List<Messaging.InboundEmail.Header>

`mimeTypeSubType`

The primary and sub MIME-type.

Signature

```
public String mimeTypeSubType {get; set;}
```

Property Value

Type: String

InboundEmailResult Class

The InboundEmailResult object is used to return the result of the email service. If this object is null, the result is assumed to be successful.

Namespace

Messaging

InboundEmailResult Properties

The following are properties for `InboundEmailResult`.

IN THIS SECTION:

message
A message that Salesforce returns in the body of a reply email. This field can be populated with text irrespective of the value returned by the `Success` field.

success
A value that indicates whether the email was successfully processed.

message

A message that Salesforce returns in the body of a reply email. This field can be populated with text irrespective of the value returned by the `Success` field.

Signature

```
public String message {get; set;}
```

Property Value

Type: String

success

A value that indicates whether the email was successfully processed.

Signature

```
public Boolean success {get; set;}
```

Property Value

Type: Boolean

Usage

If `false`, Salesforce rejects the inbound email and sends a reply email to the original sender containing the message specified in the `Message` field.

InboundEnvelope Class

The InboundEnvelope object stores the envelope information associated with the inbound email, and has the following fields.

Namespace

Messaging

InboundEnvelope Properties

The following are properties for `InboundEnvelope`.

IN THIS SECTION:

fromAddress
The name that appears in the `From` field of the envelope, if any.

toAddress
The name that appears in the `To` field of the envelope, if any.

`fromAddress`

The name that appears in the `From` field of the envelope, if any.

Signature

```
public String fromAddress {get; set;}
```

Property Value

Type: String

`toAddress`

The name that appears in the `To` field of the envelope, if any.

Signature

```
public String toAddress {get; set;}
```

Property Value

Type: String

MassEmailMessage Class

Contains methods for sending mass email.

Namespace

Messaging

Usage

MassEmailMessage extends Email and inherits all of its methods. All base email (`Email` class) methods are also available to the `MassEmailMessage` objects.

IN THIS SECTION:

MassEmailMessage Constructors

MassEmailMessage Methods

SEE ALSO:

Email Class (Base Email Methods)

MassEmailMessage Constructors

The following are constructors for `MassEmailMessage`.

IN THIS SECTION:

MassEmailMessage()
Creates a new instance of the `Messaging.MassEmailMessage` class.

`MassEmailMessage()`

Creates a new instance of the `Messaging.MassEmailMessage` class.

Signature

```
public MassEmailMessage()
```

MassEmailMessage Methods

The following are methods for `MassEmailMessage`. All are instance methods. All base email (`Email` class) methods are also available to the `MassEmailMessage` objects. These methods are described in Email Class (Base Email Methods).

IN THIS SECTION:

setDescription(description)
The description of the email.

setTargetObjectIds(targetObjectIds)
A list of IDs of the contacts, leads, or users to which the email will be sent. The IDs you specify set the context and ensure that merge fields in the template contain the correct data. The objects must be of the same type (all contacts, all leads, or all users).

setWhatIds(whatIds)
Optional. If you specify a list of contacts for the `targetObjectIds` field, you can specify a list of `whatIds` as well. This helps to further ensure that merge fields in the template contain the correct data.

`setDescription(description)`

The description of the email.

Signature

```
public Void setDescription(String description)
```

Parameters

`description`
Type: String

Return Value

Type: Void

setTargetObjectIds(targetObjectIds)

A list of IDs of the contacts, leads, or users to which the email will be sent. The IDs you specify set the context and ensure that merge fields in the template contain the correct data. The objects must be of the same type (all contacts, all leads, or all users).

Signature

```
public Void setTargetObjectIds(ID[] targetObjectIds)
```

Parameters

targetObjectIds
Type: ID[]

Return Value

Type: Void

Usage

You can list multiple IDs per email. If you specify a value for the `targetObjectIds` field, optionally specify a `whatId` as well to set the email context to a user, contact, or lead. This ensures that merge fields in the template contain the correct data. Each ID counts against the sending organization's daily mass email limit.

Do not specify the IDs of records that have the `Email Opt Out` option selected.

All emails must have a recipient value in at least one of the following fields:

- `toAddresses`
- `ccAddresses`
- `bccAddresses`
- `targetObjectId`

setWhatIds(whatIds)

Optional. If you specify a list of contacts for the `targetObjectIds` field, you can specify a list of `whatIds` as well. This helps to further ensure that merge fields in the template contain the correct data.

Signature

```
public Void setWhatIds(ID[] whatIds)
```

Parameters

whatIds
Type: ID[]

Return Value

Type: Void

Usage

The values must be one of the following types:

- Contract
- Case
- Opportunity
- Product

Note: If you specify `whatIds`, specify one for each `targetObjectId`; otherwise, you will receive an `INVALID_ID_FIELD` error.

InboundEmail.Header Class

An InboundEmail object stores RFC 2822 email header information in an InboundEmail.Header object with the following properties.

Namespace

Messaging

InboundEmail.Header Properties

The following are properties for `InboundEmail.Header`.

IN THIS SECTION:

name
The name of the header parameter, such as `Date` or `Message-ID`.

value
The value of the header.

`name`

The name of the header parameter, such as `Date` or `Message-ID`.

Signature

```
public String name {get; set;}
```

Property Value

Type: String

`value`

The value of the header.

Signature

```
public String value {get; set;}
```

Property Value

Type: String

PushNotification Class

`PushNotification` is used to configure push notifications and send them from an Apex trigger.

Namespace

Messaging

Example

This sample Apex trigger sends push notifications to the connected app named *`Test_App`*, which corresponds to a mobile app on iOS mobile clients. The trigger fires after cases have been updated and sends the push notification to two users: the case owner and the user who last modified the case.

```
trigger caseAlert on Case (after update) {

    for(Case cs : Trigger.New)
    {
        // Instantiating a notification
        Messaging.PushNotification msg =
            new Messaging.PushNotification();

        // Assembling the necessary payload parameters for Apple.
        // Apple params are:
        // (<alert text>,<alert sound>,<badge count>,
        // <free-form data>)
        // This example doesn't use badge count or free-form data.
        // The number of notifications that haven't been acted
        // upon by the intended recipient is best calculated
        // at the time of the push. This timing helps
        // ensure accuracy across multiple target devices.
        Map<String, Object> payload =
            Messaging.PushNotificationPayload.apple(
                'Case ' + cs.CaseNumber + ' status changed to: '
                + cs.Status, '', null, null);

        // Adding the assembled payload to the notification
        msg.setPayload(payload);

        // Getting recipient users
        String userId1 = cs.OwnerId;
        String userId2 = cs.LastModifiedById;

        // Adding recipient users to list
        Set<String> users = new Set<String>();
        users.add(userId1);
        users.add(userId2);

        // Sending the notification to the specified app and users.
```

```
        // Here we specify the API name of the connected app.
        msg.send('Test_App', users);
    }
}
```

IN THIS SECTION:

PushNotification Constructors

PushNotification Methods

PushNotification Constructors

The following are the constructors for `PushNotification`.

IN THIS SECTION:

PushNotification()

Creates a new instance of the `Messaging.PushNotification` class.

PushNotification(payload)

Creates a new instance of the `Messaging.PushNotification` class using the specified payload parameters as key-value pairs. When you use this constructor, you don't need to call `setPayload` to set the payload.

`PushNotification()`

Creates a new instance of the `Messaging.PushNotification` class.

Signature

```
public PushNotification()
```

`PushNotification(payload)`

Creates a new instance of the `Messaging.PushNotification` class using the specified payload parameters as key-value pairs. When you use this constructor, you don't need to call `setPayload` to set the payload.

Signature

```
public PushNotification(Map<String,Object> payload)
```

Parameters

payload

Type:Map<String, Object>

The payload, expressed as a map of key-value pairs.

PushNotification Methods

The following are the methods for `PushNotification`. All are global methods.

IN THIS SECTION:

send(application, users)
Sends a push notification message to the specified users.

setPayload(payload)
Sets the payload of the push notification message.

setTtl(ttl)
Reserved for future use.

send(application, users)

Sends a push notification message to the specified users.

Signature

```
public void send(String application, Set<String> users)
```

Parameters

application
Type: String

The connected app API name. This corresponds to the mobile client app the notification should be sent to.

users
Type: Set

A set of user IDs that correspond to the users the notification should be sent to.

Example

See the Push Notification Example.

setPayload(payload)

Sets the payload of the push notification message.

Signature

```
public void setPayload(Map<String,Object> payload)
```

Parameters

payload
Type: Map<String, Object>

The payload, expressed as a map of key-value pairs.

Payload parameters can be different for each mobile OS vendor. For more information on Apple's payload parameters, search for "Apple Push Notification Service" at https://developer.apple.com/library/mac/documentation/.

To create the payload for an Apple device, see the PushNotificationPayload Class.

Example

See the Push Notification Example.

`setTtl(ttl)`

Reserved for future use.

Signature

```
public void setTtl(Integer ttl)
```

Parameters

`ttl`
Type: Integer

Reserved for future use.

`PushNotificationPayload Class`

Contains methods to create the notification message payload for an Apple device.

Namespace

Messaging

Usage

Apple has specific requirements for the notification payload. and this class has helper methods to create the payload. For more information on Apple's payload parameters, search for "Apple Push Notification Service" at https://developer.apple.com/library/mac/documentation/.

Example

See the Push Notification Example.

IN THIS SECTION:

PushNotificationPayload Methods

PushNotificationPayload Methods

The following are the methods for `PushNotificationPayload`. All are global static methods.

IN THIS SECTION:

apple(alert, sound, badgeCount, userData)
Helper method that creates a valid Apple payload from the specified arguments.

apple(alertBody, actionLocKey, locKey, locArgs, launchImage, sound, badgeCount, userData)
Helper method that creates a valid Apple payload from the specified arguments.

apple(alert, sound, badgeCount, userData)

Helper method that creates a valid Apple payload from the specified arguments.

Signature

```
public static Map<String,Object> apple(String alert, String sound, Integer badgeCount,
Map<String,Object> userData)
```

Parameters

alert
Type: String

Notification message to be sent to the mobile client.

sound
Type: String

Name of a sound file to be played as an alert. This sound file should be in the mobile application bundle.

badgeCount
Type: Integer

Number to display as the badge of the application icon.

userData
Type: Map<String, Object>

Map of key-value pairs that contains any additional data used to provide context for the notification. For example, it can contain IDs of the records that caused the notification to be sent. The mobile client app can use these IDs to display these records.

Return Value

Type:Map<String, Object>

Returns a formatted payload that includes all of the specified arguments.

Usage

To generate a valid payload, you must provide a value for at least one of the following parameters: `alert`, `sound`, `badgeCount`.

Example

See the Push Notification Example.

apple(alertBody, actionLocKey, locKey, locArgs, launchImage, sound, badgeCount, userData)

Helper method that creates a valid Apple payload from the specified arguments.

Signature

```
public static Map<String,Object> apple(String alertBody, String actionLocKey, String
locKey, String[] locArgs, String launchImage, String sound, Integer badgeCount,
Map<String,Object> userData)
```

Parameters

alertBody
Type: String

Text of the alert message.

actionLocKey
Type: String

If a value is specified for the *actionLocKey* argument, an alert with two buttons is displayed. The value is a key to get a localized string in a `Localizable.strings` file to use for the right button's title.

locKey
Type: String

Key to an alert-message string in a `Localizable.strings` file for the current localization.

locArgs
Type: List<String>

Variable string values to appear in place of the format specifiers in *locKey*.

launchImage
Type: String

File name of an image file in the application bundle.

sound
Type: String

Name of a sound file to be played as an alert. This sound file should be in the mobile application bundle.

badgeCount
Type: Integer

Number to display as the badge of the application icon.

userData
Type: Map<String, Object>

Map of key-value pairs that contains any additional data used to provide context for the notification. For example, it can contain IDs of the records that caused the notification to be sent. The mobile client app can use these IDs to display these records.

Return Value

Type: Map<String, Object>

Returns a formatted payload that includes all of the specified arguments.

Usage

To generate a valid payload, you must provide a value for at least one of the following parameters: `alert`, `sound`, `badgeCount`.

CustomNotification Class

`CustomNotification` is used to create, configure, and send custom notifications from Apex code.

Namespace

Messaging

Usage

`CustomNotification` allows two approaches to creating and configuring a custom notification.

- Create an instance with the default constructor, and then set notification attributes using the various setter methods.
- Create an instance and configure notification parameters at the same time using the parameterized constructor.

Once the custom notification is configured, call `send()` to send the notification.

Notification Target

The *notification target* is used by the receiving client application to navigate to an appropriate record or page when a user responds to a notification. For example, when a user is notified that a record was updated, responding to the notification can open the relevant record.

You must specify a target for a notification. The target can be specified using either the `targetID` or the `targetPageRef` attribute. Neither attribute is required, but if both are omitted, `send()` throws an exception. If there's no natural target for a notification, set the `targetID` to a dummy value, such as *`000000000000000AAA`*. A dummy value prevents the exception, and also prevents automatic navigation when responding to the notification in the client app.

You can set both `targetID` and `targetPageRef` in the same notification. The client app that receives the notification determines which target, if any, to use when responding to the notification.

Important: Before Winter '21 you could set only a target record (`targetID`) for a notification. Most client applications expect to find a `targetID` in the notification payload. If you can't update a client app to handle notifications that include only a `targetPageRef`, set the `targetID` to a dummy value.

Execution Context and Notification Permissions

By default Apex code executes in system mode, and doesn't require user permissions to send notifications with `CustomNotification`. However, if your Apex code runs in a user context—for example, by executing anonymous Apex in the Developer Console—the Send Custom Notifications user permission is checked, and `send()` fails if you don't have the required permission.

Example

This example Apex class provides a static method for sending a custom notification to a recipient list. Call this method from a trigger, flow, or wherever you want to send a custom notification from Apex.

```
public without sharing class CustomNotificationFromApex {

    public static void notifyUsers(Set<String> recipientsIds, String targetId) {

        // Get the Id for our custom notification type
        CustomNotificationType notificationType =
            [SELECT Id, DeveloperName
             FROM CustomNotificationType
             WHERE DeveloperName='Custom_Notification'];

        // Create a new custom notification
        Messaging.CustomNotification notification = new Messaging.CustomNotification();

        // Set the contents for the notification
```

```
        notification.setTitle('Apex Custom Notification');
        notification.setBody('The notifications are coming from INSIDE the Apex!');

        // Set the notification type and target
        notification.setNotificationTypeId(notificationType.Id);
        notification.setTargetId(targetId);

        // Actually send the notification
        try {
            notification.send(recipientsIds);
        }
        catch (Exception e) {
            System.debug('Problem sending notification: ' + e.getMessage());
        }
    }
}
```

Note: This example uses a custom notification type with the `DeveloperName` (API name) *`Custom_Notification`*. You can create a custom notification type using Notification Builder in Setup or Tooling API. Then, use your notification type's `DeveloperName` (API name) in the query to find the ID of the notification type.

`CustomNotification.send()` can throw an exception, which is handled minimally in this example. Add more substantial error handling to code you plan to use in production.

IN THIS SECTION:

CustomNotification Constructors

CustomNotification Methods

SEE ALSO:

Salesforce Help: Send Custom Notifications

Actions Developer Guide: Custom Notification Actions

Metadata API Developer Guide: CustomNotificationType

CustomNotification Constructors

The following are constructors for `CustomNotification`.

IN THIS SECTION:

CustomNotification()

Creates a new instance of the `Messaging.CustomNotification` class.

CustomNotification(typeId, sender, title, body, targetId, targetPageRef)

Creates an instance of the `Messaging.CustomNotification` class using the specified parameters. When you use this constructor, you don't need to call the various setter methods to define the custom notification attributes.

`CustomNotification()`

Creates a new instance of the `Messaging.CustomNotification` class.

Signature

```
public CustomNotification()
```

`CustomNotification(typeId, sender, title, body, targetId, targetPageRef)`

Creates an instance of the `Messaging.CustomNotification` class using the specified parameters. When you use this constructor, you don't need to call the various setter methods to define the custom notification attributes.

Signature

```
public CustomNotification(String typeId, String sender, String title, String body,
String targetId, String targetPageRef)
```

Parameters

typeId
Type: String

The ID of the Custom Notification Type being used for the notification.

sender
Type: String

The User ID of the sender of the notification.

title
Type: String

The title of the notification. Maximum characters: 250.

body
Type: String

The body of the notification. Maximum characters: 750.

targetId
Type: String

The Record ID for the target record of the notification.

You must specify either a `targetID` or a `targetPageRef`. See Custom Notification Usage.

targetPageRef
Type: String

The `PageReference` for the navigation target of the notification. To see how to specify the target using JSON, see pageReference Types.

You must specify either a `targetID` or a `targetPageRe`. See Custom Notification Usage.

Usage

A client may see a truncated notification title or body depending on the delivery channel or app, and how the Connect API notification parameters are configured. For more information on the `trimMessages` query parameter, see Notification .

CustomNotification Methods

The following are methods for `CustomNotification`.

IN THIS SECTION:

send(users)
Sends a custom notification to the specified users.

setNotificationTypeId(id)
Sets the type of the custom notification.

setTitle(title)
Sets the title of the custom notification.

setBody(body)
Sets the body of the custom notification.

setSenderId(id)
Sets the sender of the custom notification.

setTargetId(targetId)
Sets the target record of the custom notification.

setTargetPageRef(pageRef)
Sets the target page of the custom notification.

send(users)

Sends a custom notification to the specified users.

Signature

```
public void send(Set<String> users)
```

Parameters

users
Type: Set<String>

Required. A set of recipient IDs. Each recipient ID corresponds to a recipient or recipient type that the notification should be sent to. Valid recipient or recipient type values are:

- `UserId` — The notification is sent to this user, if this user is active.
- `AccountId` — The notification is sent to all active users who are members of this account's Account Team.

  Note: This recipient type is valid if account teams are enabled for your org.

- `OpportunityId` — The notification is sent to all active users who are members of this opportunity's Opportunity Team.

  Note: This recipient type is valid if team selling is enabled for your org.

- `GroupId` — The notification is sent to all active users who are members of this group.
- `QueueId` — The notification is sent to all active users who are members of this queue.

Values can be combined in a set, up to the maximum of 500 values.

Return Value

Type: void

Example

See the Custom Notification Example.

setNotificationTypeId(id)

Sets the type of the custom notification.

Signature

```
public void setNotificationTypeId(String id)
```

Parameters

id

Type: String

The ID of the Custom Notification Type being used for the notification.

A notification type is required to send a custom notification. See Custom Notification Usage.

Return Value

Type: void

Example

See the Custom Notification Example.

setTitle(title)

Sets the title of the custom notification.

Signature

```
public void setTitle(String title)
```

Parameters

title

Type: String

The title of the notification, as it will be seen by recipients. Maximum characters: 250.

A title is required to send a custom notification. See Custom Notification Usage.

Return Value

Type: void

Example

See the Custom Notification Example.

`setBody(body)`

Sets the body of the custom notification.

Signature

```
public void setBody(String body)
```

Parameters

body
Type: String

The body of the notification, as it will be seen by recipients. Maximum characters: 750.

A body is required to send a custom notification. See Custom Notification Usage.

Return Value

Type: void

Example

See the Custom Notification Example.

`setSenderId(id)`

Sets the sender of the custom notification.

Signature

```
public void setSenderId(String id)
```

Parameters

id
Type: String

The User ID of the sender of the notification.

Setting a sender is optional. See Custom Notification Usage.

Return Value

Type: void

Example

See the Custom Notification Example.

`setTargetId(targetId)`

Sets the target record of the custom notification.

Signature

```
public void setTargetId(String targetId)
```

Parameters

targetId
Type: String

The Record ID for the target record of the notification.

Either a `targetID` or a `targetPageRef` is required to send a custom notification. See Custom Notification Usage.

Return Value

Type: void

Example

See the Custom Notification Example.

`setTargetPageRef(pageRef)`

Sets the target page of the custom notification.

Signature

```
public void setTargetPageRef(String pageRef)
```

Parameters

pageRef
Type: String

The `PageReference` for the navigation target of the notification.

Either a `targetID` or a `targetPageRef` is required to send a custom notification. See Custom Notification Usage.

Return Value

Type: void

Example

See the Custom Notification Example.

RenderEmailTemplateBodyResult Class

Contains the results for rendering email templates.

Namespace

Messaging

IN THIS SECTION:

RenderEmailTemplateBodyResult Methods

RenderEmailTemplateBodyResult Methods

The following are methods for `RenderEmailTemplateBodyResult`.

IN THIS SECTION:

getErrors()
If an error occurred during the `renderEmailTemplate` method, a `RenderEmailTemplateError` object is returned.

getMergedBody()
Returns the rendered body text with merge field references replaced with the corresponding record data.

getSuccess()
Indicates whether the operation was successful.

`getErrors()`

If an error occurred during the `renderEmailTemplate` method, a `RenderEmailTemplateError` object is returned.

Signature

```
public List<Messaging.RenderEmailTemplateError> getErrors()
```

Return Value

Type: List<Messaging.RenderEmailTemplateError>

`getMergedBody()`

Returns the rendered body text with merge field references replaced with the corresponding record data.

Signature

```
public String getMergedBody()
```

Return Value

Type: String

`getSuccess()`

Indicates whether the operation was successful.

Signature

```
public Boolean getSuccess()
```

Return Value

Type: Boolean

RenderEmailTemplateError Class

Represents an error that the `RenderEmailTemplateBodyResult` object can contain.

Namespace

Messaging

IN THIS SECTION:

RenderEmailTemplateError Methods

RenderEmailTemplateError Methods

The following are methods for `RenderEmailTemplateError`.

IN THIS SECTION:

getFieldName()
Returns the name of the merge field in the error.

getMessage()
Returns a message describing the error.

getOffset()
Returns the offset within the supplied body text where the error was discovered. If the offset cannot be determined, -1 is returned.

getStatusCode()
Returns a Salesforce API status code.

`getFieldName()`

Returns the name of the merge field in the error.

Signature

```
public String getFieldName()
```

Return Value

Type: String

`getMessage()`

Returns a message describing the error.

Signature

```
public String getMessage()
```

Return Value

Type: String

`getOffset()`

Returns the offset within the supplied body text where the error was discovered. If the offset cannot be determined, -1 is returned.

Signature

```
public Integer getOffset()
```

Return Value

Type: Integer

`getStatusCode()`

Returns a Salesforce API status code.

Signature

```
public System.StatusCode getStatusCode()
```

Return Value

Type: System.StatusCode

SendEmailError Class

Represents an error that the SendEmailResult object may contain.

Namespace

Messaging

SendEmailError Methods

The following are methods for `SendEmailError`. All are instance methods.

IN THIS SECTION:

getFields()
A list of one or more field names. Identifies which fields in the object, if any, affected the error condition.

getMessage()
The text of the error message.

getStatusCode()
Returns a code that characterizes the error.

getTargetObjectId()
The ID of the target record for which the error occurred.

getFields()

A list of one or more field names. Identifies which fields in the object, if any, affected the error condition.

Signature

```
public String[] getFields()
```

Return Value

Type: String[]

getMessage()

The text of the error message.

Signature

```
public String getMessage()
```

Return Value

Type: String

getStatusCode()

Returns a code that characterizes the error.

Signature

```
public System.StatusCode getStatusCode()
```

Return Value

Type: System.StatusCode

Usage

The full list of status codes is available in the WSDL file for your organization. For more information about accessing the WSDL file for your organization, see *Downloading Salesforce WSDLs and Client Authentication Certificates* in the Salesforce online help.

getTargetObjectId()

The ID of the target record for which the error occurred.

Signature

```
public String getTargetObjectId()
```

Return Value

Type: String

SendEmailResult Class

Contains the result of sending an email message.

Namespace

Messaging

SendEmailResult Methods

The following are methods for `SendEmailResult`. All are instance methods.

IN THIS SECTION:

getErrors()
If an error occurred during the `sendEmail` method, a `SendEmailError` object is returned.

isSuccess()
Indicates whether the email was successfully submitted for delivery (`true`) or not (`false`). Even if `isSuccess` is true, it does not mean the intended recipients received the email, as there could have been a problem with the email address or it could have bounced or been blocked by a spam blocker.

`getErrors()`

If an error occurred during the `sendEmail` method, a `SendEmailError` object is returned.

Signature

```
public SendEmailError[] getErrors()
```

Return Value

Type: Messaging.SendEmailError[]

`isSuccess()`

Indicates whether the email was successfully submitted for delivery (`true`) or not (`false`). Even if `isSuccess` is true, it does not mean the intended recipients received the email, as there could have been a problem with the email address or it could have bounced or been blocked by a spam blocker.

Signature

```
public Boolean isSuccess()
```

Return Value

Type: Boolean

SingleEmailMessage Class

Contains methods for sending single email messages.

Namespace

Messaging

Usage

SingleEmailMessage extends Email and inherits all of its methods. All base email (`Email` class) methods are also available to the `SingleEmailMessage` objects. Emails sent via SingleEmailMessage count against the sending organization's daily single email limit.

Email properties are readable and writable. Each property has corresponding setter and getter methods. For example, the `toAddresses()` property is equivalent to the `setToAddresses()` and `getToAddresses()` methods. Only the setter methods are documented. However, the `getTemplateName()` method doesn't have an equivalent setter method; use `setTemplateId()` to specify a template name.

IN THIS SECTION:

SingleEmailMessage Constructors

SingleEmailMessage Methods

SEE ALSO:

Email Class (Base Email Methods)

SingleEmailMessage Constructors

The following are constructors for `SingleEmailMessage`.

IN THIS SECTION:

SingleEmailMessage()
Creates a new instance of the `Messaging.SingleEmailMessage` class.

`SingleEmailMessage()`

Creates a new instance of the `Messaging.SingleEmailMessage` class.

Signature

```
public SingleEmailMessage()
```

SingleEmailMessage Methods

The following are methods for `SingleEmailMessage`. All are instance methods. All base email (`Email` class) methods are also available to the `SingleEmailMessage` objects. These methods are described in Email Class (Base Email Methods).

IN THIS SECTION:

getOneClickPost()

Optional. Returns a boolean value based on the value set by the `setOneClickPost` method. Default is `false`.

getTemplateName()

The name of the template used to create the email.

setBccAddresses(bccAddresses)

Optional. A list of blind carbon copy (BCC) addresses or object IDs of the contacts, leads, and users you're sending the email to. The maximum size for this field is 4,000 bytes. The maximum total of `toAddresses`, `ccAddresses`, and `bccAddresses` per email is 150. All recipients in these three fields count against the limit for email sent using Apex or the API.

setCcAddresses(ccAddresses)

Optional. A list of carbon copy (CC) addresses or object IDs of the contacts, leads, and users you're sending the email to. The maximum size for this field is 4,000 bytes. The maximum total of `toAddresses`, `ccAddresses`, and `bccAddresses` per email is 150. All recipients in these three fields count against the limit for email sent using Apex or the API.

setCharset(characterSet)

Optional. The character set for the email. If this value is null, the user's default value is used.

setDocumentAttachments(documentIds)

(Deprecated. Use `setEntityAttachments()` instead.) Optional. A list containing the ID of each document object you want to attach to the email.

setEntityAttachments(ids)

Optional. Array of IDs of Document, ContentVersion, or Attachment items to attach to the email.

setFileAttachments(fileNames)

Optional. A list containing the file names of the binary and text files you want to attach to the email.

setHtmlBody(htmlBody)

Optional. The HTML version of the email, specified by the sender. The value is encoded according to the specification associated with the organization. Specify a value for `setTemplateId`, `setHtmlBody`, or `setPlainTextBody`. Or, you can define both `setHtmlBody` and `setPlainTextBody`.

setInReplyTo(parentMessageIds)

Sets the optional In-Reply-To field of the outgoing email. This field identifies the email or emails to which this email is a reply (parent emails).

setOneClickPost(oneClickPost)

Optional. If set to true, a List-Unsubscribe-Post header is added to an email with List-Unsubscribe=One-Click. Use this method to support unsubscribe functionality in email sent via Salesforce. You can provide additional instructions on how to send unsubscribe requests by using the header. This includes specifying the HTTP method and content type to use and provides a secure way to add more info to unsubscribe requests. Default is `false`.

setOptOutPolicy(emailOptOutPolicy)

Optional. If you added recipients by ID instead of email address and the `Email Opt Out` option is set, this method determines the behavior of the `sendEmail()` call. If you add recipients by their email addresses, the opt-out settings for those recipients aren't checked and those recipients always receive the email.

setPlainTextBody(plainTextBody)

Optional. The text version of the email, specified by the sender. Specify a value for `setTemplateId`, `setHtmlBody`, or `setPlainTextBody`. Or, you can define both `setHtmlBody` and `setPlainTextBody`.

setOrgWideEmailAddressId(emailAddressId)

Optional. The ID of the organization-wide email address associated with the outgoing email. If you're using Apex to send emails from the guest user, set the sender to the verified org-wide email address or the emails are blocked. The object's `DisplayName` field cannot be set if the `setSenderDisplayName` field is already set.

setReferences(references)

Optional. The References field of the outgoing email. Identifies an email thread. Contains the parent emails' References and message IDs, and possibly the In-Reply-To fields.

setSubject(subject)

Optional. The email subject line. If you are using an email template, the subject line of the template overrides this value.

setTargetObjectId(targetObjectId)

Required if using a template, optional otherwise. The ID of the contact, lead, or user to which the email will be sent. The ID you specify sets the context and ensures that merge fields in the template contain the correct data.

setTemplateId(templateId)

Required if using a template, optional otherwise. The ID of the template used to create the email.

setToAddresses(toAddresses)

Optional. A list of email addresses or object IDs of the contacts, leads, and users you're sending the email to. The maximum size for this field is 4,000 bytes. The maximum total of `toAddresses`, `ccAddresses`, and `bccAddresses` per email is 150. All recipients in these three fields count against the limit for email sent using Apex or the API.

setTreatBodiesAsTemplate(treatAsTemplate)

Optional. If set to `true`, the subject, plain text, and HTML text bodies of the email are treated as template data. The merge fields are resolved using the `renderEmailTemplate` method. Default is `false`.

setTreatTargetObjectAsRecipient(treatAsRecipient)

Optional. If set to `true`, the `targetObjectId` (a contact, lead, or user) is the recipient of the email. If set to `false`, the `targetObjectId` is supplied as the `WhoId` field for template rendering but isn't a recipient of the email. The default is `true`.

setUnsubscribeComment(unsubscribeComment)

Optional. Sets a comment in the List-Unsubscribe email header. This comment is ignored by email clients and systems that parse the header. The comments contain human-readable notes or context for developers, administrators, or other stakeholders managing the email system.

setUnsubscribeUrls(UnsubscribeUrls)

Optional. Sets a `mailto` URI and HTTP URL of a mechanism for unsubscribing a recipient from an email list. A list of all unsubscribe URLs passed through `setUnsubscribeUrls` is added to the `List-Unsubscribe` header. A minimum of one URL is required to use this method.

setWhatId(whatId)

If you specify a contact for the `targetObjectId` field, you can specify an optional `whatId` as well. This helps to further ensure that merge fields in the template contain the correct data.

`getOneClickPost()`

Optional. Returns a boolean value based on the value set by the `setOneClickPost` method. Default is `false`.

Signature

```
public Boolean getOneClickPost()
```

Parameters

Type: Boolean

Return Value

Type: Boolean

Usage

Invoke the `setOneClickPost`method before using `getOneClickPost`. The value of `getOneClickPost` will be false if the `setOneClickPost`method is set to true only after invoking the `setUnsubscribeUrls` method.

`getTemplateName()`

The name of the template used to create the email.

Signature

```
public STRING getTemplateName()
```

Return Value

Type: String

Usage

There is no equivalent setter method for `getTemplateName()`. If the email didn't use a template, `getTemplateName()` returns nothing. If you use `setTemplateId()`, and then call `getTemplateName()`, the template name associated to the template ID is returned.

`setBccAddresses(bccAddresses)`

Optional. A list of blind carbon copy (BCC) addresses or object IDs of the contacts, leads, and users you're sending the email to. The maximum size for this field is 4,000 bytes. The maximum total of `toAddresses`, `ccAddresses`, and `bccAddresses` per email is 150. All recipients in these three fields count against the limit for email sent using Apex or the API.

Signature

```
public Void setBccAddresses(String[] bccAddresses)
```

Parameters

`bccAddresses`
Type: String[]

Return Value

Type: Void

Usage

All emails must have a recipient value in at least one of the following fields:

- `toAddresses`
- `ccAddresses`
- `bccAddresses`
- `targetObjectId`

If the BCC compliance option is set at the organization level, the user cannot add BCC addresses on standard messages. The following error code is returned: `BCC_NOT_ALLOWED_IF_BCC_ COMPLIANCE_ENABLED`. Contact your Salesforce representative for information on BCC compliance.

`setCcAddresses(ccAddresses)`

Optional. A list of carbon copy (CC) addresses or object IDs of the contacts, leads, and users you're sending the email to. The maximum size for this field is 4,000 bytes. The maximum total of `toAddresses`, `ccAddresses`, and `bccAddresses` per email is 150. All recipients in these three fields count against the limit for email sent using Apex or the API.

Signature

```
public Void setCcAddresses(String[] ccAddresses)
```

Parameters

`ccAddresses`
Type: String[]

Return Value

Type: Void

Usage

All emails must have a recipient value in at least one of the following fields:

- `toAddresses`
- `ccAddresses`
- `bccAddresses`
- `targetObjectId`

`setCharset(characterSet)`

Optional. The character set for the email. If this value is null, the user's default value is used.

Signature

```
public Void setCharset(String characterSet)
```

Parameters

characterSet
Type: String

Return Value

Type: Void

setDocumentAttachments(documentIds)

(Deprecated. Use `setEntityAttachments()` **instead.)** Optional. A list containing the ID of each document object you want to attach to the email.

Signature

```
public Void setDocumentAttachments(ID[] documentIds)
```

Parameters

documentIds
Type: ID[]

Return Value

Type: Void

Usage

You can attach multiple documents as long as the total size of all attachments does not exceed 10 MB.

setEntityAttachments(ids)

Optional. Array of IDs of Document, ContentVersion, or Attachment items to attach to the email.

Signature

```
public void setEntityAttachments(List<String> ids)
```

Parameters

ids
Type: List<String>

Return Value

Type: void

`setFileAttachments(fileNames)`

Optional. A list containing the file names of the binary and text files you want to attach to the email.

Signature

```
public Void setFileAttachments(EmailFileAttachment[] fileNames)
```

Parameters

`fileNames`
Type: Messaging.EmailFileAttachment[]

Return Value

Type: Void

Usage

You can attach multiple files as long as the total size of all attachments does not exceed 10 MB.

`setHtmlBody(htmlBody)`

Optional. The HTML version of the email, specified by the sender. The value is encoded according to the specification associated with the organization. Specify a value for `setTemplateId`, `setHtmlBody`, or `setPlainTextBody`. Or, you can define both `setHtmlBody` and `setPlainTextBody`.

Signature

```
public Void setHtmlBody(String htmlBody)
```

Parameters

`htmlBody`
Type: String

Return Value

Type: Void

`setInReplyTo(parentMessageIds)`

Sets the optional In-Reply-To field of the outgoing email. This field identifies the email or emails to which this email is a reply (parent emails).

Signature

```
public Void setInReplyTo(String parentMessageIds)
```

Parameters

parentMessageIds
Type: String

Contains one or more parent email message IDs.

Return Value

Type: Void

setOneClickPost(oneClickPost)

Optional. If set to true, a List-Unsubscribe-Post header is added to an email with List-Unsubscribe=One-Click. Use this method to support unsubscribe functionality in email sent via Salesforce. You can provide additional instructions on how to send unsubscribe requests by using the header. This includes specifying the HTTP method and content type to use and provides a secure way to add more info to unsubscribe requests. Default is `false`.

Signature

```
public void setOneClickPost(Boolean oneClickPost)
```

Parameters

oneClickPost
Type: Boolean

Return Value

Type: void

Usage

You can set the `oneClickPost` method to true only after invoking the `setUnsubscribeUrls` method. If set to true, pass at least one HTTPS unsubscribe URL to unsubscribe.

Example

This example demonstrates how to send an email using Salesforce's `Messaging.SingleEmailMessage` class with enhanced unsubscribe functionality. It creates an email message with a recipient, subject, and body, and includes an unsubscribe URL. It also enables the `oneClickPost` feature, allowing for a simplified unsubscribe process. The email message is added to a list and sent using the `Messaging.sendEmail` method.

```
Messaging.SingleEmailMessage message = new Messaging.SingleEmailMessage();
// Set the recipient's email address
// Replace IDs with valid record IDs in your org.
message.toAddresses = new String[] { '003D000000QDexS' };
message.subject = 'Test Message';
message.plainTextBody = 'This is the message body.';


// Create a list to hold unsubscribe URLs
List<String> unsubscribeUrls = new List<String>();
```

```
unsubscribeUrls.add('https://example.com/unsubscribe.html?opaque=123456789');


// Assign the unsubscribe URLs to the email message
message.unsubscribeUrls = unsubscribeUrls;


// Enable the one-click unsubscribe feature
message.oneClickPost = true;

Messaging.SingleEmailMessage[] messages =
    new List<Messaging.SingleEmailMessage> {message};
Messaging.SendEmailResult[] results = Messaging.sendEmail(messages);


if (results[0].success) {
    System.debug('The email was sent successfully.');
} else {
    System.debug('The email failed to send: '
          + results[0].errors[0].message);
}
```

setOptOutPolicy(emailOptOutPolicy)

Optional. If you added recipients by ID instead of email address and the `Email Opt Out` option is set, this method determines the behavior of the `sendEmail()` call. If you add recipients by their email addresses, the opt-out settings for those recipients aren't checked and those recipients always receive the email.

Signature

```
public void setOptOutPolicy(String emailOptOutPolicy)
```

Parameters

`emailOptOutPolicy`
Type: String

Possible values of the *`emailOptOutPolicy`* parameter are:

- `SEND` (default)—The email is sent to all recipients. The recipients' `Email Opt Out` setting is ignored. The setting Enforce email privacy settings is ignored.
- `FILTER`—No email is sent to recipients that have the `Email Opt Out` option set. Emails are sent to the other recipients. The setting Enforce email privacy settings is ignored.
- `REJECT`—If any of the recipients have the `Email Opt Out` option set, `sendEmail()` throws an error and no email is sent. The setting Enforce email privacy settings is respected, as are the selections in the data privacy record based on the Individual object. If any of the recipients have Don't Market, Don't Process, or Forget This Individual selected, `sendEmail()` throws an error and no email is sent.

Return Value

Type: void

Example

This example shows how to send an email with the opt-out setting enforced. Recipients are specified by their IDs. The `FILTER` option causes the email to be sent only to recipients that haven't opted out from email. This example uses dot notation of the email properties, which is equivalent to using the set methods.

```
Messaging.SingleEmailMessage message = new Messaging.SingleEmailMessage();
// Set recipients to two contact IDs.
// Replace IDs with valid record IDs in your org.
message.toAddresses = new String[] { '003D000000QDexS', '003D000000QDfW5' };
message.optOutPolicy = 'FILTER';
message.subject = 'Opt Out Test Message';
message.plainTextBody = 'This is the message body.';
Messaging.SingleEmailMessage[] messages =
    new List<Messaging.SingleEmailMessage> {message};
        Messaging.SendEmailResult[] results = Messaging.sendEmail(messages);
if (results[0].success) {
    System.debug('The email was sent successfully.');
} else {
    System.debug('The email failed to send: '
          + results[0].errors[0].message);
}
```

setPlainTextBody(plainTextBody)

Optional. The text version of the email, specified by the sender. Specify a value for `setTemplateId`, `setHtmlBody`, or `setPlainTextBody`. Or, you can define both `setHtmlBody` and `setPlainTextBody`.

Signature

```
public Void setPlainTextBody(String plainTextBody)
```

Parameters

plainTextBody
Type: String

Return Value

Type: Void

setOrgWideEmailAddressId(emailAddressId)

Optional. The ID of the organization-wide email address associated with the outgoing email. If you're using Apex to send emails from the guest user, set the sender to the verified org-wide email address or the emails are blocked. The object's `DisplayName` field cannot be set if the `setSenderDisplayName` field is already set.

Signature

```
public Void setOrgWideEmailAddressId(ID emailAddressId)
```

Parameters

emailAddressId
Type: ID

Usage

After you create an org-wide email address, you're sent a confirmation email to verify it. Copy the Id from the URL and use the *setOrgWideEmailAddressId(Id)* method on your instance of *Messaging.SingleEmailMessage*.

To avoid hard-coding an ID, after creating your org-wide email address, you can query them.

```
OrgWideEmailAddress[] owea = [select Id from OrgWideEmailAddress where Address =
'doNotReply@<somedomain>.com'];
Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
if ( owea.size() > 0 ) {
    mail.setOrgWideEmailAddressId(owea.get(0).Id);
}
```

Return Value

Type: Void

setReferences(references)

Optional. The References field of the outgoing email. Identifies an email thread. Contains the parent emails' References and message IDs, and possibly the In-Reply-To fields.

Signature

```
public Void setReferences(String references)
```

Parameters

references
Type: String

Return Value

Type: Void

setSubject(subject)

Optional. The email subject line. If you are using an email template, the subject line of the template overrides this value.

Signature

```
public Void setSubject(String subject)
```

Parameters

subject
Type: String

Return Value

Type: Void

setTargetObjectId(targetObjectId)

Required if using a template, optional otherwise. The ID of the contact, lead, or user to which the email will be sent. The ID you specify sets the context and ensures that merge fields in the template contain the correct data.

Signature

```
public Void setTargetObjectId(ID targetObjectId)
```

Parameters

targetObjectId
Type: ID

Return Value

Type: Void

Usage

Do not specify the IDs of records that have the `Email Opt Out` option selected.

All emails must have a recipient value in at least one of the following fields:

- `toAddresses`
- `ccAddresses`
- `bccAddresses`
- `targetObjectId`

setTemplateId(templateId)

Required if using a template, optional otherwise. The ID of the template used to create the email.

Signature

```
public Void setTemplateId(ID templateId)
```

Parameters

templateId
Type: ID

Return Value

Type: Void

setToAddresses(toAddresses)

Optional. A list of email addresses or object IDs of the contacts, leads, and users you're sending the email to. The maximum size for this field is 4,000 bytes. The maximum total of `toAddresses`, `ccAddresses`, and `bccAddresses` per email is 150. All recipients in these three fields count against the limit for email sent using Apex or the API.

Signature

```
public Void setToAddresses(String[] toAddresses)
```

Parameters

toAddresses
Type: String[]

Return Value

Type: Void

Usage

All emails must have a recipient value in at least one of the following fields:

- `toAddresses`
- `ccAddresses`
- `bccAddresses`
- `targetObjectId`

setTreatBodiesAsTemplate(treatAsTemplate)

Optional. If set to `true`, the subject, plain text, and HTML text bodies of the email are treated as template data. The merge fields are resolved using the `renderEmailTemplate` method. Default is `false`.

Signature

```
public void setTreatBodiesAsTemplate(Boolean treatAsTemplate)
```

Parameters

treatAsTemplate
Type: Boolean

Return Value

Type: void

setTreatTargetObjectAsRecipient(treatAsRecipient)

Optional. If set to `true`, the `targetObjectId` (a contact, lead, or user) is the recipient of the email. If set to `false`, the `targetObjectId` is supplied as the `WhoId` field for template rendering but isn't a recipient of the email. The default is `true`.

Signature

```
public void setTreatTargetObjectAsRecipient(Boolean treatAsRecipient)
```

Parameters

treatAsRecipient
Type: Boolean

Return Value

Type: void

Usage

Note: You can set TO, CC, and BCC addresses using the email messaging methods regardless of whether a template is used for the email or the target object is a recipient.

setUnsubscribeComment(unsubscribeComment)

Optional. Sets a comment in the List-Unsubscribe email header. This comment is ignored by email clients and systems that parse the header. The comments contain human-readable notes or context for developers, administrators, or other stakeholders managing the email system.

Signature

```
public void setUnsubscribeComment(String unsubscribeComment)
```

Parameters

unsubscribeComment
Type: String

Return Value

Type: void

Usage

Invoke the `setUnsubscribeUrls` method before using `setUnsubscribeComment`.

Example

This example demonstrates how to send an email using Salesforce's `Messaging.SingleEmailMessage` class with an option to include an unsubscribe link. It creates an email message with a recipient, subject, and body, and includes an unsubscribe URL that directs the recipient to send an unsubscribe request via email. Additionally, it sets an `unsubscribeComment` to provide context for the unsubscribe action.

```
Messaging.SingleEmailMessage message = new Messaging.SingleEmailMessage();
// Set the recipient's email address
// Replace IDs with valid record IDs in your org
message.toAddresses = new String[] { '003D000000QDexS' };
```

```
message.subject = 'Test Message';
message.plainTextBody = 'This is the message body.';


// Create a list to hold unsubscribe URLs
List<String> unsubscribeUrls = new List<String>();
unsubscribeUrls.add('mailto:listrequest@example.com?subject=unsubscribe');


// Assign the unsubscribe URLs to the email message
message.unsubscribeUrls = unsubscribeUrls;


// Set an unsubscribe comment to provide context for the unsubscribe action
message.unsubscribeComment = 'email unsubscribe support';

Messaging.SingleEmailMessage[] messages =
    new List<Messaging.SingleEmailMessage> {message};
Messaging.SendEmailResult[] results = Messaging.sendEmail(messages);


if (results[0].success) {
    System.debug('The email was sent successfully.');
} else {
    System.debug('The email failed to send: '
          + results[0].errors[0].message);
}
```

`setUnsubscribeUrls(UnsubscribeUrls)`

Optional. Sets a `mailto` URI and HTTP URL of a mechanism for unsubscribing a recipient from an email list. A list of all unsubscribe URLs passed through `setUnsubscribeUrls` is added to the `List-Unsubscribe` header. A minimum of one URL is required to use this method.

Signature

```
public void setUnsubscribeUrls(List<String> unsubscribeUrls)
```

Parameters

`UnsubscribeUrls`

Type: List<String>

Return Value

Type: void

Usage

Provide a list of URLs that support unsubscribe functionality by offering recipients multiple ways to opt-out of future communications. Each provided URL can use different protocols to allow for technical capacities of the recipient.

All `setUnsubscribeUrls` must have a value of one of these types:

- `Mailto`: Allows recipients to send an unsubscribe request via email.
  - Example: `mailto:listrequest@example.com?subject=unsubscribe`
- `HTTP`: Directs recipients to a web page where they can unsubscribe.
  - Example: `http://example.com/unsubscribe.html?opaque=123456789`
- `HTTPS`: Directs recipients to a secure web page to unsubscribe.
  - Example: `https://example.com/unsubscribe.html?opaque=123456789`

Example

This example demonstrates how to send an email using Salesforce's `Messaging.SingleEmailMessage` class that includes an option to include an unsubscribe link for a user to click. It creates an email message, sets the recipient's email address, subject, and body, and includes an unsubscribe URL. The email message is added to a list and sent using the `Messaging.sendEmail` method.

```
Messaging.SingleEmailMessage message = new Messaging.SingleEmailMessage();
// Set the recipient's email address
// Replace IDs with valid record IDs in your org.
message.toAddresses = new String[] { '003D000000QDexS' };
message.subject = 'Test Message';
message.plainTextBody = 'This is the message body.';


// Create a list to hold unsubscribe URLs
List<String> unsubscribeUrls = new List<String>();
unsubscribeUrls.add('https://example.com/unsubscribe.html?opaque=123456789');


// Assign the unsubscribe URLs to the email message
message.unsubscribeUrls = unsubscribeUrls;

Messaging.SingleEmailMessage[] messages =
    new List<Messaging.SingleEmailMessage> {message};
Messaging.SendEmailResult[] results = Messaging.sendEmail(messages);


if (results[0].success) {
    System.debug('The email was sent successfully.');
} else {
    System.debug('The email failed to send: '
          + results[0].errors[0].message);
}
```

`setWhatId(whatId)`

If you specify a contact for the `targetObjectId` field, you can specify an optional `whatId` as well. This helps to further ensure that merge fields in the template contain the correct data.

Signature

```
public Void setWhatId(ID whatId)
```

Parameters

whatId
Type: ID

Return Value

Type: Void

Usage

The value must be one of the following types:

- Account
- Asset
- Campaign
- Case
- Contract
- Opportunity
- Order
- Product
- Solution
- Custom

Metadata Namespace

The `Metadata` namespace provides classes and methods for working with custom metadata in Salesforce

Salesforce uses metadata types and components to represent org configuration and customization. Metadata is used for org settings that admins control or configuration information applied by installed apps and packages. Use the classes in the `Metadata` namespace to access metadata from within Apex code.

Metadata access in Apex is available for Apex classes using API version 40.0 and later.

For more information, see Metadata.

The following are the classes in the `Metadata` namespace.

IN THIS SECTION:

AnalyticsCloudComponentLayoutItem Class
Represents the settings for a Wave Analytics dashboard on a standard or custom page.

ConsoleComponent Class
Represents a custom console component on a section of a page layout.

Container Class
Represents a location and style in which to display more than one custom console component in the sidebars of the console.

CustomConsoleComponents Class
Represents custom console components (Visualforce pages, lookup fields, or related lists) on a page layout.

CustomMetadata Class

Represents records of custom metadata types.

CustomMetadataValue Class

Represents custom metadata values for a custom metadata component.

DeployCallback Interface

An interface for metadata deployment callback classes.

DeployCallbackContext Class

Represents context information for a deployment job.

DeployContainer Class

Represents a container for custom metadata components to be deployed.

DeployDetails Class

Contains detailed information on deployed components.

DeployMessage Class

Represents result information for the deployment of a metadata component.

DeployProblemType Enum

Describes the problem type for an unsuccessful component deploy.

DeployResult Class

Represents the results of a metadata deployment.

DeployStatus Enum

The result status of a deployment.

FeedItemTypeEnum Enum

The type of feed item in a feed-based page layout.

FeedLayout Class

Represents the values that define the feed view of a feed-based page layout. Feed-based layouts are available on Account, Case, Contact, Lead, Opportunity, custom, and external objects. They include a feed view and a detail view.

FeedLayoutComponent Class

Represents a component in the feed view of a feed-based page layout.

FeedLayoutComponentType Enum

Indicates the type of feed layout component.

FeedLayoutFilter Class

Represents a feed filter option in the feed view of a feed-based page layout. A filter can have only `standardFilter` or `feedItemType` set.

FeedLayoutFilterPosition Enum

Describes where the feed filters list is included in the layout.

FeedLayoutFilterType Enum

The type of feed layout filter.

Layout Class

Represents the metadata associated with a page layout.

LayoutColumn Class

Represents the items in a column within a layout section.

LayoutHeader Enum

Represents tagging types used for `Metadata.Layout.headers`

LayoutItem Class

Represents the valid values that define a layout item.

LayoutSection Class

Represents a section of a page layout, such as the Custom Links section.

LayoutSectionStyle Enum

Describes the possible styles for a layout section.

Metadata Class

An abstract base class that represents a custom metadata component.

MetadataType Enum

Represents the custom metadata components available in Apex.

MetadataValue Class

An abstract base class that represents a custom metadata component field.

MiniLayout Class

Represents a mini view of a record in the Console tab, hover details, and event overlays.

Operations Class

Represents a class to execute metadata operations, such as retrieving or deploying custom metadata.

PlatformActionList Class

Represents the list of actions, and their order, that display in the Salesforce mobile action bar for the layout.

PlatformActionListContextEnum Enum

Describes the different contexts of action lists.

PlatformActionListItem Class

Represents an action in the platform action list for a layout.

PlatformActionTypeEnum Enum

The type of action for a `PlatformActionListItem`.

PrimaryTabComponents Class

Represents custom console components on primary tabs in the Salesforce console.

QuickActionList Class

Represents the list of actions associated with the page layout.

QuickActionListItem Class

Represents an action in the `QuickActionList`.

RelatedContent Class

Represents the Mobile Cards section of the page layout.

RelatedContentItem Class

Represents an individual item in the `RelatedContent` list.

RelatedList Class

Represents related list custom components on the sidebars of the Salesforce console.

RelatedListItem Class

Represents an item in the related list in a page layout.

ReportChartComponentLayoutItem Class

Represents the settings for a report chart on a standard or custom page.

ReportChartComponentSize Enum

Describes the size of the displayed report chart component.

SidebarComponent Class

Represents a specific custom console component to display in a container that hosts multiple components in one of the sidebars of the Salesforce console.

SortOrder Enum

Describes the sort order of a related list.

StatusCode Enum

Describes the status code for an unsuccessful component deploy.

SubtabComponents Class

Represents custom console components on subtabs in the Salesforce console.

SummaryLayoutStyleEnum Enum

Describes the highlights panel style for a `SummaryLayout`.

SummaryLayout Class

Controls the appearance of the highlights panel, which summarizes key fields in a grid at the top of a page layout, when Case Feed is enabled.

SummaryLayoutItem Class

Controls the appearance of an individual field and its column and row position within the highlights panel grid, when Case Feed is enabled. You can have two fields per each grid in a highlights panel.

UiBehavior Enum

Describes the behavior for a layout item on a layout page.

AnalyticsCloudComponentLayoutItem Class

Represents the settings for a Wave Analytics dashboard on a standard or custom page.

Namespace

Metadata

Usage

Use this class when accessing `Metadata.Layout` metadata components. For more information, see "AnalyticsCloudComponentLayoutItem" in the *Metadata API Developer Guide*.

IN THIS SECTION:

AnalyticsCloudComponentLayoutItem Properties

AnalyticsCloudComponentLayoutItem Methods

AnalyticsCloudComponentLayoutItem Properties

The following are properties for `AnalyticsCloudComponentLayoutItem`.

IN THIS SECTION:

assetType
Specifies the type of Wave Analytics asset.

devName
Unique development name of the dashboard to add.

error
An error string that is populated only when an error occurred in the underlying dashboard.

filter
Dashboard filters for mapping data fields in the dashboard to the object's fields.

height
Specifies the height of the dashboard, in pixels.

hideOnError
Controls whether users see a dashboard that has an error.

showHeader
If `true`, includes the dashboard's header bar. If `false`, the dashboard appears without a header bar.

showSharing
If set to true, and the dashboard is shareable the dashboard shows the Share icon. If set to false, the dashboard doesn't show the Share icon.

showTitle
If true, includes the dashboard's title above the dashboard. If false, the dashboard appears without a title.

width
Specifies the width of the dashboard, in pixels or percentage.

assetType

Specifies the type of Wave Analytics asset.

Signature

```
public String assetType {get; set;}
```

Property Value

Type: String

devName

Unique development name of the dashboard to add.

Signature

```
public String devName {get; set;}
```

Property Value

Type: String

error

An error string that is populated only when an error occurred in the underlying dashboard.

Signature

```
public String error {get; set;}
```

Property Value

Type: String

filter

Dashboard filters for mapping data fields in the dashboard to the object's fields.

Signature

```
public String filter {get; set;}
```

Property Value

Type: String

height

Specifies the height of the dashboard, in pixels.

Signature

```
public Integer height {get; set;}
```

Property Value

Type: Integer

hideOnError

Controls whether users see a dashboard that has an error.

Signature

```
public Boolean hideOnError {get; set;}
```

Property Value

Type: Boolean

showHeader

If `true`, includes the dashboard's header bar. If `false`, the dashboard appears without a header bar.

Signature

```
public Boolean showHeader {get; set;}
```

Property Value

Type: Boolean

showSharing

If set to true, and the dashboard is shareable the dashboard shows the Share icon. If set to false, the dashboard doesn't show the Share icon.

Signature

```
public Boolean showSharing {get; set;}
```

Property Value

Type: Boolean

showTitle

If true, includes the dashboard's title above the dashboard. If false, the dashboard appears without a title.

Signature

```
public Boolean showTitle {get; set;}
```

Property Value

Type: Boolean

width

Specifies the width of the dashboard, in pixels or percentage.

Signature

```
public String width {get; set;}
```

Property Value

Type: String

AnalyticsCloudComponentLayoutItem Methods

The following are methods for `AnalyticsCloudComponentLayoutItem`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.AnalyticsCloudComponentLayoutItem`.

`clone()`

Makes a duplicate copy of the `Metadata.AnalyticsCloudComponentLayoutItem`.

Signature

```
public Object clone()
```

Return Value

Type: Object

ConsoleComponent Class

Represents a custom console component on a section of a page layout.

Namespace

Metadata

Usage

Use this class when accessing `Metadata.Layout` metadata components. For more information, see "ConsoleComponent" in the *Metadata API Developer Guide*.

IN THIS SECTION:

ConsoleComponent Properties

ConsoleComponent Methods

ConsoleComponent Properties

The following are properties for `ConsoleComponent`.

IN THIS SECTION:

height
The height of the custom console component in pixels.

location
The location of the custom console component on the page layout. Valid values are right, left, top, and bottom.

visualforcePage
The unique name of the custom console component.

width
The width of the custom console component in pixels.

height

The height of the custom console component in pixels.

Signature

```
public Integer height {get; set;}
```

Property Value

Type: Integer

location

The location of the custom console component on the page layout. Valid values are right, left, top, and bottom.

Signature

```
public String location {get; set;}
```

Property Value

Type: String

visualforcePage

The unique name of the custom console component.

Signature

```
public String visualforcePage {get; set;}
```

Property Value

Type: String

width

The width of the custom console component in pixels.

Signature

```
public Integer width {get; set;}
```

Property Value

Type: Integer

ConsoleComponent Methods

The following are methods for `ConsoleComponent`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.ConsoleComponent`.

`clone()`

Makes a duplicate copy of the `Metadata.ConsoleComponent`.

Signature

```
public Object clone()
```

Return Value

Type: Object

Container Class

Represents a location and style in which to display more than one custom console component in the sidebars of the console.

Namespace

Metadata

Usage

Use this class when accessing `Metadata.Layout` metadata components. For more information, see "Container" in the *Metadata API Developer Guide*.

IN THIS SECTION:

Container Properties

Container Methods

Container Properties

The following are properties for `Container`.

IN THIS SECTION:

height
The height of the component's container. The `unit` property determines the unit of measurement, in pixels or percent.

isContainerAutoSizeEnabled
If set to true, stacked console components in the sidebars autosize vertically.

region
The location of the component's container (right, left, bottom, top).

sidebarComponents
Represents a specific custom console component to display in the components' container.

style
The style of the container in which to display multiple components (stack, tab, accordion).

unit
The unit of measurement, in pixels or percent, for the height or width of the components' container.

width
The width of the component's container. The `unit` property determines the unit of measurement, in pixels or percent.

`height`

The height of the component's container. The `unit` property determines the unit of measurement, in pixels or percent.

Signature

```
public Integer height {get; set;}
```

Property Value

Type: Integer

`isContainerAutoSizeEnabled`

If set to true, stacked console components in the sidebars autosize vertically.

Signature

```
public Boolean isContainerAutoSizeEnabled {get; set;}
```

Property Value

Type: Boolean

`region`

The location of the component's container (right, left, bottom, top).

Signature

```
public String region {get; set;}
```

Property Value

Type: String

`sidebarComponents`

Represents a specific custom console component to display in the components' container.

Signature

```
public List<Metadata.SidebarComponent> sidebarComponents {get; set;}
```

Property Value

Type: List<Metadata.SidebarComponent>

`style`

The style of the container in which to display multiple components (stack, tab, accordion).

Signature

```
public String style {get; set;}
```

Property Value

Type: String

`unit`

The unit of measurement, in pixels or percent, for the height or width of the components' container.

Signature

```
public String unit {get; set;}
```

Property Value

Type: String

`width`

The width of the component's container. The `unit` property determines the unit of measurement, in pixels or percent.

Signature

```
public Integer width {get; set;}
```

Property Value

Type: Integer

Container Methods

The following are methods for `Container`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.Container`.

`clone()`

Makes a duplicate copy of the `Metadata.Container`.

Signature

```
public Object clone()
```

Return Value

Type: Object

CustomConsoleComponents Class

Represents custom console components (Visualforce pages, lookup fields, or related lists) on a page layout.

Namespace

Metadata

Usage

Use this class when accessing `Metadata.Layout` metadata components. For more information, see "CustomConsoleComponents" in the *Metadata API Developer Guide*.

IN THIS SECTION:

CustomConsoleComponents Properties

CustomConsoleComponents Methods

CustomConsoleComponents Properties

The following are properties for `CustomConsoleComponents`.

IN THIS SECTION:

primaryTabComponents
Represents custom console components on primary tabs in the Salesforce console.

subtabComponents
Represents custom console components on subtabs in the Salesforce console.

primaryTabComponents

Represents custom console components on primary tabs in the Salesforce console.

Signature

```
public Metadata.PrimaryTabComponents primaryTabComponents {get; set;}
```

Property Value

Type: Metadata.PrimaryTabComponents

subtabComponents

Represents custom console components on subtabs in the Salesforce console.

Signature

```
public Metadata.SubtabComponents subtabComponents {get; set;}
```

Property Value

Type: Metadata.SubtabComponents

CustomConsoleComponents Methods

The following are methods for `CustomConsoleComponents`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.CustomConsoleComponents`.

clone()

Makes a duplicate copy of the `Metadata.CustomConsoleComponents`.

Signature

```
public Object clone()
```

Return Value

Type: Object

CustomMetadata Class

Represents records of custom metadata types.

Warning: Protected custom metadata types behave like public custom metadata types when they are outside of a managed package. Public custom metadata types are readable for all profiles, including the guest user. Do not store secrets, personally identifying information, or any private data in these records. Use protected custom metadata types only in managed packages. Outside of a managed package, use named credentials or encrypted custom fields to store secrets like OAuth tokens, passwords, and other confidential material.

Namespace

Metadata

Usage

Use `Metadata.CustomMetadata` to represent records of custom metadata types in Apex. For more information, see Custom Metadata Types in the *Metadata API Developer Guide*.

Example

```
// Set up custom metadata to be created in the subscriber org.
    Metadata.CustomMetadata customMetadata =  new Metadata.CustomMetadata();
    customMetadata.fullName = 'ISVNamespace__MetadataTypeName.MetadataRecordName';

    Metadata.CustomMetadataValue customField = new Metadata.CustomMetadataValue();
    customField.field = 'customField__c';
    customField.value = 'New value';

    customMetadata.values.add(customField);
```

Note: When you assign namespaces to records, provide full, qualified record names to the app. If both the type and the record are in *Namespace*, use something like: `customMetadata.fullName = '`***`Namespace`***`__MetadataTypeName.`***`Namespace`***`__MetadataRecordName'`

IN THIS SECTION:

- CustomMetadata Properties
- CustomMetadata Methods

CustomMetadata Properties

The following are properties for `CustomMetadata`.

IN THIS SECTION:

description
The description of the custom metadata.

label
The label of the custom metadata record.

protected_x
Property that describes whether the custom metadata record is a protected component.

values
A list of custom metadata values, such as custom fields, for the custom metadata record.

description

The description of the custom metadata.

Signature

```
public String description {get; set;}
```

Property Value

Type: String

label

The label of the custom metadata record.

Signature

```
public String label {get; set;}
```

Property Value

Type: String

protected_x

Property that describes whether the custom metadata record is a protected component.

Signature

```
public Boolean protected_x {get; set;}
```

Property Value

Type: Boolean

values

A list of custom metadata values, such as custom fields, for the custom metadata record.

Signature

```
public List<Metadata.CustomMetadataValue> values {get; set;}
```

Property Value

Type: List<Metadata.CustomMetadataValue>

CustomMetadata Methods

The following are methods for `CustomMetadata`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.CustomMetadata`.

`clone()`

Makes a duplicate copy of the `Metadata.CustomMetadata`.

Signature

```
public Object clone()
```

Return Value

Type: Object

CustomMetadataValue Class

Represents custom metadata values for a custom metadata component.

Namespace

Metadata

Usage

Use `Metadata.CustomMetadataValue` to access values for custom fields of custom metadata records.

Supported Apex primitive types are:

- Boolean
- Date
- DateTime
- Decimal
- Double
- Integer
- Long

- String

Example

```
// Set a custom field value for a custom metadata record
Metadata.CustomMetadataValue customField = new Metadata.CustomMetadataValue();
customField.field = 'CustomField1__c';
customField.value = 'New Value';
customMetadata.values.add(customField);
```

IN THIS SECTION:

CustomMetadataValue Properties

CustomMetadataValue Methods

CustomMetadataValue Properties

The following are properties for `CustomMetadataValue`.

IN THIS SECTION:

field
The field name for the custom metadata value.

value
The field value for the custom metadata value.

field

The field name for the custom metadata value.

Signature

```
public String field {get; set;}
```

Property Value

Type: String

value

The field value for the custom metadata value.

Signature

```
public Object value {get; set;}
```

Property Value

Type: Object

Supported Apex primitive types are:

- Boolean
- Date
- DateTime
- Decimal
- Double
- Integer
- Long
- String

When setting the value for relationship fields, use the qualified API name of the related metadata, not the ID.

For more information, see Primitive Data Types.

CustomMetadataValue Methods

The following are methods for `CustomMetadataValue`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.CustomMetadataValue`.

`clone()`

Makes a duplicate copy of the `Metadata.CustomMetadataValue`.

Signature

```
public Object clone()
```

Return Value

Type: Object

DeployCallback Interface

An interface for metadata deployment callback classes.

Namespace

Metadata

Usage

You must provide a callback class for the asynchronous deployment of custom metadata through Apex. This class must implement the `Metadata.DeployCallback` interface.

Salesforce calls your `DeployCallback.handleResult()` method asynchronously once the queued deployment completes. Because the callback is called as asynchronous Apex after deployment, there may be a brief period where the deploy has completed, but your callback has not been called yet.

IN THIS SECTION:

DeployCallback Methods

DeployCallback Example Implementation

DeployCallback Methods

The following are methods for `DeployCallback`.

IN THIS SECTION:

handleResult(var1, var2)
Method that is called when the asynchronous deployment of custom metadata completes.

`handleResult(var1, var2)`

Method that is called when the asynchronous deployment of custom metadata completes.

Signature

```
public void handleResult(Metadata.DeployResult var1, Metadata.DeployCallbackContext
var2)
```

Parameters

var1
Type: Metadata.DeployResult

The results of the asynchronous deployment.

var2
Type: Metadata.DeployCallbackContext

The context for the queued asynchronous deployment job.

Return Value

Type: void

DeployCallback Example Implementation

This is an example implementation of the `Metadata.DeployCallback` interface.

```
public class MyCallback implements Metadata.DeployCallback {
    public void handleResult(Metadata.DeployResult result,
                             Metadata.DeployCallbackContext context) {
        if (result.status == Metadata.DeployStatus.Succeeded) {
            // Deployment was successful
```

```
        } else {
            // Deployment was not successful
        }
    }
}
```

The following example uses this implementation for a deployment.

```
// Setup callback and deploy
MyCallback callback = new MyCallback();
Metadata.Operations.enqueueDeployment(mdContainer, callback);
```

DeployCallbackContext Class

Represents context information for a deployment job.

Namespace

Metadata

Usage

After an asynchronous metadata deployment finishes, Salesforce provides an instance of`Metadata.DeployCallbackContext` in an asynchronous call to your implementation of `handleResult()` in your `Metadata.DeployCallback` class.

Example

```
public void handleResult(Metadata.DeployResult result,
                         Metadata.DeployCallbackContext context) {
  // Check the callback job ID for the deployment
  Id jobId = context.getCallbackJobId();
  // ...process the results...
}
```

IN THIS SECTION:

DeployCallbackContext Methods

DeployCallbackContext Methods

The following are methods for `DeployCallbackContext`.

IN THIS SECTION:

clone()

Makes a duplicate copy of the `Metadata.DeployCallbackContext`.

getCallbackJobId()

Gets the asynchronous Apex job ID for the callback job.

`clone()`

Makes a duplicate copy of the `Metadata.DeployCallbackContext`.

Signature

```
public Object clone()
```

Return Value

Type: Object

`getCallbackJobId()`

Gets the asynchronous Apex job ID for the callback job.

Signature

```
public Id getCallbackJobId()
```

Return Value

Type: Id

DeployContainer Class

Represents a container for custom metadata components to be deployed.

Namespace

Metadata

Usage

Use `Metadata.DeployContainer` to manage custom metadata components for deployment. A container must have one or more components before being deployed.

Example

```
// Use DeployContainer for deployment
Metadata.DeployContainer mdContainer = new Metadata.DeployContainer();
mdContainer.addMetadata(customMetadata);

...

// Enqueue deploy
Metadata.Operations.enqueueDeployment(mdContainer, callback);
```

IN THIS SECTION:

DeployContainer Methods

DeployContainer Methods

The following are methods for `DeployContainer`.

IN THIS SECTION:

addMetadata(md)
Add a custom metadata component to the container.

clone()
Makes a duplicate copy of the `Metadata.DeployContainer`.

getMetadata()
Retrieves a list of custom metadata components from the container.

removeMetadata(md)
Removes a metadata component from the container.

removeMetadataByFullName(fullName)
Removes a metadata component from the container using the component's full name.

`addMetadata(md)`

Add a custom metadata component to the container.

Signature

```
public void addMetadata(Metadata.Metadata md)
```

Parameters

md

Type: Metadata.Metadata

A custom metadata component class that derives from `Metadata.Metadata`. Avoid adding components to a `Metadata.DeployContainer`that have the same `Metadata.Metadata.fullName` because it causes deployment errors.

Return Value

Type: void

`clone()`

Makes a duplicate copy of the `Metadata.DeployContainer`.

Signature

```
public Object clone()
```

Return Value

Type: Object

`getMetadata()`

Retrieves a list of custom metadata components from the container.

Signature

```
public List<Metadata.Metadata> getMetadata()
```

Return Value

Type: List<Metadata.Metadata>

`removeMetadata(md)`

Removes a metadata component from the container.

Signature

```
public Boolean removeMetadata(Metadata.Metadata md)
```

Parameters

md
Type: Metadata.Metadata

Metadata component to remove.

Return Value

Type: Boolean

Returns `true` if the container changed as a result of the call.

`removeMetadataByFullName(fullName)`

Removes a metadata component from the container using the component's full name.

Signature

```
public Boolean removeMetadataByFullName(String fullName)
```

Parameters

fullName
Type: String

Full name of the component to remove.

Return Value

Type: Boolean

Returns `true` if the container changed as a result of the call.

DeployDetails Class

Contains detailed information on deployed components.

Namespace

Metadata

Usage

Use this class to obtain a list of the successfully and unsuccessfully deployed components after a completed deployment by Salesforce in your `Metadata.DeployCallback` results.

IN THIS SECTION:

DeployDetails Properties

DeployDetails Methods

DeployDetails Properties

The following are properties for `DeployDetails`.

IN THIS SECTION:

componentFailures
Contains a list of information about components that failed to deploy.

componentSuccesses
Contains a list of information about components that deployed successfully.

`componentFailures`

Contains a list of information about components that failed to deploy.

Signature

```
public List<Metadata.DeployMessage> componentFailures {get; set;}
```

Property Value

Type: List<Metadata.DeployMessage>

`componentSuccesses`

Contains a list of information about components that deployed successfully.

Signature

```
public List<Metadata.DeployMessage> componentSuccesses {get; set;}
```

Property Value

Type: List<Metadata.DeployMessage>

DeployDetails Methods

The following are methods for `DeployDetails`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.DeployDetails`.

`clone()`

Makes a duplicate copy of the `Metadata.DeployDetails`.

Signature

```
public Object clone()
```

Return Value

Type: Object

DeployMessage Class

Represents result information for the deployment of a metadata component.

Namespace

Metadata

Usage

Use `DeployMessage` to access detailed information about component deployments. Salesforce provides a list of `DeployMessages` for a completed deployment via the `DeployDetails` and `DeployResults` instances sent in the `DeployCallback.handleResult()` callback.

IN THIS SECTION:

DeployMessage Properties
DeployMessage Methods

DeployMessage Properties

The following are properties for `DeployMessage`.

IN THIS SECTION:

changed

Determines whether the component was changed after deployment. If true, the component was changed as a result of the deployment. If false, the deployed component was the same as the corresponding component already in the org.

columnNumber

Each component is represented by a text file. If an error occurs during deployment, this property represents the column of the text file where the error occurred.

componentType

The metadata type of the component in the deployment.

created

If true, the component was created as a result of the deployment. If false, the component was modified as a result of the deployment.

createdDate

The date and time when the component was created as a result of the deployment.

deleted

If true, the component was deleted as a result of the deployment. If false, the component was either new or modified as result of the deployment.

fileName

The name of the file in the metadata archive used to deploy the component.

fullName

Full name for the custom metadata component.

id

ID of the component that was deployed.

lineNumber

Each component is represented by a text file. If an error occurs during deployment, this field represents the line number of the text file where the error occurred.

problem

If an error or warning occurred, this field contains a description of the problem that caused the deployment to fail.

problemType

Indicates the problem type, for example, an error or warning.

success

Indicates whether the component was successfully deployed (true) or not (false).

changed

Determines whether the component was changed after deployment. If true, the component was changed as a result of the deployment. If false, the deployed component was the same as the corresponding component already in the org.

Signature

```
public Boolean changed {get; set;}
```

Property Value

Type: Boolean

columnNumber

Each component is represented by a text file. If an error occurs during deployment, this property represents the column of the text file where the error occurred.

Signature

```
public Integer columnNumber {get; set;}
```

Property Value

Type: Integer

componentType

The metadata type of the component in the deployment.

Signature

```
public String componentType {get; set;}
```

Property Value

Type: String

created

If true, the component was created as a result of the deployment. If false, the component was modified as a result of the deployment.

Signature

```
public Boolean created {get; set;}
```

Property Value

Type: Boolean

createdDate

The date and time when the component was created as a result of the deployment.

Signature

```
public Datetime createdDate {get; set;}
```

Property Value

Type: Datetime

`deleted`

If true, the component was deleted as a result of the deployment. If false, the component was either new or modified as result of the deployment.

Signature

```
public Boolean deleted {get; set;}
```

Property Value

Type: Boolean

`fileName`

The name of the file in the metadata archive used to deploy the component.

Signature

```
public String fileName {get; set;}
```

Property Value

Type: String

`fullName`

Full name for the custom metadata component.

Signature

```
public String fullName {get; set;}
```

Property Value

Type: String

`id`

ID of the component that was deployed.

Signature

```
public Id id {get; set;}
```

Property Value

Type: Id

`lineNumber`

Each component is represented by a text file. If an error occurs during deployment, this field represents the line number of the text file where the error occurred.

Signature

```
public Integer lineNumber {get; set;}
```

Property Value

Type: Integer

`problem`

If an error or warning occurred, this field contains a description of the problem that caused the deployment to fail.

Signature

```
public String problem {get; set;}
```

Property Value

Type: String

`problemType`

Indicates the problem type, for example, an error or warning.

Signature

```
public Metadata.DeployProblemType problemType {get; set;}
```

Property Value

Type: Metadata.DeployProblemType

`success`

Indicates whether the component was successfully deployed (true) or not (false).

Signature

```
public Boolean success {get; set;}
```

Property Value

Type: Boolean

DeployMessage Methods

The following are methods for `DeployMessage`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.DeployMessage`.

`clone()`

Makes a duplicate copy of the `Metadata.DeployMessage`.

Signature

```
public Object clone()
```

Return Value

Type: Object

DeployProblemType Enum

Describes the problem type for an unsuccessful component deploy.

Enum Values

The following are the values of the `Metadata.DeployProblemType` enum.

Value	Description
`Error`	The deploy problem is an error.
`Info`	The deploy problem is of type “Info”.
`Warning`	The deploy problem is a warning.

SEE ALSO:

StatusCode Enum

DeployResult Class

Represents the results of a metadata deployment.

Namespace

Metadata

Usage

After an asynchronous metadata deployment finishes, Salesforce provides an instance of `Metadata.DeployResult` in a call to your implementation of `handleResult()` in your `Metadata.DeployCallback` class.

Example

```
public void handleResult(Metadata.DeployResult result,
                         Metadata.DeployCallbackContext context) {
    if (result.status == Metadata.DeployStatus.Succeeded) {
        // Deployment was successful
    } else {
        // Deployment was not successful
    }
}
```

IN THIS SECTION:

DeployResult Properties

DeployResult Methods

DeployResult Properties

The following are properties for `DeployResult`.

IN THIS SECTION:

canceledBy

ID of the user who canceled the queued deployment.

canceledByName

Full name of the user who canceled the queued deployment.

checkOnly

Indicates whether the deployment checked only the validity of the deployed files without making changes in the org. A check-only deployment does not deploy components or change the org in any way.

completedDate

Date and time for when the deployment process ended.

createdBy

ID of the user who created the deployment job.

createdByName

Full name of the user who created the deployment job.

createdDate

Date and time the deployment job was first queued.

details

Provides the details for components in a completed deployment.

done

Indicates whether Salesforce finished processing the deployment.

errorMessage

Message corresponding to the values in the `errorStatusCode` property, if any.

errorStatusCode

If an error occurs during deployment, a status code is returned. The message corresponding to the status code is returned in the `errorMessagefield` property.

id

ID of the deployment job.

ignoreWarnings

Specifies whether a deployment continues, even if the deployment generates warnings.

lastModifiedDate

Date and time of the last update for the deployment process.

messages

A list of all the detail messages for a deployment.

numberComponentErrors

The number of components that generated errors during the deployment.

numberComponentsDeployed

The number of components deployed in the deployment process. Use this value with the `numberComponentsTotal` property to get an estimate of the deployment's progress.

numberComponentsTotal

The total number of components in the deployment. Use this value with the `numberComponentsDeployed` property to get an estimate of the deployment's progress.

rollbackOnError

Indicates whether any failure causes a complete rollback (true) or not (false) of the deployment.

startDate

Date and time the deployment process began.

stateDetail

Indicates which component is being deployed.

status

Indicates the current state of the deployment.

success

Indicates whether the deployment was successful (true) or not (false).

canceledBy

ID of the user who canceled the queued deployment.

Signature

```
public String canceledBy {get; set;}
```

Property Value

Type: String

canceledByName

Full name of the user who canceled the queued deployment.

Signature

```
public String canceledByName {get; set;}
```

Property Value

Type: String

checkOnly

Indicates whether the deployment checked only the validity of the deployed files without making changes in the org. A check-only deployment does not deploy components or change the org in any way.

Signature

```
public Boolean checkOnly {get; set;}
```

Property Value

Type: Boolean

completedDate

Date and time for when the deployment process ended.

Signature

```
public Datetime completedDate {get; set;}
```

Property Value

Type: Datetime

createdBy

ID of the user who created the deployment job.

Signature

```
public String createdBy {get; set;}
```

Property Value

Type: String

createdByName

Full name of the user who created the deployment job.

Signature

```
public String createdByName {get; set;}
```

Property Value

Type: String

createdDate

Date and time the deployment job was first queued.

Signature

```
public Datetime createdDate {get; set;}
```

Property Value

Type: Datetime

details

Provides the details for components in a completed deployment.

Signature

```
public Metadata.DeployDetails details {get; set;}
```

Property Value

Type: Metadata.DeployDetails

done

Indicates whether Salesforce finished processing the deployment.

Signature

```
public Boolean done {get; set;}
```

Property Value

Type: Boolean

errorMessage

Message corresponding to the values in the `errorStatusCode` property, if any.

Signature

```
public String errorMessage {get; set;}
```

Property Value

Type: String

errorStatusCode

If an error occurs during deployment, a status code is returned. The message corresponding to the status code is returned in the `errorMessagefield` property.

Signature

```
public String errorStatusCode {get; set;}
```

Property Value

Type: String

For a description of each status code value, see Core Data Types Used in API Calls in the *SOAP API Developer Guide*.

id

ID of the deployment job.

Signature

```
public Id id {get; set;}
```

Property Value

Type: Id

ignoreWarnings

Specifies whether a deployment continues, even if the deployment generates warnings.

Signature

```
public Boolean ignoreWarnings {get; set;}
```

Property Value

Type: Boolean

`lastModifiedDate`

Date and time of the last update for the deployment process.

Signature

```
public Datetime lastModifiedDate {get; set;}
```

Property Value

Type: Datetime

`messages`

A list of all the detail messages for a deployment.

Note: Removed in API version 29.0 and later.

Signature

```
public List<Metadata.DeployMessage> messages {get; set;}
```

Property Value

Type: List<Metadata.DeployMessage>

`numberComponentErrors`

The number of components that generated errors during the deployment.

Signature

```
public Integer numberComponentErrors {get; set;}
```

Property Value

Type: Integer

`numberComponentsDeployed`

The number of components deployed in the deployment process. Use this value with the `numberComponentsTotal` property to get an estimate of the deployment's progress.

Signature

```
public Integer numberComponentsDeployed {get; set;}
```

Property Value

Type: Integer

`numberComponentsTotal`

The total number of components in the deployment. Use this value with the `numberComponentsDeployed` property to get an estimate of the deployment's progress.

Signature

```
public Integer numberComponentsTotal {get; set;}
```

Property Value

Type: Integer

`rollbackOnError`

Indicates whether any failure causes a complete rollback (true) or not (false) of the deployment.

Signature

```
public Boolean rollbackOnError {get; set;}
```

Property Value

Type: Boolean

`startDate`

Date and time the deployment process began.

Signature

```
public Datetime startDate {get; set;}
```

Property Value

Type: Datetime

`stateDetail`

Indicates which component is being deployed.

Signature

```
public String stateDetail {get; set;}
```

Property Value

Type: String

`status`

Indicates the current state of the deployment.

Signature

```
public Metadata.DeployStatus status {get; set;}
```

Property Value

Type: Metadata.DeployStatus

`success`

Indicates whether the deployment was successful (true) or not (false).

Signature

```
public Boolean success {get; set;}
```

Property Value

Type: Boolean

DeployResult Methods

The following are methods for `DeployResult`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.DeployResult`.

`clone()`

Makes a duplicate copy of the `Metadata.DeployResult`.

Signature

```
public Object clone()
```

Return Value

Type: Object

DeployStatus Enum

The result status of a deployment.

Usage

`Metadata.DeployResult.status` uses this enum to describe the results of the deployment.

Enum Values

The following are the values of the `Metadata.DeployStatus` enum.

Value	Description
`Canceled`	The queued deployment was canceled.
`Canceling`	The queued deployment is being canceled.
`Failed`	The deployment failed.
`InProgress`	The deployment has been started and is in progress.
`Pending`	The deployment has been queued but not started.
`Succeeded`	The deployment succeeded.
`SucceededPartial`	The deployment succeeded, but some components might not have been successfully deployed. Check `Metadata.DeployResult` for more details.

FeedItemTypeEnum Enum

The type of feed item in a feed-based page layout.

Enum Values

The following are the values of the `Metadata.FeedItemTypeEnum` enum.

Value	Description
`ActivityEvent`	Activity on tasks and events associated with a case. Available only on Case layouts.
`AdvancedTextPost`	Group announcements posted on a feed.
`AnnouncementPost`	Not used.
`ApprovalPost`	Approvals submitted on a feed.
`AttachArticleEvent`	Activity related to attaching articles to cases.

Value	Description
`BasicTemplateFeedItem`	Activity from the Log a Call action. Available only on layouts for objects that support Activities (tasks and events).
`CallLogPost`	Activity from the Log a Call action. Available only on layouts for objects that support Activities (tasks and events).
`CanvasPost`	Posts a canvas app makes on a feed.
`CaseCommentPost`	Activity from the Case Note action. Available only on Case layouts.
`ChangeStatusPost`	Activity from the Change Status action. Available only on Case layouts.
`ChatTranscriptPost`	Activity related to attaching Chat transcripts to cases. Available only on Case layouts.
`CollaborationGroupCreated`	Creating a public group.
`CollaborationGroupUnarchived`	Not used.
`ContentPost`	Attaching a file to a post.
`CreateRecordEvent`	Creating a record from the publisher.
`DashboardComponentAlert`	Not used.
`DashboardComponentSnapshot`	Posting a dashboard snapshot on a feed.
`EmailMessageEvent`	Activity from the Email action. Available only on Case layouts.
`FacebookPost`	Not used.
`LinkPost`	Attaching a URL to a post.
`MilestoneEvent`	Changing the milestone status on a case. Available only on Case layouts.
`PollPost`	Posting a poll on a feed.
`ProfileSkillPost`	Adding skills to a user's Chatter profile.
`QuestionPost`	Posting a question on a feed.
`ReplyPost`	Activity from the Portal action. Available only on Case layouts.
`RypplePost`	Creating a Thanks badge in WDC.
`SocialPost`	Activity on Twitter from the Social Post action.
`TestItem`	Creating a text post from the publisher.
`TextPost`	Making a change or group of changes to a tracked field.
`TrackedChange`	Not used.
`Undefined`	Undefined feed item.
`UserStatus`	Not used.

FeedLayout Class

Represents the values that define the feed view of a feed-based page layout. Feed-based layouts are available on Account, Case, Contact, Lead, Opportunity, custom, and external objects. They include a feed view and a detail view.

Namespace

Metadata

Usage

Use this class when accessing `Metadata.Layout` metadata components. For more information, see "FeedLayout" in the *Metadata API Developer Guide*.

IN THIS SECTION:

FeedLayout Properties

FeedLayout Methods

FeedLayout Properties

The following are properties for `FeedLayout`.

IN THIS SECTION:

autocollapsePublisher

Specifies whether the publisher is collapsed when the page loads (true) or not (false).

compactFeed

Specifies whether the feed-based page layout uses a compact feed (true) or not (false). If set to true, feed items on the page are collapsed by default, and the feed view has an updated design.

feedFilterPosition

Indicates where the feed filters list is included in the layout.

feedFilters

The individual filters displayed in the feed filters list.

fullWidthFeed

Specifies whether the feed expands horizontally to take up all available space on the page (`true`) or not (`false`).

hideSidebar

Specifies whether the sidebar is hidden (`true`) or not (`false`).

highlightExternalFeedItems

Controls whether to highlight external feed items (true) or not (false).

leftComponents

The individual components displayed in the left column of the feed view.

rightComponents

Lists the individual components displayed in the right column of the feed view.

useInlineFiltersInConsole
Indicates whether to use inline filters in the Salesforce console.

autocollapsePublisher

Specifies whether the publisher is collapsed when the page loads (true) or not (false).

Signature

```
public Boolean autocollapsePublisher {get; set;}
```

Property Value

Type: Boolean

compactFeed

Specifies whether the feed-based page layout uses a compact feed (true) or not (false). If set to true, feed items on the page are collapsed by default, and the feed view has an updated design.

Signature

```
public Boolean compactFeed {get; set;}
```

Property Value

Type: Boolean

feedFilterPosition

Indicates where the feed filters list is included in the layout.

Signature

```
public Metadata.FeedLayoutFilterPosition feedFilterPosition {get; set;}
```

Property Value

Type: FeedLayoutFilterPosition Enum

feedFilters

The individual filters displayed in the feed filters list.

Signature

```
public List<Metadata.FeedLayoutFilter> feedFilters {get; set;}
```

Property Value

Type: List<FeedLayoutFilter Class>.

`fullWidthFeed`

Specifies whether the feed expands horizontally to take up all available space on the page (`true`) or not (`false`).

Signature

```
public Boolean fullWidthFeed {get; set;}
```

Property Value

Type: Boolean

`hideSidebar`

Specifies whether the sidebar is hidden (`true`) or not (`false`).

Signature

```
public Boolean hideSidebar {get; set;}
```

Property Value

Type: Boolean

`highlightExternalFeedItems`

Controls whether to highlight external feed items (true) or not (false).

Signature

```
public Boolean highlightExternalFeedItems {get; set;}
```

Property Value

Type: Boolean

`leftComponents`

The individual components displayed in the left column of the feed view.

Signature

```
public List<Metadata.FeedLayoutComponent> leftComponents {get; set;}
```

Property Value

Type: List<FeedLayoutComponent Class>

`rightComponents`

Lists the individual components displayed in the right column of the feed view.

Signature

```
public List<Metadata.FeedLayoutComponent> rightComponents {get; set;}
```

Property Value

Type: List<FeedLayoutComponent Class>

`useInlineFiltersInConsole`

Indicates whether to use inline filters in the Salesforce console.

Signature

```
public Boolean useInlineFiltersInConsole {get; set;}
```

Property Value

Type: Boolean

FeedLayout Methods

The following are methods for `FeedLayout`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.FeedLayout`.

`clone()`

Makes a duplicate copy of the `Metadata.FeedLayout`.

Signature

```
public Object clone()
```

Return Value

Type: Object

FeedLayoutComponent Class

Represents a component in the feed view of a feed-based page layout.

Namespace

Metadata

Usage

Use this class when accessing `Metadata.Layout` metadata components. For more information, see "FeedLayoutComponent" in the *Metadata API Developer Guide*.

IN THIS SECTION:

FeedLayoutComponent Properties

FeedLayoutComponent Methods

FeedLayoutComponent Properties

The following are properties for `FeedLayoutComponent`.

See `FeedLayoutComponent` in the *Metadata API Developer Guide*

IN THIS SECTION:

componentType
Represents a component in the feed view of a feed-based page layout. The type of component is required.

height
The height, in pixels, of the component. Doesn't apply to `standardComponents`

page_x
The name of the Visualforce page used as a custom component.

`componentType`

Represents a component in the feed view of a feed-based page layout. The type of component is required.

Signature

```
public Metadata.FeedLayoutComponentType componentType {get; set;}
```

Property Value

Type: Metadata.FeedLayoutComponentType on page 2779

`height`

The height, in pixels, of the component. Doesn't apply to `standardComponents`

Signature

```
public Integer height {get; set;}
```

Property Value

Type: Integer

page_x

The name of the Visualforce page used as a custom component.

Signature

```
public String page_x {get; set;}
```

Property Value

Type: String

FeedLayoutComponent Methods

The following are methods for `FeedLayoutComponent`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.FeedLayoutComponent`.

clone()

Makes a duplicate copy of the `Metadata.FeedLayoutComponent`.

Signature

```
public Object clone()
```

Return Value

Type: Object

FeedLayoutComponentType Enum

Indicates the type of feed layout component.

Enum Values

The following are the values of the `Metadata.FeedLayoutComponentType` enum.

Value	Description
`CaseExperts`	List of case experts.
`CaseUnifiedFiles`	List of all files attached to the case.
`CustomButtons`	Custom button.
`CustomLinks`	Custom link.
`Followers`	List of followers.

Value	Description
`Following`	Icon that toggles between a Follow button (if the user viewing a record doesn't already follow it) and a Following indicator (if the user viewing a record does follow it).
`HelpAndToolLinks`	Icons that link to the help topic for the page, the page layout, and, the printable view of the page. Available only on Case layouts.
`Milestones`	Milestone tracker, which lets users see the status of a milestone on a case. Available only on Case layouts.
`SimilarCases`	List of similar cases.
`Topics`	List of topics related to the record.
`Visualforce`	Custom Visualforce component.

FeedLayoutFilter Class

Represents a feed filter option in the feed view of a feed-based page layout. A filter can have only `standardFilter` or `feedItemType` set.

Namespace

Metadata

Usage

Use this class when accessing `Metadata.Layout` metadata components. For more information, see "FeedLayoutFilter" in the *Metadata API Developer Guide*.

IN THIS SECTION:

- FeedLayoutFilter Properties
- FeedLayoutFilter Methods

FeedLayoutFilter Properties

The following are properties for `FeedLayoutFilter`.

IN THIS SECTION:

feedFilterName
The name of a `CustomFeedFilter` component. Names are prefixed with the name of the parent object. For example, `Case.MyCustomFeedFilter`.

feedFilterType
The type of filter.

feedItemType
The type of feed item to display.

feedFilterName

The name of a `CustomFeedFilter` component. Names are prefixed with the name of the parent object. For example, `Case.MyCustomFeedFilter`.

Signature

```
public String feedFilterName {get; set;}
```

Property Value

Type: String

feedFilterType

The type of filter.

Signature

```
public Metadata.FeedLayoutFilterType feedFilterType {get; set;}
```

Property Value

Type: FeedLayoutFilterType Enum

feedItemType

The type of feed item to display.

Signature

```
public Metadata.FeedItemTypeEnum feedItemType {get; set;}
```

Property Value

Type: FeedItemTypeEnum Enum

FeedLayoutFilter Methods

The following are methods for `FeedLayoutFilter`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.FeedLayoutFilter`.

`clone()`

Makes a duplicate copy of the `Metadata.FeedLayoutFilter`.

Signature

```
public Object clone()
```

Return Value

Type: Object

FeedLayoutFilterPosition Enum

Describes where the feed filters list is included in the layout.

Enum Values

The following are the values of the `Metadata.FeedLayoutFilterPosition` enum.

Value	Description
`CenterDropDown`	As a drop-down list in the center column.
`LeftFixed`	As a fixed list in the left column.
`LeftFloat`	As a floating list in the left column.

FeedLayoutFilterType Enum

The type of feed layout filter.

Enum Values

The following are the values of the `Metadata.FeedLayoutFilterType` enum.

Value	Description
`AllUpdates`	Shows all feed items on a record.
`Custom`	Shows custom feed items.
`FeedItemType`	Shows feed items only for a particular type of activity on the record.

Layout Class

Represents the metadata associated with a page layout.

Namespace

Metadata

Usage

Use this class to access layout metadata components. For more information, see Layout in the *Metadata API Developer Guide*.

IN THIS SECTION:

Layout Properties

Layout Methods

Layout Properties

The following are properties for `Layout`.

IN THIS SECTION:

customButtons
The custom buttons for this layout.

customConsoleComponents
Represents custom console components (Visualforce pages, lookup fields, or related lists) on a page layout.

emailDefault
Default value for the email checkbox. Only relevant if the `showEmailCheckbox` property is set.

excludeButtons
List of standard buttons to exclude from this layout.

feedLayout
Represents the values that define the feed view of a feed-based page layout.

headers
Represents the layout headers used for tagging.

layoutSections
The main sections of the layout containing fields, s-controls, and custom links. The order here determines the layout order.

miniLayout
Represents a minilayout, which is used in the mini view of a record in the Console tab, hover details, and event overlays.

multilineLayoutFields
Fields for special multiline layout fields which appear in OpportunityProduct layouts.

platformActionList
The list of actions, and their order, that display in the Salesforce mobile action bar for the layout.

quickActionList
The list of quick actions that display in the full Salesforce site for the page layout.

relatedContent
The Related Content section of the page layout.

relatedLists
The related lists for the layout, listed in the order they appear in the user interface.

relatedObjects
The list of related objects that appears in the mini view of the console.

runAssignmentRulesDefault
Default value for the "run assignment rules" checkbox. Only relevant if the `showRunAssignmentRulesCheckbox` property is set.

showEmailCheckbox
Controls whether to show the email checkbox. Only allowed on Case, CaseClose, and Task layouts. The default state of checkbox is controlled by the `emailDefault` property.

showHighlightsPanel
If set, the highlights panel displays on pages in the Salesforce console.

showInteractionLogPanel
If set, the interaction log displays on pages in the Salesforce console.

showKnowledgeComponent
Only allowed on Case layouts. If set, the Knowledge sidebar displays on cases in the Salesforce console.

showRunAssignmentRulesCheckbox
Controls whether to show the Run Assignment Rules checkbox. Only allowed on Lead and Case layouts. The default state of checkbox is controlled by the `runAssignmentRulesDefault` property.

showSolutionSection
Only allowed on CaseClose layout. If set, the built-in solution information section shows up on the page.

showSubmitAndAttachButton
For Cast layouts only. If set, the Submit & Add Attachment button displays on case edit pages to portal users in the Customer Portal.

summaryLayout
The summary layout for this layout.

customButtons

The custom buttons for this layout.

Signature

```
public List<String> customButtons {get; set;}
```

Property Value

Type: List<String>

customConsoleComponents

Represents custom console components (Visualforce pages, lookup fields, or related lists) on a page layout.

Signature

```
public Metadata.CustomConsoleComponents customConsoleComponents {get; set;}
```

Property Value

Type: CustomConsoleComponents Class

`emailDefault`

Default value for the email checkbox. Only relevant if the `showEmailCheckbox` property is set.

Signature

```
public Boolean emailDefault {get; set;}
```

Property Value

Type: Boolean

`excludeButtons`

List of standard buttons to exclude from this layout.

Signature

```
public List<String> excludeButtons {get; set;}
```

Property Value

Type: List<String>

`feedLayout`

Represents the values that define the feed view of a feed-based page layout.

Signature

```
public Metadata.FeedLayout feedLayout {get; set;}
```

Property Value

Type: Metadata.FeedLayout

`headers`

Represents the layout headers used for tagging.

Signature

```
public List<Metadata.LayoutHeader> headers {get; set;}
```

Property Value

Type: List<Metadata.LayoutHeader>

layoutSections

The main sections of the layout containing fields, s-controls, and custom links. The order here determines the layout order.

Signature

```
public List<Metadata.LayoutSection> layoutSections {get; set;}
```

Property Value

Type: List<Metadata.LayoutSection>

miniLayout

Represents a minilayout, which is used in the mini view of a record in the Console tab, hover details, and event overlays.

Signature

```
public Metadata.MiniLayout miniLayout {get; set;}
```

Property Value

Type: Metadata.MiniLayout

multilineLayoutFields

Fields for special multiline layout fields which appear in OpportunityProduct layouts.

Signature

```
public List<String> multilineLayoutFields {get; set;}
```

Property Value

Type: List<String>

platformActionList

The list of actions, and their order, that display in the Salesforce mobile action bar for the layout.

Signature

```
public Metadata.PlatformActionList platformActionList {get; set;}
```

Property Value

Type: Metadata.PlatformActionList

quickActionList

The list of quick actions that display in the full Salesforce site for the page layout.

Signature

```
public Metadata.QuickActionList quickActionList {get; set;}
```

Property Value

Type: Meatadata.QuickActionL.

`relatedContent`

The Related Content section of the page layout.

Signature

```
public Metadata.RelatedContent relatedContent {get; set;}
```

Property Value

Type: Metadata.RelatedContent

`relatedLists`

The related lists for the layout, listed in the order they appear in the user interface.

Signature

```
public List<Metadata.RelatedListItem> relatedLists {get; set;}
```

Property Value

Type: List<Metadata.RelatedListItem>

`relatedObjects`

The list of related objects that appears in the mini view of the console.

Signature

```
public List<String> relatedObjects {get; set;}
```

Property Value

Type: List<String>

`runAssignmentRulesDefault`

Default value for the “run assignment rules” checkbox. Only relevant if the `showRunAssignmentRulesCheckbox` property is set.

Signature

```
public Boolean runAssignmentRulesDefault {get; set;}
```

Property Value

Type: Boolean

`showEmailCheckbox`

Controls whether to show the email checkbox. Only allowed on Case, CaseClose, and Task layouts. The default state of checkbox is controlled by the `emailDefault` property.

Signature

```
public Boolean showEmailCheckbox {get; set;}
```

Property Value

Type: Boolean

`showHighlightsPanel`

If set, the highlights panel displays on pages in the Salesforce console.

Signature

```
public Boolean showHighlightsPanel {get; set;}
```

Property Value

Type: Boolean

`showInteractionLogPanel`

If set, the interaction log displays on pages in the Salesforce console.

Signature

```
public Boolean showInteractionLogPanel {get; set;}
```

Property Value

Type: Boolean

`showKnowledgeComponent`

Only allowed on Case layouts. If set, the Knowledge sidebar displays on cases in the Salesforce console.

Signature

```
public Boolean showKnowledgeComponent {get; set;}
```

Property Value

Type: Boolean

showRunAssignmentRulesCheckbox

Controls whether to show the Run Assignment Rules checkbox. Only allowed on Lead and Case layouts. The default state of checkbox is controlled by the `runAssignmentRulesDefault` property.

Signature

```
public Boolean showRunAssignmentRulesCheckbox {get; set;}
```

Property Value

Type: Boolean

showSolutionSection

Only allowed on CaseClose layout. If set, the built-in solution information section shows up on the page.

Signature

```
public Boolean showSolutionSection {get; set;}
```

Property Value

Type: Boolean

showSubmitAndAttachButton

For Cast layouts only. If set, the Submit & Add Attachment button displays on case edit pages to portal users in the Customer Portal.

Signature

```
public Boolean showSubmitAndAttachButton {get; set;}
```

Property Value

Type: Boolean

summaryLayout

The summary layout for this layout.

Signature

```
public Metadata.SummaryLayout summaryLayout {get; set;}
```

Property Value

Type: Metadata.SummaryLayout

Layout Methods

The following are methods for `Layout`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.Layout`.

`clone()`

Makes a duplicate copy of the `Metadata.Layout`.

Signature

```
public Object clone()
```

Return Value

Type: Object

LayoutColumn Class

Represents the items in a column within a layout section.

Namespace

Metadata

Usage

Use this class when accessing `Metadata.Layout` metadata components. For more information, see "LayoutColumn" in the *Metadata API Developer Guide*.

IN THIS SECTION:

LayoutColumn Properties
LayoutColumn Methods

LayoutColumn Properties

The following are properties for `LayoutColumn`.

IN THIS SECTION:

layoutItems
The individual items within a column (ordered from top to bottom).

reserved
This field is reserved for Salesforce.

`layoutItems`

The individual items within a column (ordered from top to bottom).

Signature

```
public List<Metadata.LayoutItem> layoutItems {get; set;}
```

Property Value

Type: List<Metadata.LayoutItem>

`reserved`

This field is reserved for Salesforce.

Signature

```
public String reserved {get; set;}
```

Property Value

Type: String

LayoutColumn Methods

The following are methods for `LayoutColumn`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.LayoutColumn`.

`clone()`

Makes a duplicate copy of the `Metadata.LayoutColumn`.

Signature

```
public Object clone()
```

Return Value

Type: Object

LayoutHeader Enum

Represents tagging types used for `Metadata.Layout.headers`

Enum Values

The following are the values of the `Metadata.LayoutHeader` enum.

Value	Description
`PersonalTagging`	Tag is set to private user.
`PublicTagging`	Tag is viewable to any user who can access the record.

LayoutItem Class

Represents the valid values that define a layout item.

Namespace

Metadata

Usage

Use this class when accessing `Metadata.Layout` metadata components. For more information, see "LayoutItem" in the *Metadata API Developer Guide*.

IN THIS SECTION:

LayoutItem Properties

LayoutItem Methods

LayoutItem Properties

The following are properties for `LayoutItem`.

IN THIS SECTION:

analyticsCloudComponent

A Wave Analytics dashboard component on a page.

behavior
Determines the field behavior.

canvas
References a canvas app.

component
References a component.

customLink
The custom link reference.

emptySpace
Controls if this layout item is a blank space.

field
The field name reference, relative to the layout, for example “Description” or “MyField__c”.

height
For s-controls and pages only, the height in pixels.

page_x
Reference to a Visualforce page.

reportChartComponent
Refers to a report chart that you can add to a standard or custom object page.

scontrol
Reference to an s-control.

showLabel
For s-control and pages only, whether to show the label.

showScrollbars
For s-control and pages only, whether to show scrollbars.

width
For s-control and pages only, the width in pixels or percent. Pixel values are simply the number of pixels, for example, 500. Percentage values must include the percent sign, for example, 20%.

`analyticsCloudComponent`

A Wave Analytics dashboard component on a page.

Signature

```
public Metadata.AnalyticsCloudComponentLayoutItem analyticsCloudComponent {get; set;}
```

Property Value

Type: Metadata.AnalyticsCloudComponentLayoutItem

`behavior`

Determines the field behavior.

Signature

```
public Metadata.UiBehavior behavior {get; set;}
```

Property Value

Type: Metadata.UiBehavior

`canvas`

References a canvas app.

Signature

```
public String canvas {get; set;}
```

Property Value

Type: String

`component`

References a component.

Signature

```
public String component {get; set;}
```

Property Value

Type: String

`customLink`

The custom link reference.

Signature

```
public String customLink {get; set;}
```

Property Value

Type: String

`emptySpace`

Controls if this layout item is a blank space.

Signature

```
public Boolean emptySpace {get; set;}
```

Property Value

Type: Boolean

field

The field name reference, relative to the layout, for example “Description” or “MyField__c”.

Signature

```
public String field {get; set;}
```

Property Value

Type: String

height

For s-controls and pages only, the height in pixels.

Signature

```
public Integer height {get; set;}
```

Property Value

Type: Integer

page_x

Reference to a Visualforce page.

Signature

```
public String page_x {get; set;}
```

Property Value

Type: String

reportChartComponent

Refers to a report chart that you can add to a standard or custom object page.

Signature

```
public Metadata.ReportChartComponentLayoutItem reportChartComponent {get; set;}
```

Property Value

Type: Metadata.ReportChartComponentLayoutItem

scontrol

Reference to an s-control.

Signature

```
public String scontrol {get; set;}
```

Property Value

Type: String

showLabel

For s-control and pages only, whether to show the label.

Signature

```
public Boolean showLabel {get; set;}
```

Property Value

Type: Boolean

showScrollbars

For s-control and pages only, whether to show scrollbars.

Signature

```
public Boolean showScrollbars {get; set;}
```

Property Value

Type: Boolean

width

For s-control and pages only, the width in pixels or percent. Pixel values are simply the number of pixels, for example, 500. Percentage values must include the percent sign, for example, 20%.

Signature

```
public String width {get; set;}
```

Property Value

Type: String

LayoutItem Methods

The following are methods for `LayoutItem`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.LayoutItem`.

`clone()`

Makes a duplicate copy of the `Metadata.LayoutItem`.

Signature

```
public Object clone()
```

Return Value

Type: Object

LayoutSection Class

Represents a section of a page layout, such as the Custom Links section.

Namespace

Metadata

Usage

Use this class when accessing `Metadata.Layout` metadata components. For more information, see "LayoutSection" in the *Metadata API Developer Guide*.

IN THIS SECTION:

LayoutSection Properties

LayoutSection Methods

LayoutSection Properties

The following are properties for `LayoutSection`.

IN THIS SECTION:

customLabel
Indicates if this section's label is custom or standard (built-in).

detailHeading
Controls whether this heading appears on the detail page.

editHeading
Controls whether this heading appears on the edit page.

label
The label; either standard or custom, based on the customLabel property.

layoutColumns
Lists the layout columns. You can have one, two, or three columns, ordered left to right, are possible.

style
The style of the layout for this section.

customLabel

Indicates if this section's label is custom or standard (built-in).

Signature

```
public Boolean customLabel {get; set;}
```

Property Value

Type: Boolean

detailHeading

Controls whether this heading appears on the detail page.

Signature

```
public Boolean detailHeading {get; set;}
```

Property Value

Type: Boolean

editHeading

Controls whether this heading appears on the edit page.

Signature

```
public Boolean editHeading {get; set;}
```

Property Value

Type: Boolean

label

The label; either standard or custom, based on the customLabel property.

Signature

```
public String label {get; set;}
```

Property Value

Type: String

`layoutColumns`

Lists the layout columns. You can have one, two, or three columns, ordered left to right, are possible.

Signature

```
public List<Metadata.LayoutColumn> layoutColumns {get; set;}
```

Property Value

Type: List<Metadata.LayoutColumn>

`style`

The style of the layout for this section.

Signature

```
public Metadata.LayoutSectionStyle style {get; set;}
```

Property Value

Type: Metadata.LayoutSectionStyle

LayoutSection Methods

The following are methods for `LayoutSection`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.LayoutSection`.

`clone()`

Makes a duplicate copy of the `Metadata.LayoutSection`.

Signature

```
public Object clone()
```

Return Value

Type: Object

LayoutSectionStyle Enum

Describes the possible styles for a layout section.

Enum Values

The following are the values of the `Metadata.LayoutSectionStyle` enum.

Value	Description
`CustomLinks`	Contains custom links only
`OneColumn`	One column
`TwoColumnsLeftToRight`	Two columns, tab goes left to right
`TwoColumnsTopToBottom`	Two columns, tab goes top to bottom

Metadata Class

An abstract base class that represents a custom metadata component.

Namespace

Metadata

Usage

You can't create instances of this abstract class. Instead, create an instance of a specific custom metadata component class that derives from `Metadata.Metadata`, such as `Metadata.CustomMetadata`. For more information, see Metadata in the *Metadata API Developer Guide*.

IN THIS SECTION:

Metadata Properties

Metadata Methods

Metadata Properties

The following are properties for `Metadata`.

IN THIS SECTION:

fullName

The full name of the custom metadata, which can include the namespace, type, and component name.

`fullName`

The full name of the custom metadata, which can include the namespace, type, and component name.

Signature

```
public String fullName {get; set;}
```

Property Value

Type: String

The format of the full name can include the namespace, metadata type, and metadata component name. If you're updating components in a namespace, you also need to qualify the namespace for the component in the full name. For example, the full name for a custom metadata "MDType1__mdt" component named "Component1" that is contained in the "myPackage" namespace is "myPackage__MDType1__mdt.myPackage__Component1". For more information on full name formats for different metadata types, see reference documentation on the metadata types in the *Metadata API Developer Guide*.

Metadata Methods

The following are methods for `Metadata`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.Metadata`.

`clone()`

Makes a duplicate copy of the `Metadata.Metadata`.

Signature

```
public Object clone()
```

Return Value

Type: Object

MetadataType Enum

Represents the custom metadata components available in Apex.

Enum Values

The following are the values of the `Metadata.MetadataType` enum.

Value	Description
`CustomMetadata`	Records of custom metadata types

Value	Description
`Layout`	Layouts

MetadataValue Class

An abstract base class that represents a custom metadata component field.

Namespace

Metadata

Usage

You can't create instances of this abstract class. Instead, create an instance of a specific custom metadata component value class that derives from `Metadata.MetadataValue`, such as `Metadata.CustomMetadataValue`.

IN THIS SECTION:

MetadataValue Methods

MetadataValue Methods

The following are methods for `MetadataValue`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.MetadataValue`.

`clone()`

Makes a duplicate copy of the `Metadata.MetadataValue`.

Signature

```
public Object clone()
```

Return Value

Type: Object

MiniLayout Class

Represents a mini view of a record in the Console tab, hover details, and event overlays.

Namespace

Metadata

Usage

Use this class when accessing `Metadata.Layout` metadata components. For more information, see "MiniLayout" in the *Metadata API Developer Guide*.

IN THIS SECTION:

MiniLayout Properties

MiniLayout Methods

MiniLayout Properties

The following are properties for `MiniLayout`.

IN THIS SECTION:

fields

The fields for the mini-layout, listed in the order they appear in the UI. Fields that appear in the mini-layout must appear in the main layout.

relatedLists

The mini related lists, listed in the order they appear in the UI. You cannot set sorting on mini related lists. Fields that appear in the mini related lists must appear in the main layout.

`fields`

The fields for the mini-layout, listed in the order they appear in the UI. Fields that appear in the mini-layout must appear in the main layout.

Signature

```
public List<String> fields {get; set;}
```

Property Value

Type: List<String>

`relatedLists`

The mini related lists, listed in the order they appear in the UI. You cannot set sorting on mini related lists. Fields that appear in the mini related lists must appear in the main layout.

Signature

```
public List<Metadata.RelatedListItem> relatedLists {get; set;}
```

Property Value

Type: List<Metadata.RelatedListItem>

MiniLayout Methods

The following are methods for `MiniLayout`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.MiniLayout`.

`clone()`

Makes a duplicate copy of the `Metadata.MiniLayout`.

Signature

```
public Object clone()
```

Return Value

Type: Object

Operations Class

Represents a class to execute metadata operations, such as retrieving or deploying custom metadata.

Namespace

Metadata

Usage

Use the `Metadata.Operations` class to execute metadata operations. For more information on use cases and restrictions of metadata operations in Apex, see Metadata.

Example: Retrieve Metadata

The following example retrieves the "MyTestCustomMDType" custom metadata record from the subscriber org, and inspects the custom fields.

```
public class ReadMetadata {
  public void retrieveMetadata () {
    // List fullnames of components we want to retrieve
    List<String> componentNameList =
new List<String>{'ISVNamespace__TestCustomMDType.MyTestCustomMDType'};

    // Retrieve components that are records of custom metadata types
```

```
    // based on name
    List<Metadata.Metadata> components = Metadata.Operations.retrieve(
Metadata.MetadataType.CustomMetadata, componentNameList);
    Metadata.CustomMetadata customMetadataRecord = (Metadata.CustomMetadata)
components.get(0);

    // Check fields of retrieved component
    List<Metadata.CustomMetadataValue> values = customMetadataRecord.values;
    for (integer i = 0; i < values.size(); i++) {
      if (values.get(i).field == 'testField__c' &&
          values.get(i).value == 'desired value') {
        // ...process accordingly...
      }
    }
  }
}
```

Example: Deploy Metadata

The following example uses the Metadata API in Apex to update the customField custom field value of the MetadataRecordName custom metadata record and deploy this change into the subscriber org. Because the deployment is asynchronous, you must provide a callback class that implements the `Metadata.DeployCallback` interface, which is then used when the queued deployment completes.

```
public class CreateMetadata{
  public void updateAndDeployMetadata() {
    // Setup custom metadata to be created in the subscriber org.
    Metadata.CustomMetadata customMetadata =  new Metadata.CustomMetadata();
    customMetadata.fullName = 'ISVNamespace__MetadataTypeName.MetadataRecordName';

    Metadata.CustomMetadataValue customField = new Metadata.CustomMetadataValue();
    customField.field = 'customField__c';
    customField.value = 'New value';

    customMetadata.values.add(customField);

    Metadata.DeployContainer mdContainer = new Metadata.DeployContainer();
    mdContainer.addMetadata(customMetadata);

    // Setup deploy callback, MyDeployCallback implements
    // the Metadata.DeployCallback interface (code for
    // this class not shown in this example)
    MyDeployCallback callback = new MyDeployCallback();

    // Enqueue custom metadata deployment
    Id jobId = Metadata.Operations.enqueueDeployment(mdContainer, callback);
  }
}
```

Example: Create Two Metadata Records Synchronously

Create a metadata record along with another one that references it in the same transaction. If the parent record was installed with a namespace, prefix the developer name with *recordNs*__.

Note: No custom metadata relationship can relate records of the same type to each other.

```
public class CreateMetadata {

    public Id doCreate(
        String parentRecDevName,
        String parentRecLabel,
        String childRecDevName,
        String childRecLabel) {

        Metadata.DeployContainer mdContainer = new Metadata.DeployContainer();

        Metadata.CustomMetadata parentRecord = new Metadata.CustomMetadata();
        parentRecord.fullName = 'ParentType.' + parentRecDevName;
        parentRecord.label = parentRecLabel;
        mdContainer.addMetadata(parentRecord);

        Metadata.CustomMetadata childRecord = new Metadata.CustomMetadata();
        childRecord.fullName = 'ChildType.' + childRecDevName;
        childRecord.label = childRecLabel;
        Metadata.CustomMetadataValue relValue = new Metadata.CustomMetadataValue();
        relValue.field = 'Parent__c';
        relValue.value = parentRecDevName;
        childRecord.values.add(relValue);
        mdContainer.addMetadata(childRecord);

        Id jobId = Metadata.Operations.enqueueDeployment(mdContainer, null);
        return jobId;
    }

}
```

IN THIS SECTION:

Operations Methods

Operations Methods

The following are methods for `Operations`.

IN THIS SECTION:

clone()

Makes a duplicate copy of the `Metadata.Operations`.

enqueueDeployment(container, callback)

Deploys custom metadata components asynchronously.

retrieve(type, fullNames)

Retrieves a list of custom metadata components.

clone()

Makes a duplicate copy of the `Metadata.Operations`.

Signature

```
public Object clone()
```

Return Value

Type: Object

enqueueDeployment(container, callback)

Deploys custom metadata components asynchronously.

Signature

To preserve service function, we limit the number of Metadata API deployments originating from Apex that can be enqueued at a time. See Limit on Enqueued Deployments from Apex.

```
public static Id enqueueDeployment(Metadata.DeployContainer container,
Metadata.DeployCallback callback)
```

Parameters

container
: Type: Metadata.DeployContainer

  Container that contains the set of metadata components to deploy.

callback
: Type: Metadata.DeployCallback

  A class that implements the `Metadata.DeployCallback` interface. Used by Salesforce to return information about the deployment results.

Return Value

Type: Id

ID of deployment request.

retrieve(type, fullNames)

Retrieves a list of custom metadata components.

Signature

```
public static List<Metadata.Metadata> retrieve(Metadata.MetadataType type, List<String>
fullNames)
```

Parameters

type
Type: Metadata.MetadataType

The metadata component type.

fullNames
Type: List<String>

A list of component names to retrieve. For information on component name formats, see Metadata.fullName().

Return Value

Type: List<Metadata.Metadata>

PlatformActionList Class

Represents the list of actions, and their order, that display in the Salesforce mobile action bar for the layout.

Namespace

Metadata

Usage

Use this class when accessing `Metadata.Layout` metadata components. For more information, see "PlatformActionList" in the *Metadata API Developer Guide*.

IN THIS SECTION:

PlatformActionList Properties

PlatformActionList Methods

PlatformActionList Properties

The following are properties for `PlatformActionList`.

IN THIS SECTION:

actionListContext
The context of the action list.

platformActionListItems
The actions in the platform action list.

relatedSourceEntity
When the `actionListContext` property is "RelatedList" or" "RelatedListRecord", this field represents the API name of the related list to which the action belongs.

actionListContext

The context of the action list.

Signature

```
public Metadata.PlatformActionListContextEnum actionListContext {get; set;}
```

Property Value

Type: Metadata.PlatformActionListContextEnum

platformActionListItems

The actions in the platform action list.

Signature

```
public List<Metadata.PlatformActionListItem> platformActionListItems {get; set;}
```

Property Value

Type: List<Metadata.PlatformActionListItem>

relatedSourceEntity

When the `actionListContext` property is "RelatedList" or" "RelatedListRecord", this field represents the API name of the related list to which the action belongs.

Signature

```
public String relatedSourceEntity {get; set;}
```

Property Value

Type: String

PlatformActionList Methods

The following are methods for `PlatformActionList`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.PlatformActionList`.

clone()

Makes a duplicate copy of the `Metadata.PlatformActionList`.

Signature

```
public Object clone()
```

Return Value

Type: Object

PlatformActionListContextEnum Enum

Describes the different contexts of action lists.

Enum Values

The following are the values of the `Metadata.PlatformActionListContextEnum` enum.

Value	Description
`ActionDefinition`	Action definition context.
`Assistant`	Assistant context.
`BannerPhoto`	Banner photo context.
`Chatter`	Chatter context.
`Dockable`	Dockable context.
`FeedElement`	Feed element context.
`Flexipage`	Flexipage context.
`Global_x`	Global context.
`ListView`	Listview context.
`ListViewDefinition`	Listview definition context.
`ListViewRecord`	Listview record context.
`Lookup`	Lookup context.
`MruList`	MRU list context.
`MruRow`	MRU row context.
`ObjectHomeChart`	Object home chart context.
`Photo`	Photo context
`Record`	Record context.
`RecordEdit`	Record edit context
`RelatedList`	Related list context.
`RelatedListRecord`	Related list record context.

PlatformActionListItem Class

Represents an action in the platform action list for a layout.

Namespace

Metadata

Usage

Use this class when accessing `Metadata.Layout` metadata components. For more information, see "PlatformActionListItem" in the *Metadata API Developer Guide*.

IN THIS SECTION:

PlatformActionListItem Properties

PlatformActionListItem Methods

PlatformActionListItem Properties

The following are properties for `PlatformActionListItem`.

IN THIS SECTION:

actionName
The API name for the action in the list.

actionType
The type of action.

sortOrder
The placement of the action in the list.

subtype
The subtype of the action.

`actionName`

The API name for the action in the list.

Signature

```
public String actionName {get; set;}
```

Property Value

Type: String

`actionType`

The type of action.

Signature

```
public Metadata.PlatformActionTypeEnum actionType {get; set;}
```

Property Value

Type: Metadata.PlatformActionTypeEnum

`sortOrder`

The placement of the action in the list.

Signature

```
public Integer sortOrder {get; set;}
```

Property Value

Type: Integer

`subtype`

The subtype of the action.

Signature

```
public String subtype {get; set;}
```

Property Value

Type: String

PlatformActionListItem Methods

The following are methods for `PlatformActionListItem`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.PlatformActionListItem`.

`clone()`

Makes a duplicate copy of the `Metadata.PlatformActionListItem`.

Signature

```
public Object clone()
```

Return Value

Type: Object

PlatformActionTypeEnum Enum

The type of action for a `PlatformActionListItem`.

Enum Values

The following are the values of the `Metadata.PlatformActionTypeEnum` enum.

Value	Description
`ActionLink`	An indicator on a feed element that targets an API, a web page, or a file, represented by a button in the Salesforce Chatter feed UI.
`CustomButton`	When clicked, opens a URL or a Visualforce page in a window or executes JavaScript.
`InvocableAction`	An invocable action such as posting to Chatter.
`ProductivityAction`	Productivity actions are predefined by Salesforce and are attached to a limited set of objects. You can't edit or delete productivity actions.
`QuickAction`	A global or object-specific action.
`StandardButton`	A predefined Salesforce button such as New, Edit, and Delete.

PrimaryTabComponents Class

Represents custom console components on primary tabs in the Salesforce console.

Namespace

Metadata

Usage

Use this class when accessing `Metadata.Layout` metadata components. For more information, see "PrimaryTabComponents" in the *Metadata API Developer Guide*.

IN THIS SECTION:

PrimaryTabComponents Properties

PrimaryTabComponents Methods

PrimaryTabComponents Properties

The following are properties for `PrimaryTabComponents`.

IN THIS SECTION:

component
Represents a custom console component (Visualforce page, lookup field, or related lists) on a section of a page layout.

containers
Represents a location and style in which to display more than one custom console component on the sidebars of the Salesforce console.

component

Represents a custom console component (Visualforce page, lookup field, or related lists) on a section of a page layout.

Signature

```
public List<Metadata.ConsoleComponent> component {get; set;}
```

Property Value

Type: List<Metadata.ConsoleComponent>

containers

Represents a location and style in which to display more than one custom console component on the sidebars of the Salesforce console.

Signature

```
public List<Metadata.Container> containers {get; set;}
```

Property Value

Type: List<Metadata.Container>

PrimaryTabComponents Methods

The following are methods for `PrimaryTabComponents`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.PrimaryTabComponents`.

clone()

Makes a duplicate copy of the `Metadata.PrimaryTabComponents`.

Signature

```
public Object clone()
```

Return Value

Type: Object

QuickActionList Class

Represents the list of actions associated with the page layout.

Namespace

Metadata

Usage

Use this class when accessing `Metadata.Layout` metadata components. For more information, see "QuickActionList" in the *Metadata API Developer Guide*.

IN THIS SECTION:

QuickActionList Properties

QuickActionList Methods

QuickActionList Properties

The following are properties for `QuickActionList`.

IN THIS SECTION:

quickActionListItems
List of `QuickActionList` objects.

`quickActionListItems`

List of `QuickActionList` objects.

Signature

```
public List<Metadata.QuickActionListItem> quickActionListItems {get; set;}
```

Property Value

Type: List<Metadata.QuickActionListItem>

QuickActionList Methods

The following are methods for `QuickActionList`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.QuickActionList`.

clone()

Makes a duplicate copy of the `Metadata.QuickActionList`.

Signature

```
public Object clone()
```

Return Value

Type: Object

QuickActionListItem Class

Represents an action in the `QuickActionList`.

Namespace

Metadata

Usage

Use this class when accessing `Metadata.Layout` metadata components. For more information, see "QuickActionListItem" in the *Metadata API Developer Guide*.

IN THIS SECTION:

QuickActionListItem Properties
QuickActionListItem Methods

QuickActionListItem Properties

The following are properties for `QuickActionListItem`.

IN THIS SECTION:

quickActionName
The API name of the action.

quickActionName

The API name of the action.

Signature

```
public String quickActionName {get; set;}
```

Property Value

Type: String

QuickActionListItem Methods

The following are methods for `QuickActionListItem`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.QuickActionListItem`.

`clone()`

Makes a duplicate copy of the `Metadata.QuickActionListItem`.

Signature

```
public Object clone()
```

Return Value

Type: Object

RelatedContent Class

Represents the Mobile Cards section of the page layout.

Namespace

Metadata

Usage

Use this class when accessing `Metadata.Layout` metadata components. For more information, see "RelatedContent" in the *Metadata API Developer Guide*.

IN THIS SECTION:

RelatedContent Properties

RelatedContent Methods

RelatedContent Properties

The following are properties for `RelatedContent`.

IN THIS SECTION:

relatedContentItems
A list of layout items in the Mobile Cards section of the page layout.

`relatedContentItems`

A list of layout items in the Mobile Cards section of the page layout.

Signature

```
public List<Metadata.RelatedContentItem> relatedContentItems {get; set;}
```

Property Value

Type: List<Metadata.RelatedContentItem>

RelatedContent Methods

The following are methods for `RelatedContent`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.RelatedContent`.

`clone()`

Makes a duplicate copy of the `Metadata.RelatedContent`.

Signature

```
public Object clone()
```

Return Value

Type: Object

RelatedContentItem Class

Represents an individual item in the `RelatedContent` list.

Namespace

Metadata

Usage

Use this class when accessing `Metadata.Layout` metadata components. For more information, see "RelatedContentItem" in the *Metadata API Developer Guide*.

IN THIS SECTION:

RelatedContentItem Properties

RelatedContentItem Methods

RelatedContentItem Properties

The following are properties for `RelatedContentItem`.

IN THIS SECTION:

layoutItem
An individual layout item in the Mobile Cards section.

`layoutItem`

An individual layout item in the Mobile Cards section.

Signature

```
public Metadata.LayoutItem layoutItem {get; set;}
```

Property Value

Type: Metadata.LayoutItem

RelatedContentItem Methods

The following are methods for `RelatedContentItem`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.RelatedContentItem`.

`clone()`

Makes a duplicate copy of the `Metadata.RelatedContentItem`.

Signature

```
public Object clone()
```

Return Value

Type: Object

RelatedList Class

Represents related list custom components on the sidebars of the Salesforce console.

Namespace

Metadata

Usage

Use this class when accessing `Metadata.Layout` metadata components. For more information, see "RelatedList" in the *Metadata API Developer Guide*.

IN THIS SECTION:

RelatedList Properties

RelatedList Methods

RelatedList Properties

The following are properties for `RelatedList`.

IN THIS SECTION:

hideOnDetail

When set to true, the related list is hidden from detail pages where it appears as a component to prevent duplicate information from showing.

name

The name of the component as it appears to console users.

`hideOnDetail`

When set to true, the related list is hidden from detail pages where it appears as a component to prevent duplicate information from showing.

Signature

```
public Boolean hideOnDetail {get; set;}
```

Property Value

Type: Boolean

name

The name of the component as it appears to console users.

Signature

```
public String name {get; set;}
```

Property Value

Type: String

RelatedList Methods

The following are methods for `RelatedList`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.RelatedList`.

clone()

Makes a duplicate copy of the `Metadata.RelatedList`.

Signature

```
public Object clone()
```

Return Value

Type: Object

RelatedListItem Class

Represents an item in the related list in a page layout.

Namespace

Metadata

Usage

Use this class when accessing `Metadata.Layout` metadata components. For more information, see "RelatedListItem" in the *Metadata API Developer Guide*.

IN THIS SECTION:

RelatedListItem Properties

RelatedListItem Methods

RelatedListItem Properties

The following are properties for `RelatedListItem`.

IN THIS SECTION:

customButtons
A list of custom buttons used in the related list.

excludeButtons
A list of excluded related-list buttons.

fields
A list of fields displayed in the related list. Uses aliases instead of field or API names.

quickActions
A list of quick actions used on the related list.

relatedList
The name of the related list.

sortField
The name of the field used for sorting.

sortOrder
When `sortField` is set, the `sortOrder` property determines the sort order.

customButtons

A list of custom buttons used in the related list.

Signature

```
public List<String> customButtons {get; set;}
```

Property Value

Type: List<String>

For more information, see "Define Custom Buttons and Links" in the Salesforce online help.

excludeButtons

A list of excluded related-list buttons.

Signature

```
public List<String> excludeButtons {get; set;}
```

Property Value

Type: List<String>

fields

A list of fields displayed in the related list. Uses aliases instead of field or API names.

Signature

```
public List<String> fields {get; set;}
```

Property Value

Type: List<String>

quickActions

A list of quick actions used on the related list.

Signature

```
public List<String> quickActions {get; set;}
```

Property Value

Type: List<String>

relatedList

The name of the related list.

Signature

```
public String relatedList {get; set;}
```

Property Value

Type: String

sortField

The name of the field used for sorting.

Signature

```
public String sortField {get; set;}
```

Property Value

Type: String

sortOrder

When `sortField` is set, the `sortOrder` property determines the sort order.

Signature

```
public Metadata.SortOrder sortOrder {get; set;}
```

Property Value

Type: Metadata.SortOrder

RelatedListItem Methods

The following are methods for `RelatedListItem`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.RelatedListItem`.

`clone()`

Makes a duplicate copy of the `Metadata.RelatedListItem`.

Signature

```
public Object clone()
```

Return Value

Type: Object

ReportChartComponentLayoutItem Class

Represents the settings for a report chart on a standard or custom page.

Namespace

Metadata

Usage

Use this class when accessing `Metadata.Layout` metadata components. For more information, see "ReportChartComponentLayoutItem" in the *Metadata API Developer Guide*.

IN THIS SECTION:

ReportChartComponentLayoutItem Properties

ReportChartComponentLayoutItem Methods

ReportChartComponentLayoutItem Properties

The following are properties for `ReportChartComponentLayoutItem`.

IN THIS SECTION:

cacheData

Indicates whether to use cached data when displaying the chart. When the attribute is set to true, data is cached for 24 hours. When the attribute is set to false, the report is run every time the page is refreshed.

contextFilterableField

Unique development name of the field by which a report chart is filtered to return data relevant to the page. If set, the ID field for the parent object of the page or report type is the chart data filter. The parent object for the report type and the page must match for a chart to return relevant data.

error

Error string that is populated only when an error occurs in the underlying report.

hideOnError

Controls whether users see a chart that has an error. When an error occurs and this attribute is not set, the chart doesn't show any data except the error. Set the attribute to true to hide the chart from a page on error.

includeContext

If true, filters the report chart to return data that's relevant to the page.

reportName

Unique development name of a report that includes a chart.

showTitle

If true, applies the title from the report to the chart.

size

Size of the displayed chart. The default is medium.

cacheData

Indicates whether to use cached data when displaying the chart. When the attribute is set to true, data is cached for 24 hours. When the attribute is set to false, the report is run every time the page is refreshed.

Signature

```
public Boolean cacheData {get; set;}
```

Property Value

Type: Boolean

contextFilterableField

Unique development name of the field by which a report chart is filtered to return data relevant to the page. If set, the ID field for the parent object of the page or report type is the chart data filter. The parent object for the report type and the page must match for a chart to return relevant data.

Signature

```
public String contextFilterableField {get; set;}
```

Property Value

Type: String

error

Error string that is populated only when an error occurs in the underlying report.

Signature

```
public String error {get; set;}
```

Property Value

Type: String

hideOnError

Controls whether users see a chart that has an error. When an error occurs and this attribute is not set, the chart doesn't show any data except the error. Set the attribute to true to hide the chart from a page on error.

Signature

```
public Boolean hideOnError {get; set;}
```

Property Value

Type: Boolean

includeContext

If true, filters the report chart to return data that's relevant to the page.

Signature

```
public Boolean includeContext {get; set;}
```

Property Value

Type: Boolean

reportName

Unique development name of a report that includes a chart.

Signature

```
public String reportName {get; set;}
```

Property Value

Type: String

`showTitle`

If true, applies the title from the report to the chart.

Signature

```
public Boolean showTitle {get; set;}
```

Property Value

Type: Boolean

`size`

Size of the displayed chart. The default is medium.

Signature

```
public Metadata.ReportChartComponentSize size {get; set;}
```

Property Value

Type: Metadata.ReportChartComponentSize

ReportChartComponentLayoutItem Methods

The following are methods for `ReportChartComponentLayoutItem`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.ReportChartComponentLayoutItem`.

`clone()`

Makes a duplicate copy of the `Metadata.ReportChartComponentLayoutItem`.

Signature

```
public Object clone()
```

Return Value

Type: Object

ReportChartComponentSize Enum

Describes the size of the displayed report chart component.

Enum Values

The following are the values of the `Metadata.ReportChartComponentSize` enum.

Value	Description
`LARGE`	Large chart size.
`MEDIUM`	Medium chart size.
`SMALL`	Small chart size.

SidebarComponent Class

Represents a specific custom console component to display in a container that hosts multiple components in one of the sidebars of the Salesforce console.

Namespace

Metadata

Usage

Use this class when accessing `Metadata.Layout` metadata components. For more information, see "SidebarComponent" in the *Metadata API Developer Guide*.

IN THIS SECTION:

SidebarComponent Properties

SidebarComponent Methods

SidebarComponent Properties

The following are properties for `SidebarComponent`.

IN THIS SECTION:

componentType

Specifies the component type. Valid values are "KnowledgeOne", "Lookup", "Milestones", "RelatedList", "Topics", "Files", and "CaseExperts".

createAction
If the component is a lookup field, the name of the quick action used to create a record.

enableLinking
If the component is a lookup field, lets users associate a record with this field.

height
The height of the component in the container. The `unit` property determines the unit of measurement, in pixels or percent.

knowledgeOneEnable
Indicates if the component is enabled for Knowledge One.

label
The name of the component as it displays to console users. Available for components in a container with the style of tabs or accordion.

lookup
If the component is a lookup field, the name of the field.

page_x
If the component is a Visualforce page, the name of the Visualforce page.

relatedLists
If the component is a related list component, the list of related list names.

unit
The unit of measurement (pixels or percent) for the height and width of the component in the container.

updateAction
If the component is a lookup field, the name of the quick action used to update a record.

width
The width of the component in the container. The `unit` property determines the unit of measurement, in pixels or percent.

componentType

Specifies the component type. Valid values are "KnowledgeOne", "Lookup", "Milestones", "RelatedList", "Topics", "Files", and "CaseExperts".

Signature

```
public String componentType {get; set;}
```

Property Value

Type: String

createAction

If the component is a lookup field, the name of the quick action used to create a record.

Signature

```
public String createAction {get; set;}
```

Property Value

Type: String

enableLinking

If the component is a lookup field, lets users associate a record with this field.

Signature

```
public Boolean enableLinking {get; set;}
```

Property Value

Type: Boolean

height

The height of the component in the container. The `unit` property determines the unit of measurement, in pixels or percent.

Signature

```
public Integer height {get; set;}
```

Property Value

Type: Integer

knowledgeOneEnable

Indicates if the component is enabled for Knowledge One.

Signature

```
public Boolean knowledgeOneEnable {get; set;}
```

Property Value

Type: Boolean

label

The name of the component as it displays to console users. Available for components in a container with the style of tabs or accordion.

Signature

```
public String label {get; set;}
```

Property Value

Type: String

lookup

If the component is a lookup field, the name of the field.

Signature

```
public String lookup {get; set;}
```

Property Value

Type: String

page_x

If the component is a Visualforce page, the name of the Visualforce page.

Signature

```
public String page_x {get; set;}
```

Property Value

Type: String

relatedLists

If the component is a related list component, the list of related list names.

Signature

```
public List<Metadata.RelatedList> relatedLists {get; set;}
```

Property Value

Type: List<Metadata.RelatedList>

unit

The unit of measurement (pixels or percent) for the height and width of the component in the container.

Signature

```
public String unit {get; set;}
```

Property Value

Type: String

updateAction

If the component is a lookup field, the name of the quick action used to update a record.

Signature

```
public String updateAction {get; set;}
```

Property Value

Type: String

`width`

The width of the component in the container. The `unit` property determines the unit of measurement, in pixels or percent.

Signature

```
public Integer width {get; set;}
```

Property Value

Type: Integer

SidebarComponent Methods

The following are methods for `SidebarComponent`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.SidebarComponent`.

`clone()`

Makes a duplicate copy of the `Metadata.SidebarComponent`.

Signature

```
public Object clone()
```

Return Value

Type: Object

SortOrder Enum

Describes the sort order of a related list.

Enum Values

The following are the values of the `Metadata.SortOrder` enum.

Value	Description
`Asc_x`	Sort in ascending order.
`Desc_x`	Sort in descending order.

StatusCode Enum

Describes the status code for an unsuccessful component deploy.

Enum Values

The following are the values of the `Metadata.StatusCode` enum.

Value	Description
`INVALID_SCS_INBOUND_USER`	Log in as the RunAs user configured in your SCS setup.
`REQUIRE_CONNECTED_APP_SCS`	SCS Connected App is not installed.
`REQUIRE_CONNECTED_APP_SESSION_SCS`	To use the SCS connected app, the user must be authenticated.
`REQUIRE_RUNAS_USER`	A RunAs user must be configured in your org.

SEE ALSO:

DeployProblemType Enum

SubtabComponents Class

Represents custom console components on subtabs in the Salesforce console.

Namespace

Metadata

Usage

Use this class when accessing `Metadata.Layout` metadata components. For more information, see "SubtabComponents" in the *Metadata API Developer Guide*.

IN THIS SECTION:

SubtabComponents Properties

SubtabComponents Methods

SubtabComponents Properties

The following are properties for `SubtabComponents`.

IN THIS SECTION:

component
Represents a custom console component (Visualforce page, lookup field, or related lists) on a section of a page layout.

containers
Represents a location and style in which to display more than one custom console component on the sidebars of the Salesforce console.

component

Represents a custom console component (Visualforce page, lookup field, or related lists) on a section of a page layout.

Signature

```
public List<Metadata.ConsoleComponent> component {get; set;}
```

Property Value

Type: List<Metadata.ConsoleComponent>

containers

Represents a location and style in which to display more than one custom console component on the sidebars of the Salesforce console.

Signature

```
public List<Metadata.Container> containers {get; set;}
```

Property Value

Type: List<Metadata.Container>

SubtabComponents Methods

The following are methods for `SubtabComponents`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.SubtabComponents`.

clone()

Makes a duplicate copy of the `Metadata.SubtabComponents`.

Signature

```
public Object clone()
```

Return Value

Type: Object

SummaryLayoutStyleEnum Enum

Describes the highlights panel style for a `SummaryLayout`.

Enum Values

The following are the values of the `Metadata.SummaryLayoutStyleEnum` enum.

Value	Description
`CaseInteraction`	Case interaction style.
`ChildServiceReportTemplateStyle`	Child service report template style.
`DefaultQuoteTemplate`	Default quote template style.
`DefaultServiceReportTemplate`	Default service report style
`Default_x`	Default style.
`PathAssistant`	Path assisstant style.
`QuickActionLayoutLeftRight`	Quick action left-right layout style.
`QuickActionLayoutTopDown`	Quick action top-down layout style.
`QuoteTemplate`	Quote template style.
`ServiceReportTemplate`	Service report style.

SummaryLayout Class

Controls the appearance of the highlights panel, which summarizes key fields in a grid at the top of a page layout, when Case Feed is enabled.

Namespace

Metadata

Usage

Use this class when accessing `Metadata.Layout` metadata components. For more information, see "SummaryLayout" in the *Metadata API Developer Guide*.

IN THIS SECTION:

SummaryLayout Properties

SummaryLayout Methods

SummaryLayout Properties

The following are properties for `SummaryLayout`.

IN THIS SECTION:

masterLabel
The name of the layout label.

sizeX
Number of columns in the highlights pane, between 1 and 4 (inclusive).

sizeY
Number of rows in each column, either 1 or 2.

sizeZ
If provided, the setting is not visible to users.

summaryLayoutItems
Controls the appearance of an individual field and its column and row position within the highlights panel grid, when Case Feed is enabled. At least one is required.

summaryLayoutStyle
Specifies the panel style.

masterLabel

The name of the layout label.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Signature

```
public String masterLabel {get; set;}
```

Property Value

Type: String

sizeX

Number of columns in the highlights pane, between 1 and 4 (inclusive).

Signature

```
public Integer sizeX {get; set;}
```

Property Value

Type: Integer

sizeY

Number of rows in each column, either 1 or 2.

Signature

```
public Integer sizeY {get; set;}
```

Property Value

Type: Integer

sizeZ

If provided, the setting is not visible to users.

Signature

```
public Integer sizeZ {get; set;}
```

Property Value

Type: Integer

summaryLayoutItems

Controls the appearance of an individual field and its column and row position within the highlights panel grid, when Case Feed is enabled. At least one is required.

Signature

```
public List<Metadata.SummaryLayoutItem> summaryLayoutItems {get; set;}
```

Property Value

Type: List<Metadata.SummaryLayoutItem>

summaryLayoutStyle

Specifies the panel style.

Signature

```
public Metadata.SummaryLayoutStyleEnum summaryLayoutStyle {get; set;}
```

Property Value

Type: Metadata.SummaryLayoutStyleEnum

SummaryLayout Methods

The following are methods for `SummaryLayout`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.SummaryLayout`.

`clone()`

Makes a duplicate copy of the `Metadata.SummaryLayout`.

Signature

```
public Object clone()
```

Return Value

Type: Object

SummaryLayoutItem Class

Controls the appearance of an individual field and its column and row position within the highlights panel grid, when Case Feed is enabled. You can have two fields per each grid in a highlights panel.

Namespace

Metadata

Usage

Use this class when accessing `Metadata.Layout` metadata components. For more information, see "SummaryLayoutItem" in the *Metadata API Developer Guide*.

IN THIS SECTION:

SummaryLayoutItem Properties

SummaryLayoutItem Methods

SummaryLayoutItem Properties

The following are properties for `SummaryLayoutItem`.

IN THIS SECTION:

customLink
The custom link reference.

field
The field name reference, relative to the page layout. Must be a standard or custom field that also exists on the detail page.

posX
The item's column position in the highlights panel grid. Must be within the range of `sizeX`.

posY
The item's row position in the highlights panel grid. Must be within the range of `sizeY`.

posZ
Reserved for future use. If provided, the setting is not visible to users.

customLink

The custom link reference.

Signature

```
public String customLink {get; set;}
```

Property Value

Type: String

field

The field name reference, relative to the page layout. Must be a standard or custom field that also exists on the detail page.

Signature

```
public String field {get; set;}
```

Property Value

Type: String

posX

The item's column position in the highlights panel grid. Must be within the range of `sizeX`.

Signature

```
public Integer posX {get; set;}
```

Property Value

Type: Integer

posY

The item's row position in the highlights panel grid. Must be within the range of `sizeY`.

Signature

```
public Integer posY {get; set;}
```

Property Value

Type: Integer

posZ

Reserved for future use. If provided, the setting is not visible to users.

Signature

```
public Integer posZ {get; set;}
```

Property Value

Type: Integer

SummaryLayoutItem Methods

The following are methods for `SummaryLayoutItem`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `Metadata.SummaryLayoutItem`.

clone()

Makes a duplicate copy of the `Metadata.SummaryLayoutItem`.

Signature

```
public Object clone()
```

Return Value

Type: Object

UiBehavior Enum

Describes the behavior for a layout item on a layout page.

Enum Values

The following are the values of the `Metadata.UiBehavior` enum.

Value	Description
`Edit`	The layout field can be edited but is not required.
`Readonly`	The layout field is read-only.
`Required`	The layout field can be edited and is required.

PlaceQuote Namespace

The `PlaceQuote` namespace provides classes and methods to create or update quotes with pricing preferences and configuration options.

See PlaceQuote namespace for more information about the available classes and methods.

Pref_center Namespace

The Pref_center namespace provides an interface, classes, and methods to create and retrieve data in forms in Preference Manager. Preference Manager, previously called Preference Center, is a feature within the Privacy Center app.

The following are the classes in the `Pref_center` namespace.

IN THIS SECTION:

LoadFormData Class
Retrieve records related to the tokenized record id, and populate the values of a preference form.

LoadParameters Class
Contains methods to retrieve record Id information for parameters passed into the load-form handler.

PreferenceCenterApexHandler Interface
Pass data between your organization and a form in Preference Manager.

SubmitFormData Class
Contains methods to retrieve information on buttons and options selected in a preference form.

SubmitParameters Class
Retrieve record ID information to use with your submit-form handler.

TokenType Enum
Defines the types of values supported by the TokenUtility methods.

TokenUtility Class
Generate authentication tokens to access preference forms.

ValidationResult Class
This class is reserved for future use with Preference Manager.

LoadFormData Class

Retrieve records related to the tokenized record id, and populate the values of a preference form.

Namespace

Pref_center

Example

Use methods in the `LoadFormData` class to set available and selected values in different form components:

```
List<System.SelectOption> picklistOptions = new List<System.SelectOption>();
picklistOptions.add(new System.SelectOption('optIn', 'Opt In'));
picklistOptions.add(new System.SelectOption('optOut', 'Opt Out'));

// Set the available options for the picklist
loadFormData.setOptions('myPicklist', picklistOptions);
// Add an option to the existing options for the picklist
loadFormData.addOption('myPicklist', 'optOutAll', 'Opt Out All');
// Select the 'optIn' option in the picklist
loadFormData.setSelectedOption('myPicklist', 'optIn');


List<System.SelectOption> checkboxOptions = new List<System.SelectOption>();
checkboxOptions.add(new System.SelectOption('yes', 'Yes'));
checkboxOptions.add(new System.SelectOption('no', 'No'));


// Set available options for the checkbox group
loadFormData.setOptions('myCheckbox', checkboxOptions);
// Select the 'yes' option in the checkbox group
loadFormData.addSelectedOption('myCheckbox', 'yes');
// Also select the 'no' option in the checkbox group
loadFormData.addSelectedOption('myCheckbox', 'no');
// Another way to select both the 'yes' and 'no' options in the checkbox group
loadFormData.setSelectedOptions('myCheckbox', new List<String>{'yes', 'no'});


// Fill the value in the text input
loadFormData.setTextValue('myTextInput', 'admin@salesforce.com');
// Set the hint text for the text input
loadFormData.setTextHint('myTextInput', 'Email Address');


// Set the label for the button
loadFormData.setButtonLabel('myButton', 'Save Preferences');
```

IN THIS SECTION:

LoadFormData Constructors

LoadFormData Methods

LoadFormData Constructors

The following are constructors for `LoadFormData`.

IN THIS SECTION:

LoadFormData(data)
Creates an instance of the `LoadFormData` class for running tests on any custom Apex classes you create for Preference Manager.

`LoadFormData(data)`

Creates an instance of the `LoadFormData` class for running tests on any custom Apex classes you create for Preference Manager.

Signature

```
public LoadFormData(Map<String,pref_center.FieldProperties> data)
```

Parameters

data

Type: Map<String,pref_center.FieldProperties>Map

Maps preference form data from the field ID to the field properties.

Usage

This constructor is available in API version 56.0 and later.

LoadFormData Methods

The following are methods for `LoadFormData`.

IN THIS SECTION:

addOption(fieldId, value, label)
Add an option for a checkbox, picklist, or radio button field in a preference form using the label and value.

addOption(fieldId, option)
Add a defined, selectable option for a checkbox, picklist, or radio button field in a preference form.

addSelectedOption(fieldId, option)
Add a selected option for a checkbox field in a preference form. This requires the field on the form to have a defined option with a set value.

setButtonLabel(fieldId, label)
Set the label of a button added to the preference form.

setOptions(fieldId, options)
Add a list of selectable options for a field on a preference form.

setSelectedOption(fieldId, optionValue)
For a picklist or radio button field on a preference form that has defined values, set the value entered in the optionValue field as the selected option.

setSelectedOptions(fieldId, options)
For an existing checkbox field on a preference form that has defined values, set the values entered in the options field as the selected options. This requires the field on the form to have defined options with a set value.

setTextHint(fieldId, hintText)

Set the hint text inside a text input field. The hint text tells the user what type of information to enter, like an email address.

setTextValue(fieldId, value)

Set the value of a text field in a preference form.

addOption(fieldId, value, label)

Add an option for a checkbox, picklist, or radio button field in a preference form using the label and value.

Signature

```
public void addOption(String fieldId, String value, String label)
```

Parameters

fieldId

Type: String

Identifies a field in the preference form.

value

Type: String

Represents the selection or text entered in a preference form field.

label

Type: String

The label for the value of a field in a preference form.

Return Value

Type: void

addOption(fieldId, option)

Add a defined, selectable option for a checkbox, picklist, or radio button field in a preference form.

Signature

```
public void addOption(String fieldId, System.SelectOption option)
```

Parameters

fieldId

Type: String

Identifies a field in the preference form.

option

Type: System.SelectOption

The option selected on a field in the preference form.

Return Value

Type: void

addSelectedOption(fieldId, option)

Add a selected option for a checkbox field in a preference form. This requires the field on the form to have a defined option with a set value.

Signature

```
public void addSelectedOption(String fieldId, String option)
```

Parameters

fieldId
Type: String

Identifies a field in the preference form.

option
Type: String

The selectable option being added.

Return Value

Type: void

setButtonLabel(fieldId, label)

Set the label of a button added to the preference form.

Signature

```
public void setButtonLabel(String fieldId, String label)
```

Parameters

fieldId
Type: String

Identifies a field in the preference form.

label
Type: String

The label for a button added to the preference form.

Return Value

Type: void

setOptions(fieldId, options)

Add a list of selectable options for a field on a preference form.

Signature

```
public void setOptions(String fieldId, List<System.SelectOption> options)
```

Parameters

fieldId
Type: String

Identifies a field in the preference form.

options
Type: List<System.SelectOption>

The selectable options for a field in the preference form.

Return Value

Type: void

setSelectedOption(fieldId, optionValue)

For a picklist or radio button field on a preference form that has defined values, set the value entered in the optionValue field as the selected option.

Signature

```
public void setSelectedOption(String fieldId, String optionValue)
```

Parameters

fieldId
Type: String

Identifies a field in the preference form.

optionValue
Type: String

The value for the selected option.

Return Value

Type: void

setSelectedOptions(fieldId, options)

For an existing checkbox field on a preference form that has defined values, set the values entered in the options field as the selected options. This requires the field on the form to have defined options with a set value.

Signature

```
public void setSelectedOptions(String fieldId, List<String> options)
```

Parameters

fieldId
Type: String

Identifies the checkbox field in the preference form.

options
Type: List<String>

The selected options for a field in the preference form.

Return Value

Type: void

setTextHint(fieldId, hintText)

Set the hint text inside a text input field. The hint text tells the user what type of information to enter, like an email address.

Signature

```
public void setTextHint(String fieldId, String hintText)
```

Parameters

fieldId
Type: String

The ID of the text input field in the preference form.

hintText
Type: String

The hint text in the text input field.

Return Value

Type: void

setTextValue(fieldId, value)

Set the value of a text field in a preference form.

Signature

```
public void setTextValue(String fieldId, String value)
```

Parameters

fieldId
Type: String

Identifies a field in the preference form.

value
Type: String

Represents the value entered for the text field in a preference form.

Return Value

Type: void

LoadParameters Class

Contains methods to retrieve record Id information for parameters passed into the load-form handler.

Namespace

Pref_center

Example

```
String userId = loadParams.getRecordId();

User user = [select id, AboutMe from User where id=:userId];
```

IN THIS SECTION:

LoadParameters Methods

LoadParameters Methods

The following are methods for `LoadParameters`.

IN THIS SECTION:

getRecordId()
Returns the untokenized version of the record Id.

`getRecordId()`

Returns the untokenized version of the record Id.

Signature

```
public String getRecordId()
```

Return Value

Type: String

PreferenceCenterApexHandler Interface

Pass data between your organization and a form in Preference Manager.

Namespace

Pref_center

IN THIS SECTION:

PreferenceCenterApexHandler Methods

PreferenceCenterApexHandler Methods

The following are methods for `PreferenceCenterApexHandler`.

IN THIS SECTION:

load(loadParams, formData, validationResult)
Retrieve the record IDs and initial values from a preference form before it is edited and submitted.

submit(loadParams, formData, validationResult)
Updates the changed values of fields when the preference form is submitted.

`load(loadParams, formData, validationResult)`

Retrieve the record IDs and initial values from a preference form before it is edited and submitted.

Signature

```
public pref_center.LoadFormData load(pref_center.LoadParameters loadParams,
pref_center.LoadFormData formData, pref_center.ValidationResult validationResult)
```

Parameters

`loadParams`
Type: pref_center.LoadParameters

Retrieve the tokenized record ID.

`formData`
Type: pref_center.LoadFormData

Set the initial values of fields in a form before they are edited.

`validationResult`
Type: pref_center.ValidationResult

Reserved for future use.

Return Value

Type: pref_center.LoadFormData

`submit(loadParams, formData, validationResult)`

Updates the changed values of fields when the preference form is submitted.

Signature

```
public void submit(pref_center.SubmitParameters submitParams, pref_center.SubmitFormData formData, pref_center.ValidationResult validationResult)
```

Parameters

`submitParams`
Type: pref_center.SubmitParameters

Retrieve the tokenized record Id.

`formData`
Type: pref_center.SubmitFormData

Retrieve the values of fields in a submitted form.

`validationResult`
Type: pref_center.ValidationResult

Reserved for future use.

Return Value

Type: void

SubmitFormData Class

Contains methods to retrieve information on buttons and options selected in a preference form.

Namespace

Pref_center

Example

Use methods in the `SubmitFormData` class to retrieve the selected values in different form components:

```
String buttonClickedId = formData.getButtonClicked();
if (buttonClickedId == 'submitButton') {
// Handle form submit
} else if (buttonClickedId == 'cancelButton') {
// Handle form cancel
}

String picklistValueOld = formData.getOldSelectedValue('myPicklist');
```

```
String picklistValueNew = formData.getSelectedValue('myPicklist');
if (picklistValueOld != picklistValueNew) {
// Do something
}

List<String> checkboxValuesOld = formData.getOldSelectedValues('myCheckbox');
List<String> checkboxValuesNew = formData.getSelectedValues('myCheckbox');
if (checkboxValuesOld != null && checkboxValuesNew != null && (checkboxValuesOld.size()
!= checkboxValuesNew.size())) {
// Do something
}

String textinputValueOld = formData.getOldStringValue('myTextinput');
String textinputValueNew = formData.getStringValue('myTextinput');
if (textinputValueOld != textinputValueNew) {
// Do something
}
```

IN THIS SECTION:

SubmitFormData Methods

SubmitFormData Methods

The following are methods for `SubmitFormData`.

IN THIS SECTION:

getButtonClicked()

Returns the field ID of the button that was clicked in the preference form. For example, use this method to determine if the clicked button was **Submit** or **Cancel**.

getOldSelectedValue(fieldId)

Returns the value that was set for the specified field when the preference form was previously edited by the user. This method is used for field types such as picklist or radio buttons.

getOldSelectedValues(fieldId)

Returns a list of the string values that were set on a checkbox field when the preference form was previously edited by the user.

getOldStringValue(fieldId)

Returns the string value that was set on a field when the preference form was loaded. This method is used for field types such as text, and throws a TypeException if used with a field that can return more than one value, like a checkbox field.

getSelectedValue(fieldId)

Returns the string value that is currently selected for a picklist or radio button field in the preference form.

getSelectedValues(fieldId)

Returns a list of string values that are currently selected on a checkbox field in the preference form.

getStringValue(fieldId)

Returns the string value that was set on a field when the preference form was loaded. This method is used for field types such as text.

getButtonClicked()

Returns the field ID of the button that was clicked in the preference form. For example, use this method to determine if the clicked button was **Submit** or **Cancel**.

Signature

```
public String getButtonClicked()
```

Return Value

Type: String

getOldSelectedValue(fieldId)

Returns the value that was set for the specified field when the preference form was previously edited by the user. This method is used for field types such as picklist or radio buttons.

Signature

```
public String getOldSelectedValue(String fieldId)
```

Parameters

fieldId
Type: String

Identifies a field in the preference form.

Return Value

Type: String

getOldSelectedValues(fieldId)

Returns a list of the string values that were set on a checkbox field when the preference form was previously edited by the user.

Signature

```
public List<String> getOldSelectedValues(String fieldId)
```

Parameters

fieldId
Type: String

Identifies a field in the preference form.

Return Value

Type: List<String>

getOldStringValue(fieldId)

Returns the string value that was set on a field when the preference form was loaded. This method is used for field types such as text, and throws a TypeException if used with a field that can return more than one value, like a checkbox field.

Signature

```
public String getOldStringValue(String fieldId)
```

Parameters

fieldId
Type: String

Identifies a field in the preference form.

Return Value

Type: String

getSelectedValue(fieldId)

Returns the string value that is currently selected for a picklist or radio button field in the preference form.

Signature

```
public String getSelectedValue(String fieldId)
```

Parameters

fieldId
Type: String

Identifies a field in the preference form.

Return Value

Type: String

getSelectedValues(fieldId)

Returns a list of string values that are currently selected on a checkbox field in the preference form.

Signature

```
public List<String> getSelectedValues(String fieldId)
```

Parameters

fieldId
Type: String

Identifies a field in the preference form.

Return Value

Type: List<String>

`getStringValue(fieldId)`

Returns the string value that was set on a field when the preference form was loaded. This method is used for field types such as text.

Signature

```
public String getStringValue(String fieldId)
```

Parameters

`fieldId`
Type: String

Identifies a field in the preference form.

Return Value

Type: String

SubmitParameters Class

Retrieve record ID information to use with your submit-form handler.

Namespace

Pref_center

Example

```
String userId = submitParams.getRecordId();

User user = [select id, AboutMe from User where id=:userId];
```

IN THIS SECTION:

SubmitParameters Methods

SubmitParameters Methods

The following are methods for `SubmitParameters`.

IN THIS SECTION:

getRecordId()
Returns the untokenized version of the record ID.

`getRecordId()`

Returns the untokenized version of the record ID.

Signature

```
public String getRecordId()
```

Return Value

Type: String

TokenType Enum

Defines the types of values supported by the TokenUtility methods.

Enum Values

The following are the values of the `pref_center.TokenType` enum.

Value	Description
`EMAIL`	Identifies the token as an email address.
`STANDARD`	Identifies the token as a Salesforce record ID. This is the default token type.

TokenUtility Class

Generate authentication tokens to access preference forms.

Namespace

Pref_center

Example

Call the `generateToken()` method to generate a single token for a specified Salesforce record ID:

```
Individual individual = [SELECT Id FROM Individual LIMIT 1];
String token = pref_center.TokenUtility.generateToken(individual.Id);
// Do something with the token
System.debug(token)
```

Call the `generateTokens()` method to generate tokens in bulk when given a list of Salesforce record IDs:

```
List<Id> individualIds = new List<Id>();
// Get Ids of Individuals who have not opted out of tracking
for (Individual individual : [SELECT Id FROM Individual WHERE HasOptedOutTracking = false])
 {
    individualIds.add(individual.Id);
}
```

```
// Generate tokens for the list of Individual record Ids
Map<String, String> tokens = pref_center.TokenUtility.generateTokens(individualIds);
String firstIndividualId = individualIds[0];
// The returned Map has the input record Id as key and the corresponding token as value
String tokenForFirstIndividual = tokens.get(firstIndividualId);
// Do something with the token
System.debug(tokenForFirstIndividual);
```

IN THIS SECTION:

TokenUtility Methods

TokenUtility Methods

The following are methods for `TokenUtility`.

IN THIS SECTION:

generateToken(tokenValue, tokenType)

Returns the authentication token for the specified token value using the given token type.

generateToken(tokenValue)

Returns the authentication token for the specified token value using the default `standard` token type.

generateTokens(tokenValues, tokenType)

Returns the authentication tokens in the form of a map, where the map key is the input value to be tokenized and the map value is the corresponding token. The given token type is used to generate the tokens.

generateTokens(tokenValues)

Returns the generated tokens in the form of a map. This method uses the default standard token type to generate the tokens.

`generateToken(tokenValue, tokenType)`

Returns the authentication token for the specified token value using the given token type.

Signature

```
public static String generateToken(String tokenValue, pref_center.TokenType tokenType)
```

Parameters

`tokenValue`

Type: String

The value passed to `LoadParameters.getRecordId()` and `SubmitParameters.getRecordId()`. Identifies the entity that the preference form is acting on.

`tokenType`

Type: pref_center.TokenType

Specifies the type of the value to be encrypted with authentication tokens.

Return Value

Type: String

generateToken(tokenValue)

Returns the authentication token for the specified token value using the default `standard` token type.

Signature

```
public static String generateToken(String tokenValue)
```

Parameters

tokenValue
Type: String

Identifies the entity that the preference form is acting on. The value passed to `LoadParameters.getRecordId()` and `SubmitParameters.getRecordId()`.

Return Value

Type: String

generateTokens(tokenValues, tokenType)

Returns the authentication tokens in the form of a map, where the map key is the input value to be tokenized and the map value is the corresponding token. The given token type is used to generate the tokens.

Signature

```
public static Map<String,String> generateTokens(List<String> tokenValues,
pref_center.TokenType tokenType)
```

Parameters

tokenValues
Type: List<String>

The values passed to `LoadParameters.getRecordId()` and `SubmitParameters.getRecordId()`. Identifies the entity that the preference form is acting on. Contains multiple values to be encrypted with authentication tokens.

tokenType
Type: pref_center.TokenType

Specifies the type of the value to be encrypted with authentication tokens.

Return Value

Type: Map<String,String>

generateTokens(tokenValues)

Returns the generated tokens in the form of a map. This method uses the default standard token type to generate the tokens.

Signature

```
public static Map<String,String> generateTokens(List<String> tokenValues)
```

Parameters

tokenValues
Type: List<String>

The list of string values passed to `LoadParameters.getRecordId()` and `SubmitParameters.getRecordId()`. Contains multiple values to be encrypted with authentication tokens.

Return Value

Type: Map<String,String>, where the map key is the input value to be tokenized and the map value is the corresponding token.

ValidationResult Class

This class is reserved for future use with Preference Manager.

Namespace

Pref_center

Process Namespace

The `Process` namespace provides an interface and classes for passing data between your organization and a flow.

The following are the interfaces and classes in the `Process` namespace.

IN THIS SECTION:

Plugin Interface
Allows you to pass data between your organization and a specified flow.

PluginDescribeResult Class
Describes the input and output parameters for `Process.PluginResult`.

PluginDescribeResult.InputParameter Class
Describes the input parameter for `Process.PluginResult`.

PluginDescribeResult.OutputParameter Class
Describes the output parameter for `Process.PluginResult`.

PluginRequest Class
Passes input parameters from the class that implements the `Process.Plugin` interface to the flow.

PluginResult Class
Returns output parameters from the class that implements the `Process.Plugin` interface to the flow.

Plugin Interface

Allows you to pass data between your organization and a specified flow.

Namespace

Process

Tip: We recommend using the `@InvocableMethod` annotation instead of the `Process.Plugin` interface.

- The interface doesn't support Blob, Collection, sObject, and Time data types, and it doesn't support bulk operations. After you implement the interface on a class, the class can be referenced only from flows.
- The annotation supports all data types and bulk operations. After you implement the annotation on a class, the class can be referenced from flows, processes, and the Custom Invocable Actions REST API endpoint.
- Legacy Apex actions aren't supported in auto-layout in Flow Builder. Legacy Apex actions are only available to be added in free-form in Flow Builder. Existing actions can be edited in both auto-layout and free-form mode.

IN THIS SECTION:

Plugin Methods

Plugin Example Implementation

Plugin Methods

The following are instance methods for `Plugin`.

IN THIS SECTION:

describe()

Returns a `Process.PluginDescribeResult` object that describes this method call.

invoke(request)

Primary method that the system invokes when the class that implements the interface is instantiated.

`describe()`

Returns a `Process.PluginDescribeResult` object that describes this method call.

Signature

```
public Process.PluginDescribeResult describe()
```

Return Value

Type: Process.PluginDescribeResult

`invoke(request)`

Primary method that the system invokes when the class that implements the interface is instantiated.

Signature

```
public Process.PluginResult invoke(Process.PluginRequest request)
```

Parameters

request
Type: Process.PluginRequest

Return Value

Type: Process.PluginResult

Plugin Example Implementation

```
global class flowChat implements Process.Plugin {

// The main method to be implemented. The Flow calls this at run time.
global Process.PluginResult invoke(Process.PluginRequest request) {
        // Get the subject of the Chatter post from the flow
        String subject = (String) request.inputParameters.get('subject');

        // Use the Chatter APIs to post it to the current user's feed
        FeedItem fItem = new FeedItem();
        fItem.ParentId = UserInfo.getUserId();
        fItem.Body = 'Flow Update: ' + subject;
        insert fItem;

        // return to Flow
        Map<String,Object> result = new Map<String,Object>();
        return new Process.PluginResult(result);
    }

    // Returns the describe information for the interface
    global Process.PluginDescribeResult describe() {
        Process.PluginDescribeResult result = new Process.PluginDescribeResult();
        result.Name = 'flowchatplugin';
        result.Tag = 'chat';
        result.inputParameters = new
           List<Process.PluginDescribeResult.InputParameter>{
               new Process.PluginDescribeResult.InputParameter('subject',
               Process.PluginDescribeResult.ParameterType.STRING, true)
            };
        result.outputParameters = new
           List<Process.PluginDescribeResult.OutputParameter>{ };
        return result;
    }
}
```

Test Class

The following is a test class for the above class.

```
@isTest
private class flowChatTest {

    static testmethod void flowChatTests() {

        flowChat plugin = new flowChat();
        Map<String,Object> inputParams = new Map<String,Object>();

        string feedSubject = 'Flow is alive';
        InputParams.put('subject', feedSubject);

        Process.PluginRequest request = new Process.PluginRequest(inputParams);


        plugin.invoke(request);
    }
}
```

PluginDescribeResult Class

Describes the input and output parameters for `Process.PluginResult`.

Namespace

Process

Tip: We recommend using the `@InvocableMethod` annotation instead of the `Process.Plugin` interface.

- The interface doesn't support Blob, Collection, sObject, and Time data types, and it doesn't support bulk operations. After you implement the interface on a class, the class can be referenced only from flows.
- The annotation supports all data types and bulk operations. After you implement the annotation on a class, the class can be referenced from flows, processes, and the Custom Invocable Actions REST API endpoint.
- Legacy Apex actions aren't supported in auto-layout in Flow Builder. Legacy Apex actions are only available to be added in free-form in Flow Builder. Existing actions can be edited in both auto-layout and free-form mode.

IN THIS SECTION:

PluginDescribeResult Constructors

PluginDescribeResult Properties

PluginDescribeResult Constructors

The following are constructors for `PluginDescribeResult`.

IN THIS SECTION:

PluginDescribeResult()
Creates a new instance of the `Process.PluginDescribeResult` class.

`PluginDescribeResult()`

Creates a new instance of the `Process.PluginDescribeResult` class.

Signature

```
public PluginDescribeResult()
```

PluginDescribeResult Properties

The following are properties for `PluginDescribeResult`.

IN THIS SECTION:

description
This optional field describes the purpose of the plug-in.

inputParameters
The input parameters passed by the `Process.PluginRequest` class from a flow to the class that implements the `Process.Plugin` interface.

name
Unique name of the plug-in.

outputParameters
The output parameters passed by the `Process.PluginResult` class from the class that implements the `Process.Plugin` interface to the flow.

`description`

This optional field describes the purpose of the plug-in.

Signature

```
public String description {get; set;}
```

Property Value

Type: String

Usage

Size limit: 255 characters.

`inputParameters`

The input parameters passed by the `Process.PluginRequest` class from a flow to the class that implements the `Process.Plugin` interface.

Signature

```
public List<Process.PluginDescribeResult.InputParameter> inputParameters {get; set;}
```

Property Value

Type: List<Process.PluginDescribeResult.InputParameter>

`name`

Unique name of the plug-in.

Signature

```
public String name {get; set;}
```

Property Value

Type: String

Usage

Size limit: 40 characters.

`outputParameters`

The output parameters passed by the `Process.PluginResult` class from the class that implements the `Process.Plugin` interface to the flow.

Signature

```
public List<Process.PluginDescribeResult.OutputParameter> outputParameters {get; set;}
```

Property Value

Type: List<Process.PluginDescribeResult.OutputParameter>

PluginDescribeResult.InputParameter Class

Describes the input parameter for `Process.PluginResult`.

Namespace

Process

Tip: We recommend using the `@InvocableMethod` annotation instead of the `Process.Plugin` interface.

- The interface doesn't support Blob, Collection, sObject, and Time data types, and it doesn't support bulk operations. After you implement the interface on a class, the class can be referenced only from flows.
- The annotation supports all data types and bulk operations. After you implement the annotation on a class, the class can be referenced from flows, processes, and the Custom Invocable Actions REST API endpoint.
- Legacy Apex actions aren't supported in auto-layout in Flow Builder. Legacy Apex actions are only available to be added in free-form in Flow Builder. Existing actions can be edited in both auto-layout and free-form mode.

IN THIS SECTION:

PluginDescribeResult.InputParameter Constructors

PluginDescribeResult.InputParameter Properties

PluginDescribeResult.InputParameter Constructors

The following are constructors for `PluginDescribeResult.InputParameter`.

IN THIS SECTION:

PluginDescribeResult.InputParameter(name, description, parameterType, required)

Creates a new instance of the `Process.PluginDescribeResult.InputParameter` class using the specified name, description, parameter type, and required option.

PluginDescribeResult.InputParameter(name, parameterType, required)

Creates a new instance of the `Process.PluginDescribeResult.InputParameter` class using the specified name, parameter type, and required option.

`PluginDescribeResult.InputParameter(name, description, parameterType, required)`

Creates a new instance of the `Process.PluginDescribeResult.InputParameter` class using the specified name, description, parameter type, and required option.

Signature

```
public PluginDescribeResult.InputParameter(String name, String description,
Process.PluginDescribeResult.ParameterType parameterType, Boolean required)
```

Parameters

name

Type: String

Unique name of the plug-in.

description

Type: String

Describes the purpose of the plug-in.

parameterType
Type: Process.PluginDescribeResult.ParameterType

The data type of the input parameter.

required
Type: Boolean

Set to `true` for required and `false` otherwise.

`PluginDescribeResult.InputParameter(name, parameterType, required)`

Creates a new instance of the `Process.PluginDescribeResult.InputParameter` class using the specified name, parameter type, and required option.

Signature

```
public PluginDescribeResult.InputParameter(String name,
Process.PluginDescribeResult.ParameterType parameterType, Boolean required)
```

Parameters

name
Type: String

Unique name of the plug-in.

parameterType
Type: Process.PluginDescribeResult.ParameterType

The data type of the input parameter.

required
Type: Boolean

Set to `true` for required and `false` otherwise.

PluginDescribeResult.InputParameter Properties

The following are properties for `PluginDescribeResult.InputParameter`.

IN THIS SECTION:

Description
This optional field describes the purpose of the plug-in.

Name
Unique name of the plug-in.

ParameterType
The data type of the input parameter.

Required
Set to `true` for required and `false` otherwise.

Description

This optional field describes the purpose of the plug-in.

Signature

```
public String Description {get; set;}
```

Property Value

Type: String

Usage

Size limit: 255 characters.

Name

Unique name of the plug-in.

Signature

```
public String Name {get; set;}
```

Property Value

Type: String

Usage

Size limit: 40 characters.

ParameterType

The data type of the input parameter.

Signature

```
public Process.PluginDescribeResult.ParameterType ParameterType {get; set;}
```

Property Value

Type: Process.PluginDescribeResult.ParameterType

Required

Set to `true` for required and `false` otherwise.

Signature

```
public Boolean Required {get; set;}
```

Property Value

Type: Boolean

PluginDescribeResult.OutputParameter Class

Describes the output parameter for `Process.PluginResult`.

Namespace

Process

Tip: We recommend using the `@InvocableMethod` annotation instead of the `Process.Plugin` interface.

- The interface doesn't support Blob, Collection, sObject, and Time data types, and it doesn't support bulk operations. After you implement the interface on a class, the class can be referenced only from flows.
- The annotation supports all data types and bulk operations. After you implement the annotation on a class, the class can be referenced from flows, processes, and the Custom Invocable Actions REST API endpoint.
- Legacy Apex actions aren't supported in auto-layout in Flow Builder. Legacy Apex actions are only available to be added in free-form in Flow Builder. Existing actions can be edited in both auto-layout and free-form mode.

IN THIS SECTION:

PluginDescribeResult.OutputParameter Constructors

PluginDescribeResult.OutputParameter Properties

PluginDescribeResult.OutputParameter Constructors

The following are constructors for `PluginDescribeResult.OutputParameter`.

IN THIS SECTION:

PluginDescribeResult.OutputParameter(name, description, parameterType)

Creates a new instance of the `Process.PluginDescribeResult.OutputParameter` class using the specified name, description, and parameter type.

PluginDescribeResult.OutputParameter(name, parameterType)

Creates a new instance of the `Process.PluginDescribeResult.OutputParameter` class using the specified name, description, and parameter type.

`PluginDescribeResult.OutputParameter(name, description, parameterType)`

Creates a new instance of the `Process.PluginDescribeResult.OutputParameter` class using the specified name, description, and parameter type.

Signature

```
public PluginDescribeResult.OutputParameter(String name, String description,
Process.PluginDescribeResult.ParameterType parameterType)
```

Parameters

name
Type: String

Unique name of the plug-in.

description
Type: String

Describes the purpose of the plug-in.

parameterType
Type: Process.PluginDescribeResult.ParameterType

The data type of the input parameter.

`PluginDescribeResult.OutputParameter(name, parameterType)`

Creates a new instance of the `Process.PluginDescribeResult.OutputParameter` class using the specified name, description, and parameter type.

Signature

```
public PluginDescribeResult.OutputParameter(String name,
Process.PluginDescribeResult.ParameterType parameterType)
```

Parameters

name
Type: String

Unique name of the plug-in.

parameterType
Type: Process.PluginDescribeResult.ParameterType

The data type of the input parameter.

PluginDescribeResult.OutputParameter Properties

The following are properties for `PluginDescribeResult.OutputParameter`.

IN THIS SECTION:

Description
This optional field describes the purpose of the plug-in.

Name
Unique name of the plug-in.

ParameterType
The data type of the input parameter.

Description

This optional field describes the purpose of the plug-in.

Signature

```
public String Description {get; set;}
```

Property Value

Type: String

Usage

Size limit: 255 characters.

Name

Unique name of the plug-in.

Signature

```
public String Name {get; set;}
```

Property Value

Type: String

Usage

Size limit: 40 characters.

ParameterType

The data type of the input parameter.

Signature

```
public Process.PluginDescribeResult.ParameterType ParameterType {get; set;}
```

Property Value

Type: Process.PluginDescribeResult.ParameterType

PluginRequest Class

Passes input parameters from the class that implements the `Process.Plugin` interface to the flow.

Namespace

Process

Tip: We recommend using the `@InvocableMethod` annotation instead of the `Process.Plugin` interface.

- The interface doesn't support Blob, Collection, sObject, and Time data types, and it doesn't support bulk operations. After you implement the interface on a class, the class can be referenced only from flows.
- The annotation supports all data types and bulk operations. After you implement the annotation on a class, the class can be referenced from flows, processes, and the Custom Invocable Actions REST API endpoint.
- Legacy Apex actions aren't supported in auto-layout in Flow Builder. Legacy Apex actions are only available to be added in free-form in Flow Builder. Existing actions can be edited in both auto-layout and free-form mode.

PluginRequest Properties

The following are properties for `PluginRequest`.

IN THIS SECTION:

inputParameters
Input parameters that are passed from the class that implements the `Process.Plugin` interface to the flow.

`inputParameters`

Input parameters that are passed from the class that implements the `Process.Plugin` interface to the flow.

Signature

```
public MAP<String,ANY> inputParameters {get; set;}
```

Property Value

Type: Map<String, Object>

PluginResult Class

Returns output parameters from the class that implements the `Process.Plugin` interface to the flow.

Tip: We recommend using the `@InvocableMethod` annotation instead of the `Process.Plugin` interface.

- The interface doesn't support Blob, Collection, sObject, and Time data types, and it doesn't support bulk operations. After you implement the interface on a class, the class can be referenced only from flows.
- The annotation supports all data types and bulk operations. After you implement the annotation on a class, the class can be referenced from flows, processes, and the Custom Invocable Actions REST API endpoint.
- Legacy Apex actions aren't supported in auto-layout in Flow Builder. Legacy Apex actions are only available to be added in free-form in Flow Builder. Existing actions can be edited in both auto-layout and free-form mode.

Namespace

Process

PluginResult Properties

The following are properties for `PluginResult`.

IN THIS SECTION:

outputParameters
Output parameters returned from the class that implements the interface to the flow.

`outputParameters`

Output parameters returned from the class that implements the interface to the flow.

Signature

```
public MAP<String, ANY> outputParameters {get; set;}
```

Property Value

Type: Map<String, Object>

QuickAction Namespace

The `QuickAction` namespace provides classes and methods for quick actions.

The following are the classes in the `QuickAction` namespace.

IN THIS SECTION:

DescribeAvailableQuickActionResult Class
Contains describe metadata information for a quick action that is available for a specified parent.

DescribeLayoutComponent Class
Represents the smallest unit in a layout—a field or a separator.

DescribeLayoutItem Class
Represents an individual item in a `QuickAction.DescribeLayoutRow`.

DescribeLayoutRow Class
Represents a row in a `QuickAction.DescribeLayoutSection`.

DescribeLayoutSection Class
Represents a section of a layout and consists of one or more columns and one or more rows (an array of `QuickAction.DescribeLayoutRow`).

DescribeQuickActionDefaultValue Class
Returns a default value for a quick action.

DescribeQuickActionParameter Class
Represents the parameters corresponding to a quick action.

DescribeQuickActionResult Class
Contains describe metadata information for a quick action.

QuickActionDefaults Class

Represents an abstract Apex class that provides the context for running the standard Email Action on Case Feed and the container of the Email Message fields for the action payload. You can override the target fields before the standard Email Action is rendered.

QuickActionDefaultsHandler Interface

The `QuickAction.QuickActionDefaultsHandler` interface lets you specify the default values for the standard Email and Send Email actions in the case feed. You can use this interface to specify the From address, CC address, BCC address, subject, and email body for the Email action in the case feed. You can use the interface to pre-populate these fields based on the context where the action is displayed, such as the case origin (for example, country) and subject.

QuickActionRequest Class

Use the `QuickAction.QuickActionRequest` class for providing action information for quick actions to be performed by `QuickAction` class methods. Action information includes the action name, context record ID, and record.

QuickActionResult Class

After you initiate a quick action with the `QuickAction` class, use the `QuickActionResult` class for processing action results.

SendEmailQuickActionDefaults Class

Represents an Apex class that provides: the From address list; the original email's email message ID, provided that the reply action was invoked on the email message feed item; and methods to specify related settings on templates. You can override these fields before the standard Email Action is rendered.

DescribeAvailableQuickActionResult Class

Contains describe metadata information for a quick action that is available for a specified parent.

Namespace

QuickAction

Usage

The QuickAction `describeAvailableQuickActions` method returns an array of available quick action describe result objects (`QuickAction.DescribeAvailableQuickActionResult`).

DescribeAvailableQuickActionResult Methods

The following are methods for `DescribeAvailableQuickActionResult`. All are instance methods.

IN THIS SECTION:

getActionEnumOrId()

Returns the unique ID for the action. If the action doesn't have an ID, its API name is used.

getLabel()

The quick action label.

getName()

The quick action name.

getType()
The quick action type.

getActionEnumOrId()

Returns the unique ID for the action. If the action doesn't have an ID, its API name is used.

Signature

```
public String getActionEnumOrId()
```

Return Value

Type: String

getLabel()

The quick action label.

Signature

```
public String getLabel()
```

Return Value

Type: String

getName()

The quick action name.

Signature

```
public String getName()
```

Return Value

Type: String

getType()

The quick action type.

Signature

```
public String getType()
```

Return Value

Type: String

DescribeLayoutComponent Class

Represents the smallest unit in a layout—a field or a separator.

Namespace

QuickAction

DescribeLayoutComponent Methods

The following are methods for `DescribeLayoutComponent`. All are instance methods.

IN THIS SECTION:

getDisplayLines()
Returns the vertical lines displayed for a field. Applies to `textarea` and multi-select picklist fields.

getTabOrder()
Returns the tab order for the item in the row.

getType()
Returns the name of the `QuickAction.DescribeLayoutComponent` type for this component.

getValue()
Returns the name of the field if the type for `QuickAction.DescribeLayoutComponent` is `textarea`.

`getDisplayLines()`

Returns the vertical lines displayed for a field. Applies to `textarea` and multi-select picklist fields.

Signature

```
public Integer getDisplayLines()
```

Return Value

Type: Integer

`getTabOrder()`

Returns the tab order for the item in the row.

Signature

```
public Integer getTabOrder()
```

Return Value

Type: Integer

`getType()`

Returns the name of the `QuickAction.DescribeLayoutComponent` type for this component.

Signature

```
public String getType()
```

Return Value

Type: String

`getValue()`

Returns the name of the field if the type for `QuickAction.DescribeLayoutComponent` is `textarea`.

Signature

```
public String getValue()
```

Return Value

Type: String

DescribeLayoutItem Class

Represents an individual item in a `QuickAction.DescribeLayoutRow`.

Namespace

QuickAction

Usage

For most fields on a layout, there is only one component per layout item. However, in a display-only view, the `QuickAction.DescribeLayoutItem` might be a composite of the individual fields (for example, an address can consist of street, city, state, country, and postal code data). On the corresponding edit view, each component of the address field would be split up into separate `QuickAction.DescribeLayoutItems`.

DescribeLayoutItem Methods

The following are methods for `DescribeLayoutItem`. All are instance methods.

IN THIS SECTION:

getLabel()
Returns the label text for this item.

getLayoutComponents()
Returns a list of `QuickAction.DescribeLayoutComponents` for this item.

isEditableForNew()
Indicates whether this item can be edited for new (`true`) or not (`false`).

isEditableForUpdate()
Indicates whether this item can be edited for update(`true`) or not (`false`).

isPlaceholder()
Indicates whether this item is a placeholder (`true`) or not (`false`). If `true`, then this item is blank.

isRequired()
Indicates whether this item is required (`true`) or not (`false`).

`getLabel()`

Returns the label text for this item.

Signature

```
public String getLabel()
```

Return Value

Type: String

`getLayoutComponents()`

Returns a list of `QuickAction.DescribeLayoutComponents` for this item.

Signature

```
public List<QuickAction.DescribeLayoutComponent> getLayoutComponents()
```

Return Value

Type: List<QuickAction.DescribeLayoutComponent>

`isEditableForNew()`

Indicates whether this item can be edited for new (`true`) or not (`false`).

Signature

```
public Boolean isEditableForNew()
```

Return Value

Type: Boolean

`isEditableForUpdate()`

Indicates whether this item can be edited for update(`true`) or not (`false`).

Signature

```
public Boolean isEditableForUpdate()
```

Return Value

Type: Boolean

`isPlaceholder()`

Indicates whether this item is a placeholder (`true`) or not (`false`). If `true`, then this item is blank.

Signature

```
public Boolean isPlaceholder()
```

Return Value

Type: Boolean

`isRequired()`

Indicates whether this item is required (`true`) or not (`false`).

Signature

```
public Boolean isRequired()
```

Return Value

Type: Boolean

Usage

This is useful if, for example, you want to render required fields in a contrasting color.

DescribeLayoutRow Class

Represents a row in a `QuickAction.DescribeLayoutSection`.

Namespace

QuickAction

Usage

A `QuickAction.DescribeLayoutRow` consists of one or more `QuickAction.DescribeLayoutItem` objects. For each `QuickAction.DescribeLayoutRow`, a `QuickAction.DescribeLayoutItem` refers either to a specific field or to an "empty" `QuickAction.DescribeLayoutItem` (one that contains no `QuickAction.DescribeLayoutComponent`

objects). An empty `QuickAction.DescribeLayoutItem` can be returned when a given `QuickAction.DescribeLayoutRow` is sparse (for example, containing more fields on the right column than on the left column).

DescribeLayoutRow Methods

The following are methods for `DescribeLayoutRow`. All are instance methods.

IN THIS SECTION:

getLayoutItems()
Returns either a specific field or an empty `QuickAction.DescribeLayoutItem` (one that contains no `QuickAction.DescribeLayoutComponent` objects).

getNumItems()
Returns the number of `QuickAction.DescribeLayoutItem`.

`getLayoutItems()`

Returns either a specific field or an empty `QuickAction.DescribeLayoutItem` (one that contains no `QuickAction.DescribeLayoutComponent` objects).

Signature

```
public List<QuickAction.DescribeLayoutItem> getLayoutItems()
```

Return Value

Type: List<QuickAction.DescribeLayoutItem>

`getNumItems()`

Returns the number of `QuickAction.DescribeLayoutItem`.

Signature

```
public Integer getNumItems()
```

Return Value

Type: Integer

DescribeLayoutSection Class

Represents a section of a layout and consists of one or more columns and one or more rows (an array of `QuickAction.DescribeLayoutRow`).

Namespace

QuickAction

DescribeLayoutSection Properties

The following are properties for `DescribeLayoutSection`.

`collapsed`

The current view of the record details section: collapsed (`true`) or expanded (`false`).

Signature

```
public Boolean collapsed {get; set;}
```

Property Value

Type: Boolean

`layoutsectionid`

The unique ID of the record details section in the layout.

Signature

```
public Id layoutsectionid {get; set;}
```

Property Value

Type: Id

DescribeLayoutSection Methods

The following are methods for `DescribeLayoutSection`.

IN THIS SECTION:

getColumns()
Returns the number of columns in the `QuickAction.DescribeLayoutSection`.

getHeading()
The heading text (label) for the `QuickAction.DescribeLayoutSection`.

getLayoutRows()
Returns an array of one or more `QuickAction.DescribeLayoutRow` objects.

getLayoutSectionId()
Returns the ID of the record details section in the layout.

getParentLayoutId()
Returns the ID of the layout upon which this `DescribeLayoutSection` resides.

getRows()
Returns the number of rows in the `QuickAction.DescribeLayoutSection`.

isCollapsed()
Indicates whether the record details section is collapsed (`true`) or expanded (`false`). If you build your own app, you can use this method to see whether the current user collapsed a section, and respect that preference in your own UI.

isUseCollapsibleSection()
Indicates whether the `QuickAction.DescribeLayoutSection` is a collapsible section (`true`) or not (`false`).

isUseHeading()
Indicates whether to use the `heading` (`true`) or not (`false`).

`getColumns()`

Returns the number of columns in the `QuickAction.DescribeLayoutSection`.

Signature

```
public Integer getColumns()
```

Return Value

Type: Integer

`getHeading()`

The heading text (label) for the `QuickAction.DescribeLayoutSection`.

Signature

```
public String getHeading()
```

Return Value

Type: String

`getLayoutRows()`

Returns an array of one or more `QuickAction.DescribeLayoutRow` objects.

Signature

```
public List<QuickAction.DescribeLayoutRow> getLayoutRows()
```

Return Value

Type: List<QuickAction.DescribeLayoutRow>

`getLayoutSectionId()`

Returns the ID of the record details section in the layout.

Signature

```
public Id getLayoutSectionId()
```

Return Value

Type: Id

`getParentLayoutId()`

Returns the ID of the layout upon which this `DescribeLayoutSection` resides.

Signature

```
public Id getParentLayoutId()
```

Return Value

Type: Id

`getRows()`

Returns the number of rows in the `QuickAction.DescribeLayoutSection`.

Signature

```
public Integer getRows()
```

Return Value

Type: Integer

`isCollapsed()`

Indicates whether the record details section is collapsed (`true`) or expanded (`false`). If you build your own app, you can use this method to see whether the current user collapsed a section, and respect that preference in your own UI.

Signature

```
public Boolean isCollapsed()
```

Return Value

Type: Boolean

`isUseCollapsibleSection()`

Indicates whether the `QuickAction.DescribeLayoutSection` is a collapsible section (`true`) or not (`false`).

Signature

```
public Boolean isUseCollapsibleSection()
```

Return Value

Type: Boolean

`isUseHeading()`

Indicates whether to use the `heading` (`true`) or not (`false`).

Signature

```
public Boolean isUseHeading()
```

Return Value

Type: Boolean

DescribeQuickActionDefaultValue Class

Returns a default value for a quick action.

Namespace

QuickAction

Usage

Represents the default values of fields to use in default layouts.

DescribeQuickActionDefaultValue Methods

The following are methods for `DescribeQuickActionDefaultValue`. All are instance methods.

IN THIS SECTION:

getDefaultValue()
Returns the default value of the quick action.

getField()
Returns the field name of the action.

`getDefaultValue()`

Returns the default value of the quick action.

Signature

```
public String getDefaultValue()
```

Return Value

Type: String

`getField()`

Returns the field name of the action.

Signature

```
public String getField()
```

Return Value

Type: String

DescribeQuickActionParameter Class

Represents the parameters corresponding to a quick action.

Namespace

QuickAction

IN THIS SECTION:

DescribeQuickActionParameter Properties

Learn more about the available properties with the `CalculateTaxRequest` class.

DescribeQuickActionParameter Methods

DescribeQuickActionParameter Properties

Learn more about the available properties with the `CalculateTaxRequest` class.

The following are properties for `DescribeQuickActionParameter`.The following are properties for `DescribeQuickActionParameter`.

IN THIS SECTION:

parametername

Describes the name of the parameter that can be associated with a specific quick action type. For example, User Utterance is a parameter that is associated with agent quick actions.

parametertype

Describes the type of quick action. The type can either be Input or Output.

parametervalue
Describes the value of the parameter associated with the quick action.

parametername

Describes the name of the parameter that can be associated with a specific quick action type. For example, User Utterance is a parameter that is associated with agent quick actions.

Signature

```
public String parametername {get; set;}
```

Property Value

Type: String

parametertype

Describes the type of quick action. The type can either be Input or Output.

Signature

```
public String parametertype {get; set;}
```

Property Value

Type: String

parametervalue

Describes the value of the parameter associated with the quick action.

Signature

```
public String parametervalue {get; set;}
```

Property Value

Type: String

DescribeQuickActionParameter Methods

The following are methods for `DescribeQuickActionParameter`.

IN THIS SECTION:

getParameterName()
Returns the name of the parameter associated with the quick action.

getParameterType()
Returns the type of the parameter associated with the quick action. This can either be Input or Output.

getParameterValue()
Returns the value of the parameter associated with the quick action.

`getParameterName()`

Returns the name of the parameter associated with the quick action.

Signature

```
public String getParameterName()
```

Return Value

Type: String

`getParameterType()`

Returns the type of the parameter associated with the quick action. This can either be Input or Output.

Signature

```
public String getParameterType()
```

Return Value

Type: String

`getParameterValue()`

Returns the value of the parameter associated with the quick action.

Signature

```
public String getParameterValue()
```

Return Value

Type: String

DescribeQuickActionResult Class

Contains describe metadata information for a quick action.

Namespace

QuickAction

Usage

The QuickAction `describeQuickActions` method returns an array of quick action describe result objects (`QuickAction.DescribeQuickActionResult`).

IN THIS SECTION:

DescribeQuickActionResult Properties

DescribeQuickActionResult Methods

DescribeQuickActionResult Properties

The following are properties for `DescribeQuickActionResult`.

IN THIS SECTION:

canvasapplicationname
The name of the Canvas application invoked by the custom action.

colors
Array of color information. Each color is associated with a theme.

contextsobjecttype
The object used for the action. Was `getsourceSobjectType()` in API version 29.0 and earlier.

defaultvalues
The action's default values.

flowdevname
If the custom action invokes a flow, the fully qualified name of the flow.

flowrecordidvar
If the custom action invokes a flow, the input variable that the custom action passes the record's ID to.

height
The height in pixels of the action pane.

iconname
The name of the icon used for the action. If a custom icon is not used, this value isn't set.

icons
Array of icons. Each icon is associated with a theme.

iconurl
The URL of the icon used for the action. This icon URL corresponds to the 32x32 icon used for the current Salesforce theme, introduced in Spring '10, or the custom icon, if there is one.

layout
The section of the layout where the action resides.

lightningcomponentbundleid
If the custom action invokes an Aura component, the ID of the Aura component bundle to which the component belongs.

lightningcomponentbundlename
If the custom action invokes an Aura component, the name of the Aura component bundle to which the component belongs.

lightningcomponentqualifiedname

The fully qualified name of the Aura component invoked by the custom action.

lightningwebcomponentbundleid

If the custom action invokes a Lightning web component, the ID of the Lightning web component bundle to which the component belongs.

lightningwebcomponentbundlename

If the custom action invokes a Lightning web component, the name of the Lightning web component bundle to which the component belongs.

lightningwebcomponentqualifiedname

The fully qualified name of the Lightning web component invoked by the custom action.

miniiconurl

The icon's URL. This icon URL corresponds to the 16x16 icon used for the current Salesforce theme, introduced in Spring '10, or the custom icon, if there is one.

showquickactionlcheader

Indicates whether the Lightning component quick action header and footer are shown. If `false`, then both the header containing the quick action title and the footer containing the Save and Cancel buttons aren't displayed.

showquickactionvfheader

Indicates whether the Visualforce quick action header and footer should be shown. If `false`, then both the header containing the quick action title and the footer containing the Save and Cancel buttons aren't displayed.

targetparentfield

The parent object type of the action. Links the target object to the parent object. For example, the value is Account if the target object is Contact and the parent object is Account.

targetrecordtypeid

The record type of the target record.

targetsobjecttype

The action's target object type.

visualforcepagename

The name of the Visualforce page associated with the custom action.

visualforcepageurl

The URL of the Visualforce page associated with the action.

width

The width in pixels of the action pane, for custom actions that call Visualforce pages, Canvas apps, or Lightning components.

`canvasapplicationname`

The name of the Canvas application invoked by the custom action.

Signature

```
public String canvasapplicationname {get; set;}
```

Property Value

Type: String

colors

Array of color information. Each color is associated with a theme.

Signature

```
public List<Schema.DescribeColorResult> colors {get; set;}
```

Property Value

Type: List<Schema.DescribeColorResult> on page 3094

contextsobjecttype

The object used for the action. Was `getsourceSobjectType()` in API version 29.0 and earlier.

Signature

```
public String contextsobjecttype {get; set;}
```

Property Value

Type: String

defaultvalues

The action's default values.

Signature

```
public List<QuickAction.DescribeQuickActionDefaultValue> defaultvalues {get; set;}
```

Property Value

Type: List<QuickAction.DescribeQuickActionDefaultValue>

flowdevname

If the custom action invokes a flow, the fully qualified name of the flow.

Signature

```
public String flowdevname {get; set;}
```

Property Value

Type: String

flowrecordidvar

If the custom action invokes a flow, the input variable that the custom action passes the record's ID to.

Signature

```
public String flowrecordidvar {get; set;}
```

Property Value

Type: String

Valid values are *null* or *recordId*.

height

The height in pixels of the action pane.

Signature

```
public Integer height {get; set;}
```

Property Value

Type: Integer

iconname

The name of the icon used for the action. If a custom icon is not used, this value isn't set.

Signature

```
public String iconname {get; set;}
```

Property Value

Type: String

icons

Array of icons. Each icon is associated with a theme.

Signature

```
public List<Schema.DescribeIconResult> icons {get; set;}
```

Property Value

Type: List<Schema.DescribeIconResult> on page 3118>

If no custom icon was associated with the quick action and the quick action creates a specific object, the icons will correspond to the icons used for the created object. For example, if the quick action creates an Account, the icon array will contain the icons used for Account.

If a custom icon was associated with the quick action, the array will contain that custom icon.

iconurl

The URL of the icon used for the action. This icon URL corresponds to the 32x32 icon used for the current Salesforce theme, introduced in Spring '10, or the custom icon, if there is one.

Signature

```
public String iconurl {get; set;}
```

Property Value

Type: String

layout

The section of the layout where the action resides.

Signature

```
public QuickAction.DescribeLayoutSection layout {get; set;}
```

Property Value

Type: QuickAction.DescribeLayoutSection on page 2878

lightningcomponentbundleid

If the custom action invokes an Aura component, the ID of the Aura component bundle to which the component belongs.

Signature

```
public String lightningcomponentbundleid {get; set;}
```

Property Value

Type: String

lightningcomponentbundlename

If the custom action invokes an Aura component, the name of the Aura component bundle to which the component belongs.

Signature

```
public String lightningcomponentbundlename {get; set;}
```

Property Value

Type: String

lightningcomponentqualifiedname

The fully qualified name of the Aura component invoked by the custom action.

Signature

```
public String lightningcomponentqualifiedname {get; set;}
```

Property Value

Type: String

lightningwebcomponentbundleid

If the custom action invokes a Lightning web component, the ID of the Lightning web component bundle to which the component belongs.

Signature

```
public String lightningwebcomponentbundleid {get; set;}
```

Property Value

Type: String

lightningwebcomponentbundlename

If the custom action invokes a Lightning web component, the name of the Lightning web component bundle to which the component belongs.

Signature

```
public String lightningwebcomponentbundlename {get; set;}
```

Property Value

Type: String

lightningwebcomponentqualifiedname

The fully qualified name of the Lightning web component invoked by the custom action.

Signature

```
public String lightningwebcomponentqualifiedname {get; set;}
```

Property Value

Type: String

miniiconurl

The icon's URL. This icon URL corresponds to the 16x16 icon used for the current Salesforce theme, introduced in Spring '10, or the custom icon, if there is one.

Signature

```
public String miniiconurl {get; set;}
```

Property Value

Type: String

showquickactionlcheader

Indicates whether the Lightning component quick action header and footer are shown. If `false`, then both the header containing the quick action title and the footer containing the Save and Cancel buttons aren't displayed.

Signature

```
public Boolean showquickactionlcheader {get; set;}
```

Property Value

Type: Boolean

showquickactionvfheader

Indicates whether the Visualforce quick action header and footer should be shown. If `false`, then both the header containing the quick action title and the footer containing the Save and Cancel buttons aren't displayed.

Signature

```
public Boolean showquickactionvfheader {get; set;}
```

Property Value

Type: Boolean

targetparentfield

The parent object type of the action. Links the target object to the parent object. For example, the value is Account if the target object is Contact and the parent object is Account.

Signature

```
public String targetparentfield {get; set;}
```

Property Value

Type: String

targetrecordtypeid

The record type of the target record.

Signature

```
public String targetrecordtypeid {get; set;}
```

Property Value

Type: String

targetsobjecttype

The action's target object type.

Signature

```
public String targetsobjecttype {get; set;}
```

Property Value

Type: String

visualforcepagename

The name of the Visualforce page associated with the custom action.

Signature

```
public String visualforcepagename {get; set;}
```

Property Value

Type: String

visualforcepageurl

The URL of the Visualforce page associated with the action.

Signature

```
public String visualforcepageurl {get; set;}
```

Property Value

Type: String

width

The width in pixels of the action pane, for custom actions that call Visualforce pages, Canvas apps, or Lightning components.

Signature

```
public Integer width {get; set;}
```

Property Value

Type: Integer

DescribeQuickActionResult Methods

The following are methods for `DescribeQuickActionResult`. All are instance methods.

IN THIS SECTION:

getActionEnumOrId()
Returns the unique ID for the action. If the action doesn't have an ID, its API name is used.

getCanvasApplicationName()
Returns the name of the Canvas application, if used.

getColors()
Returns an array of color information. Each color is associated with a theme.

getContextSobjectType()
Returns the object used for the action. Replaces `getsourceSobjectType()` in API version 30.0 and later.

getDefaultValues()
Returns the default values for a action.

getFlowDevName()
If the custom action invokes a flow, returns the fully qualified name of the flow invoked by the custom action.

getFlowRecordIdVar()
If the custom action invokes a flow, returns the input variable that the custom action passes the record's ID to.

getHeight()
Returns the height in pixels of the action pane.

getIconName()
Returns the actions' icon name.

getIconUrl()
Returns the URL of the 32x32 icon used for the action.

getIcons()
Returns a list of `Schema.DescribeIconResult` objects that describe colors used in a tab.

getLabel()
Returns the action label.

getLayout()
Returns the layout sections that comprise an action.

getLightningComponentBundleId()
If the custom action invokes an Aura component, returns the ID of the Aura component bundle to which the component belongs.

getLightningComponentBundleName()
If the custom action invokes an Aura component, returns the name of the Aura component bundle to which the component belongs.

getLightningComponentQualifiedName()
If the custom action invokes an Aura component, returns the fully qualified name of the Aura component invoked by the custom action.

getLightningWebComponentBundleId()
If the custom action invokes a Lightning web component, returns the ID of the Lightning web component bundle to which the component belongs.

getLightningWebComponentBundleName()
If the custom action invokes a Lightning web component, returns the name of the Lightning web component bundle to which the component belongs.

getLightningWebComponentQualifiedName()
If the custom action invokes a Lightning web component, returns the fully qualified name of the Lightning web component invoked by the custom action.

getMiniIconUrl()
Returns the 16x16 icon URL.

getName()
Returns the action name.

getShowQuickActionLcHeader()
Returns an indication of whether the Lightning component quick action header and footer are shown.

getShowQuickActionVfHeader()
Returns an indication of whether the Visualforce quick action header and footer should be shown.

getSourceSobjectType()
Returns the object type used for the action.

getTargetParentField()
Returns the parent object's type for the action.

getTargetRecordTypeId()
Returns the record type of the targeted record.

getTargetSobjectType()
Returns the action's target object type.

getType()
Returns a create or custom Visualforce action.

getVisualforcePageName()
If Visualforce is used, returns the name of the associated page for the action.

getVisualforcePageUrl()
Returns the URL of the Visualforce page associated with the action.

getWidth()
If a custom action is created, returns the width in pixels of the action pane.

`getActionEnumOrId()`

Returns the unique ID for the action. If the action doesn't have an ID, its API name is used.

Signature

```
public String getActionEnumOrId()
```

Return Value

Type: String

getCanvasApplicationName()

Returns the name of the Canvas application, if used.

Syntax

```
public String getCanvasApplicationName()
```

Return Value

Type: String

getColors()

Returns an array of color information. Each color is associated with a theme.

Signature

```
public List<Schema.DescribeColorResult> getColors()
```

Return Value

Type: List <Schema.DescribeColorResult>

getContextSobjectType()

Returns the object used for the action. Replaces `getsourceSobjectType()` in API version 30.0 and later.

Signature

```
public String getContextSobjectType()
```

Return Value

Type: String

getDefaultValues()

Returns the default values for a action.

Signature

```
public List<QuickAction.DescribeQuickActionDefaultValue> getDefaultValues()
```

Return Value

Type: List<QuickAction.DescribeQuickActionDefaultValue>

`getFlowDevName()`

If the custom action invokes a flow, returns the fully qualified name of the flow invoked by the custom action.

Signature

```
public String getFlowDevName()
```

Return Value

Type: String

`getFlowRecordIdVar()`

If the custom action invokes a flow, returns the input variable that the custom action passes the record's ID to.

Signature

```
public String getFlowRecordIdVar()
```

Return Value

Type: String

`getHeight()`

Returns the height in pixels of the action pane.

Signature

```
public Integer getHeight()
```

Return Value

Type: Integer

`getIconName()`

Returns the actions' icon name.

Signature

```
public String getIconName()
```

Return Value

Type: String

getIconUrl()

Returns the URL of the 32x32 icon used for the action.

Signature

```
public String getIconUrl()
```

Return Value

Type: String

getIcons()

Returns a list of `Schema.DescribeIconResult` objects that describe colors used in a tab.

Signature

```
public List<Schema.DescribeIconResult> getIcons()
```

Return Value

Type: List<Schema.DescribeIconResult>

getLabel()

Returns the action label.

Signature

```
public String getLabel()
```

Return Value

Type: String

getLayout()

Returns the layout sections that comprise an action.

Signature

```
public QuickAction.DescribeLayoutSection getLayout()
```

Return Value

Type: QuickAction.DescribeLayoutSection

getLightningComponentBundleId()

If the custom action invokes an Aura component, returns the ID of the Aura component bundle to which the component belongs.

Signature

```
public String getLightningComponentBundleId()
```

Return Value

Type: String

`getLightningComponentBundleName()`

If the custom action invokes an Aura component, returns the name of the Aura component bundle to which the component belongs.

Signature

```
public String getLightningComponentBundleName()
```

Return Value

Type: String

`getLightningComponentQualifiedName()`

If the custom action invokes an Aura component, returns the fully qualified name of the Aura component invoked by the custom action.

Signature

```
public String getLightningComponentQualifiedName()
```

Return Value

Type: String

`getLightningWebComponentBundleId()`

If the custom action invokes a Lightning web component, returns the ID of the Lightning web component bundle to which the component belongs.

Signature

```
public String getLightningWebComponentBundleId()
```

Return Value

Type: String

`getLightningWebComponentBundleName()`

If the custom action invokes a Lightning web component, returns the name of the Lightning web component bundle to which the component belongs.

Signature

```
public String getLightningWebComponentBundleName()
```

Return Value

Type: String

`getLightningWebComponentQualifiedName()`

If the custom action invokes a Lightning web component, returns the fully qualified name of the Lightning web component invoked by the custom action.

Signature

```
public String getLightningWebComponentQualifiedName()
```

Return Value

Type: String

`getMiniIconUrl()`

Returns the 16x16 icon URL.

Signature

```
public String getMiniIconUrl()
```

Return Value

Type: String

`getName()`

Returns the action name.

Signature

```
public String getName()
```

Return Value

Type: String

`getShowQuickActionLcHeader()`

Returns an indication of whether the Lightning component quick action header and footer are shown.

Signature

```
public Boolean getShowQuickActionLcHeader()
```

Return Value

Type: Boolean

If `false`, then both the header containing the quick action title and the footer containing the Save and Cancel buttons aren't displayed.

`getShowQuickActionVfHeader()`

Returns an indication of whether the Visualforce quick action header and footer should be shown.

Signature

```
public Boolean getShowQuickActionVfHeader()
```

Return Value

Type: Boolean

If `false`, then both the header containing the quick action title and the footer containing the Save and Cancel buttons aren't displayed.

`getSourceSobjectType()`

Returns the object type used for the action.

Signature

```
public String getSourceSobjectType()
```

Return Value

Type: String

`getTargetParentField()`

Returns the parent object's type for the action.

Signature

```
public String getTargetParentField()
```

Return Value

Type: String

`getTargetRecordTypeId()`

Returns the record type of the targeted record.

Signature

```
public String getTargetRecordTypeId()
```

Return Value

Type: String

getTargetSobjectType()

Returns the action's target object type.

Signature

```
public String getTargetSobjectType()
```

Return Value

Type: String

getType()

Returns a create or custom Visualforce action.

Signature

```
public String getType()
```

Return Value

Type: String

getVisualforcePageName()

If Visualforce is used, returns the name of the associated page for the action.

Signature

```
public String getVisualforcePageName()
```

Return Value

Type: String

getVisualforcePageUrl()

Returns the URL of the Visualforce page associated with the action.

Signature

```
public String getVisualforcePageUrl()
```

Return Value

Type: String

`getWidth()`

If a custom action is created, returns the width in pixels of the action pane.

Signature

```
public Integer getWidth()
```

Return Value

Type: Integer

QuickActionDefaults Class

Represents an abstract Apex class that provides the context for running the standard Email Action on Case Feed and the container of the Email Message fields for the action payload. You can override the target fields before the standard Email Action is rendered.

Namespace

QuickAction

Usage

Note: You cannot extend this abstract class. You can use the getter methods when using it in the context of QuickAction.QuickActionDefaultsHandler. Salesforce provides a class that extends this class (See QuickAction.SendEmailQuickActionDefaults.)

IN THIS SECTION:

QuickActionDefaults Methods

QuickActionDefaults Methods

The following are methods for `QuickActionDefaults`.

IN THIS SECTION:

getActionName()

Returns the name of the standard Email Action on Case Feed (Case.Email).

getActionType()

Returns the type of the standard Email Action on Case Feed (Email).

getContextId()

The ID of the context related to the standard Email Action on Case Feed (Case ID).

getTargetSObject()
The target object of the standard Email Action on Case Feed (EmailMessage).

getActionName()

Returns the name of the standard Email Action on Case Feed (Case.Email).

Signature

```
public String getActionName()
```

Return Value

Type: String

getActionType()

Returns the type of the standard Email Action on Case Feed (Email).

Signature

```
public String getActionType()
```

Return Value

Type: String

getContextId()

The ID of the context related to the standard Email Action on Case Feed (Case ID).

Signature

```
public Id getContextId()
```

Return Value

Type: Id

getTargetSObject()

The target object of the standard Email Action on Case Feed (EmailMessage).

Signature

```
public SObject getTargetSObject()
```

Return Value

Type: SObject

QuickActionDefaultsHandler Interface

The `QuickAction.QuickActionDefaultsHandler` interface lets you specify the default values for the standard Email and Send Email actions in the case feed. You can use this interface to specify the From address, CC address, BCC address, subject, and email body for the Email action in the case feed. You can use the interface to pre-populate these fields based on the context where the action is displayed, such as the case origin (for example, country) and subject.

Namespace

QuickAction

Usage

To specify default values for the standard Email action in the case feed, create a class that implements `QuickAction.QuickActionDefaultsHandler`.

The `QuickAction.QuickActionDefaultsHandler` interface works in Salesforce Classic and Lightning Experience.

When working in Lightning Experience, keep the following things in mind:

- The interface overrides email values set up with predefined IDs.
- The interface works with the out-of-the-box Email action provided on cases. You can also use the interface with custom Email actions for the case object.
- The interface in Lightning Experience doesn't support:
  - Email attachments
  - Custom email fields
  - Visualforce email templates, which are a type of email template available in Salesforce Classic
- The From field determines the from address picklist. While you can't customize this picklist in Send Email action types via the QuickActionDefaultsHandler interface, you can customize the From Address field. To customize this field, remove the From field from the SendEmail quick action layout and add the From Address field instead. Then provide a valid and verified from address in the QuickActionDefaultsHandler code. This address must be the current user's address, an organization-wide email address that the current user has access to, or an Email-to-Case routing address.
- If your Apex interface adds content to the email body, merge fields display as unresolved. During preview and send, the merge fields resolve.

When you implement this interface, provide an empty parameterless constructor.

IN THIS SECTION:

QuickActionDefaultsHandler Methods

QuickActionDefaultsHandler Example Implementations

These examples are implementations of the `QuickAction.QuickActionDefaultsHandler` interface.

QuickActionDefaultsHandler Methods

The following are methods for `QuickActionDefaultsHandler`.

IN THIS SECTION:

onInitDefaults(actionDefaults)
Implement this method to provide default values for the standard Email action in the case feed.

onInitDefaults(actionDefaults)

Implement this method to provide default values for the standard Email action in the case feed.

Signature

```
public void onInitDefaults(QuickAction.QuickActionDefaults[] actionDefaults)
```

Parameters

actionDefaults
Type: QuickAction.QuickActionDefaults[]

This array contains only one item of type `QuickAction.SendEmailQuickActionDefaults`.

Return Value

Type: void

QuickActionDefaultsHandler Example Implementations

These examples are implementations of the `QuickAction.QuickActionDefaultsHandler` interface.

In this example, the `onInitDefaults` method checks whether the element passed in the array is for the standard Email action in the case feed. Then, it performs a query to retrieve the case that corresponds to the context ID. Next, it sets the value of the BCC address of the corresponding email message to a default value. The default value is based on the case reason. Finally, it sets the default values of the email template properties. The `onInitDefaults` method determines the default values based on two criteria: first, whether a reply action on an email message initiated the call to the method, and second, whether any previous emails attached to the case are associated with the call.

```
global class EmailPublisherLoader implements QuickAction.QuickActionDefaultsHandler {
    // Empty constructor
    global EmailPublisherLoader() {
    }

    // The main interface method
    global void onInitDefaults(QuickAction.QuickActionDefaults[] defaults) {
        QuickAction.SendEmailQuickActionDefaults sendEmailDefaults = null;


        // Check if the quick action is the standard case feed Email action
        for (Integer j = 0; j < defaults.size(); j++) {
            if (defaults.get(j) instanceof QuickAction.SendEmailQuickActionDefaults &&
               defaults.get(j).getTargetSObject().getSObjectType() ==
                   EmailMessage.sObjectType &&
               defaults.get(j).getActionName().equals('Case.Email') &&
               defaults.get(j).getActionType().equals('Email')) {
                   sendEmailDefaults =
```

```
                    (QuickAction.SendEmailQuickActionDefaults)defaults.get(j);
                break;
            }
        }

        if (sendEmailDefaults != null) {
            Case c = [SELECT Status, Reason FROM Case
                      WHERE Id=:sendEmailDefaults.getContextId()];

          EmailMessage emailMessage = (EmailMessage)sendEmailDefaults.getTargetSObject();

            // Set BCC address to make sure each email goes for audit
            emailMessage.BccAddress = getBccAddress(c.Reason);

            /*
            Set Template related fields
            when the In Reply To Id field is null we know the interface
            is called on page load. Here we check if
            there are any previous emails attached to the case and load
            the 'New_Case_Created' or 'Automatic_Response' template.
            When the In Reply To Id field is not null we know that
            the interface is called on click of reply/reply all
            of an email and we load the 'Default_reply_template' template
            */
            if (sendEmailDefaults.getInReplyToId() == null) {
                Integer emailCount = [SELECT count() FROM EmailMessage
                                      WHERE ParentId=:sendEmailDefaults.getContextId()];
                if (emailCount!= null && emailCount > 0) {
                    sendEmailDefaults.setTemplateId(
                        getTemplateIdHelper('Automatic_Response'));
                } else {
                    sendEmailDefaults.setTemplateId(
                        getTemplateIdHelper('New_Case_Created'));
                }
                sendEmailDefaults.setInsertTemplateBody(false);
                sendEmailDefaults.setIgnoreTemplateSubject(false);
            } else {
                sendEmailDefaults.setTemplateId(
                    getTemplateIdHelper('Default_reply_template'));
                sendEmailDefaults.setInsertTemplateBody(false);
                sendEmailDefaults.setIgnoreTemplateSubject(true);
            }
        }
    }

    private Id getTemplateIdHelper(String templateApiName) {
        Id templateId = null;
        try {
            templateId = [select id, name from EmailTemplate
                          where developername = : templateApiName].id;
        } catch (Exception e) {
            system.debug('Unble to locate EmailTemplate using name: ' +
                templateApiName + ' refer to Setup | Communications Templates '
                    + templateApiName);
```

```
        }
        return templateId;
    }
private String getBccAddress(String reason) {
        if (reason != null && reason.equals('Technical'))
            { return 'support_technical@mycompany.com'; }
        else if (reason != null && reason.equals('Billing'))
            { return 'support_billing@mycompany.com'; }
        else { return 'support@mycompany.com'; }
    }


}
```

In this example, the `onInitDefaults` method checks whether the element passed in the array is for the standard Email action in the case feed. Then it performs a query to determine if the case Priority is set to *`High`*. If the Priority is set to *`High`*, the email address *`managers@acme.com`* is appended to the BCC field.

```
global class EmailPublisherForHighPriorityCases implements
QuickAction.QuickActionDefaultsHandler {
    // Empty constructor
    global EmailPublisherForHighPriorityCases() {
    }

    // The main interface method
    global void onInitDefaults(QuickAction.QuickActionDefaults[] defaults) {
        QuickAction.SendEmailQuickActionDefaults sendEmailDefaults =
(QuickAction.SendEmailQuickActionDefaults)defaults.get(0);
        EmailMessage emailMessage = (EmailMessage)sendEmailDefaults.getTargetSObject();

        Case c = [SELECT CaseNumber, Priority FROM Case WHERE
Id=:sendEmailDefaults.getContextId()];

        // If case severity is "High," append "managers@acme.com" to the existing (and
possibly blank) BCC field
        if (c.Priority != null && c.Priority.equals('High')) { // Priority is 'High'
            emailMessage.BccAddress = 'managers@acme.com';
        }
    }
}
```

In this example, the `onInitDefaults` method checks whether the element passed in the array is for the standard Email action in the case feed. Then it performs a query to determine if the case Type is set to *`Problem`*. If the type is set to *`Problem`*, the *`First Response`* email template is inserted into the body of the email.

```
global class EmailPublisherForCaseType implements QuickAction.QuickActionDefaultsHandler
{
    // Empty constructor
    global EmailPublisherForCaseType() {
    }

    // The main interface method
    global void onInitDefaults(QuickAction.QuickActionDefaults[] defaults) {
    QuickAction.SendEmailQuickActionDefaults sendEmailDefaults =
(QuickAction.SendEmailQuickActionDefaults)defaults.get(0);
```

```
    EmailMessage emailMessage = (EmailMessage)sendEmailDefaults.getTargetSObject();

    Case c = [SELECT CaseNumber, Type FROM Case WHERE Id=:sendEmailDefaults.getContextId()];


    // If case type is "Problem," insert the "First Response" email template
    if (c.CaseNumber != null && c.Type.equals('Problem')) {
        sendEmailDefaults.setTemplateId('Insert Email Template ID Here');   // Set the
template Id corresponding to First Response
        sendEmailDefaults.setInsertTemplateBody(true);
        sendEmailDefaults.setIgnoreTemplateSubject(false);
    }
}
```

In this example, the `onInitDefaults` method checks whether the element passed in the array is for the standard Email action in the case feed. Then it performs a query to determine if the email is a Reply or Reply All email. If the email is a Reply or Reply All email, the corresponding email templates for these emails are inserted into the body of the email.

```
global class EmailPublisherForReplyAndReplyAll implements
QuickAction.QuickActionDefaultsHandler {

    // Empty constructor
    global EmailPublisherForReplyAndReplyAll() {

    }

    // The main interface method
    global void onInitDefaults(QuickAction.QuickActionDefaults[] defaults) {

    QuickAction.SendEmailQuickActionDefaults sendEmailDefaults =
(QuickAction.SendEmailQuickActionDefaults)defaults.get(0);
    EmailMessage emailMessage = (EmailMessage)sendEmailDefaults.getTargetSObject();

    // If the email is a "Reply" email, insert the "Reply Email Template" to the email
body
    if (sendEmailDefaults.getActionName().equals('EmailMessage._Reply')) {
        sendEmailDefaults.setTemplateId('Insert Reply Email Template ID Here');

        sendEmailDefaults.setInsertTemplateBody(true);
        sendEmailDefaults.setIgnoreTemplateSubject(false);

    // If the email is a "Reply All" email, insert the "Reply All Email Template" to the
email body
    } else if (sendEmailDefaults.getActionName().equals('EmailMessage._ReplyAll')) {
        sendEmailDefaults.setTemplateId('Insert Reply All Email Template ID Here');

        sendEmailDefaults.setInsertTemplateBody(true);
        sendEmailDefaults.setIgnoreTemplateSubject(false);

}
```

QuickActionRequest Class

Use the `QuickAction.QuickActionRequest` class for providing action information for quick actions to be performed by `QuickAction` class methods. Action information includes the action name, context record ID, and record.

Namespace

QuickAction

Usage

For Apex saved using Salesforce API version 28.0, a parent ID is associated with the QuickActionRequest instead of the context ID.

The constructor of this class takes no arguments:

```
QuickAction.QuickActionRequest qar = new QuickAction.QuickActionRequest();
```

Example

In this sample, a new quick action is created to create a contact and assign a record to it.

```
QuickAction.QuickActionRequest req = new QuickAction.QuickActionRequest();
// Some quick action name
req.quickActionName = Schema.Account.QuickAction.AccountCreateContact;

// Define a record for the quick action to create
Contact c = new Contact();
c.lastname = 'last name';
req.record = c;

// Provide the context ID (or parent ID). In this case, it is an Account record.
req.contextid = '001xx000003DGcO';

QuickAction.QuickActionResult res = QuickAction.performQuickAction(req);
```

IN THIS SECTION:

QuickActionRequest Constructors

QuickActionRequest Methods

SEE ALSO:

QuickAction Class

QuickActionRequest Constructors

The following are constructors for `QuickActionRequest`.

IN THIS SECTION:

QuickActionRequest()
Creates a new instance of the `QuickAction.QuickActionRequest` class.

`QuickActionRequest()`

Creates a new instance of the `QuickAction.QuickActionRequest` class.

Signature

```
public QuickActionRequest()
```

QuickActionRequest Methods

The following are methods for `QuickActionRequest`. All are instance methods.

IN THIS SECTION:

getContextId()
Returns this QuickAction's context record ID.

getQuickActionName()
Returns this QuickAction's name.

getRecord()
Returns the QuickAction's associated record.

setContextId(contextId)
Sets this QuickAction's context ID. Returned by `getContextId`.

setQuickActionName(name)
Sets this QuickAction's name. Returned by `getQuickActionName`.

setRecord(record)
Sets a record for this QuickAction. Returned by `getRecord`.

`getContextId()`

Returns this QuickAction's context record ID.

Signature

```
public Id getContextId()
```

Return Value

Type: ID

`getQuickActionName()`

Returns this QuickAction's name.

Signature

```
public String getQuickActionName()
```

Return Value

Type: String

getRecord()

Returns the QuickAction's associated record.

Signature

```
public SObject getRecord()
```

Return Value

Type: sObject

setContextId(contextId)

Sets this QuickAction's context ID. Returned by `getContextId`.

Signature

```
public Void setContextId(Id contextId)
```

Parameters

contextId
Type: ID

Return Value

Type: Void

Usage

For Apex saved using Salesforce API version 28.0, sets this QuickAction's parent ID and is returned by `getParentId`.

setQuickActionName(name)

Sets this QuickAction's name. Returned by `getQuickActionName`.

Signature

```
public Void setQuickActionName(String name)
```

Parameters

name
Type: String

Return Value

Type: Void

`setRecord(record)`

Sets a record for this QuickAction. Returned by `getRecord`.

Signature

```
public Void setRecord(SObject record)
```

Parameters

record
Type: sObject

Return Value

Type: Void

QuickActionResult Class

After you initiate a quick action with the `QuickAction` class, use the `QuickActionResult` class for processing action results.

Namespace

QuickAction

SEE ALSO:
QuickAction Class

QuickActionResult Methods

The following are methods for `QuickActionResult`. All are instance methods.

IN THIS SECTION:

getErrors()
If an error occurs, an array of one or more database error objects, along with error codes and descriptions, is returned.

getIds()
The IDs of the QuickActions being processed.

getSuccessMessage()
Returns the success message associated with the quick action.

isCreated()
Returns `true` if the action is created; otherwise, `false`.

isSuccess()
Returns `true` if the action completes successfully; otherwise, `false`.

`getErrors()`

If an error occurs, an array of one or more database error objects, along with error codes and descriptions, is returned.

Signature

```
public List<Database.Error> getErrors()
```

Return Value

Type: List<Database.Error>

`getIds()`

The IDs of the QuickActions being processed.

Signature

```
public List<Id> getIds()
```

Return Value

Type: List<Id>

`getSuccessMessage()`

Returns the success message associated with the quick action.

Signature

```
public String getSuccessMessage()
```

Return Value

Type: String

`isCreated()`

Returns `true` if the action is created; otherwise, `false`.

Signature

```
public Boolean isCreated()
```

Return Value

Type: Boolean

`isSuccess()`

Returns `true` if the action completes successfully; otherwise, `false`.

Signature

```
public Boolean isSuccess()
```

Return Value

Type: Boolean

SendEmailQuickActionDefaults Class

Represents an Apex class that provides: the From address list; the original email's email message ID, provided that the reply action was invoked on the email message feed item; and methods to specify related settings on templates. You can override these fields before the standard Email Action is rendered.

Namespace

QuickAction

Usage

Note: You cannot instantiate this class. One can use the getters/setters when using it in the context of `QuickAction.QuickActionDefaultsHandler`.

IN THIS SECTION:

SendEmailQuickActionDefaults Methods

SendEmailQuickActionDefaults Methods

The following are methods for `SendEmailQuickActionDefaults`.

IN THIS SECTION:

getFromAddressList()
Returns a list of email addresses that are available in the From: address drop-down menu for the standard Email Action.

getInReplyToId()
Returns the email message ID of the email to which the reply/reply all action has been invoked.

setIgnoreTemplateSubject(useOriginalSubject)
Specifies whether the template subject should be ignored (true), thus using the original subject, or whether the template subject should replace the original subject (false).

setInsertTemplateBody(keepOriginalBodyContent)

Specifies whether the template body should be inserted above the original body content (true) or whether it should replace the entire content with the template body (false).

setTemplateId(templateId)

Sets the email template ID to load into the email body.

getFromAddressList()

Returns a list of email addresses that are available in the From: address drop-down menu for the standard Email Action.

Signature

```
public List<String> getFromAddressList()
```

Return Value

Type: List<String>

getInReplyToId()

Returns the email message ID of the email to which the reply/reply all action has been invoked.

Signature

```
public Id getInReplyToId()
```

Return Value

Type: Id

setIgnoreTemplateSubject(useOriginalSubject)

Specifies whether the template subject should be ignored (true), thus using the original subject, or whether the template subject should replace the original subject (false).

Signature

```
public void setIgnoreTemplateSubject(Boolean useOriginalSubject)
```

Parameters

useOriginalSubject

Type: Boolean

Return Value

Type: void

`setInsertTemplateBody(keepOriginalBodyContent)`

Specifies whether the template body should be inserted above the original body content (true) or whether it should replace the entire content with the template body (false).

Signature

```
public void setInsertTemplateBody(Boolean keepOriginalBodyContent)
```

Parameters

keepOriginalBodyContent
Type: Boolean

Return Value

Type: void

`setTemplateId(templateId)`

Sets the email template ID to load into the email body.

Signature

```
public void setTemplateId(Id templateId)
```

Parameters

templateId
Type: Id

The template ID.

Return Value

Type: void

Reports Namespace

The `Reports` namespace provides classes for accessing the same data as is available in the Salesforce Reports and Dashboards REST API.

The following are the classes in the `Reports` namespace.

IN THIS SECTION:

AggregateColumn Class
Contains methods for describing summary fields such as Record Count, Sum, Average, Max, Min, and custom summary formulas. Includes name, label, data type, and grouping context.

BucketField Class

Contains methods and constructors to work with information about a bucket field, including bucket type, name, and bucketed values.

BucketFieldValue Class

Contains information about the report values included in a bucket field.

BucketType Enum

The types of values included in a bucket.

ColumnDataType Enum

The `Reports.ColumnDataType` enum describes the type of data in a column. It is returned by the `getDataType` method.

ColumnSortOrder Enum

The `Reports.ColumnSortOrder` enum describes the order that the grouping column uses to sort data.

CrossFilter Class

Contains methods and constructors used to work with information about a cross filter.

CsfGroupType Enum

The group level at which the custom summary format aggregate is displayed in a report.

DateGranularity Enum

The `Reports.DateGranularity` enum describes the date interval that is used for grouping.

DetailColumn Class

Contains methods for describing fields that contain detailed data. Detailed data fields are also listed in the report metadata.

Dimension Class

Contains information for each row or column grouping.

EvaluatedCondition Class

Contains the individual components of an evaluated condition for a report notification, such as the aggregate name and label, the operator, and the value that the aggregate is compared to.

EvaluatedConditionOperator Enum

The `Reports.EvaluatedConditionOperator` enum describes the type of operator used to compare an aggregate to a value. It is returned by the `getOperator` method.

FilterOperator Class

Contains information about a filter operator, such as display name and API name.

FilterValue Class

Contains information about a filter value, such as the display name and API name.

FormulaType Enum

The format of the numbers in a custom summary formula.

GroupingColumn Class

Contains methods for describing fields that are used for column grouping.

GroupingInfo Class

Contains methods for describing fields that are used for grouping.

GroupingValue Class

Contains grouping values for a row or column, including the key, label, and value.

NotificationAction Interface

Implement this interface to trigger a custom Apex class when the conditions for a report notification are met.

NotificationActionContext Class

Contains information about the report instance and condition threshold for a report notification.

ReportCsf Class

Contains methods and constructors for working with information about a custom summary formula (CSF).

ReportCurrency Class

Contains information about a currency value, including the amount and currency code.

ReportDataCell Class

Contains the data for a cell in the report, including the display label and value.

ReportDescribeResult Class

Contains report, report type, and extended metadata for a tabular, summary, or matrix report.

ReportDetailRow Class

Contains data cells for a detail row of a report.

ReportDivisionInfo Class

Contains information about the divisions that can be used to filter a report.

ReportExtendedMetadata Class

Contains report extended metadata for a tabular, summary, or matrix report.

ReportFact Class

Contains the fact map for the report, which represents the report's data values.

ReportFactWithDetails Class

Contains the detailed fact map for the report, which represents the report's data values.

ReportFactWithSummaries Class

Contains the fact map for the report, which represents the report's data values, and includes summarized fields.

ReportFilter Class

Contains information about a report filter, including column, operator, and value.

ReportFormat Enum

Contains the possible report format types.

ReportFilterType Enum

The types of values included in a report filter type.

ReportInstance Class

Returns an instance of a report that was run asynchronously. Retrieves the results for that instance.

ReportManager Class

Runs a report synchronously or asynchronously and with or without details.

ReportMetadata Class

Contains report metadata for a tabular, summary, or matrix report.

ReportResults Class

Contains the results of running a report.

ReportScopeInfo Class

Contains information about possible scope values that you can choose. Scope values depend on the report type. For example, you can set the scope for opportunity reports to `All opportunities`, `My team's opportunities`, or `My opportunities`.

ReportScopeValue Class

Contains information about a possible scope value. Scope values depend on the report type. For example, you can set the scope for opportunity reports to `All opportunities`, `My team's opportunities`, or `My opportunities`.

ReportType Class

Contains the unique API name and display name for the report type.

ReportTypeColumn Class

Contains detailed report type metadata about a field, including data type, display name, and filter values.

ReportTypeColumnCategory Class

Information about categories of fields in a report type.

ReportTypeMetadata Class

Contains report type metadata, which gives you information about the fields that are available in each section of the report type, plus filter information for those fields.

SortColumn Class

Contains information about the sort column used in the report.

StandardDateFilter Class

Contains information about standard date filter available in the report—for example, the API name, start date, and end date of the standard date filter duration as well as the API name of the date field on which the filter is placed.

StandardDateFilterDuration Class

Contains information about each standard date filter—also referred to as a relative date filter. It contains the API name and display label of the standard date filter duration as well as the start and end dates.

StandardDateFilterDurationGroup Class

Contains information about the standard date filter groupings, such as the grouping display label and all standard date filters that fall under the grouping. Groupings include `Calendar Year`, `Calendar Quarter`, `Calendar Month`, `Calendar Week`, `Fiscal Year`, `Fiscal Quarter`, `Day`, and custom values based on user-defined date ranges.

StandardFilter Class

Contains information about the standard filter defined in the report, such as the filter field API name and filter value.

StandardFilterInfo Class

Is an abstract base class for an object that provides standard filter information.

StandardFilterInfoPicklist Class

Contains information about the standard filter picklist, such as the display name and type of the filter field, the default picklist value, and a list of all possible picklist values.

StandardFilterType Enum

The `StandardFilterType` enum describes the type of standard filters in a report. The `getType()` method returns a `Reports.StandardFilterType` enum value.

SummaryValue Class

Contains summary data for a cell of the report.

ThresholdInformation Class

Contains a list of evaluated conditions for a report notification.

TopRows Class

Contains methods and constructors for working with information about a row limit filter.

Reports Exceptions

The `Reports` namespace contains exception classes.

AggregateColumn Class

Contains methods for describing summary fields such as Record Count, Sum, Average, Max, Min, and custom summary formulas. Includes name, label, data type, and grouping context.

Namespace

Reports

AggregateColumn Methods

The following are methods for `AggregateColumn`. All are instance methods.

IN THIS SECTION:

getName()
Returns the unique API name of the summary field.

getLabel()
Returns the localized display name for the summarized or custom summary formula field.

getDataType()
Returns the data type of the summarized or custom summary formula field.

getAcrossGroupingContext()
Returns the column grouping in the report where the summary field is displayed.

getDownGroupingContext()
Returns the row grouping in the report where the summary field is displayed.

`getName()`

Returns the unique API name of the summary field.

Syntax

```
public String getName()
```

Return Value

Type: String

`getLabel()`

Returns the localized display name for the summarized or custom summary formula field.

Syntax

```
public String getLabel()
```

Return Value

Type: String

`getDataType()`

Returns the data type of the summarized or custom summary formula field.

Syntax

```
public Reports.ColumnDataType getDataType()
```

Return Value

Type: Reports.ColumnDataType

`getAcrossGroupingContext()`

Returns the column grouping in the report where the summary field is displayed.

Syntax

```
public String getAcrossGroupingContext()
```

Return Value

Type: String

`getDownGroupingContext()`

Returns the row grouping in the report where the summary field is displayed.

Syntax

```
public String getDownGroupingContext()
```

Return Value

Type: String

BucketField Class

Contains methods and constructors to work with information about a bucket field, including bucket type, name, and bucketed values.

Namespace

Reports

IN THIS SECTION:

BucketField Constructors

BucketField Methods

BucketField Constructors

The following are constructors for `BucketField`.

IN THIS SECTION:

BucketField(bucketType, devloperName, label, nullTreatedAsZero, otherBucketLabel, sourceColumnName, values)
Creates an instance of the `Reports.BucketField` class using the specified parameters.

BucketField()
Creates an instance of the `Reports.BucketField` class. You can then set values by using the class's `set` methods.

`BucketField(bucketType, devloperName, label, nullTreatedAsZero, otherBucketLabel, sourceColumnName, values)`

Creates an instance of the `Reports.BucketField` class using the specified parameters.

Signature

```
public BucketField(Reports.BucketType bucketType, String devloperName, String label,
Boolean nullTreatedAsZero, String otherBucketLabel, String sourceColumnName,
List<Reports.BucketFieldValue> values)
```

Parameters

`bucketType`
Type: Reports.BucketType

The type of bucket.

`devloperName`
Type: String

API name of the bucket.

`label`
Type: String

User-facing name of the bucket.

`nullTreatedAsZero`
Type: Boolean

Specifies whether null values are converted to zero (`true`) or not (`false`).

`otherBucketLabel`
Type: String

Name of the fields grouped as `Other` (in buckets of `BucketType PICKLIST`).

`sourceColumnName`
Type: String

Name of the bucketed field.

values
Type: List<Reports.BucketType>

Types of the values included in the bucket.

BucketField()

Creates an instance of the `Reports.BucketField` class. You can then set values by using the class's `set` methods.

Signature

```
public BucketField()
```

BucketField Methods

The following are methods for `BucketField`.

IN THIS SECTION:

getBucketType()
Returns the bucket type.

getDevloperName()
Returns the bucket's API name.

getLabel()
Returns the user-facing name of the bucket.

getNullTreatedAsZero()
Returns `true` if null values are converted to the number zero, otherwise returns `false`.

getOtherBucketLabel()
Returns the name of fields grouped as `Other` in buckets of type `PICKLIST`.

getSourceColumnName()
Returns the API name of the bucketed field.

getValues()
Returns the report values grouped by the bucket field.

setBucketType(value)
Sets the `BucketType` of the bucket.

setBucketType(bucketType)
Sets the `BucketType` of the bucket.

setDevloperName(devloperName)
Sets the API name of the bucket.

setLabel(label)
Sets the user-facing name of the bucket.

setNullTreatedAsZero(nullTreatedAsZero)
Specifies whether null values in the bucket are converted to zero (`true`) or not (`false`).

setOtherBucketLabel(otherBucketLabel)
Sets the name of the fields grouped as `Other` (in buckets of `BucketType PICKLIST`).

setSourceColumnName(sourceColumnName)
Specifies the name of the bucketed field.

setValues(values)
Specifies which type of values are included in the bucket.

toString()
Returns a string.

getBucketType()

Returns the bucket type.

Signature

```
public Reports.BucketType getBucketType()
```

Return Value

Type: Reports.BucketType

getDevloperName()

Returns the bucket's API name.

Signature

```
public String getDevloperName()
```

Return Value

Type: String

getLabel()

Returns the user-facing name of the bucket.

Signature

```
public String getLabel()
```

Return Value

Type: String

getNullTreatedAsZero()

Returns `true` if null values are converted to the number zero, otherwise returns `false`.

Signature

```
public Boolean getNullTreatedAsZero()
```

Return Value

Type: Boolean

getOtherBucketLabel()

Returns the name of fields grouped as `Other` in buckets of type `PICKLIST`.

Signature

```
public String getOtherBucketLabel()
```

Return Value

Type: String

getSourceColumnName()

Returns the API name of the bucketed field.

Signature

```
public String getSourceColumnName()
```

Return Value

Type: String

getValues()

Returns the report values grouped by the bucket field.

Signature

```
public List<Reports.BucketFieldValue> getValues()
```

Return Value

Type: List on page 3603<Reports.BucketFieldValue>

setBucketType(value)

Sets the `BucketType` of the bucket.

Signature

```
public void setBucketType(String value)
```

Parameters

value
Type: String

See the Reports.BucketType enum for valid values.

Return Value

Type: void

setBucketType(bucketType)

Sets the `BucketType` of the bucket.

Signature

```
public void setBucketType(Reports.BucketType bucketType)
```

Parameters

bucketType
Type: Reports.BucketType

Return Value

Type: void

setDevloperName(devloperName)

Sets the API name of the bucket.

Signature

```
public void setDevloperName(String devloperName)
```

Parameters

devloperName
Type: String

The API name to assign to the bucket.

Return Value

Type: void

setLabel(label)

Sets the user-facing name of the bucket.

Signature

```
public void setLabel(String label)
```

Parameters

label
Type: String

Return Value

Type: void

setNullTreatedAsZero(nullTreatedAsZero)

Specifies whether null values in the bucket are converted to zero (`true`) or not (`false`).

Signature

```
public void setNullTreatedAsZero(Boolean nullTreatedAsZero)
```

Parameters

nullTreatedAsZero
Type: Boolean

Return Value

Type: void

setOtherBucketLabel(otherBucketLabel)

Sets the name of the fields grouped as `Other` (in buckets of `BucketType PICKLIST`).

Signature

```
public void setOtherBucketLabel(String otherBucketLabel)
```

Parameters

otherBucketLabel
Type: String

Return Value

Type: void

setSourceColumnName(sourceColumnName)

Specifies the name of the bucketed field.

Signature

```
public void setSourceColumnName(String sourceColumnName)
```

Parameters

sourceColumnName
Type: String

Return Value

Type: void

setValues(values)

Specifies which type of values are included in the bucket.

Signature

```
public void setValues(List<Reports.BucketFieldValue> values)
```

Parameters

values
Type: List on page 3603<Reports.BucketFieldValue>

Return Value

Type: void

toString()

Returns a string.

Signature

```
public String toString()
```

Return Value

Type: String

BucketFieldValue Class

Contains information about the report values included in a bucket field.

Namespace

Reports

IN THIS SECTION:

BucketFieldValue Constructors

BucketFieldValue Methods

BucketFieldValue Constructors

The following are constructors for `BucketFieldValue`.

IN THIS SECTION:

BucketFieldValue(label, sourceDimensionValues, rangeUpperBound)

Creates an instance of the `Reports.BucketFieldValue` class using the specified parameters.

BucketFieldValue()

Creates an instance of the `Reports.BucketFieldValue` class. You can then set values by using the class's `set` methods.

`BucketFieldValue(label, sourceDimensionValues, rangeUpperBound)`

Creates an instance of the `Reports.BucketFieldValue` class using the specified parameters.

Signature

```
public BucketFieldValue(String label, List<String> sourceDimensionValues, Double
rangeUpperBound)
```

Parameters

`label`
Type: String

The user-facing name of the bucket.

`sourceDimensionValues`
Type: List on page 3603<String>

A list of the values from the source field included in this bucket category (in buckets of type `PICKLIST` and buckets of type `TEXT`).

`rangeUpperBound`
Type: Double

The greatest range limit under which values are included in this bucket category (in buckets of type `NUMBER`).

`BucketFieldValue()`

Creates an instance of the `Reports.BucketFieldValue` class. You can then set values by using the class's `set` methods.

Signature

```
public BucketFieldValue()
```

BucketFieldValue Methods

The following are methods for `BucketFieldValue`.

IN THIS SECTION:

getLabel()
Returns the user-facing name of the bucket category.

getRangeUpperBound()
Returns the greatest range limit under which values are included in this bucket category (in buckets of type `NUMBER`).

getSourceDimensionValues()
Returns a list of the values from the source field included in this bucket category (in buckets of type `PICKLIST` and buckets of type `TEXT`).

setLabel(label)
Set the user-facing name of the bucket category.

setRangeUpperBound(rangeUpperBound)
Sets the greatest limit of a range under which values are included in this bucket category (in buckets of type `NUMBER`).

setSourceDimensionValues(sourceDimensionValues)
Specifies the values from the source field included in this bucket category (in buckets of type `PICKLIST` and buckets of type `TEXT`).

toString()
Returns a string.

`getLabel()`

Returns the user-facing name of the bucket category.

Signature

```
public String getLabel()
```

Return Value

Type: String

`getRangeUpperBound()`

Returns the greatest range limit under which values are included in this bucket category (in buckets of type `NUMBER`).

Signature

```
public Double getRangeUpperBound()
```

Return Value

Type: Double

`getSourceDimensionValues()`

Returns a list of the values from the source field included in this bucket category (in buckets of type `PICKLIST` and buckets of type `TEXT`).

Signature

```
public List<String> getSourceDimensionValues()
```

Return Value

Type: List<String>

`setLabel(label)`

Set the user-facing name of the bucket category.

Signature

```
public void setLabel(String label)
```

Parameters

label
Type: String

Return Value

Type: void

`setRangeUpperBound(rangeUpperBound)`

Sets the greatest limit of a range under which values are included in this bucket category (in buckets of type `NUMBER`).

Signature

```
public void setRangeUpperBound(Double rangeUpperBound)
```

Parameters

rangeUpperBound
Type: Double

Return Value

Type: void

`setSourceDimensionValues(sourceDimensionValues)`

Specifies the values from the source field included in this bucket category (in buckets of type `PICKLIST` and buckets of type `TEXT`).

Signature

```
public void setSourceDimensionValues(List<String> sourceDimensionValues)
```

Parameters

sourceDimensionValues
Type: List<String>

Return Value

Type: void

toString()

Returns a string.

Signature

```
public String toString()
```

Return Value

Type: String

BucketType Enum

The types of values included in a bucket.

Enum Values

The following are the values of the `Reports.BucketType` enum.

Value	Description
`NUMBER`	Numeric values
`PICKLIST`	Picklist values
`TEXT`	String values

ColumnDataType Enum

The `Reports.ColumnDataType` enum describes the type of data in a column. It is returned by the `getDataType` method.

Namespace

Reports

Enum Values

The following are the values of the `Reports.ColumnDataType` enum.

Value	Description
BOOLEAN_DATA	Boolean (true or false) values
COMBOBOX_DATA	Comboboxes, which provide a set of enumerated values and enable the user to specify a value that is not in the list
CURRENCY_DATA	Currency values
DATETIME_DATA	DateTime values
DATE_DATA	Date values
DOUBLE_DATA	Double values
EMAIL_DATA	Email addresses
ID_DATA	An object's Salesforce ID
INT_DATA	Integer values
MULTIPICKLIST_DATA	Multi-select picklists, which provide a set of enumerated values from which multiple values can be selected
PERCENT_DATA	Percent values
PHONE_DATA	Phone numbers. Values can include alphabetic characters. Client applications are responsible for phone number formatting.
PICKLIST_DATA	Single-select picklists, which provide a set of enumerated values from which only one value can be selected
REFERENCE_DATA	Cross-references to another object, analogous to a foreign key field
STRING_DATA	String values
TEXTAREA_DATA	String values that are displayed as multiline text fields
TIME_DATA	Time values
URL_DATA	URL values that are displayed as hyperlinks

ColumnSortOrder Enum

The `Reports.ColumnSortOrder` enum describes the order that the grouping column uses to sort data.

Namespace

Reports

Usage

The `GroupingInfo.getColumnSortOrder()` method returns a `Reports.ColumnSortOrder` enum value. The `GroupingInfo.setColumnSortOrder()` method takes the enum value as an argument.

Enum Values

The following are the values of the `Reports.ColumnSortOrder` enum.

Value	Description
`ASCENDING`	Sort data in ascending order (A–Z)
`DESCENDING`	Sort data in descending order (Z–A)

CrossFilter Class

Contains methods and constructors used to work with information about a cross filter.

Namespace

Reports

IN THIS SECTION:

CrossFilter Constructors

CrossFilter Methods

CrossFilter Constructors

The following are constructors for `CrossFilter`.

IN THIS SECTION:

CrossFilter(criteria, includesObject, primaryEntityField, relatedEntity, relatedEntityJoinField)

Creates an instance of the `Reports.CrossFilter` class using the specified parameters.

CrossFilter()

Creates an instance of the `Reports.CrossFilter` class. You can then set values by using the class's `set` methods.

`CrossFilter(criteria, includesObject, primaryEntityField, relatedEntity, relatedEntityJoinField)`

Creates an instance of the `Reports.CrossFilter` class using the specified parameters.

Signature

```
public CrossFilter(List<Reports.ReportFilter> criteria, Boolean includesObject, String primaryEntityField, String relatedEntity, String relatedEntityJoinField)
```

Parameters

criteria

Type: List<Reports.ReportFilter>

Information about how to filter the `relatedEntity`. Relates the primary entity with a subset of the `relatedEntity`.

`includesObject`
Type: Boolean

Specifies whether objects returned have a relationship with the `relatedEntity` (`true`) or not (`false`).

`primaryEntityField`
Type: String

The name of the object on which the cross filter is evaluated.

`relatedEntity`
Type: String

The name of the object that the `primaryEntityField` is evaluated against—the right-hand side of the cross filter.

`relatedEntityJoinField`
Type: String

The name of the field used to join the `primaryEntityField` and `relatedEntity`.

`CrossFilter()`

Creates an instance of the `Reports.CrossFilter` class. You can then set values by using the class's `set` methods.

Signature

```
public CrossFilter()
```

CrossFilter Methods

The following are methods for `CrossFilter`.

IN THIS SECTION:

getCriteria()
Returns information about how to filter the `relatedEntity`. Describes the subset of the `relatedEntity` which the primary entity is evaluated against.

getIncludesObject()
Returns `true` if primary object has a relationship with the `relatedEntity`, otherwise returns `false`.

getPrimaryEntityField()
Returns the name of the object on which the cross filter is evaluated.

getRelatedEntity()
Returns name of the object that the `primaryEntityField` is evaluated against—the right-hand side of the cross filter.

getRelatedEntityJoinField()
Returns the name of the field used to join the `primaryEntityField` and `relatedEntity`.

setCriteria(criteria)
Specifis how to filter the `relatedEntity`. Relates the primary entity with a subset of the `relatedEntity`.

setIncludesObject(includesObject)
Specifies whether objects returned have a relationship with the `relatedEntity` (`true`) or not (`false`).

setPrimaryEntityField(primaryEntityField)
Specifies the name of the object on which the cross filter is evaluated.

setRelatedEntity(relatedEntity)
Specifies the name of the object that the `primaryEntityField` is evaluated against—the right-hand side of the cross filter.

setRelatedEntityJoinField(relatedEntityJoinField)
Specifies the name of the field used to join the `primaryEntityField` and `relatedEntity`.

toString()
Returns a string.

getCriteria()

Returns information about how to filter the `relatedEntity`. Describes the subset of the `relatedEntity` which the primary entity is evaluated against.

Signature

```
public List<Reports.ReportFilter> getCriteria()
```

Return Value

Type: List<Reports.ReportFilter>

getIncludesObject()

Returns `true` if primary object has a relationship with the `relatedEntity`, otherwise returns `false`.

Signature

```
public Boolean getIncludesObject()
```

Return Value

Type: Boolean

getPrimaryEntityField()

Returns the name of the object on which the cross filter is evaluated.

Signature

```
public String getPrimaryEntityField()
```

Return Value

Type: String

getRelatedEntity()

Returns name of the object that the `primaryEntityField` is evaluated against—the right-hand side of the cross filter.

Signature

```
public String getRelatedEntity()
```

Return Value

Type: String

getRelatedEntityJoinField()

Returns the name of the field used to join the `primaryEntityField` and `relatedEntity`.

Signature

```
public String getRelatedEntityJoinField()
```

Return Value

Type: String

setCriteria(criteria)

Specifis how to filter the `relatedEntity`. Relates the primary entity with a subset of the `relatedEntity`.

Signature

```
public void setCriteria(List<Reports.ReportFilter> criteria)
```

Parameters

criteria
Type: List<Reports.ReportFilter>

Return Value

Type: void

setIncludesObject(includesObject)

Specifies whether objects returned have a relationship with the `relatedEntity` (`true`) or not (`false`).

Signature

```
public void setIncludesObject(Boolean includesObject)
```

Parameters

includesObject
Type: Boolean

Return Value

Type: void

`setPrimaryEntityField(primaryEntityField)`

Specifies the name of the object on which the cross filter is evaluated.

Signature

```
public void setPrimaryEntityField(String primaryEntityField)
```

Parameters

`primaryEntityField`
Type: String

Return Value

Type: void

`setRelatedEntity(relatedEntity)`

Specifies the name of the object that the `primaryEntityField` is evaluated against—the right-hand side of the cross filter.

Signature

```
public void setRelatedEntity(String relatedEntity)
```

Parameters

`relatedEntity`
Type: String

Return Value

Type: void

`setRelatedEntityJoinField(relatedEntityJoinField)`

Specifies the name of the field used to join the `primaryEntityField` and `relatedEntity`.

Signature

```
public void setRelatedEntityJoinField(String relatedEntityJoinField)
```

Parameters

`relatedEntityJoinField`
Type: String

Return Value

Type: void

`toString()`

Returns a string.

Signature

```
public String toString()
```

Return Value

Type: String

CsfGroupType Enum

The group level at which the custom summary format aggregate is displayed in a report.

Enum Values

The following are the values of the `Reports.CsfGroupType` enum.

Value	Description
`ALL`	The aggregate is displayed at the end of every summary row.
`CUSTOM`	The aggregate is displayed at specified grouping levels.
`GRAND_TOTAL`	The aggregate is displayed only at the grand total level.

DateGranularity Enum

The `Reports.DateGranularity` enum describes the date interval that is used for grouping.

Namespace

Reports

Usage

The `GroupingInfo.getDateGranularity` method returns a `Reports.DateGranularity` enum value. The `GroupingInfo.setDateGranularity` method takes the enum value as an argument.

Enum Values

The following are the values of the `Reports.DateGranularity` enum.

Value	Description
DAY	The day of the week (Monday–Sunday)
DAY_IN_MONTH	The day of the month (1–31)
FISCAL_PERIOD	The fiscal period
FISCAL_QUARTER	The fiscal quarter
FISCAL_WEEK	The fiscal week
FISCAL_YEAR	The fiscal year
MONTH	The month (January–December)
MONTH_IN_YEAR	The month number (1–12)
NONE	No date grouping
QUARTER	The quarter number (1–4)
WEEK	The week number (1–52)
YEAR	The year number (####)

DetailColumn Class

Contains methods for describing fields that contain detailed data. Detailed data fields are also listed in the report metadata.

Namespace

Reports

DetailColumn Instance Methods

The following are instance methods for `DetailColumn`. All are instance methods.

IN THIS SECTION:

getName()
Returns the unique API name of the detail column field.

getLabel()
Returns the localized display name of a standard field, the ID of a custom field, or the API name of a bucket field that has detailed data.

getDataType()
Returns the data type of a detail column field.

`getName()`

Returns the unique API name of the detail column field.

Syntax

```
public String getName()
```

Return Value

Type: String

`getLabel()`

Returns the localized display name of a standard field, the ID of a custom field, or the API name of a bucket field that has detailed data.

Syntax

```
public String getLabel()
```

Return Value

Type: String

`getDataType()`

Returns the data type of a detail column field.

Syntax

```
public Reports.ColumnDataType getDataType()
```

Return Value

Type: Reports.ColumnDataType

Dimension Class

Contains information for each row or column grouping.

Namespace

Reports

Dimension Methods

The following are methods for `Dimension`. All are instance methods.

IN THIS SECTION:

getGroupings()
Returns information for each row or column grouping as a list.

`getGroupings()`

Returns information for each row or column grouping as a list.

Syntax

```
public List<Reports.GroupingValue> getGroupings()
```

Return Value

Type: List<Reports.GroupingValue>

EvaluatedCondition Class

Contains the individual components of an evaluated condition for a report notification, such as the aggregate name and label, the operator, and the value that the aggregate is compared to.

Namespace

Reports

IN THIS SECTION:

EvaluatedCondition Constructors

EvaluatedCondition Methods

EvaluatedCondition Constructors

The following are constructors for `EvaluatedCondition`.

IN THIS SECTION:

EvaluatedCondition(aggregateName, aggregateLabel, compareToValue, aggregateValue, displayCompareTo, displayValue, operator)
Creates a new instance of the `Reports.EvaluatedConditions` class using the specified parameters.

`EvaluatedCondition(aggregateName, aggregateLabel, compareToValue, aggregateValue, displayCompareTo, displayValue, operator)`

Creates a new instance of the `Reports.EvaluatedConditions` class using the specified parameters.

Signature

```
public EvaluatedCondition(String aggregateName, String aggregateLabel, Double
compareToValue, Double aggregateValue, String displayCompareTo, String displayValue,
Reports.EvaluatedConditionOperator operator)
```

Parameters

aggregateName
Type: String

The unique API name of the aggregate.

aggregateLabel
Type: String

The localized display name of the aggregate.

compareToValue
Type: Double

The value that the aggregate is compared to in the condition.

aggregateValue
Type: Double

The actual value of the aggregate when the report is run.

displayCompareTo
Type: String

The value that the aggregate is compared to in the condition, formatted for display. For example, a display value for a currency is $20.00 or USD20.00 instead of 20.00.

displayValue
Type: String

The value of the aggregate when the report is run, formatted for display. For example, a display value for a currency is $20.00 or USD20.00 instead of 20.00.

operator
Type: Reports.EvaluatedConditionOperator

The operator used in the condition.

EvaluatedCondition Methods

The following are methods for `EvaluatedCondition`.

IN THIS SECTION:

getAggregateLabel()
Returns the localized display name of the aggregate.

getAggregateName()
Returns the unique API name of the aggregate.

getCompareTo()
Returns the value that the aggregate is compared to in the condition.

getDisplayCompareTo()
Returns the value that the aggregate is compared to in the condition, formatted for display. For example, a display value for a currency is $20.00 or USD20.00 instead of 20.00.

getDisplayValue()
Returns the value of the aggregate when the report is run, formatted for display. For example, a display value for a currency is $20.00 or USD20.00 instead of 20.00.

getOperator()
Returns the operator used in the condition.

getValue()
Returns the actual value of the aggregate when the report is run.

getAggregateLabel()

Returns the localized display name of the aggregate.

Signature

```
public String getAggregateLabel()
```

Return Value

Type: String

getAggregateName()

Returns the unique API name of the aggregate.

Signature

```
public String getAggregateName()
```

Return Value

Type: String

getCompareTo()

Returns the value that the aggregate is compared to in the condition.

Signature

```
public Double getCompareTo()
```

Return Value

Type: Double

getDisplayCompareTo()

Returns the value that the aggregate is compared to in the condition, formatted for display. For example, a display value for a currency is $20.00 or USD20.00 instead of 20.00.

Signature

```
public String getDisplayCompareTo()
```

Return Value

Type: String

`getDisplayValue()`

Returns the value of the aggregate when the report is run, formatted for display. For example, a display value for a currency is $20.00 or USD20.00 instead of 20.00.

Signature

```
public String getDisplayValue()
```

Return Value

Type: String

`getOperator()`

Returns the operator used in the condition.

Signature

```
public Reports.EvaluatedConditionOperator getOperator()
```

Return Value

Type: Reports.EvaluatedConditionOperator

`getValue()`

Returns the actual value of the aggregate when the report is run.

Signature

```
public Double getValue()
```

Return Value

Type: Double

EvaluatedConditionOperator Enum

The `Reports.EvaluatedConditionOperator` enum describes the type of operator used to compare an aggregate to a value. It is returned by the `getOperator` method.

Namespace

Reports

Enum Values

The following are the values of the `Reports.EvaluatedConditionOperator` enum.

Value	Description
`EQUAL`	Equality operator.
`GREATER_THAN`	Greater than operator.
`GREATER_THAN_EQUAL`	Greater than or equal to operator.
`LESS_THAN`	Less than operator.
`LESS_THAN_EQUAL`	Less than or equal to operator.
`NOT_EQUAL`	Inequality operator.

FilterOperator Class

Contains information about a filter operator, such as display name and API name.

Namespace

Reports

FilterOperator Methods

The following are methods for `FilterOperator`. All are instance methods.

IN THIS SECTION:

getLabel()
Returns the localized display name of the filter operator. Possible values for this name are restricted based on the data type of the column being filtered.

getName()
Returns the unique API name of the filter operator. Possible values for this name are restricted based on the data type of the column being filtered. For example `multipicklist` fields can use the following filter operators: “equals,” “not equal to,” “includes,” and “excludes.” Bucket fields are considered to be of the `String` type.

`getLabel()`

Returns the localized display name of the filter operator. Possible values for this name are restricted based on the data type of the column being filtered.

Syntax

```
public String getLabel()
```

Return Value

Type: String

`getName()`

Returns the unique API name of the filter operator. Possible values for this name are restricted based on the data type of the column being filtered. For example `multipicklist` fields can use the following filter operators: "equals," "not equal to," "includes," and "excludes." Bucket fields are considered to be of the `String` type.

Syntax

```
public String getName()
```

Return Value

Type: String

FilterValue Class

Contains information about a filter value, such as the display name and API name.

Namespace

Reports

FilterValue Methods

The following are methods for `FilterValue`. All are instance methods.

IN THIS SECTION:

getLabel()
Returns the localized display name of the filter value. Possible values for this name are restricted based on the data type of the column being filtered.

getName()
Returns the unique API name of the filter value. Possible values for this name are restricted based on the data type of the column being filtered.

`getLabel()`

Returns the localized display name of the filter value. Possible values for this name are restricted based on the data type of the column being filtered.

Syntax

```
public String getLabel()
```

Return Value

Type: String

`getName()`

Returns the unique API name of the filter value. Possible values for this name are restricted based on the data type of the column being filtered.

Syntax

```
public String getName()
```

Return Value

Type: String

FormulaType Enum

The format of the numbers in a custom summary formula.

Enum Values

The following are the values of the `Reports.FormulaType` enum.

Value	Description
`CURRENCY`	Formatted as currency. For example, $100.00.
`NUMBER`	Formatted as numbers. For example, 100.
`PERCENT`	Formatted as percentages. For example, 100%.

GroupingColumn Class

Contains methods for describing fields that are used for column grouping.

Namespace

Reports

The `GroupingColumn` class provides basic information about column grouping fields. The `GroupingInfo` class includes additional methods for describing and updating grouping fields.

GroupingColumn Methods

The following are methods for `GroupingColumn`. All are instance methods.

IN THIS SECTION:

getName()
Returns the unique API name of the field or bucket field that is used for column grouping.

getLabel()
Returns the localized display name of the field that is used for column grouping.

getDataType()
Returns the data type of the field that is used for column grouping.

getGroupingLevel()
Returns the level of grouping for the column.

`getName()`

Returns the unique API name of the field or bucket field that is used for column grouping.

Syntax

```
public String getName()
```

Return Value

Type: String

`getLabel()`

Returns the localized display name of the field that is used for column grouping.

Syntax

```
public String getLabel()
```

Return Value

Type: String

`getDataType()`

Returns the data type of the field that is used for column grouping.

Syntax

```
public Reports.ColumnDataType getDataType()
```

Return Value

Type: Reports.ColumnDataType

`getGroupingLevel()`

Returns the level of grouping for the column.

Syntax

```
public Integer getGroupingLevel()
```

Return Value

Type: Integer

Usage

- In a summary report, 0, 1, or 2 indicates grouping at the first, second, or third row level.
- In a matrix report, 0 or 1 indicates grouping at the first or second row or column level.

GroupingInfo Class

Contains methods for describing fields that are used for grouping.

Namespace

Reports

GroupingInfo Methods

The following are methods for `GroupingInfo`. All are instance methods.

IN THIS SECTION:

getName()
Returns the unique API name of the field or bucket field that is used for row or column grouping.

getSortOrder()
Returns the order that is used to sort data in a row or column grouping (`ASCENDING` or `DESCENDING`).

getDateGranularity()
Returns the date interval that is used for row or column grouping.

getSortAggregate()
Returns the summary field that is used to sort data within a grouping in a summary report. The value is null when data within a grouping is not sorted by a summary field.

`getName()`

Returns the unique API name of the field or bucket field that is used for row or column grouping.

Syntax

```
public String getName()
```

Return Value

Type: String

`getSortOrder()`

Returns the order that is used to sort data in a row or column grouping (`ASCENDING` or `DESCENDING`).

Syntax

```
public Reports.ColumnSortOrder getSortOrder()
```

Return Value

Type: Reports.ColumnSortOrder

`getDateGranularity()`

Returns the date interval that is used for row or column grouping.

Syntax

```
public Reports.DateGranularity getDateGranularity()
```

Return Value

Type: Reports.DateGranularity

`getSortAggregate()`

Returns the summary field that is used to sort data within a grouping in a summary report. The value is null when data within a grouping is not sorted by a summary field.

Syntax

```
public String getSortAggregate()
```

Return Value

Type: String

GroupingValue Class

Contains grouping values for a row or column, including the key, label, and value.

Namespace

Reports

GroupingValue Methods

The following are methods for `GroupingValue`. All are instance methods.

IN THIS SECTION:

getGroupings()
Returns a list of second- or third-level row or column groupings. If there are none, the value is an empty array.

getKey()
Returns the unique identifier for a row or column grouping. The identifier is used by the fact map to specify data values within each grouping.

getLabel()
Returns the localized display name of a row or column grouping. For date and time fields, the label is the localized date or time.

getValue()
Returns the value of the field that is used as a row or column grouping.

`getGroupings()`

Returns a list of second- or third-level row or column groupings. If there are none, the value is an empty array.

Syntax

```
public LIST<Reports.GroupingValue> getGroupings()
```

Return Value

Type: List<Reports.GroupingValue>

`getKey()`

Returns the unique identifier for a row or column grouping. The identifier is used by the fact map to specify data values within each grouping.

Syntax

```
public String getKey()
```

Return Value

Type: String

`getLabel()`

Returns the localized display name of a row or column grouping. For date and time fields, the label is the localized date or time.

Syntax

```
public String getLabel()
```

Return Value

Type: String

`getValue()`

Returns the value of the field that is used as a row or column grouping.

Syntax

```
public Object getValue()
```

Return Value

Type: Object

Usage

The value depends on the field's data type.

- Currency fields:
  - `amount`: Of type currency. A data cell's value.
  - `currency`: Of type picklist. The ISO 4217 currency code, if available; for example, USD for US dollars or CNY for Chinese yuan. (If the grouping is on the converted currency, this value is the currency code for the report and not for the record.)
- Picklist fields: API name. For example, a custom picklist field—`Type of Business` with values 1, 2, and 3 for Consulting, Services, and Add-On Business respectively—has `1`, `2`, or `3` as the grouping value.
- ID fields: API name.
- Record type fields: API name.
- Date and time fields: Date or time in ISO-8601 format.
- Lookup fields: Unique API name. For example, for the `Opportunity Owner` lookup field, the ID of each opportunity owner's Chatter profile page can be a grouping value.

NotificationAction Interface

Implement this interface to trigger a custom Apex class when the conditions for a report notification are met.

Namespace

Reports

Usage

Report notifications for reports that users have subscribed to can trigger a custom Apex class, which must implement the `Reports.NotificationAction` interface. The `execute` method in this interface receives a `NotificationActionContext` object as a parameter, which contains information about the report instance and the conditions that must be met for a notification to be triggered.

IN THIS SECTION:

NotificationAction Methods

NotificationAction Example Implementation

NotificationAction Methods

The following are methods for `NotificationAction`.

IN THIS SECTION:

execute(context)
Executes the custom Apex action specified in the `context` parameter of the context object, `NotificationActionContext`. The object contains information about the report instance and the conditions that must be met for a notification to be triggered. The method executes whenever the specified conditions are met.

`execute(context)`

Executes the custom Apex action specified in the `context` parameter of the context object, `NotificationActionContext`. The object contains information about the report instance and the conditions that must be met for a notification to be triggered. The method executes whenever the specified conditions are met.

Signature

```
public void execute(Reports.NotificationActionContext context)
```

Parameters

context
Type: Reports.NotificationActionContext

Return Value

Type: Void

NotificationAction Example Implementation

This is an example implementation of the `Reports.NotificationAction` interface.

```
public class AlertOwners implements Reports.NotificationAction {

    public void execute(Reports.NotificationActionContext context) {
        Reports.ReportResults results = context.getReportInstance().getReportResults();
        for(Reports.GroupingValue g: results.getGroupingsDown().getGroupings()) {
            FeedItem t = new FeedItem();
            t.ParentId = (Id)g.getValue();
            t.Body = 'This record needs attention. Please view the report.';
            t.Title = 'Needs Attention: '+ results.getReportMetadata().getName();
            t.LinkUrl = '/' + results.getReportMetadata().getId();
            insert t;
        }
    }
}
```

NotificationActionContext Class

Contains information about the report instance and condition threshold for a report notification.

Namespace

Reports

IN THIS SECTION:

NotificationActionContext Constructors

NotificationActionContext Methods

NotificationActionContext Constructors

The following are constructors for `NotificationActionContext`.

IN THIS SECTION:

NotificationActionContext(reportInstance, thresholdInformation)
Creates a new instance of the `Reports.NotificationActionContext` class using the specified parameters.

`NotificationActionContext(reportInstance, thresholdInformation)`

Creates a new instance of the `Reports.NotificationActionContext` class using the specified parameters.

Signature

```
public NotificationActionContext(Reports.ReportInstance reportInstance,
Reports.ThresholdInformation thresholdInformation)
```

Parameters

`reportInstance`
Type: Reports.ReportInstance

An instance of a report.

`thresholdInformation`
Type: Reports.ThresholdInformation

The evaluated conditions for the notification.

NotificationActionContext Methods

The following are methods for `NotificationActionContext`.

IN THIS SECTION:

getReportInstance()
Returns the report instance associated with the notification.

getThresholdInformation()
Returns the threshold information associated with the notification.

`getReportInstance()`

Returns the report instance associated with the notification.

Signature

```
public Reports.ReportInstance getReportInstance()
```

Return Value

Type: Reports.ReportInstance

`getThresholdInformation()`

Returns the threshold information associated with the notification.

Signature

```
public Reports.ThresholdInformation getThresholdInformation()
```

Return Value

Type: Reports.ThresholdInformation

ReportCsf Class

Contains methods and constructors for working with information about a custom summary formula (CSF).

Namespace

Reports

IN THIS SECTION:

ReportCsf Constructors

ReportCsf Methods

ReportCsf Constructors

The following are constructors for `ReportCsf`.

IN THIS SECTION:

ReportCsf(label, description, formulaType, decimalPlaces, downGroup, downGroupType, acrossGroup, acrossGroupType, formula)
Creates an instance of the `Reports.ReportCsf` class using the specified parameters.

ReportCsf()

Creates an instance of the `Reports.ReportCsf` class. You can then set values by using the class's `set` methods.

ReportCsf(label, description, formulaType, decimalPlaces, downGroup, downGroupType, acrossGroup, acrossGroupType, formula)

Creates an instance of the `Reports.ReportCsf` class using the specified parameters.

Signature

```
public ReportCsf(String label, String description, Reports.FormulaType formulaType,
Integer decimalPlaces, String downGroup, Reports.CsfGroupType downGroupType, String
acrossGroup, Reports.CsfGroupType acrossGroupType, String formula)
```

Parameters

label

Type: String

The user-facing name of the custom summary formula.

description

Type: String

The user-facing description of the custom summary formula.

formulaType

Type: Reports.FormulaType

The format of the numbers in the custom summary formula.

decimalPlaces

Type: Integer

The number of decimal places to include in numbers.

downGroup

Type: String

The name of a row grouping when the `downGroupType` is `CUSTOM`; `null` otherwise.

downGroupType

Type: Reports.CsfGroupType

Where to display the aggregate of the custom summary formula.

acrossGroup

Type: String

The name of a column grouping when the `accrossGroupType` is `CUSTOM`; `null` otherwise.

acrossGroupType

Type: Reports.CsfGroupType

Where to display the aggregate of the custom summary formula.

formula

Type: String

The operations performed on values in the custom summary formula.

ReportCsf()

Creates an instance of the `Reports.ReportCsf` class. You can then set values by using the class's `set` methods.

Signature

```
public ReportCsf()
```

ReportCsf Methods

The following are methods for `ReportCsf`.

IN THIS SECTION:

getAcrossGroup()
Returns the name of a column grouping when the `acrossGroupType` is `CUSTOM`. Otherwise, returns `null`.

getAcrossGroupType()
Returns where to display the aggregate.

getDecimalPlaces()
Returns the number of decimal places that numbers in the custom summary formula have.

getDescription()
Returns the user-facing description of a custom summary formula.

getDownGroup()
Returns the name of a row grouping when the `downGroupType` is `CUSTOM`. Otherwise, returns `null`.

getDownGroupType()
Returns where to display the aggregate of the custom summary formula.

getFormula()
Returns the operations performed on values in the custom summary formula.

getFormulaType()
Returns the formula type.

getLabel()
Returns the user-facing name of the custom summary formula.

setAcrossGroup(acrossGroup)
Specifies the column for the across grouping.

setAcrossGroupType(value)
Sets where to display the aggregate.

setAcrossGroupType(acrossGroupType)
Sets where to display the aggregate.

setDecimalPlaces(decimalPlaces)
Sets the number of decimal places in numbers.

setDescription(description)
Sets the user-facing description of the custom summary formula.

setDownGroup(downGroup)
Sets the name of a row grouping when the `downGroupType` is `CUSTOM`.

setDownGroupType(value)
Sets where to display the aggregate.

setDownGroupType(downGroupType)
Sets where to display the aggregate.

setFormula(formula)
Sets the operations to perform on values in the custom summary formula.

setFormulaType(value)
Sets the format of the numbers in the custom summary formula.

setFormulaType(formulaType)
Sets the format of numbers used in the custom summary formula.

setLabel(label)
Sets the user-facing name of the custom summary formula.

toString()
Returns a string.

getAcrossGroup()

Returns the name of a column grouping when the `acrossGroupType` is `CUSTOM`. Otherwise, returns `null`.

Signature

```
public String getAcrossGroup()
```

Return Value

Type: String

getAcrossGroupType()

Returns where to display the aggregate.

Signature

```
public Reports.CsfGroupType getAcrossGroupType()
```

Return Value

Type: Reports.CsfGroupType

getDecimalPlaces()

Returns the number of decimal places that numbers in the custom summary formula have.

Signature

```
public Integer getDecimalPlaces()
```

Return Value

Type: Integer

`getDescription()`

Returns the user-facing description of a custom summary formula.

Signature

```
public String getDescription()
```

Return Value

Type: String

`getDownGroup()`

Returns the name of a row grouping when the `downGroupType` is `CUSTOM`. Otherwise, returns `null`.

Signature

```
public String getDownGroup()
```

Return Value

Type: String

`getDownGroupType()`

Returns where to display the aggregate of the custom summary formula.

Signature

```
public Reports.CsfGroupType getDownGroupType()
```

Return Value

Type: Reports.CsfGroupType

`getFormula()`

Returns the operations performed on values in the custom summary formula.

Signature

```
public String getFormula()
```

Return Value

Type: String

`getFormulaType()`

Returns the formula type.

Signature

```
public Reports.FormulaType getFormulaType()
```

Return Value

Type: Reports.FormulaType

`getLabel()`

Returns the user-facing name of the custom summary formula.

Signature

```
public String getLabel()
```

Return Value

Type: String

`setAcrossGroup(acrossGroup)`

Specifies the column for the across grouping.

Signature

```
public void setAcrossGroup(String acrossGroup)
```

Parameters

acrossGroup
Type: String

Return Value

Type: void

`setAcrossGroupType(value)`

Sets where to display the aggregate.

Signature

```
public void setAcrossGroupType(String value)
```

Parameters

value
Type: String

For possible values, see Reports.CsfGroupType.

Return Value

Type: void

setAcrossGroupType(acrossGroupType)

Sets where to display the aggregate.

Signature

```
public void setAcrossGroupType(Reports.CsfGroupType acrossGroupType)
```

Parameters

acrossGroupType
Type: Reports.CsfGroupType

Return Value

Type: void

setDecimalPlaces(decimalPlaces)

Sets the number of decimal places in numbers.

Signature

```
public void setDecimalPlaces(Integer decimalPlaces)
```

Parameters

decimalPlaces
Type: Integer

Return Value

Type: void

setDescription(description)

Sets the user-facing description of the custom summary formula.

Signature

```
public void setDescription(String description)
```

Parameters

description
Type: String

Return Value

Type: void

setDownGroup(downGroup)

Sets the name of a row grouping when the `downGroupType` is `CUSTOM`.

Signature

```
public void setDownGroup(String downGroup)
```

Parameters

downGroup
Type: String

Return Value

Type: void

setDownGroupType(value)

Sets where to display the aggregate.

Signature

```
public void setDownGroupType(String value)
```

Parameters

value
Type: String

For valid values, see Reports.CsfGroupType.

Return Value

Type: void

setDownGroupType(downGroupType)

Sets where to display the aggregate.

Signature

```
public void setDownGroupType(Reports.CsfGroupType downGroupType)
```

Parameters

downGroupType
Type: Reports.CsfGroupType

Return Value

Type: void

setFormula(formula)

Sets the operations to perform on values in the custom summary formula.

Signature

```
public void setFormula(String formula)
```

Parameters

formula
Type: String

Return Value

Type: void

setFormulaType(value)

Sets the format of the numbers in the custom summary formula.

Signature

```
public void setFormulaType(String value)
```

Parameters

value
Type: String

For valid values, see Reports.FormulaType.

Return Value

Type: void

setFormulaType(formulaType)

Sets the format of numbers used in the custom summary formula.

Signature

```
public void setFormulaType(Reports.FormulaType formulaType)
```

Parameters

formulaType
Type: Reports.FormulaType

Return Value

Type: void

setLabel(label)

Sets the user-facing name of the custom summary formula.

Signature

```
public void setLabel(String label)
```

Parameters

label
Type: String

Return Value

Type: void

toString()

Returns a string.

Signature

```
public String toString()
```

Return Value

Type: String

ReportCurrency Class

Contains information about a currency value, including the amount and currency code.

Namespace

Reports

ReportCurrency Methods

The following are methods for `ReportCurrency`. All are instance methods.

IN THIS SECTION:

getAmount()
Returns the amount of the currency value.

getCurrencyCode()
Returns the report currency code, such as USD, EUR, or GBP, for an organization that has multicurrency enabled. The value is `null` if the organization does not have multicurrency enabled.

`getAmount()`

Returns the amount of the currency value.

Syntax

```
public Decimal getAmount()
```

Return Value

Type: Decimal

`getCurrencyCode()`

Returns the report currency code, such as USD, EUR, or GBP, for an organization that has multicurrency enabled. The value is `null` if the organization does not have multicurrency enabled.

Syntax

```
public String getCurrencyCode()
```

Return Value

Type: String

ReportDataCell Class

Contains the data for a cell in the report, including the display label and value.

Namespace

Reports

ReportDataCell Methods

The following are methods for `ReportDataCell`. All are instance methods.

IN THIS SECTION:

getLabel()
Returns the localized display name of the value of a specified cell in the report.

getValue()
Returns the value of a specified cell of a detail row of a report.

`getLabel()`

Returns the localized display name of the value of a specified cell in the report.

Syntax

```
public String getLabel()
```

Return Value

Type: String

`getValue()`

Returns the value of a specified cell of a detail row of a report.

Syntax

```
public Object getValue()
```

Return Value

Type: Object

ReportDescribeResult Class

Contains report, report type, and extended metadata for a tabular, summary, or matrix report.

Namespace

Reports

ReportDescribeResult Methods

The following are methods for `ReportDescribeResult`. All are instance methods.

IN THIS SECTION:

getReportExtendedMetadata()
Returns additional information about grouping and summaries.

getReportMetadata()
Returns unique identifiers for groupings and summaries.

getReportTypeMetadata()
Returns the fields in each section of a report type, plus filtering information for those fields.

`getReportExtendedMetadata()`

Returns additional information about grouping and summaries.

Syntax

```
public Reports.ReportExtendedMetadata getReportExtendedMetadata()
```

Return Value

Type: Reports.ReportExtendedMetadata

`getReportMetadata()`

Returns unique identifiers for groupings and summaries.

Syntax

```
public Reports.ReportMetadata getReportMetadata()
```

Return Value

Type: Reports.ReportMetadata

`getReportTypeMetadata()`

Returns the fields in each section of a report type, plus filtering information for those fields.

Syntax

```
public Reports.ReportTypeMetadata getReportTypeMetadata()
```

Return Value

Type: Reports.ReportTypeMetadata

ReportDetailRow Class

Contains data cells for a detail row of a report.

Namespace

Reports

ReportDetailRow Methods

The following are methods for `ReportDetailRow`. All are instance methods.

IN THIS SECTION:

getDataCells()
Returns a list of data cells for a detail row.

getDataCells()

Returns a list of data cells for a detail row.

Syntax

```
public LIST<Reports.ReportDataCell> getDataCells()
```

Return Value

Type: List<Reports.ReportDataCell>

ReportDivisionInfo Class

Contains information about the divisions that can be used to filter a report.

Available only if your organization uses divisions to segment data and you have the "Affected by Divisions" permission. If you do not have the "Affected by Divisions" permission, your reports include records in all divisions.

Namespace

Reports

Usage

Use to filter records in the report based on a division, like West Coast and East Coast.

ReportDivisionInfo Methods

The following are methods for `ReportDivisionInfo`.

getDefaultValue()

Returns the default division for the report.

Signature

```
public String getDefaultValue()
```

Return Value

Type: String

getValues()

Returns a list of all possible divisions for the report.

Signature

```
public List<Reports.FilterValue> getValues()
```

Return Value

Type: List<Reports.FilterValue>

ReportExtendedMetadata Class

Contains report extended metadata for a tabular, summary, or matrix report.

Namespace

Reports

Report extended metadata provides additional, detailed metadata about summary and grouping fields, including data type and label information.

ReportExtendedMetadata Methods

The following are methods for `ReportExtendedMetadata`. All are instance methods.

IN THIS SECTION:

getAggregateColumnInfo()
Returns all report summaries such as `Record Count`, `Sum`, `Average`, `Max`, `Min`, and custom summary formulas. Contains values for each summary that is listed in the report metadata.

getDetailColumnInfo()
Returns a map of two properties for each field that has detailed data identified by its unique API name. The detailed data fields are also listed in the report metadata.

getGroupingColumnInfo()
Returns a map of each row or column grouping to its metadata. Contains values for each grouping that is identified in the groupingsDown and groupingsAcross lists.

`getAggregateColumnInfo()`

Returns all report summaries such as `Record Count`, `Sum`, `Average`, `Max`, `Min`, and custom summary formulas. Contains values for each summary that is listed in the report metadata.

Syntax

```
public MAP<String,Reports.AggregateColumn> getAggregateColumnInfo()
```

Return Value

Type: Map<String,Reports.AggregateColumn>

getDetailColumnInfo()

Returns a map of two properties for each field that has detailed data identified by its unique API name. The detailed data fields are also listed in the report metadata.

Syntax

```
public MAP<String,Reports.DetailColumn> getDetailColumnInfo()
```

Return Value

Type: Map<String,Reports.DetailColumn>

getGroupingColumnInfo()

Returns a map of each row or column grouping to its metadata. Contains values for each grouping that is identified in the groupingsDown and groupingsAcross lists.

Syntax

```
public MAP<String,Reports.GroupingColumn> getGroupingColumnInfo()
```

Return Value

Type: Map<String,Reports.GroupingColumn>

ReportFact Class

Contains the fact map for the report, which represents the report's data values.

Namespace

Reports

Usage

`ReportFact` is the parent class of `ReportFactWithDetails` and `ReportFactWithSummaries`. If `includeDetails` is `true` when the report is run, the fact map is a `ReportFactWithDetails` object. If `includeDetails` is `false` when the report is run, the fact map is a `ReportFactWithSummaries` object.

ReportFact Methods

The following are methods for `ReportFact`. All are instance methods.

IN THIS SECTION:

getAggregates()
Returns summary-level data for a report, including the record count.

getKey()
Returns the unique identifier for a row or column grouping. This identifier can be used to index specific data values within each grouping.

`getAggregates()`

Returns summary-level data for a report, including the record count.

Syntax

```
public LIST<Reports.SummaryValue> getAggregates()
```

Return Value

Type: List<Reports.SummaryValue>

`getKey()`

Returns the unique identifier for a row or column grouping. This identifier can be used to index specific data values within each grouping.

Syntax

```
public String getKey()
```

Return Value

Type: String

ReportFactWithDetails Class

Contains the detailed fact map for the report, which represents the report's data values.

Namespace

Reports

Usage

The `ReportFactWithDetails` class extends the `ReportFact` class. A `ReportFactWithDetails` object is returned if `includeDetails` is set to `true` when the report is run. To access the detail values, you'll need to cast the return value of the `ReportResults.getFactMap` method to a `ReportFactWithDetails` object.

ReportFactWithDetails Methods

The following are methods for `ReportFactWithDetails`. All are instance methods.

IN THIS SECTION:

getAggregates()
Returns summary-level data for a report, including the record count.

getKey()
Returns the unique identifier for a row or column grouping. This identifier can be used to index specific data values within each grouping.

getRows()
Returns a list of detailed report data in the order of the detail columns that are provided by the report metadata.

`getAggregates()`

Returns summary-level data for a report, including the record count.

Syntax

```
public LIST<Reports.SummaryValue> getAggregates()
```

Return Value

Type: List<Reports.SummaryValue>

`getKey()`

Returns the unique identifier for a row or column grouping. This identifier can be used to index specific data values within each grouping.

Syntax

```
public String getKey()
```

Return Value

Type: String

`getRows()`

Returns a list of detailed report data in the order of the detail columns that are provided by the report metadata.

Syntax

```
public LIST<Reports.ReportDetailRow> getRows()
```

Return Value

Type: List<Reports.ReportDetailRow>

ReportFactWithSummaries Class

Contains the fact map for the report, which represents the report's data values, and includes summarized fields.

Namespace

Reports

Usage

The `ReportFactWithSummaries` class extends the `ReportFact` class. A `ReportFactWithSummaries` object is returned if `includeDetails` is set to `false` when the report is run.

ReportFactWithSummaries Methods

The following are methods for `ReportFactWithSummaries`. All are instance methods.

IN THIS SECTION:

getAggregates()
Returns summary-level data for a report, including the record count.

getKey()
Returns the unique identifier for a row or column grouping. This identifier can be used to index specific data values within each grouping.

toString()
Returns a string.

`getAggregates()`

Returns summary-level data for a report, including the record count.

Syntax

```
public LIST<Reports.SummaryValue> getAggregates()
```

Return Value

Type: List<Reports.SummaryValue>

`getKey()`

Returns the unique identifier for a row or column grouping. This identifier can be used to index specific data values within each grouping.

Syntax

```
public String getKey()
```

Return Value

Type: String

`toString()`

Returns a string.

Signature

```
public String toString()
```

Return Value

Type: String

ReportFilter Class

Contains information about a report filter, including column, operator, and value.

Namespace

Reports

IN THIS SECTION:

ReportFilter Constructors
ReportFilter Methods

ReportFilter Constructors

The following are constructors for `ReportFilter`.

IN THIS SECTION:

ReportFilter()
Creates a new instance of the `Reports.ReportFilter` class. You can then set values by using the "set" methods.
ReportFilter(column, operator, value)
Creates a new instance of the `Reports.ReportFilter` class by using the specified parameters.
ReportFilter(column, operator, value, filterType)
Creates a new instance of the `Reports.ReportFilter` class by using the specified parameters.
ReportFilter(column, operator, value, filterType, entityName)
Creates a new instance of the `Reports.ReportFilter` class by using the specified parameters.

`ReportFilter()`

Creates a new instance of the `Reports.ReportFilter` class. You can then set values by using the "set" methods.

Signature

```
public ReportFilter()
```

ReportFilter(column, operator, value)

Creates a new instance of the `Reports.ReportFilter` class by using the specified parameters.

Signature

```
public ReportFilter(String column, String operator, String value)
```

Parameters

column
Type: String

operator
Type: String

value
Type: String

ReportFilter(column, operator, value, filterType)

Creates a new instance of the `Reports.ReportFilter` class by using the specified parameters.

Syntax

```
public ReportFilterType(String column, String operator, String value,
Reports.ReportFilterType filterType)
```

Parameters

column
Type: String

operator
Type: String

value
Type: String

filterType
Type: ReportFilterType Enum on page 2982

ReportFilter(column, operator, value, filterType, entityName)

Creates a new instance of the `Reports.ReportFilter` class by using the specified parameters.

Syntax

```
public ReportFilterType(String column, String operator, String value,
Reports.ReportFilterType filterType, String entityName)
```

Parameters

column
Type: String

operator
Type: String

value
Type: String

filterType
Type: ReportFilterType Enum on page 2982

entityName
Type: String

ReportFilter Methods

The following are methods for `ReportFilter`. All are instance methods.

IN THIS SECTION:

getColumn()
Returns the unique API name for the field that's being filtered.

getEntityName()
Returns the entity name used in the report filter. Use the entity name to handle ambiguous field names across entities, specifically when using cross filters.

getFilterType()
Returns the type of report filter.

getOperator()
Returns the unique API name for the condition that is used to filter a field, such as "greater than" or "not equal to." Filter conditions depend on the data type of the field.

getValue()
Returns the value that the field is being filtered by. For example, the field `Age` can be filtered by a numeric value.

setColumn(column)
Sets the unique API name for the field that's being filtered.

setEntityName(entityName)
Sets the entity name to use in the report filter. Use the entity name to handle ambiguous field names across entities, specifically when using cross filters.

setFilterType()
Sets the type of report filter.

setOperator(operator)
Sets the unique API name for the condition that is used to filter a field, such as "greater than" or "not equal to." Filter conditions depend on the data type of the field.

setValue(value)
Sets the value by which a field can be filtered. For example, the field `Age` can be filtered by a numeric value.

toString(column)
Returns a string representation of the filter.

getColumn()

Returns the unique API name for the field that's being filtered.

Syntax

```
public String getColumn()
```

Return Value

Type: String

getEntityName()

Returns the entity name used in the report filter. Use the entity name to handle ambiguous field names across entities, specifically when using cross filters.

Syntax

```
public String getEntityName()
```

Return Value

Type: String

getFilterType()

Returns the type of report filter.

Syntax

```
public String getFilterType()
```

Return Value

Type: ReportFilterType Enum on page 2982

getOperator()

Returns the unique API name for the condition that is used to filter a field, such as "greater than" or "not equal to." Filter conditions depend on the data type of the field.

Syntax

```
public String getOperator()
```

Return Value

Type: String

`getValue()`

Returns the value that the field is being filtered by. For example, the field `Age` can be filtered by a numeric value.

Syntax

```
public String getValue()
```

Return Value

Type: String

`setColumn(column)`

Sets the unique API name for the field that's being filtered.

Syntax

```
public Void setColumn(String column)
```

Parameters

column
Type: String

Return Value

Type: Void

`setEntityName(entityName)`

Sets the entity name to use in the report filter. Use the entity name to handle ambiguous field names across entities, specifically when using cross filters.

Syntax

```
public Void setEntityName(String entityName)
```

Parameters

operator
Type: String

Return Value

Type: Void

`setFilterType()`

Sets the type of report filter.

Syntax

```
public Void setFilterType(String column)
```

Parameters

column
Type: String

Return Value

Type: Void

`setOperator(operator)`

Sets the unique API name for the condition that is used to filter a field, such as “greater than” or “not equal to.” Filter conditions depend on the data type of the field.

Syntax

```
public Void setOperator(String operator)
```

Parameters

operator
Type: String

Return Value

Type: Void

`setValue(value)`

Sets the value by which a field can be filtered. For example, the field `Age` can be filtered by a numeric value.

Syntax

```
public Void setValue(String value)
```

Parameters

value
Type: String

Return Value

Type: Void

`toString(column)`

Returns a string representation of the filter.

Signature

```
public String toString()
```

Return Value

Type: String

ReportFormat Enum

Contains the possible report format types.

Namespace

Reports

Enum Values

The following are the values of the `Reports.ReportFormat` enum.

Value	Description
`MATRIX`	Matrix report format
`SUMMARY`	Summary report format
`TABULAR`	Tabular report format

ReportFilterType Enum

The types of values included in a report filter type.

Enum Values

The following are the values of the `Reports.ReportFilterType` enum.

Value	Description
`fieldToField`	Field-to-field filter
`fieldValue`	Field-to-value filter

ReportInstance Class

Returns an instance of a report that was run asynchronously. Retrieves the results for that instance.

Namespace

Reports

ReportInstance Methods

The following are methods for `ReportInstance`. All are instance methods.

IN THIS SECTION:

getCompletionDate()
Returns the date and time when the instance of the report finished running. The completion date is available only if the report instance ran successfully or couldn't be run because of an error. Date and time information is in ISO-8601 format.

getId()
Returns the unique ID for an instance of a report that was run asynchronously.

getOwnerId()
Returns the ID of the user who created the report instance.

getReportId()
Returns the unique ID of the report this instance is based on.

getReportResults()
Retrieves results for an instance of an asynchronous report. When you request your report, you can specify whether to summarize data or include details.

getRequestDate()
Returns the date and time when an instance of the report was run. Date and time information is in ISO-8601 format.

getStatus()
Returns the status of a report.

`getCompletionDate()`

Returns the date and time when the instance of the report finished running. The completion date is available only if the report instance ran successfully or couldn't be run because of an error. Date and time information is in ISO-8601 format.

Syntax

```
public Datetime getCompletionDate()
```

Return Value

Type: Datetime

`getId()`

Returns the unique ID for an instance of a report that was run asynchronously.

Syntax

```
public Id getId()
```

Return Value

Type: Id

getOwnerId()

Returns the ID of the user who created the report instance.

Syntax

```
public Id getOwnerId()
```

Return Value

Type: Id

getReportId()

Returns the unique ID of the report this instance is based on.

Syntax

```
public Id getReportId()
```

Return Value

Type: Id

getReportResults()

Retrieves results for an instance of an asynchronous report. When you request your report, you can specify whether to summarize data or include details.

Syntax

```
public Reports.ReportResults getReportResults()
```

Return Value

Type: Reports.ReportResults

getRequestDate()

Returns the date and time when an instance of the report was run. Date and time information is in ISO-8601 format.

Syntax

```
public Datetime getRequestDate()
```

Return Value

Type: Datetime

`getStatus()`

Returns the status of a report.

Syntax

```
public String getStatus()
```

Return Value

Type: String

Usage

- `New` if the report run was recently triggered through a request.
- `Success` if the report ran.
- `Running` if the report is being run.
- `Error` if the report run failed. The instance of a report run can return an error if, for example, your permission to access the report was removed after you requested the run.

ReportManager Class

Runs a report synchronously or asynchronously and with or without details.

Namespace

Reports

Usage

Gets instances of reports and describes the metadata of Reports.

ReportManager Methods

The following are methods for `ReportManager`. All methods are static.

IN THIS SECTION:

describeReport(reportId)
Retrieves report, report type, and extended metadata for a tabular, summary, or matrix report.

getDatatypeFilterOperatorMap()
Lists the field data types that you can use to filter the report.

getReportInstance(instanceId)

Retrieves results for an instance of a report that has been run asynchronously. The settings you use when you run your asynchronous report determine whether you can retrieve summary data or detailed data.

getReportInstances(reportId)

Returns a list of instances for a report that was run asynchronously. Each item in the list represents a separate instance of the report, with metadata for the time at which the report was run.

runAsyncReport(reportId, reportMetadata, includeDetails)

Runs a report asynchronously with the report ID. Includes details if `includeDetails` is set to `true`. Filters the report based on the report metadata in `reportMetadata`.

runAsyncReport(reportId, includeDetails)

Runs a report asynchronously with the report ID. Includes details if `includeDetails` is set to `true`.

runAsyncReport(reportId, reportMetadata)

Runs a report asynchronously with the report ID. Filters the results based on the report metadata in `reportMetadata`.

runAsyncReport(reportId)

Runs a report asynchronously with the report ID.

runReport(reportId, reportMetadata, includeDetails)

Runs a report immediately with the report ID. Includes details if `includeDetails` is set to `true`. Filters the results based on the report metadata in `reportMetadata`.

runReport(reportId, includeDetails)

Runs a report immediately with the report ID. Includes details if `includeDetails` is set to `true`.

runReport(reportId, reportMetadata)

Runs a report immediately with the report ID. Filters the results based on the report metadata in `rmData`.

runReport(reportId)

Runs a report immediately with the report ID.

`describeReport(reportId)`

Retrieves report, report type, and extended metadata for a tabular, summary, or matrix report.

Syntax

```
public static Reports.ReportDescribeResult describeReport(Id reportId)
```

Parameters

reportId
Type: Id

Return Value

Type: Reports.ReportDescribeResult

`getDatatypeFilterOperatorMap()`

Lists the field data types that you can use to filter the report.

Syntax

```
public static MAP<String,LIST<Reports.FilterOperator>> getDatatypeFilterOperatorMap()
```

Return Value

Type: Map<String, List<Reports.FilterOperator>>

`getReportInstance(instanceId)`

Retrieves results for an instance of a report that has been run asynchronously. The settings you use when you run your asynchronous report determine whether you can retrieve summary data or detailed data.

Syntax

```
public static Reports.ReportInstance getReportInstance(Id instanceId)
```

Parameters

instanceId
Type: Id

Return Value

Type: Reports.ReportInstance

`getReportInstances(reportId)`

Returns a list of instances for a report that was run asynchronously. Each item in the list represents a separate instance of the report, with metadata for the time at which the report was run.

Syntax

```
public static LIST<Reports.ReportInstance> getReportInstances(Id reportId)
```

Parameters

reportId
Type: Id

Return Value

Type: List<Reports.ReportInstance>

`runAsyncReport(reportId, reportMetadata, includeDetails)`

Runs a report asynchronously with the report ID. Includes details if *includeDetails* is set to `true`. Filters the report based on the report metadata in *reportMetadata*.

Syntax

```
public static Reports.ReportInstance runAsyncReport(Id reportId, Reports.ReportMetadata
reportMetadata, Boolean includeDetails)
```

Parameters

reportId
Type: Id

reportMetadata
Type: Reports.ReportMetadata

includeDetails
Type: Boolean

Return Value

Type: Reports.ReportInstance

runAsyncReport(reportId, includeDetails)

Runs a report asynchronously with the report ID. Includes details if *includeDetails* is set to `true`.

Syntax

```
public static Reports.ReportInstance runAsyncReport(Id reportId, Boolean includeDetails)
```

Parameters

reportId
Type: Id

includeDetails
Type: Boolean

Return Value

Type: Reports.ReportInstance

runAsyncReport(reportId, reportMetadata)

Runs a report asynchronously with the report ID. Filters the results based on the report metadata in *reportMetadata*.

Syntax

```
public static Reports.ReportInstance runAsyncReport(Id reportId, Reports.ReportMetadata
reportMetadata)
```

Parameters

reportId
Type: Id

reportMetadata
Type: Reports.ReportMetadata

Return Value

Type: Reports.ReportInstance

runAsyncReport(reportId)

Runs a report asynchronously with the report ID.

Syntax

```
public static Reports.ReportInstance runAsyncReport(Id reportId)
```

Parameters

reportId
Type: Id

Return Value

Type: Reports.ReportInstance

runReport(reportId, reportMetadata, includeDetails)

Runs a report immediately with the report ID. Includes details if *includeDetails* is set to `true`. Filters the results based on the report metadata in *reportMetadata*.

Syntax

```
public static Reports.ReportResults runReport(Id reportId, Reports.ReportMetadata
reportMetadata, Boolean includeDetails)
```

Parameters

reportId
Type: Id

reportMetadata
Type: Reports.ReportMetadata

includeDetails
Type: Boolean

Return Value

Type: Reports.ReportResults

runReport(reportId, includeDetails)

Runs a report immediately with the report ID. Includes details if *includeDetails* is set to `true`.

Syntax

```
public static Reports.ReportResults runReport(Id reportId, Boolean includeDetails)
```

Parameters

reportId
Type: Id

includeDetails
Type: Boolean

Return Value

Type: Reports.ReportResults

runReport(reportId, reportMetadata)

Runs a report immediately with the report ID. Filters the results based on the report metadata in *rmData*.

Syntax

```
public static Reports.ReportResults runReport(Id reportId, Reports.ReportMetadata
reportMetadata)
```

Parameters

reportId
Type: Id

reportMetadata
Type: Reports.ReportMetadata Reports.ReportMetadata

Return Value

Type: Reports.ReportResults

runReport(reportId)

Runs a report immediately with the report ID.

Syntax

```
public static Reports.ReportResults runReport(Id reportId)
```

Parameters

reportId
Type: Id

Return Value

Type: Reports.ReportResults

ReportMetadata Class

Contains report metadata for a tabular, summary, or matrix report.

Namespace

Reports

Usage

Report metadata gives information about the report as a whole, such as the report type, format, summary fields, row or column groupings, and filters that are saved to the report. You can use the `ReportMetadata` class to retrieve report metadata and to set metadata that can be used to filter a report.

ReportMetadata Methods

The following are methods for `ReportMetadata`. All are instance methods.

IN THIS SECTION:

getAggregates()
Returns unique identifiers for summary or custom summary formula fields in the report.

getBuckets()
Returns a list of bucket fields in the report.

getCrossFilters()
Returns information about cross filters applied to a report.

getCurrencyCode()
Returns report currency, such as USD, EUR, or GBP, for an organization that has multicurrency enabled. The value is `null` if the organization does not have multicurrency enabled.

getCustomSummaryFormula()
Returns information about custom summary formulas in a report.

getDescription()
Returns the description of the report.

getDetailColumns()
Returns unique API names (column names) for the fields that contain detailed data. For example, the method might return the following values: "OPPORTUNITY_NAME, TYPE, LEAD_SOURCE, AMOUNT."

getDeveloperName()
Returns the report API name. For example, the method might return the following value: "Closed_Sales_This_Quarter."

getDivision()
Returns the division specified in the report.

getGroupingsAcross()
Returns column groupings in a report.

getGroupingsDown()
Returns row groupings for a report.

getHasDetailRows()
Indicates whether the report has detail rows.

getHasRecordCount()
Indicates whether the report shows the total number of records.

getHistoricalSnapshotDates()
Returns a list of historical snapshot dates.

getId()
Returns the unique report ID.

getName()
Returns the report name.

getReportBooleanFilter()
Returns logic to parse custom field filters. The value is `null` when filter logic is not specified.

getReportFilters()
Returns a list of each custom filter in the report along with the field name, filter operator, and filter value.

getReportFormat()
Returns the format of the report.

getReportType()
Returns the unique API name and display name for the report type.

getScope()
Returns the API name for the scope defined for the report. Scope values depend on the report type.

getShowGrandTotal()
Indicates whether the report shows the grand total.

getShowSubtotals()
Indicates whether the report shows subtotals, such as column or row totals.

getSortBy()
Returns the list of columns on which the report is sorted. Currently, you can sort on only one column.

getStandardDateFilter()
Returns information about the standard date filter for the report, such as the start date, end date, date range, and date field API name.

getStandardFilters()
Returns a list of standard filters for the report.

getTopRows()
Returns information about a row limit filter, including the number of rows returned and the sort order.

setAggregates(aggregates)
Sets unique identifiers for standard or custom summary formula fields in the report.

setBuckets(buckets)
Creates bucket fields in a report.

setCrossFilters(crossFilters)
Applies cross filters to a report.

setCurrencyCode(currencyCode)
Sets the currency, such as USD, EUR, or GBP, for report summary fields in an organization that has multicurrency enabled.

setCustomSummaryFormula(customSummaryFormula)

Adds a custom summary formula to a report.

setDescription(description)

Sets the description of the report.

setDetailColumns(detailColumns)

Sets the unique API names for the fields that contain detailed data—for example, `OPPORTUNITY_NAME`, `TYPE`, `LEAD_SOURCE`, or `AMOUNT`.

setDeveloperName(developerName)

Sets the report API name—for example, `Closed_Sales_This_Quarter`.

setDivision(division)

Sets the division of the report.

setGroupingsAcross(groupingInfo)

Sets column groupings in a report.

setGroupingsDown(groupingInfo)

Sets row groupings for a report.

setHasDetailRows(hasDetailRows)

Specifies whether the report has detail rows.

setHasRecordCount(hasRecordCount)

Specifies whether the report is configured to show the total number of records.

setHistoricalSnapshotDates(historicalSnapshot)

Sets a list of historical snapshot dates.

setId(id)

Sets the unique report ID.

setName(name)

Sets the report name.

setReportBooleanFilter(reportBooleanFilter)

Sets logic to parse custom field filters.

setReportFilters(reportFilters)

Sets a list of each custom filter in the report along with the field name, filter operator, and filter value.

setReportFormat(format)

Sets the format of the report.

setReportType(reportType)

Sets the unique API name and display name for the report type.

setScope(scopeName)

Sets the API name for the scope defined for the report. Scope values depend on the report type.

setShowGrandTotal(showGrandTotal)

Specifies whether the report shows the grand total.

setShowSubtotals(showSubtotals)

Specifies whether the report shows subtotals, such as column or row totals.

setSortBy(column)

Sets the list of columns on which the report is sorted. Currently, you can only sort on one column.

setStandardDateFilter(dateFilter)

Sets the standard date filter—which includes the start date, end date, date range, and date field API name—for the report.

setStandardFilters(filters)

Sets one or more standard filters on the report.

setTopRows(topRows)

Applies a row limit filter to a report.

getAggregates()

Returns unique identifiers for summary or custom summary formula fields in the report.

Syntax

```
public LIST<String> getAggregates()
```

Return Value

Type: List<String>

Usage

For example:

- `a!Amount` represents the average for the `Amount` column.
- `s!Amount` represents the sum of the `Amount` column.
- `m!Amount` represents the minimum value of the `Amount` column.
- `x!Amount` represents the maximum value of the `Amount` column.
- `s!<customfieldID>` represents the sum of a custom field column. For custom fields and custom report types, the identifier is a combination of the summary type and the field ID.

getBuckets()

Returns a list of bucket fields in the report.

Signature

```
public List<Reports.BucketField> getBuckets()
```

Return Value

Type: List<Reports.BucketField>

getCrossFilters()

Returns information about cross filters applied to a report.

Signature

```
public Reports.CrossFilter getCrossFilters()
```

Return Value

Type: List<Reports.CrossFilter>

getCurrencyCode()

Returns report currency, such as USD, EUR, or GBP, for an organization that has multicurrency enabled. The value is `null` if the organization does not have multicurrency enabled.

Syntax

```
public String getCurrencyCode()
```

Return Value

Type: String

getCustomSummaryFormula()

Returns information about custom summary formulas in a report.

Signature

```
public Map<String,Reports.ReportCsf> getCustomSummaryFormula()
```

Return Value

Type: Map<String,Reports.ReportCsf>

getDescription()

Returns the description of the report.

Signature

```
public String getDescription()
```

Return Value

Type: String

getDetailColumns()

Returns unique API names (column names) for the fields that contain detailed data. For example, the method might return the following values: "OPPORTUNITY_NAME, TYPE, LEAD_SOURCE, AMOUNT."

Syntax

```
public LIST<String> getDetailColumns()
```

Return Value

Type: List<String>

`getDeveloperName()`

Returns the report API name. For example, the method might return the following value: "Closed_Sales_This_Quarter."

Syntax

```
public String getDeveloperName()
```

Return Value

Type: String

`getDivision()`

Returns the division specified in the report.

Note: Reports that use standard filters (such as My Cases or My Team's Accounts) show records in all divisions. These reports can't be further limited to a specific division.

Signature

```
public String getDivision()
```

Return Value

Type: String

`getGroupingsAcross()`

Returns column groupings in a report.

Syntax

```
public LIST<Reports.GroupingInfo> getGroupingsAcross()
```

Return Value

Type: List<Reports.GroupingInfo>

Usage

The identifier is:

- An empty array for reports in summary format, because summary reports don't include column groupings
- `BucketField_(ID)` for bucket fields
- The ID of a custom field when the custom field is used for a column grouping

getGroupingsDown()

Returns row groupings for a report.

Syntax

```
public LIST<Reports.GroupingInfo> getGroupingsDown()
```

Return Value

Type: List<Reports.GroupingInfo>

Usage

The identifier is:

- `BucketField_(ID)` for bucket fields
- The ID of a custom field when the custom field is used for grouping

getHasDetailRows()

Indicates whether the report has detail rows.

Signature

```
public Boolean getHasDetailRows()
```

Return Value

Type: Boolean

getHasRecordCount()

Indicates whether the report shows the total number of records.

Signature

```
public Boolean getHasRecordCount()
```

Return Value

Type: Boolean

getHistoricalSnapshotDates()

Returns a list of historical snapshot dates.

Syntax

```
public LIST<String> getHistoricalSnapshotDates()
```

Return Value

Type: List<String>

getId()

Returns the unique report ID.

Syntax

```
public Id getId()
```

Return Value

Type: Id

getName()

Returns the report name.

Syntax

```
public String getName()
```

Return Value

Type: String

getReportBooleanFilter()

Returns logic to parse custom field filters. The value is `null` when filter logic is not specified.

Syntax

```
public String getReportBooleanFilter()
```

Return Value

Type: String

getReportFilters()

Returns a list of each custom filter in the report along with the field name, filter operator, and filter value.

Syntax

```
public LIST<Reports.ReportFilter> getReportFilters()
```

Return Value

Type: List<Reports.ReportFilter>

getReportFormat()

Returns the format of the report.

Syntax

```
public Reports.ReportFormat getReportFormat()
```

Return Value

Type: Reports.ReportFormat

Usage

This value can be:

- TABULAR
- SUMMARY
- MATRIX

getReportType()

Returns the unique API name and display name for the report type.

Syntax

```
public Reports.ReportType getReportType()
```

Return Value

Type: Reports.ReportType

getScope()

Returns the API name for the scope defined for the report. Scope values depend on the report type.

Signature

```
public String getScope()
```

Return Value

Type: String

getShowGrandTotal()

Indicates whether the report shows the grand total.

Signature

```
public Boolean getShowGrandTotal()
```

Return Value

Type: Boolean

getShowSubtotals()

Indicates whether the report shows subtotals, such as column or row totals.

Signature

```
public Boolean getShowSubtotals()
```

Return Value

Type: Boolean

getSortBy()

Returns the list of columns on which the report is sorted. Currently, you can sort on only one column.

Signature

```
public List<Reports.SortColumn> getSortBy()
```

Return Value

Type: List<Reports.SortColumn>

getStandardDateFilter()

Returns information about the standard date filter for the report, such as the start date, end date, date range, and date field API name.

Signature

```
public Reports.StandardDateFilter getStandardDateFilter()
```

Return Value

Type: Reports.StandardDateFilter

getStandardFilters()

Returns a list of standard filters for the report.

Signature

```
public List<Reports.StandardFilter> getStandardFilters()
```

Return Value

Type: List<Reports.StandardFilter>

getTopRows()

Returns information about a row limit filter, including the number of rows returned and the sort order.

Signature

```
public Reports.TopRows getTopRows()
```

Return Value

Type: Reports.TopRows

setAggregates(aggregates)

Sets unique identifiers for standard or custom summary formula fields in the report.

Signature

```
public void setAggregates(List<String> aggregates)
```

Parameters

aggregates
Type: List<String>

Return Value

Type: void

setBuckets(buckets)

Creates bucket fields in a report.

Signature

```
public void setBuckets(List<Reports.BucketField> buckets)
```

Parameters

buckets
Type: List<Reports.BucketField>

Return Value

Type: void

setCrossFilters(crossFilters)

Applies cross filters to a report.

Signature

```
public void setCrossFilters(List<Reports.CrossFilter> crossFilters)
```

Parameters

crossFilter
Type: List<Reports.CrossFilter>

Return Value

Type: void

setCurrencyCode(currencyCode)

Sets the currency, such as USD, EUR, or GBP, for report summary fields in an organization that has multicurrency enabled.

Signature

```
public void setCurrencyCode(String currencyCode)
```

Parameters

currencyCode
Type: String

Return Value

Type: void

setCustomSummaryFormula(customSummaryFormula)

Adds a custom summary formula to a report.

Signature

```
public void setCustomSummaryFormula(MAP<String,Reports.ReportCsf> customSummaryFormula)
```

Parameters

customSummaryFormula
Type: Map<String, Reports.ReportCsf>

Return Value

Type: void

setDescription(description)

Sets the description of the report.

Signature

```
public void setDescription(String description)
```

Parameters

description
Type: String

Return Value

Type: void

setDetailColumns(detailColumns)

Sets the unique API names for the fields that contain detailed data—for example, `OPPORTUNITY_NAME`, `TYPE`, `LEAD_SOURCE`, or `AMOUNT`.

Signature

```
public void setDetailColumns(List<String> detailColumns)
```

Parameters

detailColumns
Type: List<String>

Return Value

Type: void

setDeveloperName(developerName)

Sets the report API name—for example, `Closed_Sales_This_Quarter`.

Signature

```
public void setDeveloperName(String developerName)
```

Parameters

developerName
Type: String

Return Value

Type: void

setDivision(division)

Sets the division of the report.

Note: Reports that use standard filters (such as My Cases or My Team's Accounts) show records in all divisions. These reports can't be further limited to a specific division.

Signature

```
public void setDivision(String division)
```

Parameters

division
Type: String

Return Value

Type: void

setGroupingsAcross(groupingInfo)

Sets column groupings in a report.

Signature

```
public void setGroupingsAcross(List<Reports.GroupingInfo> groupingInfo)
```

Parameters

groupingInfo
Type: List<Reports.GroupingInfo>

Return Value

Type: void

setGroupingsDown(groupingInfo)

Sets row groupings for a report.

Signature

```
public void setGroupingsDown(List<Reports.GroupingInfo> groupingInfo)
```

Parameters

groupingInfo
Type: List<Reports.GroupingInfo>

Return Value

Type: void

setHasDetailRows(hasDetailRows)

Specifies whether the report has detail rows.

Signature

```
public void setHasDetailRows(Boolean hasDetailRows)
```

Parameters

hasDetailRows
Type: Boolean

Return Value

Type: void

setHasRecordCount(hasRecordCount)

Specifies whether the report is configured to show the total number of records.

Signature

```
public void setHasRecordCount(Boolean hasRecordCount)
```

Parameters

hasRecordCount
Type: Boolean

Return Value

Type: void

setHistoricalSnapshotDates(historicalSnapshot)

Sets a list of historical snapshot dates.

Syntax

```
public Void setHistoricalSnapshotDates(LIST<String> historicalSnapshot)
```

Parameters

historicalSnapshot
Type: List<String>

Return Value

Type: Void

`setId(id)`

Sets the unique report ID.

Signature

```
public void setId(Id id)
```

Parameters

id
Type: Id

Return Value

Type: void

`setName(name)`

Sets the report name.

Signature

```
public void setName(String name)
```

Parameters

name
Type: String

Return Value

Type: void

`setReportBooleanFilter(reportBooleanFilter)`

Sets logic to parse custom field filters.

Syntax

```
public Void setReportBooleanFilter(String reportBooleanFilter)
```

Parameters

reportBooleanFilter
Type: String

Return Value

Type: Void

setReportFilters(reportFilters)

Sets a list of each custom filter in the report along with the field name, filter operator, and filter value.

Syntax

```
public Void setReportFilters(LIST<Reports.ReportFilter> reportFilters)
```

Parameters

reportFilters
Type: List<Reports.ReportFilter>

Return Value

Type: Void

setReportFormat(format)

Sets the format of the report.

Signature

```
public void setReportFormat(Reports.ReportFormat format)
```

Parameters

format
Type: Reports.ReportFormat

Return Value

Type: void

setReportType(reportType)

Sets the unique API name and display name for the report type.

Signature

```
public void setReportType(Reports.ReportType reportType)
```

Parameters

reportType
Type: Reports.ReportType

Return Value

Type: void

setScope(scopeName)

Sets the API name for the scope defined for the report. Scope values depend on the report type.

Signature

```
public void setScope(String scopeName)
```

Parameters

scopeName
Type: String

Return Value

Type: void

setShowGrandTotal(showGrandTotal)

Specifies whether the report shows the grand total.

Signature

```
public void setShowGrandTotal(Boolean showGrandTotal)
```

Parameters

showGrandTotal
Type: Boolean

Return Value

Type: void

setShowSubtotals(showSubtotals)

Specifies whether the report shows subtotals, such as column or row totals.

Signature

```
public void setShowSubtotals(Boolean showSubtotals)
```

Parameters

showSubtotals
Type: Boolean

Return Value

Type: void

setSortBy(column)

Sets the list of columns on which the report is sorted. Currently, you can only sort on one column.

Signature

```
public void setSortBy(List<Reports.SortColumn> column)
```

Parameters

column
Type: List<Reports.SortColumn>

Return Value

Type: void

setStandardDateFilter(dateFilter)

Sets the standard date filter—which includes the start date, end date, date range, and date field API name—for the report.

Signature

```
public void setStandardDateFilter(Reports.StandardDateFilter dateFilter)
```

Parameters

dateFilter
Type: Reports.StandardDateFilter

Return Value

Type: void

setStandardFilters(filters)

Sets one or more standard filters on the report.

Signature

```
public void setStandardFilters(List<Reports.StandardFilter> filters)
```

Parameters

filters
Type: List<Reports.StandardFilter>

Return Value

Type: void

`setTopRows(topRows)`

Applies a row limit filter to a report.

Signature

```
public Reports.TopRows setTopRows(Reports.TopRows topRows)
```

Parameters

`topRows`
Type: Reports.TopRows

Return Value

Type: void

ReportResults Class

Contains the results of running a report.

Namespace

Reports

ReportResults Methods

The following are methods for `ReportResults`. All are instance methods.

IN THIS SECTION:

getAllData()
Returns all report data.

getFactMap()
Returns summary-level data or summary and detailed data for each row or column grouping. Detailed data is available if the `includeDetails` parameter is set to `true` when the report is run.

getGroupingsAcross()
Returns a collection of column groupings, keys, and values.

getGroupingsDown()
Returns a collection of row groupings, keys, and values.

getHasDetailRows()
Returns information about whether the fact map has detail rows.

getReportExtendedMetadata()
Returns additional, detailed metadata about the report, including data type and label information for groupings and summaries.

getReportMetadata()
Returns metadata about the report, including grouping and summary information.

getAllData()

Returns all report data.

Syntax

```
public Boolean getAllData()
```

Return Value

Type: Boolean

Usage

When `true`, indicates that all report results are returned.

When `false`, indicates that results are returned for the same number of rows as in a report run in Salesforce.

Note: For reports that contain too many records, use filters to refine results.

getFactMap()

Returns summary-level data or summary and detailed data for each row or column grouping. Detailed data is available if the `includeDetails` parameter is set to `true` when the report is run.

Syntax

```
public MAP<String,Reports.ReportFact> getFactMap()
```

Return Value

Type: Map<String,Reports.ReportFact>

getGroupingsAcross()

Returns a collection of column groupings, keys, and values.

Syntax

```
public Reports.Dimension getGroupingsAcross()
```

Return Value

Type: Reports.Dimension

getGroupingsDown()

Returns a collection of row groupings, keys, and values.

Syntax

```
public Reports.Dimension getGroupingsDown()
```

Return Value

Type: Reports.Dimension

`getHasDetailRows()`

Returns information about whether the fact map has detail rows.

Syntax

```
public Boolean getHasDetailRows()
```

Return Value

Type: Boolean

Usage

- When `true`, indicates that the fact map returns values for summary-level and record-level data.
- When `false`, indicates that the fact map returns summary values.

`getReportExtendedMetadata()`

Returns additional, detailed metadata about the report, including data type and label information for groupings and summaries.

Syntax

```
public Reports.ReportExtendedMetadata getReportExtendedMetadata()
```

Return Value

Type: Reports.ReportExtendedMetadata

`getReportMetadata()`

Returns metadata about the report, including grouping and summary information.

Syntax

```
public Reports.ReportMetadata getReportMetadata()
```

Return Value

Type: Reports.ReportMetadata

ReportScopeInfo Class

Contains information about possible scope values that you can choose. Scope values depend on the report type. For example, you can set the scope for opportunity reports to `All opportunities`, `My team's opportunities`, or `My opportunities`.

Namespace

Reports

IN THIS SECTION:

ReportScopeInfo Methods

ReportScopeInfo Methods

The following are methods for `ReportScopeInfo`.

IN THIS SECTION:

getDefaultValue()
Returns the default scope of the data to display in the report.

getValues()
Returns a list of scope values specified for the report.

`getDefaultValue()`

Returns the default scope of the data to display in the report.

Signature

```
public String getDefaultValue()
```

Return Value

Type: String

`getValues()`

Returns a list of scope values specified for the report.

Signature

```
public List<Reports.ReportScopeValue> getValues()
```

Return Value

Type: List<Reports.ReportScopeValue>

ReportScopeValue Class

Contains information about a possible scope value. Scope values depend on the report type. For example, you can set the scope for opportunity reports to `All opportunities`, `My team's opportunities`, or `My opportunities`.

Namespace

Reports

IN THIS SECTION:

ReportScopeValue Methods

ReportScopeValue Methods

The following are methods for `ReportScopeValue`.

IN THIS SECTION:

getAllowsDivision()
Returns a boolean value that indicates whether you can segment the report by this scope.

getLabel()
Returns the display name of the scope of the report.

getValue()
Returns the scope value for the report.

`getAllowsDivision()`

Returns a boolean value that indicates whether you can segment the report by this scope.

Signature

```
public Boolean getAllowsDivision()
```

Return Value

Type: Boolean

`getLabel()`

Returns the display name of the scope of the report.

Signature

```
public String getLabel()
```

Return Value

Type: String

`getValue()`

Returns the scope value for the report.

Signature

```
public String getValue()
```

Return Value

Type: String

ReportType Class

Contains the unique API name and display name for the report type.

Namespace

Reports

ReportType Methods

The following are methods for `ReportType`. All are instance methods.

IN THIS SECTION:

getLabel()
Returns the localized display name of the report type.

getType()
Returns the unique identifier of the report type.

`getLabel()`

Returns the localized display name of the report type.

Syntax

```
public String getLabel()
```

Return Value

Type: String

`getType()`

Returns the unique identifier of the report type.

Syntax

```
public String getType()
```

Return Value

Type: String

ReportTypeColumn Class

Contains detailed report type metadata about a field, including data type, display name, and filter values.

Namespace

Reports

ReportTypeColumn Methods

The following are methods for `ReportTypeColumn`. All are instance methods.

IN THIS SECTION:

getDataType()
Returns the data type of the field.

getFilterValues()
If the field data type is picklist, multi-select picklist, boolean, or checkbox, returns all filter values for a field. For example, checkbox fields always have a value of `true` or `false`. For fields of other data types, the filter value is an empty array, because their values can't be determined.

getFilterable()
If the field is of a type that can't be filtered, returns `False`. For example, fields of the type `Encrypted Text` can't be filtered.

getLabel()
Returns the localized display name of the field.

getName()
Returns the unique API name of the field.

`getDataType()`

Returns the data type of the field.

Syntax

```
public Reports.ColumnDataType getDataType()
```

Return Value

Type: Reports.ColumnDataType

`getFilterValues()`

If the field data type is picklist, multi-select picklist, boolean, or checkbox, returns all filter values for a field. For example, checkbox fields always have a value of `true` or `false`. For fields of other data types, the filter value is an empty array, because their values can't be determined.

Syntax

```
public LIST<Reports.FilterValue> getFilterValues()
```

Return Value

Type: List<Reports.FilterValue>

`getFilterable()`

If the field is of a type that can't be filtered, returns `False`. For example, fields of the type `Encrypted Text` can't be filtered.

Syntax

```
public Boolean getFilterable()
```

Return Value

Type: Boolean

`getLabel()`

Returns the localized display name of the field.

Syntax

```
public String getLabel()
```

Return Value

Type: String

`getName()`

Returns the unique API name of the field.

Syntax

```
public String getName()
```

Return Value

Type: String

ReportTypeColumnCategory Class

Information about categories of fields in a report type.

Namespace

Reports

Usage

A report type column category is a set of fields that the report type grants access to. For example, an opportunity report has categories like *Opportunity Information* and *Primary Contact*. The Opportunity Information category has fields like *Amount*, *Probability*, and *Close Date*.

Get category information about a report by first getting the report metadata:

```
// Get the report ID
List <Report> reportList = [SELECT Id,DeveloperName FROM Report where DeveloperName =
'Q1_Opportunities2'];

String reportId = (String)reportList.get(0).get('Id');

// Describe the report
Reports.ReportDescribeResult describeResults =
Reports.ReportManager.describeReport(reportId);

// Get report type metadata
Reports.ReportTypeMetadata reportTypeMetadata = describeResults.getReportTypeMetadata();

// Get report type column categories
List<Reports.ReportTypeColumnCategory> reportTypeColumnCategories =
reportTypeMetadata.getCategories();

System.debug('reportTypeColumnCategories: ' + reportTypeColumnCategories);
```

ReportTypeColumnCategory Methods

The following are methods for `ReportTypeColumnCategory`. All are instance methods.

IN THIS SECTION:

getColumns()
Returns information for all fields in the report type. The information is organized by each section's unique API name.

getLabel()
Returns the localized display name of a section in the report type under which fields are organized. For example, in an Accounts with Contacts custom report type, `Account General` is the display name of the section that contains fields on general account information.

`getColumns()`

Returns information for all fields in the report type. The information is organized by each section's unique API name.

Syntax

```
public MAP<String,Reports.ReportTypeColumn> getColumns()
```

Return Value

Type: Map<String,Reports.ReportTypeColumn>

getLabel()

Returns the localized display name of a section in the report type under which fields are organized. For example, in an Accounts with Contacts custom report type, `Account General` is the display name of the section that contains fields on general account information.

Syntax

```
public String getLabel()
```

Return Value

Type: String

ReportTypeMetadata Class

Contains report type metadata, which gives you information about the fields that are available in each section of the report type, plus filter information for those fields.

Namespace

Reports

IN THIS SECTION:

ReportTypeMetadata Methods

ReportTypeMetadata Methods

The following are methods for `ReportTypeMetadata`. All are instance methods.

IN THIS SECTION:

getCategories()
Returns all fields in the report type. The fields are organized by section.

getDivisionInfo()
Returns the default division and a list of all possible divisions that can be applied to this type of report.

getScopeInfo()
Returns information about the scopes that can be applied to this type of report.

getStandardDateFilterDurationGroups()
Returns information about the standard date filter groupings that can be applied to this type of report. Standard date filter groupings include Calendar Year, Calendar Quarter, Calendar Month, Calendar Week, Fiscal Year, Fiscal Quarter, Day and a custom value based on a user-defined date range.

getStandardFilterInfos()
Returns information about standard date filters that can be applied to this type of report.

getCategories()

Returns all fields in the report type. The fields are organized by section.

Syntax

```
public LIST<Reports.ReportTypeColumnCategory> getCategories()
```

Return Value

Type: List<Reports.ReportTypeColumnCategory>

getDivisionInfo()

Returns the default division and a list of all possible divisions that can be applied to this type of report.

Signature

```
public Reports.ReportDivisionInfo getDivisionInfo()
```

Return Value

Type: Reports.ReportDivisionInfo

getScopeInfo()

Returns information about the scopes that can be applied to this type of report.

Signature

```
public Reports.ReportScopeInfo getScopeInfo()
```

Return Value

Type: Reports.ReportScopeInfo

getStandardDateFilterDurationGroups()

Returns information about the standard date filter groupings that can be applied to this type of report. Standard date filter groupings include Calendar Year, Calendar Quarter, Calendar Month, Calendar Week, Fiscal Year, Fiscal Quarter, Day and a custom value based on a user-defined date range.

Signature

```
public List<Reports.StandardDateFilterDurationGroup>
getStandardDateFilterDurationGroups()
```

Return Value

Type: List<Reports.StandardDateFilterDurationGroup>

getStandardFilterInfos()

Returns information about standard date filters that can be applied to this type of report.

Signature

```
public Map<String,Reports.StandardFilterInfo> getStandardFilterInfos()
```

Return Value

Type: Map<String,Reports.StandardFilterInfo>

SortColumn Class

Contains information about the sort column used in the report.

Namespace

Reports

IN THIS SECTION:

SortColumn Methods

SortColumn Methods

The following are methods for `SortColumn`.

IN THIS SECTION:

getSortColumn()
Returns the column used to sort the records in the report.

getSortOrder()
Returns the the sort order— ascending or descending—for the sort column.

setSortColumn(sortColumn)
Sets the column used to sort the records in the report.

setSortOrder(SortOrder)
Sets the sort order— ascending or descending—for the sort column.

`getSortColumn()`

Returns the column used to sort the records in the report.

Signature

```
public String getSortColumn()
```

Return Value

Type: String

`getSortOrder()`

Returns the the sort order— ascending or descending—for the sort column.

Signature

```
public Reports.ColumnSortOrder getSortOrder()
```

Return Value

Type: Reports.ColumnSortOrder

`setSortColumn(sortColumn)`

Sets the column used to sort the records in the report.

Signature

```
public void setSortColumn(String sortColumn)
```

Parameters

`sortColumn`
Type: String

Return Value

Type: void

`setSortOrder(SortOrder)`

Sets the sort order— ascending or descending—for the sort column.

Signature

```
public void setSortOrder(Reports.ColumnSortOrder sortOrder)
```

Parameters

`sortOrder`
Type: Reports.ColumnSortOrder

Return Value

Type: void

StandardDateFilter Class

Contains information about standard date filter available in the report—for example, the API name, start date, and end date of the standard date filter duration as well as the API name of the date field on which the filter is placed.

Namespace

Reports

IN THIS SECTION:

StandardDateFilter Methods

StandardDateFilter Methods

The following are methods for `StandardDateFilter`.

IN THIS SECTION:

getColumn()
Returns the API name of the standard date filter column.

getDurationValue()
Returns duration information about a standard date filter, such as start date, end date, and display name and API name of the date filter.

getEndDate()
Returns the end date of the standard date filter.

getStartDate()
Returns the start date for the standard date filter.

setColumn(standardDateFilterColumnName)
Sets the API name of the standard date filter column.

setDurationValue(durationName)
Sets the API name of the standard date filter.

setEndDate(endDate)
Sets the end date for the standard date filter.

setStartDate(startDate)
Sets the start date for the standard date filter.

`getColumn()`

Returns the API name of the standard date filter column.

Signature

```
public String getColumn()
```

Return Value

Type: String

`getDurationValue()`

Returns duration information about a standard date filter, such as start date, end date, and display name and API name of the date filter.

Signature

```
public String getDurationValue()
```

Return Value

Type: String

getEndDate()

Returns the end date of the standard date filter.

Signature

```
public String getEndDate()
```

Return Value

Type: String

getStartDate()

Returns the start date for the standard date filter.

Signature

```
public String getStartDate()
```

Return Value

Type: String

setColumn(standardDateFilterColumnName)

Sets the API name of the standard date filter column.

Signature

```
public void setColumn(String standardDateFilterColumnName)
```

Parameters

standardDateFilterColumnName
Type: String

Return Value

Type: void

`setDurationValue(durationName)`

Sets the API name of the standard date filter.

Signature

```
public void setDurationValue(String durationName)
```

Parameters

durationName
Type: String

Return Value

Type: void

`setEndDate(endDate)`

Sets the end date for the standard date filter.

Signature

```
public void setEndDate(String endDate)
```

Parameters

endDate
Type: String

Return Value

Type: void

`setStartDate(startDate)`

Sets the start date for the standard date filter.

Signature

```
public void setStartDate(String startDate)
```

Parameters

startDate
Type: String

Return Value

Type: void

StandardDateFilterDuration Class

Contains information about each standard date filter—also referred to as a relative date filter. It contains the API name and display label of the standard date filter duration as well as the start and end dates.

Namespace

Reports

IN THIS SECTION:

StandardDateFilterDuration Methods

StandardDateFilterDuration Methods

The following are methods for `StandardDateFilterDuration`.

IN THIS SECTION:

getEndDate()
Returns the end date of the date filter.

getLabel()
Returns the display name of the date filter. Possible values are relative date filters—like `Current FY` and `Current FQ`—and custom date filters.

getStartDate()
Returns the start date of the date filter.

getValue()
Returns the API name of the date filter. Possible values are relative date filters—like `THIS_FISCAL_YEAR` and `NEXT_FISCAL_QUARTER`—and custom date filters.

`getEndDate()`

Returns the end date of the date filter.

Signature

```
public String getEndDate()
```

Return Value

Type: String

`getLabel()`

Returns the display name of the date filter. Possible values are relative date filters—like `Current FY` and `Current FQ`—and custom date filters.

Signature

```
public String getLabel()
```

Return Value

Type: String

`getStartDate()`

Returns the start date of the date filter.

Signature

```
public String getStartDate()
```

Return Value

Type: String

`getValue()`

Returns the API name of the date filter. Possible values are relative date filters—like `THIS_FISCAL_YEAR` and `NEXT_FISCAL_QUARTER`—and custom date filters.

Signature

```
public String getValue()
```

Return Value

Type: String

StandardDateFilterDurationGroup Class

Contains information about the standard date filter groupings, such as the grouping display label and all standard date filters that fall under the grouping. Groupings include `Calendar Year`, `Calendar Quarter`, `Calendar Month`, `Calendar Week`, `Fiscal Year`, `Fiscal Quarter`, `Day`, and custom values based on user-defined date ranges.

Namespace

Reports

IN THIS SECTION:

StandardDateFilterDurationGroup Methods

StandardDateFilterDurationGroup Methods

The following are methods for `StandardDateFilterDurationGroup`.

IN THIS SECTION:

getLabel()
Returns the display label for the standard date filter grouping.

getStandardDateFilterDurations()
Returns the standard date filter groupings.

getLabel()

Returns the display label for the standard date filter grouping.

Signature

```
public String getLabel()
```

Return Value

Type: String

getStandardDateFilterDurations()

Returns the standard date filter groupings.

Signature

```
public List<Reports.StandardDateFilterDuration> getStandardDateFilterDurations()
```

Return Value

Type: List<Reports.StandardDateFilterDuration>

For example, a standard filter date grouping might look like this:

```
Reports.StandardDateFilterDuration[endDate=2015-12-31, label=Current FY,
startDate=2015-01-01, value=THIS_FISCAL_YEAR],
Reports.StandardDateFilterDuration[endDate=2014-12-31, label=Previous FY,
startDate=2014-01-01, value=LAST_FISCAL_YEAR],
Reports.StandardDateFilterDuration[endDate=2014-12-31, label=Previous 2 FY,
startDate=2013-01-01, value=LAST_N_FISCAL_YEARS:2]
```

StandardFilter Class

Contains information about the standard filter defined in the report, such as the filter field API name and filter value.

Namespace

Reports

Usage

Use to get or set standard filters on a report. Standard filters vary by report type. For example, standard filters for reports on the Opportunity object are Show, Opportunity Status, and Probability.

IN THIS SECTION:

StandardFilter Methods

StandardFilter Methods

The following are methods for `StandardFilter`.

IN THIS SECTION:

getName()
Return the API name of the standard filter.

getValue()
Returns the standard filter value.

setName(name)
Sets the API name of the standard filter.

setValue(value)
Sets the standard filter value.

`getName()`

Return the API name of the standard filter.

Signature

```
public String getName()
```

Return Value

Type: String

`getValue()`

Returns the standard filter value.

Signature

```
public String getValue()
```

Return Value

Type: String

`setName(name)`

Sets the API name of the standard filter.

Signature

```
public void setName(String name)
```

Parameters

name
Type: String

Return Value

Type: void

`setValue(value)`

Sets the standard filter value.

Signature

```
public void setValue(String value)
```

Parameters

value
Type: String

Return Value

Type: void

StandardFilterInfo Class

Is an abstract base class for an object that provides standard filter information.

Namespace

Reports

IN THIS SECTION:

StandardFilterInfo Methods

StandardFilterInfo Methods

The following are methods for `StandardFilterInfo`.

IN THIS SECTION:

getLabel()
Returns the display label of the standard filter.

getType()
Returns the type of standard filter.

`getLabel()`

Returns the display label of the standard filter.

Signature

```
public String getLabel()
```

Return Value

Type: String

`getType()`

Returns the type of standard filter.

Signature

```
public Reports.StandardFilterType getType()
```

Return Value

Type: Reports.StandardFilterType

StandardFilterInfoPicklist Class

Contains information about the standard filter picklist, such as the display name and type of the filter field, the default picklist value, and a list of all possible picklist values.

Namespace

Reports

IN THIS SECTION:

StandardFilterInfoPicklist Methods

StandardFilterInfoPicklist Methods

The following are methods for `StandardFilterInfoPicklist`.

IN THIS SECTION:

getDefaultValue()
Returns the default value for the standard filter picklist.

getFilterValues()
Returns a list of standard filter picklist values.

getLabel()
Returns the display name of the standard filter picklist.

getType()
Returns the type of the standard filter picklist.

getDefaultValue()

Returns the default value for the standard filter picklist.

Signature

```
public String getDefaultValue()
```

Return Value

Type: String

getFilterValues()

Returns a list of standard filter picklist values.

Signature

```
public List<Reports.FilterValue> getFilterValues()
```

Return Value

Type: List<Reports.FilterValue>

getLabel()

Returns the display name of the standard filter picklist.

Signature

```
public String getLabel()
```

Return Value

Type: String

getType()

Returns the type of the standard filter picklist.

Signature

```
public Reports.StandardFilterType getType()
```

Return Value

Type: Reports.StandardFilterType

StandardFilterType Enum

The `StandardFilterType` enum describes the type of standard filters in a report. The `getType()` method returns a `Reports.StandardFilterType` enum value.

Namespace

Reports

Enum Values

The following are the values of the `Reports.StandardFilterType` enum.

Value	Description
`PICKLIST`	Values for the standard filter type.
`STRING`	String values.

SummaryValue Class

Contains summary data for a cell of the report.

Namespace

Reports

SummaryValue Methods

The following are methods for `SummaryValue`. All are instance methods.

IN THIS SECTION:

getLabel()
Returns the formatted summary data for a specified cell.

getValue()
Returns the numeric value of the summary data for a specified cell.

`getLabel()`

Returns the formatted summary data for a specified cell.

Syntax

```
public String getLabel()
```

Return Value

Type: String

`getValue()`

Returns the numeric value of the summary data for a specified cell.

Syntax

```
public Object getValue()
```

Return Value

Type: Object

ThresholdInformation Class

Contains a list of evaluated conditions for a report notification.

Namespace

Reports

IN THIS SECTION:

ThresholdInformation Constructors

ThresholdInformation Methods

ThresholdInformation Constructors

The following are constructors for `ThresholdInformation`.

IN THIS SECTION:

ThresholdInformation(evaluatedConditions)
Creates a new instance of the `Reports.EvaluatedCondition` class.

`ThresholdInformation(evaluatedConditions)`

Creates a new instance of the `Reports.EvaluatedCondition` class.

Signature

```
public ThresholdInformation(List<Reports.EvaluatedCondition> evaluatedConditions)
```

Parameters

evaluatedConditions
Type: List<Reports.EvaluatedCondition>

A list of `Reports.EvaluatedCondition` objects.

ThresholdInformation Methods

The following are methods for `ThresholdInformation`.

IN THIS SECTION:

getEvaluatedConditions()
Returns a list of evaluated conditions for a report notification.

`getEvaluatedConditions()`

Returns a list of evaluated conditions for a report notification.

Signature

```
public List<Reports.EvaluatedCondition> getEvaluatedConditions()
```

Return Value

Type: List<Reports.EvaluatedCondition>

TopRows Class

Contains methods and constructors for working with information about a row limit filter.

Namespace

Reports

IN THIS SECTION:

TopRows Constructors

TopRows Methods

TopRows Constructors

The following are constructors for `TopRows`.

IN THIS SECTION:

TopRows(rowLimit, direction)
Creates an instance of the `Reports.TopRows` class using the specified parameters.

TopRows()

Creates an instance of the `Reports.TopRows` class. You can then set values by using the class's `set` methods.

`TopRows(rowLimit, direction)`

Creates an instance of the `Reports.TopRows` class using the specified parameters.

Signature

```
public TopRows(Integer rowLimit, Reports.ColumnSortOrder direction)
```

Parameters

`rowLimit`

Type: Integer

The number of rows returned in the report.

`direction`

Type: Reports.ColumnSortOrder

The sort order of the report rows.

`TopRows()`

Creates an instance of the `Reports.TopRows` class. You can then set values by using the class's `set` methods.

Signature

```
public TopRows()
```

TopRows Methods

The following are methods for `TopRows`.

IN THIS SECTION:

getDirection()

Returns the sort order of the report rows.

getRowLimit()

Returns the maximum number of rows shown in the report.

setDirection(value)

Sets the sort order of the report's rows.

setDirection(direction)

Sets the sort order of the report's rows.

setRowLimit(rowLimit)

Sets the maximum number of rows included in the report.

toString()

Returns a string.

getDirection()

Returns the sort order of the report rows.

Signature

```
public Reports.ColumnSortOrder getDirection()
```

Return Value

Type: Reports.ColumnSortOrder

getRowLimit()

Returns the maximum number of rows shown in the report.

Signature

```
public Integer getRowLimit()
```

Return Value

Type: Integer

setDirection(value)

Sets the sort order of the report's rows.

Signature

```
public void setDirection(String value)
```

Parameters

value
Type: String

For possible values, see Reports.ColumnSortOrder.

Return Value

Type: void

setDirection(direction)

Sets the sort order of the report's rows.

Signature

```
public void setDirection(Reports.ColumnSortOrder direction)
```

Parameters

direction
Type: Reports.ColumnSortOrder

Return Value

Type: void

setRowLimit(rowLimit)

Sets the maximum number of rows included in the report.

Signature

```
public void setRowLimit(Integer rowLimit)
```

Parameters

rowLimit
Type: Integer

Return Value

Type: void

toString()

Returns a string.

Signature

```
public String toString()
```

Return Value

Type: String

Reports Exceptions

The `Reports` namespace contains exception classes.

All exception classes support built-in methods for returning the error message and exception type. See Exception Class and Built-In Exceptions on page 3499.

The `Reports` namespace contains these exceptions:

Exception	Description	Methods
`Reports.FeatureNotSupportedException`	Invalid report format	
`Reports.InstanceAccessException`	Unable to access report instance	

Exception	Description	Methods
`Reports.InvalidFilterException`	Filter validation error	`List<String> getFilterErrors()` returns a list of filter errors
`Reports.InvalidReportMetadataException`	Missing metadata for filters	`List<String> getReportMetadataErrors()` returns a list of metadata errors
`Reports.InvalidSnapshotDateException`	Invalid historical report format	`List<String> getSnapshotDateErrors()` returns a list of snapshot date errors
`Reports.MetadataException`	No selected report columns	
`Reports.ReportRunException`	Error running report	
`Reports.UnsupportedOperationException`	Missing permissions for running reports	

RevSalesTrxn Namespace

The `RevSalesTrxn` namespace provides classes and methods to create a sales transaction, such as a quote or an order, with integrated pricing and configuration.

The `RevSalesTrxn` namespace includes these classes.

- ConfigurationOptionsInput Class
- GraphRequest Class
- PlaceSalesTransactionException Class
- PlaceSalesTransactionExecutor Class
- PlaceSalesTransactionResponse Class
- RecordResource Class
- RecordWithReferenceRequest Class

RichMessaging Namespace

Provides objects and methods for handling content in enhanced Messaging channels.

The following are the classes in the `RichMessaging` namespace.

IN THIS SECTION:

AbstractTiming Class
Parent class for other RichMessaging timing classes.

AddressableContact Class
Represents an addressable contact.

AuthRequestHandler Interface
Use this interface to handle authorization request responses.

AuthRequestResponse Class

This class contains authorization request response data.

AuthRequestResult Class

This class contains the result from handling the authorization request response.

AuthRequestResultStatus Enum

This enum describes the authentication result status.

DeferredTiming Class

Represents timing for a transaction that occurs in the future.

MessageDefinitionInputParameter Class

Represents a messaging component parameter value. This class is used to provide parameter payloads that can be translated to structured content payloads in rich content messages.

PaymentItemStatus Enum

Represents the status of a payment item in payment requests sent in enhanced Messaging channels.

PaymentLineItem Class

Represents a payment line item in payment requests sent in enhanced Messaging channels.

PaymentMethod Class

Represents a payment method.

PostalAddress Class

Represents the postal address.

ProcessPaymentHandler Interface

Interface used to process payment requests.

ProcessPaymentRequest Class

Represents a request to process a payment.

ProcessPaymentResult Class

Represents the result of a payment processing operation.

ProcessPaymentResultStatus Enum

Represents the status of a payment processing result.

RecurringTiming Class

Represents a payment that occurs on a regular basis.

ShippingMethod Class

Represents a shipping method listed in payment requests sent in enhanced Messaging channels.

TimeSlotOption Class

Represents a complex time slot option type. This class is used to provide time option payloads that can be translated to structured content payloads in rich content messages.

TimingIntervalUnit Enum

Represents an enumerated type that describes the timing interval.

TimingType Enum

Represents an enumerated type that describes the type of timing.

AbstractTiming Class

Parent class for other RichMessaging timing classes.

Namespace

RichMessaging

SEE ALSO:

- DeferredTiming Class
- RecurringTiming Class

AddressableContact Class

Represents an addressable contact.

Namespace

RichMessaging

IN THIS SECTION:

- AddressableContact Constructors
- AddressableContact Properties

AddressableContact Constructors

The following are constructors for `AddressableContact`.

IN THIS SECTION:

AddressableContact(givenName, phoneticGivenName, familyName, phoneticFamilyName, emailAddress, phoneNumber, postalAddress)
Creates a new instance of the `RichMessaging.AddressableContact` class.

`AddressableContact(givenName, phoneticGivenName, familyName, phoneticFamilyName, emailAddress, phoneNumber, postalAddress)`

Creates a new instance of the `RichMessaging.AddressableContact` class.

Signature

```
public AddressableContact(String givenName, String phoneticGivenName, String familyName,
String phoneticFamilyName, String emailAddress, String phoneNumber,
RichMessaging.PostalAddress postalAddress)
```

Parameters

givenName
Type: String

The contact's first name.

phoneticGivenName
Type: String

The phonetic spelling of the contact's first name.

familyName
Type: String

The contact's surname.

phoneticFamilyName
Type: String

The phonetic spelling of the contact's surname.

emailAddress
Type: String

The contact's email address.

phoneNumber
Type: String

The contact's phone number.

postalAddress
Type: RichMessaging.PostalAddress

The contact's postal address.

AddressableContact Properties

The following are properties for `AddressableContact`.

IN THIS SECTION:

emailAddress
The contact's email address.

familyName
The contact's surname.

givenName
The contact's first name.

phoneNumber
The contact's phone number.

phoneticFamilyName
The phonetic spelling of the contact's surname.

phoneticGivenName
The phonetic spelling of the contact's first name.

postalAddress
The contact's postal address.

emailAddress

The contact's email address.

Signature

```
public String emailAddress {get; set;}
```

Property Value

Type: String

familyName

The contact's surname.

Signature

```
public String familyName {get; set;}
```

Property Value

Type: String

givenName

The contact's first name.

Signature

```
public String givenName {get; set;}
```

Property Value

Type: String

phoneNumber

The contact's phone number.

Signature

```
public String phoneNumber {get; set;}
```

Property Value

Type: String

phoneticFamilyName

The phonetic spelling of the contact's surname.

Signature

```
public String phoneticFamilyName {get; set;}
```

Property Value

Type: String

phoneticGivenName

The phonetic spelling of the contact's first name.

Signature

```
public String phoneticGivenName {get; set;}
```

Property Value

Type: String

postalAddress

The contact's postal address.

Signature

```
public RichMessaging.PostalAddress postalAddress {get; set;}
```

Property Value

Type: RichMessaging.PostalAddress

AuthRequestHandler Interface

Use this interface to handle authorization request responses.

Namespace

RichMessaging on page 3039

Usage

When using this interface, the following limits are overridden. See Execution Governors and Limits in the Apex Developer Guide.

Table 1: Overridden Limits

Limit Name	Overridden Value
Total number of SOQL queries issued	2
Total number of records retrieved by a single SOSL query	2
Total number of DML statements issued	1
Total number of records processed as a result of DML statements	1
Total number of callouts (HTTP requests or web services calls) in a transaction	2

IN THIS SECTION:

AuthRequestHandler Methods

AuthRequestHandler Example Implementation

AuthRequestHandler Methods

The following are methods for `AuthRequestHandler`.

IN THIS SECTION:

handleAuthRequest(var1)
Handles authorization request response.

`handleAuthRequest(var1)`

Handles authorization request response.

Signature

```
public RichMessaging.AuthRequestResult
handleAuthRequest(RichMessaging.AuthRequestResponse var1)
```

Parameters

`var1`

Type: RichMessaging.AuthRequestResponse on page 3047

The authorization response.

Return Value

Type: RichMessaging.AuthRequestResult on page 3049

AuthRequestHandler Example Implementation

This is an example implementation of the `RichMessaging.AuthRequestHandler` interface.

```
global class SampleAuthRequestHandler implements RichMessaging.AuthRequestHandler {

    global RichMessaging.AuthRequestResult
handleAuthRequest(RichMessaging.AuthRequestResponse authReqResponse) {

        // Get contact email from messaging session
        String sessionId = authReqResponse.getContextRecordId();
        String contactEmail = [select MessagingSession.EndUserContact.Email from
MessagingSession where id = :sessionId].EndUserContact.Email;

        RichMessaging.AuthRequestResultStatus authRequestStatus =
RichMessaging.AuthRequestResultStatus.DECLINED;
        DateTime dt = DateTime.now();

        // Get user info if there's a valid contact email
        if (!String.isBlank(contactEmail)) {
            String userInfoUrl = 'https://api.MY_AUTH_DOMAIN.com/v1/';

            HttpRequest req = new HttpRequest();

            req.setEndpoint(userInfoUrl);
            req.setHeader('Content-Type','application/json');
            req.setMethod('GET');
            req.setHeader('Authorization', 'Bearer '+authReqResponse.getAccessToken());
            Http http = new Http();
            HTTPResponse res = http.send(req);

            String responseBody = res.getBody();

            UserWrapper userInfo = (UserWrapper)System.JSON.deserialize(responseBody,
UserWrapper.class);

            if (userInfo.email == contactEmail) {
                authRequestStatus = RichMessaging.AuthRequestResultStatus.AUTHENTICATED;
                dt = dt.addHours(6);
            }
         }

        return new RichMessaging.AuthRequestResult(
            null,
            authRequestStatus,
            dt);
    }

    public class UserWrapper{
        public String href;
        public String display_name;
        public String type;
        public String country;
        public String product;
        public String email;
```

```
    }
}
```

AuthRequestResponse Class

This class contains authorization request response data.

Namespace

RichMessaging

IN THIS SECTION:

AuthRequestResponse Constructors

AuthRequestResponse Methods

AuthRequestResponse Constructors

The following are constructors for `AuthRequestResponse`.

IN THIS SECTION:

AuthRequestResponse(accessToken, contextRecordId, authProviderName)

Creates a new instance of the `RichMessaging.AuthRequestResponse` class.

`AuthRequestResponse(accessToken, contextRecordId, authProviderName)`

Creates a new instance of the `RichMessaging.AuthRequestResponse` class.

Signature

```
public AuthRequestResponse(String accessToken, String contextRecordId, String
authProviderName)
```

Parameters

`accessToken`

Type: String

The authorization access token.

`contextRecordId`

Type: String

The context record ID.

`authProviderName`

Type: String

The provider name.

AuthRequestResponse Methods

The following are methods for `AuthRequestResponse`.

IN THIS SECTION:

getAccessToken()
Gets the authorization access token.

getAuthProviderName()
Get the authorization provider name.

getContextRecordId()
Gets the context record ID.

`getAccessToken()`

Gets the authorization access token.

Signature

```
public String getAccessToken()
```

Return Value

Type: String

The access token.

`getAuthProviderName()`

Get the authorization provider name.

Signature

```
public String getAuthProviderName()
```

Return Value

Type: String

The authorization provider name.

`getContextRecordId()`

Gets the context record ID.

Signature

```
public String getContextRecordId()
```

Return Value

Type: String

The context record ID.

AuthRequestResult Class

This class contains the result from handling the authorization request response.

Namespace

RichMessaging

IN THIS SECTION:

AuthRequestResult Constructors

AuthRequestResult Properties

AuthRequestResult Constructors

The following are constructors for `AuthRequestResult`.

IN THIS SECTION:

AuthRequestResult(redirectPageReference, resultStatus, expirationDateTime)

Creates a new instance of the `RichMessaging.AuthRequestResult` class.

`AuthRequestResult(redirectPageReference, resultStatus, expirationDateTime)`

Creates a new instance of the `RichMessaging.AuthRequestResult` class.

Signature

```
public AuthRequestResult(System.PageReference redirectPageReference,
RichMessaging.AuthRequestResultStatus resultStatus, Datetime expirationDateTime)
```

Parameters

`redirectPageReference`

Type: System.PageReference on page 3696

The reference to the redirect page.

`resultStatus`

Type: RichMessaging.AuthRequestResultStatus on page 3051

The result status value.

`expirationDateTime`

Type: Datetime

The expiration time.

AuthRequestResult Properties

The following are properties for `AuthRequestResult`.

IN THIS SECTION:

expirationDateTime
The expiration date and time.

redirectPageReference
The reference to the redirect page.

resultStatus
The result status value.

`expirationDateTime`

The expiration date and time.

Signature

```
public Datetime expirationDateTime {get; set;}
```

Property Value

Type: Datetime

`redirectPageReference`

The reference to the redirect page.

Signature

```
public System.PageReference redirectPageReference {get; set;}
```

Property Value

Type: System.PageReference on page 3696

`resultStatus`

The result status value.

Signature

```
public RichMessaging.AuthRequestResultStatus resultStatus {get; set;}
```

Property Value

Type: RichMessaging.AuthRequestResultStatus on page 3051

AuthRequestResultStatus Enum

This enum describes the authentication result status.

Enum Values

The following are the values of the `RichMessaging.AuthRequestResultStatus` enum.

Value	Description
`AUTHENTICATED`	Authenticated result.
`DECLINED`	Declined result.

DeferredTiming Class

Represents timing for a transaction that occurs in the future.

Namespace

RichMessaging

IN THIS SECTION:

DeferredTiming Constructors

DeferredTiming Properties

DeferredTiming Constructors

The following are constructors for `DeferredTiming`.

IN THIS SECTION:

DeferredTiming(deferredDate)

Creates a new instance of the `RichMessaging.DeferredTiming` class.

DeferredTiming()

Creates a new instance of the `RichMessaging.DeferredTiming` class.

`DeferredTiming(deferredDate)`

Creates a new instance of the `RichMessaging.DeferredTiming` class.

Signature

```
public DeferredTiming(Datetime deferredDate)
```

Parameters

deferredDate
Type: Datetime

The deferred date.

DeferredTiming()

Creates a new instance of the `RichMessaging.DeferredTiming` class.

Signature

```
public DeferredTiming()
```

DeferredTiming Properties

The following are properties for `DeferredTiming`.

IN THIS SECTION:

deferredDate
The deferred date. Invocable variable.

deferredDateValue
The deferred date. Enabled for Lightning components.

timingType
Always returns "DeferredTiming".

deferredDate

The deferred date. Invocable variable.

Signature

```
public Datetime deferredDate {get; set;}
```

Property Value

Type: Datetime

deferredDateValue

The deferred date. Enabled for Lightning components.

Signature

```
public Datetime deferredDateValue {get; set;}
```

Property Value

Type: Datetime

`timingType`

Always returns "DeferredTiming".

Signature

```
public String timingType {get; set;}
```

Property Value

Type: String

MessageDefinitionInputParameter Class

Represents a messaging component parameter value. This class is used to provide parameter payloads that can be translated to structured content payloads in rich content messages.

Namespace

RichMessaging

IN THIS SECTION:

MessageDefinitionInputParameter Properties

MessageDefinitionInputParameter Properties

The following are properties for `MessageDefinitionInputParameter`.

IN THIS SECTION:

booleanValue
A boolean input parameter.

booleanValues
A list of boolean parameters.

dateTimeValue
A datetime input parameter.

dateTimeValues
A list of datetime input parameters.

dateValue
A date input parameter.

dateValues
A list of date input parameters.

name
A name input parameter.

numberValue
A number input parameter.

numberValues
A list of number input parameters.

recordIdValue
A record ID input parameter.

recordIdValues
A list of record ID input parameters.

textValue
A text input parameter.

textValues
A list of text input parameters.

booleanValue

A boolean input parameter.

Signature

```
public Boolean booleanValue {get; set;}
```

Property Value

Type: Boolean

booleanValues

A list of boolean parameters.

Signature

```
public List<Boolean> booleanValues {get; set;}
```

Property Value

Type: List on page 3603<Boolean>

dateTimeValue

A datetime input parameter.

Signature

```
public Datetime dateTimeValue {get; set;}
```

Property Value

Type: Datetime

dateTimeValues

A list of datetime input parameters.

Signature

```
public List<Datetime> dateTimeValues {get; set;}
```

Property Value

Type: List on page 3603<Datetime>

dateValue

A date input parameter.

Signature

```
public Date dateValue {get; set;}
```

Property Value

Type: Date

dateValues

A list of date input parameters.

Signature

```
public List<Date> dateValues {get; set;}
```

Property Value

Type: List on page 3603<Date>

name

A name input parameter.

Signature

```
public String name {get; set;}
```

Property Value

Type: String

numberValue

A number input parameter.

Signature

```
public Double numberValue {get; set;}
```

Property Value

Type: Double

`numberValues`

A list of number input parameters.

Signature

```
public List<Double> numberValues {get; set;}
```

Property Value

Type: List on page 3603<Double>

`recordIdValue`

A record ID input parameter.

Signature

```
public String recordIdValue {get; set;}
```

Property Value

Type: String

`recordIdValues`

A list of record ID input parameters.

Signature

```
public List<String> recordIdValues {get; set;}
```

Property Value

Type: List on page 3603<String>

`textValue`

A text input parameter.

Signature

```
public String textValue {get; set;}
```

Property Value

Type: String

`textValues`

A list of text input parameters.

Signature

```
public List<String> textValues {get; set;}
```

Property Value

Type: List on page 3603<String>

PaymentItemStatus Enum

Represents the status of a payment item in payment requests sent in enhanced Messaging channels.

Enum Values

The following are the values of the `RichMessaging.PaymentItemStatus` enum.

Value	Description
`FinalCost`	Indicates that the payment item's cost is final and has been determined.
`PendingCost`	Indicates that the payment item's cost is pending and has not been determined yet.

PaymentLineItem Class

Represents a payment line item in payment requests sent in enhanced Messaging channels.

Namespace

RichMessaging

Example

```
public with sharing class MessagingPaymentLineItems {

    @InvocableMethod
    public static List<List<RichMessaging.PaymentLineItem>> getLineItems() {
        Double amount = 0.25;
        List<List<RichMessaging.PaymentLineItem>> result = new
List<List<RichMessaging.PaymentLineItem>>();
        RichMessaging.PaymentLineItem pizza = new RichMessaging.PaymentLineItem('pizza',
amount);
```

```
        RichMessaging.PaymentLineItem pasta = new RichMessaging.PaymentLineItem('pasta',
amount);
        pizza.statusValue = RichMessaging.PaymentItemStatus.FinalCost;
        pasta.statusValue = RichMessaging.PaymentItemStatus.FinalCost;

        List<RichMessaging.PaymentLineItem> options = new
List<RichMessaging.PaymentLineItem>{
            pizza, pasta
        };
        result.add(options);
        return result;
    }

}
```

IN THIS SECTION:

PaymentLineItem Constructors

PaymentLineItem Properties

PaymentLineItem Methods

PaymentLineItem Constructors

The following are constructors for `PaymentLineItem`.

IN THIS SECTION:

PaymentLineItem(label, amount, timing)
Creates a new instance of the `RichMessaging.PaymentLineItem` class.

PaymentLineItem(label, amount)
Creates a new instance of the `RichMessaging.PaymentLineItem` class.

PaymentLineItem()
Creates a new instance of the `RichMessaging.PaymentLineItem` class.

`PaymentLineItem(label, amount, timing)`

Creates a new instance of the `RichMessaging.PaymentLineItem` class.

Signature

```
public PaymentLineItem(String label, Double amount, RichMessaging.AbstractTiming timing)
```

Parameters

`label`
Type: String

The label of the payment line item.

amount
Type: Double

The amount of the payment line item.

timing
Type: RichMessaging.AbstractTiming

The timing of the payment line item.

PaymentLineItem(label, amount)

Creates a new instance of the `RichMessaging.PaymentLineItem` class.

Signature

```
public PaymentLineItem(String label, Double amount)
```

Parameters

label
Type: String

The label of the payment line item.

amount
Type: Double

The amount of the payment line item.

PaymentLineItem()

Creates a new instance of the `RichMessaging.PaymentLineItem` class.

Signature

```
public PaymentLineItem()
```

PaymentLineItem Properties

The following are properties for `PaymentLineItem`.

IN THIS SECTION:

amount
The amount of the payment line item.

amountValue
The amount value of the payment line item.

automaticReloadPaymentThresholdAmount
The automatic reload payment threshold amount of the payment line item.

automaticReloadPaymentThresholdAmountValue
The automatic reload payment threshold amount value of the payment line item.

label
The label of the payment line item.

labelValue
The label value of the payment line item.

lineItemType
The line item type of the payment line item. Read-only variable.

status
The status of the payment line item.

statusValue
The status value of the payment line item.

timing
The timing of the payment line item.

timingValue
The timing value of the payment line item.

amount

The amount of the payment line item.

Signature

```
public Double amount {get; set;}
```

Property Value

Type: Double

amountValue

The amount value of the payment line item.

Signature

```
public Double amountValue {get; set;}
```

Property Value

Type: Double

automaticReloadPaymentThresholdAmount

The automatic reload payment threshold amount of the payment line item.

Signature

```
public Double automaticReloadPaymentThresholdAmount {get; set;}
```

Property Value

Type: Double

automaticReloadPaymentThresholdAmountValue

The automatic reload payment threshold amount value of the payment line item.

Signature

```
public Double automaticReloadPaymentThresholdAmountValue {get; set;}
```

Property Value

Type: Double

label

The label of the payment line item.

Signature

```
public String label {get; set;}
```

Property Value

Type: String

labelValue

The label value of the payment line item.

Signature

```
public String labelValue {get; set;}
```

Property Value

Type: String

lineItemType

The line item type of the payment line item. Read-only variable.

Signature

```
public String lineItemType {get; set;}
```

Property Value

Type: String

status

The status of the payment line item.

Signature

```
public String status {get; set;}
```

Property Value

Type: String

statusValue

The status value of the payment line item.

Signature

```
public RichMessaging.PaymentItemStatus statusValue {get; set;}
```

Property Value

Type: RichMessaging.PaymentItemStatus

timing

The timing of the payment line item.

Signature

```
public RichMessaging.AbstractTiming timing {get; set;}
```

Property Value

Type: RichMessaging.AbstractTiming

timingValue

The timing value of the payment line item.

Signature

```
public RichMessaging.AbstractTiming timingValue {get; set;}
```

Property Value

Type: RichMessaging.AbstractTiming

PaymentLineItem Methods

The following are methods for `PaymentLineItem`.

PaymentMethod Class

Represents a payment method.

Namespace

RichMessaging

IN THIS SECTION:

PaymentMethod Constructors

PaymentMethod Properties

PaymentMethod Constructors

The following are constructors for `PaymentMethod`.

IN THIS SECTION:

PaymentMethod(network, paymentType, displayName)
Creates a new instance of the `RichMessaging.PaymentMethod` class.

`PaymentMethod(network, paymentType, displayName)`

Creates a new instance of the `RichMessaging.PaymentMethod` class.

Signature

```
public PaymentMethod(String network, String paymentType, String displayName)
```

Parameters

`network`
Type: String

The network associated with the payment method.

`paymentType`
Type: String

The payment type of the payment method.

`displayName`
Type: String

The display name of the payment method.

PaymentMethod Properties

The following are properties for `PaymentMethod`.

IN THIS SECTION:

displayName
The display name of the payment method.

network
The network associated with the payment method.

paymentType
The payment type of the payment method.

displayName

The display name of the payment method.

Signature

```
public String displayName {get; set;}
```

Property Value

Type: String

network

The network associated with the payment method.

Signature

```
public String network {get; set;}
```

Property Value

Type: String

paymentType

The payment type of the payment method.

Signature

```
public String paymentType {get; set;}
```

Property Value

Type: String

PostalAddress Class

Represents the postal address.

Namespace

RichMessaging

IN THIS SECTION:

PostalAddress Constructors

PostalAddress Properties

PostalAddress Constructors

The following are constructors for `PostalAddress`.

IN THIS SECTION:

PostalAddress(addressLines, subLocality, locality, postalCode, subAdministrativeArea, administrativeArea, country, countryCode)
Creates a new instance of the `RichMessaging.PostalAddress` class.

`PostalAddress(addressLines, subLocality, locality, postalCode, subAdministrativeArea, administrativeArea, country, countryCode)`

Creates a new instance of the `RichMessaging.PostalAddress` class.

Signature

```
public PostalAddress(List<String> addressLines, String subLocality, String locality,
String postalCode, String subAdministrativeArea, String administrativeArea, String
country, String countryCode)
```

Parameters

`addressLines`
Type: List<String>

The street address.

`subLocality`
Type: String

The sub-locality of the address.

`locality`
Type: String

The locality of the address.

`postalCode`
Type: String

The postal code.

`subAdministrativeArea`
Type: String

The sub-administrative area.

administrativeArea
Type: String

The administrative area.

country
Type: String

The country.

countryCode
Type: String

The country code.

PostalAddress Properties

The following are properties for `PostalAddress`.

IN THIS SECTION:

addressLines
The street address.

administrativeArea
The administrative area.

country
The country.

countryCode
The country code.

locality
The locality of the address.

postalCode
The postal code.

subAdministrativeArea
The sub-administrative area.

subLocality
The sub-locality of the address.

addressLines

The street address.

Signature

```
public List<String> addressLines {get; set;}
```

Property Value

Type: List<String>

`administrativeArea`

The administrative area.

Signature

```
public String administrativeArea {get; set;}
```

Property Value

Type: String

`country`

The country.

Signature

```
public String country {get; set;}
```

Property Value

Type: String

`countryCode`

The country code.

Signature

```
public String countryCode {get; set;}
```

Property Value

Type: String

`locality`

The locality of the address.

Signature

```
public String locality {get; set;}
```

Property Value

Type: String

`postalCode`

The postal code.

Signature

```
public String postalCode {get; set;}
```

Property Value

Type: String

`subAdministrativeArea`

The sub-administrative area.

Signature

```
public String subAdministrativeArea {get; set;}
```

Property Value

Type: String

`subLocality`

The sub-locality of the address.

Signature

```
public String subLocality {get; set;}
```

Property Value

Type: String

ProcessPaymentHandler Interface

Interface used to process payment requests.

Namespace

RichMessaging

IN THIS SECTION:

ProcessPaymentHandler Methods

ProcessPaymentHandler Example Implementation

ProcessPaymentHandler Methods

The following are methods for `ProcessPaymentHandler`.

IN THIS SECTION:

processPaymentRequest(var1)
Processes a payment request.

processPaymentRequest(var1)

Processes a payment request.

Signature

```
public RichMessaging.ProcessPaymentResult
processPaymentRequest(RichMessaging.ProcessPaymentRequest var1)
```

Parameters

var1

Type: RichMessaging.ProcessPaymentRequest

The payment request.

Return Value

Type: RichMessaging.ProcessPaymentResult

ProcessPaymentHandler Example Implementation

This is an example implementation of the `RichMessaging.ProcessPaymentHandler` interface.

```
global class MyProcessPaymentHandler implements Richmessaging.ProcessPaymentHandler {

   global RichMessaging.ProcessPaymentResult
processPaymentRequest(RichMessaging.ProcessPaymentRequest paymentRequest) {

       // TODO: Reach out to your payment processor here and return success or failure
based on the result of that request

       return new
RichMessaging.ProcessPaymentResult(RichMessaging.ProcessPaymentResultStatus.SUCCESS);
   }
}
```

ProcessPaymentRequest Class

Represents a request to process a payment.

Namespace

RichMessaging

IN THIS SECTION:

ProcessPaymentRequest Constructors

ProcessPaymentRequest Properties

ProcessPaymentRequest Constructors

The following are constructors for `ProcessPaymentRequest`.

IN THIS SECTION:

ProcessPaymentRequest(transactionIdentifier, paymentData, billingContact, shippingContact, paymentMethod, shippingMethod, contextRecordId)
Creates a new instance of the `RichMessaging.ProcessPaymentRequest` class.

`ProcessPaymentRequest(transactionIdentifier, paymentData, billingContact, shippingContact, paymentMethod, shippingMethod, contextRecordId)`

Creates a new instance of the `RichMessaging.ProcessPaymentRequest` class.

Signature

```
public ProcessPaymentRequest(String transactionIdentifier, String paymentData,
RichMessaging.AddressableContact billingContact, RichMessaging.AddressableContact
shippingContact, RichMessaging.PaymentMethod paymentMethod, RichMessaging.ShippingMethod
shippingMethod, String contextRecordId)
```

Parameters

`transactionIdentifier`
Type: String

The transaction identifier associated with the payment request.

`paymentData`
Type: String

The encrypted payment data for the payment request.

`billingContact`
Type: RichMessaging.AddressableContact

The billing contact information for the payment request.

`shippingContact`
Type: RichMessaging.AddressableContact

The shipping contact information for the payment request.

`paymentMethod`
Type: RichMessaging.PaymentMethod

The payment method for the payment request.

`shippingMethod`
Type: RichMessaging.ShippingMethod

The shipping method for the payment request.

contextRecordId
Type: String

The context record ID associated with the payment request.

ProcessPaymentRequest Properties

The following are properties for `ProcessPaymentRequest`.

IN THIS SECTION:

billingContact
The billing contact information for the payment request.

contextRecordId
The context record ID associated with the payment request.

paymentData
The encrypted payment data for the payment request.

paymentMethod
The payment method for the payment request.

shippingContact
The shipping contact information for the payment request.

shippingMethod
The shipping method for the payment request.

transactionIdentifier
The transaction identifier associated with the payment request.

billingContact

The billing contact information for the payment request.

Signature

```
public RichMessaging.AddressableContact billingContact {get; set;}
```

Property Value

Type: RichMessaging.AddressableContact

contextRecordId

The context record ID associated with the payment request.

Signature

```
public String contextRecordId {get; set;}
```

Property Value

Type: String

paymentData

The encrypted payment data for the payment request.

Signature

```
public String paymentData {get; set;}
```

Property Value

Type: String

paymentMethod

The payment method for the payment request.

Signature

```
public RichMessaging.PaymentMethod paymentMethod {get; set;}
```

Property Value

Type: RichMessaging.PaymentMethod

shippingContact

The shipping contact information for the payment request.

Signature

```
public RichMessaging.AddressableContact shippingContact {get; set;}
```

Property Value

Type: RichMessaging.AddressableContact

shippingMethod

The shipping method for the payment request.

Signature

```
public RichMessaging.ShippingMethod shippingMethod {get; set;}
```

Property Value

Type: RichMessaging.ShippingMethod

`transactionIdentifier`

The transaction identifier associated with the payment request.

Signature

```
public String transactionIdentifier {get; set;}
```

Property Value

Type: String

ProcessPaymentResult Class

Represents the result of a payment processing operation.

Namespace

RichMessaging

IN THIS SECTION:

ProcessPaymentResult Constructors

ProcessPaymentResult Properties

ProcessPaymentResult Constructors

The following are constructors for `ProcessPaymentResult`.

IN THIS SECTION:

ProcessPaymentResult(resultStatus, errorMessage)

Creates a new instance of the `RichMessaging.ProcessPaymentResult` class.

ProcessPaymentResult(resultStatus)

Creates a new instance of the `RichMessaging.ProcessPaymentResult` class.

`ProcessPaymentResult(resultStatus, errorMessage)`

Creates a new instance of the `RichMessaging.ProcessPaymentResult` class.

Signature

```
public ProcessPaymentResult(RichMessaging.ProcessPaymentResultStatus resultStatus,
String errorMessage)
```

Parameters

`resultStatus`

Type: RichMessaging.ProcessPaymentResultStatus

The status of the payment processing result.

errorMessage
Type: String

The error message associated with the payment processing result, if any.

ProcessPaymentResult(resultStatus)

Creates a new instance of the `RichMessaging.ProcessPaymentResult` class.

Signature

```
public ProcessPaymentResult(RichMessaging.ProcessPaymentResultStatus resultStatus)
```

Parameters

resultStatus
Type: RichMessaging.ProcessPaymentResultStatus

The status of the payment processing result.

ProcessPaymentResult Properties

The following are properties for `ProcessPaymentResult`.

IN THIS SECTION:

errorMessage
The error message associated with the payment processing result, if any.

resultStatus
The status of the payment processing result.

errorMessage

The error message associated with the payment processing result, if any.

Signature

```
public String errorMessage {get; set;}
```

Property Value

Type: String

resultStatus

The status of the payment processing result.

Signature

```
public RichMessaging.ProcessPaymentResultStatus resultStatus {get; set;}
```

Property Value

Type: RichMessaging.ProcessPaymentResultStatus

ProcessPaymentResultStatus Enum

Represents the status of a payment processing result.

Enum Values

The following are the values of the `RichMessaging.ProcessPaymentResultStatus` enum.

Value	Description
`PROCESSOR_ERROR`	Indicates an error occurred during payment processing at the processor level.
`SUCCESS`	Indicates a successful payment processing result.

RecurringTiming Class

Represents a payment that occurs on a regular basis.

Namespace

RichMessaging

IN THIS SECTION:

- RecurringTiming Constructors
- RecurringTiming Properties

RecurringTiming Constructors

The following are constructors for `RecurringTiming`.

IN THIS SECTION:

- RecurringTiming(startDate, endDate, intervalCount, intervalUnit)
  Creates a new instance of the `RichMessaging.RecurringTiming` class.
- RecurringTiming()
  Creates a new instance of the `RichMessaging.RecurringTiming` class.

`RecurringTiming(startDate, endDate, intervalCount, intervalUnit)`

Creates a new instance of the `RichMessaging.RecurringTiming` class.

Signature

```
public RecurringTiming(Date startDate, Date endDate, Integer intervalCount,
RichMessaging.TimingIntervalUnit intervalUnit)
```

Parameters

startDate
Type: Date

The start date. Invocable variable.

endDate
Type: Date

The end date. Invocable variable.

intervalCount
Type: Integer

The number of interval units that make up the total payment interval. Invocable variable.

intervalUnit
Type: RichMessaging.TimingIntervalUnit

The amount of time—in calendar units, such as day, month, or year—that represents a fraction of the total payment interval. Invocable variable.

`RecurringTiming()`

Creates a new instance of the `RichMessaging.RecurringTiming` class.

Signature

```
public RecurringTiming()
```

RecurringTiming Properties

The following are properties for `RecurringTiming`.

IN THIS SECTION:

endDate
The end date. Invocable variable.

endDateValue
The end date. Enabled for Lightning components.

intervalCount
The number of interval units that make up the total payment interval. Invocable variable.

intervalCountValue
The number of interval units that make up the total payment interval. Enabled for Lightning components.

intervalUnit
The amount of time—in calendar units, such as day, month, or year—that represents a fraction of the total payment interval. Invocable variable.

intervalUnitValue
The amount of time—in calendar units, such as day, month, or year—that represents a fraction of the total payment interval. Enabled for Lightning components.

startDate
The start date. Invocable variable.

startDateValue
The start date. Enabled for Lightning components.

timingType
Always returns "RecurringTiming".

endDate

The end date. Invocable variable.

Signature

```
public Date endDate {get; set;}
```

Property Value

Type: Date

endDateValue

The end date. Enabled for Lightning components.

Signature

```
public Date endDateValue {get; set;}
```

Property Value

Type: Date

intervalCount

The number of interval units that make up the total payment interval. Invocable variable.

Signature

```
public Integer intervalCount {get; set;}
```

Property Value

Type: Integer

intervalCountValue

The number of interval units that make up the total payment interval. Enabled for Lightning components.

Signature

```
public Integer intervalCountValue {get; set;}
```

Property Value

Type: Integer

`intervalUnit`

The amount of time—in calendar units, such as day, month, or year—that represents a fraction of the total payment interval. Invocable variable.

Signature

```
public String intervalUnit {get; set;}
```

Property Value

Type: String

`intervalUnitValue`

The amount of time—in calendar units, such as day, month, or year—that represents a fraction of the total payment interval. Enabled for Lightning components.

Signature

```
public RichMessaging.TimingIntervalUnit intervalUnitValue {get; set;}
```

Property Value

Type: RichMessaging.TimingIntervalUnit

`startDate`

The start date. Invocable variable.

Signature

```
public Date startDate {get; set;}
```

Property Value

Type: Date

`startDateValue`

The start date. Enabled for Lightning components.

Signature

```
public Date startDateValue {get; set;}
```

Property Value

Type: Date

`timingType`

Always returns "RecurringTiming".

Signature

```
public String timingType {get; set;}
```

Property Value

Type: String

ShippingMethod Class

Represents a shipping method listed in payment requests sent in enhanced Messaging channels.

Namespace

RichMessaging

Example

```
public with sharing class MessagingShippingMethods {

    @InvocableMethod
    public static List<List<RichMessaging.ShippingMethod>> getShippingMethods(){
        Double amount = 0.25;
        List<List<RichMessaging.ShippingMethod>> result = new
List<List<RichMessaging.ShippingMethod>>();

       List<RichMessaging.ShippingMethod> options = new List<RichMessaging.ShippingMethod>{

            new RichMessaging.ShippingMethod('doordash', amount, '1 hour delivery to your
 door', 'ddash'),
            new RichMessaging.ShippingMethod('UPS', amount, '2 days delivery', 'UPS')
        };
        result.add(options);
        return result;
    }

}
```

IN THIS SECTION:

ShippingMethod Constructors

ShippingMethod Properties

ShippingMethod Constructors

The following are constructors for `ShippingMethod`.

IN THIS SECTION:

ShippingMethod(label, amount, detail, identifier)

Creates a new instance of the `RichMessaging.ShippingMethod` class.

ShippingMethod()

Creates a new instance of the `RichMessaging.ShippingMethod` class.

`ShippingMethod(label, amount, detail, identifier)`

Creates a new instance of the `RichMessaging.ShippingMethod` class.

Signature

```
public ShippingMethod(String label, Double amount, String detail, String identifier)
```

Parameters

`label`

Type: String

The label of the shipping method.

`amount`

Type: Double

The amount of the shipping method.

`detail`

Type: String

Details about the shipping method.

`identifier`

Type: String

The identifier of the shipping method.

`ShippingMethod()`

Creates a new instance of the `RichMessaging.ShippingMethod` class.

Signature

```
public ShippingMethod()
```

ShippingMethod Properties

The following are properties for `ShippingMethod`.

IN THIS SECTION:

amount
The amount of the shipping method.

amountValue
The amount value of the shipping method.

detail
Details about the shipping method.

detailValue
The detail value of the shipping method.

identifier
The identifier of the shipping method.

identifierValue
The identifier value of the shipping method.

label
The label of the shipping method.

labelValue
The label value of the shipping method.

shippingMethodType
The shipping method type. Read only.

`amount`

The amount of the shipping method.

Signature

```
public Double amount {get; set;}
```

Property Value

Type: Double

`amountValue`

The amount value of the shipping method.

Signature

```
public Double amountValue {get; set;}
```

Property Value

Type: Double

`detail`

Details about the shipping method.

Signature

```
public String detail {get; set;}
```

Property Value

Type: String

`detailValue`

The detail value of the shipping method.

Signature

```
public String detailValue {get; set;}
```

Property Value

Type: String

`identifier`

The identifier of the shipping method.

Signature

```
public String identifier {get; set;}
```

Property Value

Type: String

`identifierValue`

The identifier value of the shipping method.

Signature

```
public String identifierValue {get; set;}
```

Property Value

Type: String

label

The label of the shipping method.

Signature

```
public String label {get; set;}
```

Property Value

Type: String

labelValue

The label value of the shipping method.

Signature

```
public String labelValue {get; set;}
```

Property Value

Type: String

shippingMethodType

The shipping method type. Read only.

Signature

```
public String shippingMethodType {get; set;}
```

Property Value

Type: String

TimeSlotOption Class

Represents a complex time slot option type. This class is used to provide time option payloads that can be translated to structured content payloads in rich content messages.

Namespace

RichMessaging

IN THIS SECTION:

TimeSlotOption Constructors

TimeSlotOption Properties

TimeSlotOption Constructors

The following are constructors for `TimeSlotOption`.

IN THIS SECTION:

TimeSlotOption(startTime, endTime)
Creates a `TimeSlotOption` object with a start and end time.

TimeSlotOption(startTime, duration)
Creates a `TimeSlotOption` object with a start time and a duration.

TimeSlotOption()
Creates a `TimeSlotOption` object.

`TimeSlotOption(startTime, endTime)`

Creates a `TimeSlotOption` object with a start and end time.

Signature

```
public TimeSlotOption(Datetime startTime, Datetime endTime)
```

Parameters

`startTime`
Type: Datetime

Start time.

`endTime`
Type: Datetime

End time.

`TimeSlotOption(startTime, duration)`

Creates a `TimeSlotOption` object with a start time and a duration.

Signature

```
public TimeSlotOption(Datetime startTime, Integer duration)
```

Parameters

`startTime`
Type: Datetime

Start time.

`duration`
Type: Integer

Duration in seconds.

`TimeSlotOption()`

Creates a `TimeSlotOption` object.

Signature

```
public TimeSlotOption()
```

TimeSlotOption Properties

The following are properties for `TimeSlotOption`.

IN THIS SECTION:

duration
The duration in seconds.

durationValue
The duration in seconds. Enabled for Lightning components.

endTimeValue
The end time. Enabled for Lightning components.

startTime
The start time.

startTimeValue
The start time. Enabled for Lightning components.

`duration`

The duration in seconds.

Signature

```
public Integer duration {get; set;}
```

Property Value

Type: Integer

`durationValue`

The duration in seconds. Enabled for Lightning components.

Signature

```
public Integer durationValue {get; set;}
```

Property Value

Type: Integer

endTimeValue

The end time. Enabled for Lightning components.

Signature

```
public Datetime endTimeValue {get; set;}
```

Property Value

Type: Datetime

startTime

The start time.

Signature

```
public Datetime startTime {get; set;}
```

Property Value

Type: Datetime

startTimeValue

The start time. Enabled for Lightning components.

Signature

```
public Datetime startTimeValue {get; set;}
```

Property Value

Type: Datetime

TimingIntervalUnit Enum

Represents an enumerated type that describes the timing interval.

Enum Values

The following are the values of the `RichMessaging.TimingIntervalUnit` enum.

Value	Description
`Day`	Day interval.
`Hour`	Hour interval.
`Minute`	Minute interval.

Value	Description
`Month`	Month interval.
`Year`	Year interval.

TimingType Enum

Represents an enumerated type that describes the type of timing.

Enum Values

The following are the values of the `RichMessaging.TimingType` enum.

Value	Description
`DeferredTiming`	Indicates that the timing is deferred. See DeferredTiming Class.
`RecurringTiming`	Indicates that the timing recurs. See RecurringTiming Class.

Salesforce_Backup Namespace

The `Salesforce_Backup` namespace provides classes and methods for Salesforce Backup.

This class is in the `Salesforce_Backup` namespace.

IN THIS SECTION:

SalesforceBackupIntegrationUtil Class
This class and its methods are for internal use only.

SalesforceBackupIntegrationUtil Class

This class and its methods are for internal use only.

Namespace

Salesforce_Backup

Schema Namespace

The `Schema` namespace provides classes and methods for schema metadata information.

The following are the classes in the `Schema` namespace.

IN THIS SECTION:

ChildRelationship Class
Contains methods for accessing the child relationship as well as the child sObject for a parent sObject.

DataCategory Class
Represents the categories within a category group.

DataCategoryGroupSobjectTypePair Class
Specifies a category group and an associated object.

DescribeColorResult Class
Contains color metadata information for a tab.

DescribeDataCategoryGroupResult Class
Contains the list of the category groups associated with KnowledgeArticleVersion and Question.

DescribeDataCategoryGroupStructureResult Class
Contains the category groups and categories associated with KnowledgeArticleVersion and Question.

DescribeFieldResult Class
Contains methods for describing sObject fields.

DescribeIconResult Class
Contains icon metadata information for a tab.

DescribeSObjectResult Class
Contains methods for describing SObjects. None of the methods take an argument.

DescribeTabResult Class
Contains tab metadata information for a tab in a standard or custom app available in the Salesforce user interface.

DescribeTabSetResult Class
Contains metadata information about a Salesforce Classic standard or custom app available in the Salesforce user interface.

DisplayType Enum
A `Schema.DisplayType` enum value is returned by the field describe result's `getType` method.

FieldDescribeOptions Enum
A `Schema.FieldDescribeOptions` enum value is a parameter in the `SObjectType.getDescribe` method.

FieldSet Class
Contains methods for discovering and retrieving the details of field sets created on sObjects.

FieldSetMember Class
Contains methods for accessing the metadata for field set member fields.

PicklistEntry Class
Represents a picklist entry.

RecordTypeInfo Class
Contains methods for accessing record type information for an sObject with associated record types.

SOAPType Enum
A `Schema.SOAPType` enum value is returned by the field describe result `getSoapType` method.

SObjectDescribeOptions Enum
A `Schema.SObjectDescribeOptions` enum value is a parameter in the `SObjectType.getDescribe` method.

SObjectField Class

A `Schema.sObjectField` object is returned from the field describe result using the `getController` and `getSObjectField` methods.

SObjectType Class

A `Schema.sObjectType` object is returned from the field describe result using the `getReferenceTo` method, or from the sObject describe result using the `getSObjectType` method.

ChildRelationship Class

Contains methods for accessing the child relationship as well as the child sObject for a parent sObject.

Namespace

Schema

Example

A ChildRelationship object is returned from the sObject describe result using the `getChildRelationship` method. For example:

```
Schema.DescribeSObjectResult R = Account.SObjectType.getDescribe();
List<Schema.ChildRelationship> C = R.getChildRelationships();
```

ChildRelationship Methods

The following are methods for `ChildRelationship`. All are instance methods.

IN THIS SECTION:

getChildSObject()

Returns the token of the child sObject on which there is a foreign key back to the parent sObject.

getField()

Returns the token of the field that has a foreign key back to the parent sObject.

getRelationshipName()

Returns the name of the relationship.

isCascadeDelete()

Returns `true` if the child object is deleted when the parent object is deleted, `false` otherwise.

isDeprecatedAndHidden()

Reserved for future use.

isRestrictedDelete()

Returns `true` if the parent object can't be deleted because it is referenced by a child object, `false` otherwise.

`getChildSObject()`

Returns the token of the child sObject on which there is a foreign key back to the parent sObject.

Signature

```
public Schema.SObjectType getChildSObject()
```

Return Value

Type: Schema.SObjectType

`getField()`

Returns the token of the field that has a foreign key back to the parent sObject.

Signature

```
public Schema.SObjectField getField()
```

Return Value

Type: Schema.SObjectField

`getRelationshipName()`

Returns the name of the relationship.

Signature

```
public String getRelationshipName()
```

Return Value

Type: String

`isCascadeDelete()`

Returns `true` if the child object is deleted when the parent object is deleted, `false` otherwise.

Signature

```
public Boolean isCascadeDelete()
```

Return Value

Type: Boolean

`isDeprecatedAndHidden()`

Reserved for future use.

Signature

```
public Boolean isDeprecatedAndHidden()
```

Return Value

Type: Boolean

`isRestrictedDelete()`

Returns `true` if the parent object can't be deleted because it is referenced by a child object, `false` otherwise.

Signature

```
public Boolean isRestrictedDelete()
```

Return Value

Type: Boolean

DataCategory Class

Represents the categories within a category group.

Namespace

Schema

Usage

The `Schema.DataCategory` object is returned by the `getTopCategories` method.

DataCategory Methods

The following are methods for `DataCategory`. All are instance methods.

IN THIS SECTION:

getChildCategories()
Returns a recursive object that contains the visible sub categories in the data category.

getLabel()
Returns the label for the data category used in the Salesforce user interface.

getName()
Returns the unique name used by the API to access to the data category.

`getChildCategories()`

Returns a recursive object that contains the visible sub categories in the data category.

Signature

```
public Schema.DataCategory getChildCategories()
```

Return Value

Type: List<Schema.DataCategory>

`getLabel()`

Returns the label for the data category used in the Salesforce user interface.

Signature

```
public String getLabel()
```

Return Value

Type: String

`getName()`

Returns the unique name used by the API to access to the data category.

Signature

```
public String getName()
```

Return Value

Type: String

DataCategoryGroupSobjectTypePair Class

Specifies a category group and an associated object.

Namespace

Schema

Usage

Schema.DataCategoryGroupSobjectTypePair is used by the `describeDataCategory GroupStructures` method to return the categories available to this object.

IN THIS SECTION:

DataCategoryGroupSobjectTypePair Constructors

DataCategoryGroupSobjectTypePair Methods

DataCategoryGroupSobjectTypePair Constructors

The following are constructors for `DataCategoryGroupSobjectTypePair`.

IN THIS SECTION:

DataCategoryGroupSobjectTypePair()
Creates a new instance of the `Schema.DataCategoryGroupSobjectTypePair` class.

DataCategoryGroupSobjectTypePair()

Creates a new instance of the `Schema.DataCategoryGroupSobjectTypePair` class.

Signature

```
public DataCategoryGroupSobjectTypePair()
```

DataCategoryGroupSobjectTypePair Methods

The following are methods for `DataCategoryGroupSobjectTypePair`. All are instance methods.

IN THIS SECTION:

getDataCategoryGroupName()
Returns the unique name used by the API to access the data category group.

getSobject()
Returns the object name associated with the data category group.

setDataCategoryGroupName(name)
Specifies the unique name used by the API to access the data category group.

setSobject(sObjectName)
Sets the sObject associated with the data category group.

getDataCategoryGroupName()

Returns the unique name used by the API to access the data category group.

Signature

```
public String getDataCategoryGroupName()
```

Return Value

Type: String

getSobject()

Returns the object name associated with the data category group.

Signature

```
public String getSobject()
```

Return Value

Type: String

`setDataCategoryGroupName(name)`

Specifies the unique name used by the API to access the data category group.

Signature

```
public String setDataCategoryGroupName(String name)
```

Parameters

name
Type: String

Return Value

Type: Void

`setSobject(sObjectName)`

Sets the sObject associated with the data category group.

Signature

```
public Void setSobject(String sObjectName)
```

Parameters

`sObjectName`
Type: String

The *`sObjectName`* is the object name associated with the data category group. Valid values are:

- `KnowledgeArticleVersion`—for article types.
- `Question`—for questions from Answers.

Return Value

Type: Void

DescribeColorResult Class

Contains color metadata information for a tab.

Namespace

Schema

Usage

The `getColors` method of the `Schema.DescribeTabResult` class returns a list of `Schema.DescribeColorResult` objects that describe colors used in a tab.

The methods in the `Schema.DescribeColorResult` class can be called using their property counterparts. For each method starting with `get`, you can omit the `get` prefix and the ending parentheses `()` to call the property counterpart. For example, `colorResultObj.color` is equivalent to `colorResultObj.getColor()`.

Example

This sample shows how to get the color information in the Sales app for the first tab's first color.

```
// Get tab set describes for each app
List<Schema.DescribeTabSetResult> tabSetDesc = Schema.DescribeTabs();

// Iterate through each tab set describe for each app and display the info
for(Schema.DescribeTabSetResult tsr : tabSetDesc) {
    // Display tab info for the Sales app
    if (tsr.getLabel() == 'Sales') {
        // Get color information for the first tab
        List<Schema.DescribeColorResult> colorDesc = tsr.getTabs()[0].getColors();
        // Display the icon color, theme, and context of the first color returned
        System.debug('Color: ' + colorDesc[0].getColor());
        System.debug('Theme: ' + colorDesc[0].getTheme());
        System.debug('Context: ' + colorDesc[0].getContext());
    }
}

// Example debug statement output
// DEBUG|Color: 1797C0
// DEBUG|Theme: theme4
// DEBUG|Context: primary
```

DescribeColorResult Methods

The following are methods for `DescribeColorResult`. All are instance methods.

IN THIS SECTION:

getColor()

Returns the Web RGB color code, such as `00FF00`.

getContext()

Returns the color context. The context determines whether the color is the main color for the tab or not.

getTheme()

Returns the color theme.

`getColor()`

Returns the Web RGB color code, such as `00FF00`.

Signature

```
public String getColor()
```

Return Value

Type: String

`getContext()`

Returns the color context. The context determines whether the color is the main color for the tab or not.

Signature

```
public String getContext()
```

Return Value

Type: String

`getTheme()`

Returns the color theme.

Signature

```
public String getTheme()
```

Return Value

Type: String

Possible theme values include `theme3`, `theme4`, and `custom`.

- `theme3` is the Salesforce theme introduced during Spring '10.
- `theme4` is the Salesforce theme introduced in Winter '14 for the mobile touchscreen version of Salesforce.
- `custom` is the theme name associated with a custom icon.

DescribeDataCategoryGroupResult Class

Contains the list of the category groups associated with KnowledgeArticleVersion and Question.

Namespace

Schema

Usage

The `describeDataCategoryGroups` method returns a `Schema.DescribeDataCategoryGroupResult` object containing the list of the category groups associated with the specified object.

For additional information and code examples using `describeDataCategoryGroups`, see Accessing All Data Categories Associated with an sObject.

Example

The following is an example of how to instantiate a data category group describe result object:

```
List <String> objType = new List<String>();
objType.add('KnowledgeArticleVersion');
objType.add('Question');

List<Schema.DescribeDataCategoryGroupResult> describeCategoryResult =
   Schema.describeDataCategoryGroups(objType);
```

DescribeDataCategoryGroupResult Methods

The following are methods for `DescribeDataCategoryGroupResult`. All are instance methods.

IN THIS SECTION:

getCategoryCount()
Returns the number of visible data categories in the data category group.

getDescription()
Returns the description of the data category group.

getLabel()
Returns the label for the data category group used in the Salesforce user interface.

getName()
Returns the unique name used by the API to access to the data category group.

getSobject()
Returns the object name associated with the data category group.

`getCategoryCount()`

Returns the number of visible data categories in the data category group.

Signature

```
public Integer getCategoryCount()
```

Return Value

Type: Integer

`getDescription()`

Returns the description of the data category group.

Signature

```
public String getDescription()
```

Return Value

Type: String

`getLabel()`

Returns the label for the data category group used in the Salesforce user interface.

Signature

```
public String getLabel()
```

Return Value

Type: String

`getName()`

Returns the unique name used by the API to access to the data category group.

Signature

```
public String getName()
```

Return Value

Type: String

`getSobject()`

Returns the object name associated with the data category group.

Signature

```
public String getSobject()
```

Return Value

Type: String

DescribeDataCategoryGroupStructureResult Class

Contains the category groups and categories associated with KnowledgeArticleVersion and Question.

Namespace

Schema

Usage

The `describeDataCategoryGroupStructures` method returns a list of `Schema.DescribeDataCategoryGroupStructureResult` objects containing the category groups and categories associated with the specified object.

For additional information and code examples, see Accessing All Data Categories Associated with an sObject.

Example

The following is an example of how to instantiate a data category group structure describe result object:

```
List <DataCategoryGroupSobjectTypePair> pairs =
      new List<DataCategoryGroupSobjectTypePair>();

DataCategoryGroupSobjectTypePair pair1 =
      new DataCategoryGroupSobjectTypePair();
pair1.setSobject('KnowledgeArticleVersion');
pair1.setDataCategoryGroupName('Regions');

DataCategoryGroupSobjectTypePair pair2 =
      new DataCategoryGroupSobjectTypePair();
pair2.setSobject('Questions');
pair2.setDataCategoryGroupName('Regions');

pairs.add(pair1);
pairs.add(pair2);

List<Schema.DescribeDataCategoryGroupStructureResult>results =
      Schema.describeDataCategoryGroupStructures(pairs, true);
```

DescribeDataCategoryGroupStructureResult Methods

The following are methods for `DescribeDataCategoryGroupStructureResult`. All are instance methods.

IN THIS SECTION:

getDescription()

Returns the description of the data category group.

getLabel()

Returns the label for the data category group used in the Salesforce user interface.

getName()

Returns the unique name used by the API to access to the data category group.

getSobject()

Returns the name of object associated with the data category group.

getTopCategories()

Returns a `Schema.DataCategory` object, that contains the top categories visible depending on the user's data category group visibility settings.

`getDescription()`

Returns the description of the data category group.

Signature

```
public String getDescription()
```

Return Value

Type: String

`getLabel()`

Returns the label for the data category group used in the Salesforce user interface.

Signature

```
public String getLabel()
```

Return Value

Type: String

`getName()`

Returns the unique name used by the API to access to the data category group.

Signature

```
public String getName()
```

Return Value

Type: String

`getSobject()`

Returns the name of object associated with the data category group.

Signature

```
public String getSobject()
```

Return Value

Type: String

`getTopCategories()`

Returns a `Schema.DataCategory` object, that contains the top categories visible depending on the user's data category group visibility settings.

Signature

```
public List<Schema.DataCategory> getTopCategories()
```

Return Value

Type: List<Schema.DataCategory>

Usage

For more information on data category group visibility, see "Data Category Visibility" in the Salesforce online help.

DescribeFieldResult Class

Contains methods for describing sObject fields.

Namespace

Schema

Usage

Instances of field describe results on the same `DescribeFieldResult` aren't guaranteed to be equal because the state and behavior of a describe object is determined by various factors including the API version used. To compare describe results, call the `getSObjectField()` method on the field describe results and use the equality operator (==) to compare the `SObjectField` values.

Example

The following is an example of how to instantiate a field describe result object:

```
Schema.DescribeFieldResult dfr = Account.Description.getDescribe();
```

DescribeFieldResult Methods

The following are methods for `DescribeFieldResult`. All are instance methods.

IN THIS SECTION:

getByteLength()

For variable-length fields (including binary fields), returns the maximum size of the field, in bytes.

getCalculatedFormula()

Returns the formula specified for this field.

getController()

Returns the token of the controlling field.

getDefaultValue()

Returns the default value for this field.

getDefaultValueFormula()

Returns the default formula value that is specified for this formula field.

getDigits()

Returns the maximum number of digits specified for the field. This method is only valid with Integer fields.

getInlineHelpText()

Returns the content of the field-level help.

getLabel()

Returns the text label that is displayed next to the field in the Salesforce user interface. This label can be localized.

getLength()

Returns the maximum size of the field for the DescribeFieldResult object in Unicode characters (not bytes).

getLocalName()

Returns the name of the field, similar to the `getName` method. However, if the field is part of the current namespace, the namespace portion of the name is omitted.

getName()

Returns the field name used in Apex.

getPicklistValues()

Returns a list of active PicklistEntry objects. A runtime error is returned if the field isn't a picklist. Only active picklist values are returned.

getPrecision()

For fields of type Double, returns the maximum number of digits that can be stored, including all numbers to the left and to the right of the decimal point (but excluding the decimal point character).

getReferenceTargetField()

Returns the name of the custom field on the parent standard or custom object whose values are matched against the values of the child external object's indirect lookup relationship field. The match is done to determine which records are related to each other.

getReferenceTo()

Returns a list of Schema.sObjectType objects for the parent objects of this field. If the `isNamePointing` method returns `true`, there is more than one entry in the list, otherwise there is only one.

getRelationshipName()

Returns the name of the child-to-parent relationship.

getRelationshipOrder()

Returns 0 if the field is the primary relationship field or 1 if the field is the secondary relationship field.

getScale()

For fields of type Double, returns the number of digits to the right of the decimal point.

getSOAPType()

Returns one of the SoapType enum values, depending on the type of field.

getSObjectField()

Returns the token for this field.

getSObjectType()
Returns the Salesforce object type from which this field originates.

getType()
Returns one of the DisplayType enum values, depending on the type of field.

isAccessible()
Returns `true` if the current user can see this field, `false` otherwise.

isAiPredictionField() (Beta)
Returns `true` if the current field is enabled to display Einstein prediction data, `false` otherwise.

isAutoNumber()
Returns `true` if the field is an Auto Number field, `false` otherwise.

isCalculated()
Returns `true` if the field is a custom formula field, `false` otherwise. Note that custom formula fields are always read-only.

isCascadeDelete()
Returns `true` if the child object is deleted when the parent object is deleted, `false` otherwise.

isCaseSensitive()
Returns `true` if the field is case sensitive, `false` otherwise.

isCreateable()
Returns `true` if the field can be created by the current user, `false` otherwise.

isCustom()
Returns `true` if the field is a custom field, `false` if it is a standard field, such as `Name`.

isDefaultedOnCreate()
Returns `true` if the field receives a default value when created, `false` otherwise.

isDependentPicklist()
Returns `true` if the picklist is a dependent picklist, `false` otherwise.

isDeprecatedAndHidden()
Reserved for future use.

isEncrypted()
Returns `true` if the field is encrypted with Shield Platform Encryption, `false` otherwise.

isExternalID()
Returns `true` if the field is used as an external ID, `false` otherwise.

isFilterable()
Returns `true` if the field can be used as part of the filter criteria of a `WHERE` statement, `false` otherwise.

isFormulaTreatNullNumberAsZero()
Returns `true` if `null` is treated as zero in a formula field, `false` otherwise.

isGroupable()
Returns `true` if the field can be included in the `GROUP BY` clause of a SOQL query, `false` otherwise. This method is only available for Apex classes and triggers saved using API version 18.0 and higher.

isHtmlFormatted()
Returns `true` if the field has been formatted for HTML and should be encoded for display in HTML, `false` otherwise. One example of a field that returns `true` for this method is a hyperlink custom formula field. Another example is a custom formula field that has an `IMAGE` text function.

isIdLookup()

Returns `true` if the field can be used to specify a record in an `upsert` method, `false` otherwise.

isNameField()

Returns `true` if the field is a name field, `false` otherwise.

isNamePointing()

Returns `true` if the field can have multiple types of objects as parents. For example, a task can have both the `Contact/Lead ID` (`WhoId`) field and the `Opportunity/Account ID` (`WhatId`) field return `true` for this method. because either of those objects can be the parent of a particular task record. This method returns `false` otherwise.

isNillable()

Returns `true` if the field is nillable, `false` otherwise. A nillable field can have empty content. A non-nillable field must have a value for the object to be created or saved.

isPermissionable()

Returns `true` if field permissions can be specified for the field, `false` otherwise.

isRestrictedDelete()

Returns `true` if the parent object can't be deleted because it is referenced by a child object, `false` otherwise.

isRestrictedPicklist()

Returns `true` if the field is a restricted picklist, `false` otherwise

isSearchPrefilterable()

Returns `true` if a foreign key can be included in prefiltering when used in a SOSL `WHERE` clause, `false` otherwise.

isSortable()

Returns `true` if a query can sort on the field, `false` otherwise

isUnique()

Returns `true` if the value for the field must be unique, `false` otherwise

isUpdateable()

Returns `true` if the field can be edited by the current user, or child records in a master-detail relationship field on a custom object can be reparented to different parent records; `false` otherwise.

isWriteRequiresMasterRead()

Returns `true` if writing to the detail object requires read sharing instead of read/write sharing of the parent.

`getByteLength()`

For variable-length fields (including binary fields), returns the maximum size of the field, in bytes.

Signature

```
public Integer getByteLength()
```

Return Value

Type: Integer

`getCalculatedFormula()`

Returns the formula specified for this field.

Signature

```
public String getCalculatedFormula()
```

Return Value

Type: String

`getController()`

Returns the token of the controlling field.

Signature

```
public Schema.sObjectField getController()
```

Return Value

Type: Schema.SObjectField

`getDefaultValue()`

Returns the default value for this field.

Signature

```
public Object getDefaultValue()
```

Return Value

Type: Object

`getDefaultValueFormula()`

Returns the default formula value that is specified for this formula field.

Signature

```
public String getDefaultValueFormula()
```

Return Value

Type: String

`getDigits()`

Returns the maximum number of digits specified for the field. This method is only valid with Integer fields.

Signature

```
public Integer getDigits()
```

Return Value

Type: Integer

`getInlineHelpText()`

Returns the content of the field-level help.

Signature

```
public String getInlineHelpText()
```

Return Value

Type: String

Usage

For more information, see "Define Field-Level Help" in the Salesforce online help.

`getLabel()`

Returns the text label that is displayed next to the field in the Salesforce user interface. This label can be localized.

Signature

```
public String getLabel()
```

Return Value

Type: String

Usage

Note: For the Type field on standard objects, `getLabel` returns a label different from the default label. It returns a label of the form *`Object`* `Type`, where *Object* is the standard object label. For example, for the Type field on Account, `getLabel` returns `Account Type` instead of the default label `Type`. If the Type label is renamed, `getLabel` returns the new label. You can check or change the labels of all standard object fields from Setup by entering *`Rename Tabs and Labels`* in the `Quick Find box`, then selecting **Rename Tabs and Labels**.

`getLength()`

Returns the maximum size of the field for the DescribeFieldResult object in Unicode characters (not bytes).

Signature

```
public Integer getLength()
```

Return Value

Type: Integer

getLocalName()

Returns the name of the field, similar to the `getName` method. However, if the field is part of the current namespace, the namespace portion of the name is omitted.

Signature

```
public String getLocalName()
```

Return Value

Type: String

getName()

Returns the field name used in Apex.

Signature

```
public String getName()
```

Return Value

Type: String

getPicklistValues()

Returns a list of active PicklistEntry objects. A runtime error is returned if the field isn't a picklist. Only active picklist values are returned.

Signature

```
public List<Schema.PicklistEntry> getPicklistValues()
```

Return Value

Type: List<Schema.PicklistEntry>

getPrecision()

For fields of type Double, returns the maximum number of digits that can be stored, including all numbers to the left and to the right of the decimal point (but excluding the decimal point character).

Signature

```
public Integer getPrecision()
```

Return Value

Type: Integer

`getReferenceTargetField()`

Returns the name of the custom field on the parent standard or custom object whose values are matched against the values of the child external object's indirect lookup relationship field. The match is done to determine which records are related to each other.

Signature

```
public String getReferenceTargetField()
```

Return Value

Type: String

Usage

For information about indirect lookup relationships, see "Indirect Lookup Relationship Fields on External Objects" in the Salesforce Help.

`getReferenceTo()`

Returns a list of Schema.sObjectType objects for the parent objects of this field. If the `isNamePointing` method returns `true`, there is more than one entry in the list, otherwise there is only one.

Signature

```
public List <Schema.sObjectType> getReferenceTo()
```

Return Value

Type: List<Schema.sObjectType>

Versioned Behavior Changes

In API version 51.0 and later, the `getReferenceTo()` method returns referenced objects that aren't accessible to the context user. If the context user has access to an object's field that references another object, irrespective of the context user's access to the cross-referenced object, the method returns references. In API version 50.0 and earlier, if the context user doesn't have access to the cross-referenced object, the method returns an empty list.

`getRelationshipName()`

Returns the name of the child-to-parent relationship.

Signature

```
public String getRelationshipName()
```

Return Value

Type: String

Usage

For more information about relationships and relationship names, see Understanding Relationship Names in the *SOQL and SOSL Reference*.

getRelationshipOrder()

Returns 0 if the field is the primary relationship field or 1 if the field is the secondary relationship field.

Signature

```
public Integer getRelationshipOrder()
```

Return Value

Type: Integer

Usage

For more information about primary and secondary relationships, see Considerations for Object Relationships. For more information about relationships and relationship names, see Understanding Relationship Names in the *SOQL and SOSL Reference*.

getScale()

For fields of type Double, returns the number of digits to the right of the decimal point.

Signature

```
public Integer getScale()
```

Return Value

Type: Integer

getSOAPType()

Returns one of the SoapType enum values, depending on the type of field.

Signature

```
public Schema.SOAPType getSOAPType()
```

Return Value

Type: Schema.SOAPType

getSObjectField()

Returns the token for this field.

Signature

```
public Schema.sObjectField getSObjectField()
```

Return Value

Type: Schema.SObjectField

getSObjectType()

Returns the Salesforce object type from which this field originates.

Signature

```
public Schema.SObjectType getSObjectType()
```

Return Value

Type: Schema.SObjectType

Example

```
Schema.DescribeFieldResult f = Account.Industry.getDescribe();
Schema.sObjectType sourceType = f.getSObjectType();
Assert.areEqual(Account.sObjectType, sourceType);
```

getType()

Returns one of the DisplayType enum values, depending on the type of field.

Signature

```
public Schema.DisplayType getType()
```

Return Value

Type: Schema.DisplayType

isAccessible()

Returns `true` if the current user can see this field, `false` otherwise.

Signature

```
public Boolean isAccessible()
```

Return Value

Type: Boolean

`isAiPredictionField()` (Beta)

Returns `true` if the current field is enabled to display Einstein prediction data, `false` otherwise.

Signature

```
public Boolean isAiPredictionField()
```

Return Value

Type: Boolean

Usage

Custom number fields can be set to display Einstein prediction values. If you are participating in the Einstein Prediction Builder Beta program, use Einstein Prediction Builder to set up the value to display. Use this method to find out if a field is enabled to display an Einstein prediction value.

`isAutoNumber()`

Returns `true` if the field is an Auto Number field, `false` otherwise.

Signature

```
public Boolean isAutoNumber()
```

Return Value

Type: Boolean

Usage

Analogous to a SQL IDENTITY type, Auto Number fields are read-only, non-createable text fields with a maximum length of 30 characters. Auto Number fields are used to provide a unique ID that is independent of the internal object ID (such as a purchase order number or invoice number). Auto Number fields are configured entirely in the Salesforce user interface.

`isCalculated()`

Returns `true` if the field is a custom formula field, `false` otherwise. Note that custom formula fields are always read-only.

Signature

```
public Boolean isCalculated()
```

Return Value

Type: Boolean

`isCascadeDelete()`

Returns `true` if the child object is deleted when the parent object is deleted, `false` otherwise.

Signature

```
public Boolean isCascadeDelete()
```

Return Value

Type: Boolean

`isCaseSensitive()`

Returns `true` if the field is case sensitive, `false` otherwise.

Signature

```
public Boolean isCaseSensitive()
```

Return Value

Type: Boolean

`isCreateable()`

Returns `true` if the field can be created by the current user, `false` otherwise.

Signature

```
public Boolean isCreateable()
```

Return Value

Type: Boolean

`isCustom()`

Returns `true` if the field is a custom field, `false` if it is a standard field, such as `Name`.

Signature

```
public Boolean isCustom()
```

Return Value

Type: Boolean

`isDefaultedOnCreate()`

Returns `true` if the field receives a default value when created, `false` otherwise.

Signature

```
public Boolean isDefaultedOnCreate()
```

Return Value

Type: Boolean

Usage

If this method returns `true`, Salesforce implicitly assigns a value for this field when the object is created, even if a value for this field is not passed in on the create call. For example, in the Opportunity object, the Probability field has this attribute because its value is derived from the Stage field. Similarly, the Owner has this attribute on most objects because its value is derived from the current user (if the Owner field is not specified).

`isDependentPicklist()`

Returns `true` if the picklist is a dependent picklist, `false` otherwise.

Signature

```
public Boolean isDependentPicklist()
```

Return Value

Type: Boolean

`isDeprecatedAndHidden()`

Reserved for future use.

Signature

```
public Boolean isDeprecatedAndHidden()
```

Return Value

Type: Boolean

`isEncrypted()`

Returns `true` if the field is encrypted with Shield Platform Encryption, `false` otherwise.

Signature

```
public Boolean isEncrypted()
```

Return Value

Type: Boolean

`isExternalID()`

Returns `true` if the field is used as an external ID, `false` otherwise.

Signature

```
public Boolean isExternalID()
```

Return Value

Type: Boolean

isFilterable()

Returns `true` if the field can be used as part of the filter criteria of a `WHERE` statement, `false` otherwise.

Signature

```
public Boolean isFilterable()
```

Return Value

Type: Boolean

isFormulaTreatNullNumberAsZero()

Returns `true` if `null` is treated as zero in a formula field, `false` otherwise.

Signature

```
public Boolean isFormulaTreatNullNumberAsZero()
```

Return Value

Type: Boolean

isGroupable()

Returns `true` if the field can be included in the `GROUP BY` clause of a SOQL query, `false` otherwise. This method is only available for Apex classes and triggers saved using API version 18.0 and higher.

Signature

```
public Boolean isGroupable()
```

Return Value

Type: Boolean

isHtmlFormatted()

Returns `true` if the field has been formatted for HTML and should be encoded for display in HTML, `false` otherwise. One example of a field that returns `true` for this method is a hyperlink custom formula field. Another example is a custom formula field that has an `IMAGE` text function.

Signature

```
public Boolean isHtmlFormatted()
```

Return Value

Type: Boolean

isIdLookup()

Returns `true` if the field can be used to specify a record in an `upsert` method, `false` otherwise.

Signature

```
public Boolean isIdLookup()
```

Return Value

Type: Boolean

isNameField()

Returns `true` if the field is a name field, `false` otherwise.

Signature

```
public Boolean isNameField()
```

Return Value

Type: Boolean

Usage

This method is used to identify the name field for standard objects (such as `AccountName` for an Account object) and custom objects. Objects can only have one name field, except where the `FirstName` and `LastName` fields are used instead (such as on the Contact object).

If a compound name is present, for example, the `Name` field on a person account, `isNameField` is set to `true` for that record.

isNamePointing()

Returns `true` if the field can have multiple types of objects as parents. For example, a task can have both the `Contact/Lead ID` (`WhoId`) field and the `Opportunity/Account ID` (`WhatId`) field return `true` for this method. because either of those objects can be the parent of a particular task record. This method returns `false` otherwise.

Signature

```
public Boolean isNamePointing()
```

Return Value

Type: Boolean

isNillable()

Returns `true` if the field is nillable, `false` otherwise. A nillable field can have empty content. A non-nillable field must have a value for the object to be created or saved.

Signature

```
public Boolean isNillable()
```

Return Value

Type: Boolean

isPermissionable()

Returns `true` if field permissions can be specified for the field, `false` otherwise.

Signature

```
public Boolean isPermissionable()
```

Return Value

Type: Boolean

isRestrictedDelete()

Returns `true` if the parent object can't be deleted because it is referenced by a child object, `false` otherwise.

Signature

```
public Boolean isRestrictedDelete()
```

Return Value

Type: Boolean

isRestrictedPicklist()

Returns `true` if the field is a restricted picklist, `false` otherwise

Signature

```
public Boolean isRestrictedPicklist()
```

Return Value

Type: Boolean

`isSearchPrefilterable()`

Returns `true` if a foreign key can be included in prefiltering when used in a SOSL `WHERE` clause, `false` otherwise.

Signature

```
public Boolean isSearchPrefilterable()
```

Return Value

Type: Boolean

Usage

Prefiltering means to filter by a specific field value before executing the full search query. Prefiltering is supported only in `WHERE` clauses with the equals (=) operator.

`isSortable()`

Returns `true` if a query can sort on the field, `false` otherwise

Signature

```
public Boolean isSortable()
```

Return Value

Type: Boolean

`isUnique()`

Returns `true` if the value for the field must be unique, `false` otherwise

Signature

```
public Boolean isUnique()
```

Return Value

Type: Boolean

`isUpdateable()`

Returns `true` if the field can be edited by the current user, or child records in a master-detail relationship field on a custom object can be reparented to different parent records; `false` otherwise.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Signature

```
public Boolean isUpdateable()
```

Return Value

Type: Boolean

`isWriteRequiresMasterRead()`

Returns `true` if writing to the detail object requires read sharing instead of read/write sharing of the parent.

Signature

```
public Boolean isWriteRequiresMasterRead()
```

Return Value

Type: Boolean

DescribeIconResult Class

Contains icon metadata information for a tab.

Namespace

Schema

Usage

The `getIcons` method of the `Schema.DescribeTabResult` class returns a list of `Schema.DescribeIconResult` objects that describe colors used in a tab.

The methods in the `Schema.DescribeIconResult` class can be called using their property counterparts. For each method starting with `get`, you can omit the `get` prefix and the ending parentheses `()` to call the property counterpart. For example, `iconResultObj.url` is equivalent to `iconResultObj.getUrl()`.

Example

This sample shows how to get the icon information in the Sales app for the first tab's first icon.

```
// Get tab set describes for each app
List<Schema.DescribeTabSetResult> tabSetDesc = Schema.describeTabs();

// Iterate through each tab set
for(Schema.DescribeTabSetResult tsr : tabSetDesc) {
    // Get tab info for the Sales app
    if (tsr.getLabel() == 'Sales') {
        // Get icon information for the first tab
        List<Schema.DescribeIconResult> iconDesc = tsr.getTabs()[0].getIcons();
        // Display the icon height and width of the first icon
        System.debug('Height: ' + iconDesc[0].getHeight());
```

```
        System.debug('Width: ' + iconDesc[0].getWidth());
    }
}

// Example debug statement output
// DEBUG|Height: 32
// DEBUG|Width: 32
```

DescribeIconResult Methods

The following are methods for `DescribeIconResult`. All are instance methods.

IN THIS SECTION:

getContentType()
Returns the tab icon's content type, such as `image/png`.

getHeight()
Returns the tab icon's height in pixels.

getTheme()
Returns the tab's icon theme.

getUrl()
Returns the tab's icon fully qualified URL.

getWidth()
Returns the tab's icon width in pixels.

`getContentType()`

Returns the tab icon's content type, such as `image/png`.

Signature

```
public String getContentType()
```

Return Value

Type: String

`getHeight()`

Returns the tab icon's height in pixels.

Signature

```
public Integer getHeight()
```

Return Value

Type: Integer

Usage

Note: If the icon content type is SVG, the icon won't have a size and its height is zero.

getTheme()

Returns the tab's icon theme.

Signature

```
public String getTheme()
```

Return Value

Type: String

Possible theme values include `theme3`, `theme4`, and `custom`.

- `theme3` is the Salesforce theme introduced during Spring '10.
- `theme4` is the Salesforce theme introduced in Winter '14 for the mobile touchscreen version of Salesforce.
- `custom` is the theme name associated with a custom icon.

getUrl()

Returns the tab's icon fully qualified URL.

Signature

```
public String getUrl()
```

Return Value

Type: String

getWidth()

Returns the tab's icon width in pixels.

Signature

```
public Integer getWidth()
```

Return Value

Type: Integer

Usage

Note: If the icon content type is SVG, the icon won't have a size and its width is zero.

DescribeSObjectResult Class

Contains methods for describing SObjects. None of the methods take an argument.

Namespace

Schema

Usage

Instances of describe results on the same `DescribeSObjectResult` aren't guaranteed to be equal because the state and behavior of a describe object is determined by various factors including the API version used. To compare describe results, call the `getSObjectType()` method on the SObject describe results and use the equality operator (==) to compare the `SObjectType` values.

DescribeSObjectResult Properties

The following are properties for `DescribeSObjectResult`.

accessible

Indicates whether the current user has access to the SObject.

Signature

```
public Boolean accessible {get; set;}
```

Property Value

Type: Boolean

associateentitytype

The type of associated object. For example, `History` or `Share`.

Signature

```
public String associateentitytype {get; set;}
```

Property Value

Type: String

associateparententity

The parent object of an associated object.

Signature

```
public String associateparententity {get; set;}
```

Property Value

Type: String

childrelationships

A list of child relationships, which is the name of the sObject that has a foreign key to the sObject being described.

Signature

```
public List<Schema.ChildRelationship> childrelationships {get; set;}
```

Property Value

Type: List<Schema.ChildRelationship on page 3089>

createable

Indicates whether the SObject can be created by the current user.

Signature

```
public Boolean createable {get; set;}
```

Property Value

Type: Boolean

custom

Indicates whether the SObject is a custom object.

Signature

```
public Boolean custom {get; set;}
```

Property Value

Type: Boolean

customsetting

Indicates whether the SObject is a custom setting.

Signature

```
public Boolean customsetting {get; set;}
```

Property Value

Type: Boolean

datatranslationenabled

Indicates whether data translation is enabled for the SObject. This property is available in API version 49.0 and later.

Signature

```
public Boolean datatranslationenabled {get; set;}
```

Property Value

Type: Boolean

defaultimplementation

Reserved for future use.

Signature

```
public String defaultimplementation {get; set;}
```

Property Value

Type: String

deletable

Indicates whether the SObject can be deleted by the current user.

Signature

```
public Boolean deletable {get; set;}
```

Property Value

Type: Boolean

deprecatedandhidden

Reserved for future use.

Signature

```
public Boolean deprecatedandhidden {get; set;}
```

Property Value

Type: Boolean

feedenabled

Indicates whether Chatter feeds are enabled for the SObject.

Signature

```
public Boolean feedenabled {get; set;}
```

Property Value

Type: Boolean

fields

A list of fields associated with the SObject.

Signature

```
public Schema.SObjectTypeFields fields {get; set;}
```

Property Value

Type: Schema.SObjectTypeFields

Follow `fields` with the `getMap` method.

Example

This sample code shows how to use `fields`. To get a custom field, specify the custom field name.

```
Schema.DescribeFieldResult dfr = Schema.SObjectType.Account.fields.Name;
```

fieldSets

Represents field sets, which is a grouping of the SObject fields.

Signature

```
public Schema.SObjectTypeFieldSets fieldsets {get; set;}
```

Property Value

Type: Schema.SObjectTypeFieldSets

Follow `fieldSets` with a field set name or with the `getMap` method.

Example

This sample code shows how to use `fieldSet`.

```
Schema.DescribeSObjectResult d =
  Account.sObjectType.getDescribe();
Map<String, Schema.FieldSet> FsMap =
  d.fieldSets.getMap();
```

hassubtypes

Reserved for future use.

Signature

```
public Boolean hassubtypes {get; set;}
```

Property Value

Type: Boolean

implementedby

Reserved for future use.

Signature

```
public String implementedby {get; set;}
```

Property Value

Type: String

implementsinterfaces

Reserved for future use.

Signature

```
public String implementsinterfaces {get; set;}
```

Property Value

Type: String

isinterface

Reserved for future use.

Signature

```
public Boolean isinterface {get; set;}
```

Property Value

Type: Boolean

keyprefix

The three-character prefix code in the SObject ID.

Signature

```
public String keyprefix {get; set;}
```

Property Value

Type: String

`label`

The SObject's label, which may or may not match the object name. For example, an organization representing a medical vertical might rename Account to Patient. Tabs and fields can be renamed in the Salesforce user interface.

Signature

```
public String label {get; set;}
```

Property Value

Type: String

`labelplural`

The SObject's plural label, which may or may not match the object name. For example, Accounts.

Signature

```
public String labelplural {get; set;}
```

Property Value

Type: String

`localname`

The name of the SObject. If the object is part of the current namespace, the namespace portion of the name is omitted.

Signature

```
public String localname {get; set;}
```

Property Value

Type: String

`mergeable`

Indicates whether the SObject can be merged with other objects of its type by the current user. This is set to `true` for leads, contacts, and accounts.

Signature

```
public Boolean mergeable {get; set;}
```

Property Value

Type: Boolean

mruenabled

Indicates whether Most Recently Used (MRU) list functionality is enabled for the SObject.

Signature

```
public Boolean mruenabled {get; set;}
```

Property Value

Type: Boolean

name

The name field of the SObject.

Signature

```
public String name {get; set;}
```

Property Value

Type: String

queryable

Indicates whether the SObject can be queried by the current user.

Signature

```
public Boolean queryable {get; set;}
```

Property Value

Type: Boolean

recordtypeinfos

A list of the record types supported by the SObject.

Signature

```
public List<Schema.RecordTypeInfo> recordtypeinfos {get; set;}
```

Property Value

Type: List<Schema.RecordTypeInfo>

recordtypeinfosbydevelopername

A map that matches developer names to their associated record type.

Signature

```
public Map<String,Schema.RecordTypeInfo> recordtypeinfosbydevelopername {get; set;}
```

Property Value

Type: Map<String, Schema.RecordTypeInfo>

recordtypeinfosbyid

A map that matches record IDs to their associated record types.

Signature

```
public Map<Id,Schema.RecordTypeInfo> recordtypeinfosbyid {get; set;}
```

Property Value

Type: Map<ID, Schema.RecordTypeInfo>

recordtypeinfosbyname

A map that matches record labels to their associated record type.

Signature

```
public Map<String,Schema.RecordTypeInfo> recordtypeinfosbyname {get; set;}
```

Property Value

Type: Map<String, Schema.RecordTypeInfo>

searchable

Indicates whether the SObject can be searched by the current user.

Signature

```
public Boolean searchable {get; set;}
```

Property Value

Type: Boolean

sobjectdescribeoption

The effective describe option used by the system for the SObject.

Signature

```
public Schema.SObjectDescribeOptions sobjectdescribeoption {get; set;}
```

Property Value

Type: Schema.SObjectDescribeOptions

sobjecttype

The Schema.SObjectType object for the SObject.

Signature

```
public Schema.SObjectType sobjecttype {get; set;}
```

Property Value

Type: Schema.SObjectType

undeletable

Indicates whether the SObject can be undeleted by the current user.

Signature

```
public Boolean undeletable {get; set;}
```

Property Value

Type: Boolean

updateable

Indicates whether the SObject can be updated by the current user.

Signature

```
public Boolean updateable {get; set;}
```

Property Value

Type: Boolean

DescribeSObjectResult Methods

The following are methods for `DescribeSObjectResult`. All are instance methods.

IN THIS SECTION:

equals(obj)
Compares the SObject to the specified object and returns true if both are equal. Otherwise, returns false.

getAssociateEntityType()
Returns additional metadata for an associated object of a specified parent but only if it's a specific associated object type. Used in combination with the `getAssociateParentEntity()` method to get the parent object. For example, invoking the method on AccountHistory returns the parent object as `Account` and the type of associated object as `History`.

getAssociateParentEntity()
Returns additional metadata for an associated object but only if it's associated to a specific parent object. Used in combination with the `getAssociateEntityType()` method to get the type of associated object. For example, invoking the method on AccountHistory returns the parent object as `Account` and the type of associated object as `History`.

getChildRelationships()
Returns a list of child relationships, which are the names of the sObjects that have a foreign key to the sObject being described.

getDataTranslationEnabled()
Returns true if data translation is enabled for the SObject. Otherwise, returns false.

getDefaultImplementation()
Reserved for future use.

getFields()
Returns the fields that make up the SObject being described.

getFieldSets()
Returns field sets, which is a grouping of the SObject fields.

getHasSubtypes()
Reserved for future use.

getImplementedBy()
Reserved for future use.

getImplementsInterfaces()
Reserved for future use.

getIsInterface()
Reserved for future use.

getKeyPrefix()
Returns the three-character prefix code for the object. Record IDs are prefixed with three-character codes that specify the type of the object (for example, accounts have a prefix of `001` and opportunities have a prefix of `006`).

getLabel()
Returns the object's label, which may or may not match the object name.

getLabelPlural()
Returns the object's plural label, which may or may not match the object name.

getLocalName()
Returns the name of the object, similar to the `getName` method. However, if the object is part of the current namespace, the namespace portion of the name is omitted.

getName()
Returns the name of the object.

getRecordTypeInfos()

Returns a list of the record types supported by this object. The current user is not required to have access to a record type to see it in this list.

getRecordTypeInfosByDeveloperName()

Returns a map that matches developer names to their associated record type. The current user is not required to have access to a record type to see it in this map.

getRecordTypeInfosById()

Returns a map that matches record IDs to their associated record types. The current user is not required to have access to a record type to see it in this map.

getRecordTypeInfosByName()

Returns a map that matches record labels to their associated record type. The current user is not required to have access to a record type to see it in this map.

getSObjectDescribeOption()

Returns the effective describe option used by the system for the SObject.

getSobjectType()

Returns the Schema.SObjectType object for the sObject. You can use this to create a similar sObject.

getHasSubtypes()

Reserved for future use.

hashCode()

Returns the hash code for the SObject.

isAccessible()

Returns `true` if the current user can see this object, `false` otherwise.

isCreateable()

Returns `true` if the object can be created by the current user, `false` otherwise.

isCustom()

Returns `true` if the object is a custom object, `false` if it is a standard object.

isCustomSetting()

Returns `true` if the object is a custom setting, `false` otherwise.

isDeletable()

Returns `true` if the object can be deleted by the current user, `false` otherwise.

isDeprecatedAndHidden()

Reserved for future use.

isFeedEnabled()

Returns `true` if Chatter feeds are enabled for the object, `false` otherwise. This method is only available for Apex classes and triggers saved using SalesforceAPI version 19.0 and later.

isMergeable()

Returns `true` if the object can be merged with other objects of its type by the current user, `false` otherwise. `true` is returned for leads, contacts, and accounts.

isMruEnabled()

Returns `true` if Most Recently Used (MRU) list functionality is enabled for the object, `false` otherwise.

isQueryable()
Returns `true` if the object can be queried by the current user, `false` otherwise

isSearchable()
Returns `true` if the object can be searched by the current user, `false` otherwise.

isUndeletable()
Returns `true` if the object can be undeleted by the current user, `false` otherwise.

isUpdateable()
Returns `true` if the object can be updated by the current user, `false` otherwise.

toString()
Returns a string that represents the SObject.

`equals(obj)`

Compares the SObject to the specified object and returns true if both are equal. Otherwise, returns false.

Signature

```
public Boolean equals(Object obj)
```

Parameters

obj
Type: Object
The object with which to compare.

Return Value

Type: Boolean

`getAssociateEntityType()`

Returns additional metadata for an associated object of a specified parent but only if it's a specific associated object type. Used in combination with the `getAssociateParentEntity()` method to get the parent object. For example, invoking the method on AccountHistory returns the parent object as `Account` and the type of associated object as `History`.

Signature

```
public String associateentitytype {get; set;}
```

Return Value

Type: String

SEE ALSO:
DescribeSObjectResult Properties

getAssociateParentEntity()

Returns additional metadata for an associated object but only if it's associated to a specific parent object. Used in combination with the `getAssociateEntityType()` method to get the type of associated object. For example, invoking the method on AccountHistory returns the parent object as `Account` and the type of associated object as `History`.

Signature

```
public String getAssociateParentEntity()
```

Return Value

Type: String

SEE ALSO:

DescribeSObjectResult Properties

getChildRelationships()

Returns a list of child relationships, which are the names of the sObjects that have a foreign key to the sObject being described.

Signature

```
public Schema.ChildRelationship getChildRelationships()
```

Return Value

Type: List<Schema.ChildRelationship>

Example

For example, the Account object includes `Contacts` and `Opportunities` as child relationships.

getDataTranslationEnabled()

Returns true if data translation is enabled for the SObject. Otherwise, returns false.

Signature

```
public Boolean getDataTranslationEnabled()
```

Return Value

Type: Boolean

getDefaultImplementation()

Reserved for future use.

Signature

```
public String getDefaultImplementation()
```

Return Value

Type: String

getFields()

Returns the fields that make up the SObject being described.

Signature

```
public Schema.SObjectTypeFields getFields()
```

Return Value

Type: Schema.SObjectTypeFields

The return value is a special data type. Call the `getMap()` method to get a map of Strings and SObjectFields.

Usage

When you describe SObjects and their fields from within an Apex class, custom fields of new field types are returned regardless of the API version that the class is saved in. If a field type, such as the geolocation field type, is available only in a recent API version, components of a geolocation field are returned even if the class is saved in an earlier API version.

SEE ALSO:

Apex Developer Guide: Using Field Tokens

Apex Developer Guide: Describing sObjects Using Schema Method

Apex Developer Guide: Understanding Apex Describe Information

getFieldSets()

Returns field sets, which is a grouping of the SObject fields.

Signature

```
public Schema.SObjectTypeFieldSets getFieldSets()
```

Return Value

Type: Schema.SObjectTypeFieldSets

The return value is a special data type. Call the `getMap()` method to get a map of Strings and SObjectFieldSets.

SEE ALSO:

Apex Developer Guide: Using Field Tokens

Apex Developer Guide: Describing sObjects Using Schema Method

Apex Developer Guide: Understanding Apex Describe Information

getHasSubtypes()

Reserved for future use.

To check if Person Accounts are enabled for the current org, use this code snippet:

```
Schema.SObjectType.Account.fields.getMap().containsKey( 'isPersonAccount' );
```

Signature

```
public Boolean getHasSubtypes()
```

Return Value

Type: Boolean

getImplementedBy()

Reserved for future use.

Signature

```
public String getImplementedBy()
```

Return Value

Type: String

getImplementsInterfaces()

Reserved for future use.

Signature

```
public String getImplementsInterfaces()
```

Return Value

Type: String

getIsInterface()

Reserved for future use.

Signature

```
public Boolean getIsInterface()
```

Return Value

Type: Boolean

`getKeyPrefix()`

Returns the three-character prefix code for the object. Record IDs are prefixed with three-character codes that specify the type of the object (for example, accounts have a prefix of `001` and opportunities have a prefix of `006`).

Signature

```
public String getKeyPrefix()
```

Return Value

Type: String

Usage

The DescribeSobjectResult object returns a value for objects that have a stable prefix. For object types that do not have a stable or predictable prefix, this field is blank. Client applications that rely on these codes can use this way of determining object type to ensure forward compatibility.

`getLabel()`

Returns the object's label, which may or may not match the object name.

Signature

```
public String getLabel()
```

Return Value

Type: String

Usage

The object's label might not always match the object name. For example, an organization in the medical industry might change the label for Account to Patient. This label is then used in the Salesforce user interface. See the Salesforce online help for more information.

`getLabelPlural()`

Returns the object's plural label, which may or may not match the object name.

Signature

```
public String getLabelPlural()
```

Return Value

Type: String

Usage

The object's plural label might not always match the object name. For example, an organization in the medical industry might change the plural label for Account to Patients. This label is then used in the Salesforce user interface. See the Salesforce online help for more information.

getLocalName()

Returns the name of the object, similar to the `getName` method. However, if the object is part of the current namespace, the namespace portion of the name is omitted.

Signature

```
public String getLocalName()
```

Return Value

Type: String

getName()

Returns the name of the object.

Signature

```
public String getName()
```

Return Value

Type: String

getRecordTypeInfos()

Returns a list of the record types supported by this object. The current user is not required to have access to a record type to see it in this list.

Signature

```
public List<Schema.RecordTypeInfo> getRecordTypeInfos()
```

Return Value

Type: List<Schema.RecordTypeInfo>

getRecordTypeInfosByDeveloperName()

Returns a map that matches developer names to their associated record type. The current user is not required to have access to a record type to see it in this map.

Signature

```
public Map<String, Schema.RecordTypeInfo> getRecordTypeInfosByDeveloperName()
```

Return Value

Type: Map<String, Schema.RecordTypeInfo>

getRecordTypeInfosById()

Returns a map that matches record IDs to their associated record types. The current user is not required to have access to a record type to see it in this map.

Signature

```
public Schema.RecordTypeInfo getRecordTypeInfosById()
```

Return Value

Type: Map<ID, Schema.RecordTypeInfo>

getRecordTypeInfosByName()

Returns a map that matches record labels to their associated record type. The current user is not required to have access to a record type to see it in this map.

Signature

```
public Schema.RecordTypeInfo getRecordTypeInfosByName()
```

Return Value

Type: Map<String, Schema.RecordTypeInfo>

getSObjectDescribeOption()

Returns the effective describe option used by the system for the SObject.

Signature

```
public Schema.SObjectDescribeOptions getSObjectDescribeOption()
```

Return Value

Type: Schema.SObjectDescribeOptions

Valid values are:

- `SObjectDescribeOptions.FULL`: Indicates eager-load all elements of the describe, including child relationships, up-front at the time of method invocation.
- `SObjectDescribeOptions.DEFERRED`: Indicates lazy-load child relationships. This means that all child relationships will not be loaded at the time of first invocation of the method.

`getSobjectType()`

Returns the Schema.SObjectType object for the sObject. You can use this to create a similar sObject.

Signature

```
public Schema.SObjectType getSobjectType()
```

Return Value

Type: Schema.SObjectType

`getHasSubtypes()`

Reserved for future use.

To check if Person Accounts are enabled for the current org, use this code snippet:

```
Schema.SObjectType.Account.fields.getMap().containsKey( 'isPersonAccount' );
```

Signature

```
public Boolean getHasSubtypes()
```

Return Value

Type: Boolean

`hashCode()`

Returns the hash code for the SObject.

Signature

```
public Integer hashCode()
```

Return Value

Type: Integer

`isAccessible()`

Returns `true` if the current user can see this object, `false` otherwise.

Signature

```
public Boolean isAccessible()
```

Return Value

Type: Boolean

Versioned Behavior Changes

In API version 54.0 and later, for custom settings and custom metadata type objects, `DescribeSObjectResult.isAccessible()` returns `false` if the user doesn't have permissions to access the queried objects. In API version 53.0 and earlier, the method returns `true` even if the user doesn't have the required permissions.

`isCreateable()`

Returns `true` if the object can be created by the current user, `false` otherwise.

Signature

```
public Boolean isCreateable()
```

Return Value

Type: Boolean

`isCustom()`

Returns `true` if the object is a custom object, `false` if it is a standard object.

Signature

```
public Boolean isCustom()
```

Return Value

Type: Boolean

`isCustomSetting()`

Returns `true` if the object is a custom setting, `false` otherwise.

Signature

```
public Boolean isCustomSetting()
```

Return Value

Type: Boolean

`isDeletable()`

Returns `true` if the object can be deleted by the current user, `false` otherwise.

Signature

```
public Boolean isDeletable()
```

Return Value

Type: Boolean

isDeprecatedAndHidden()

Reserved for future use.

Signature

```
public Boolean isDeprecatedAndHidden()
```

Return Value

Type: Boolean

isFeedEnabled()

Returns `true` if Chatter feeds are enabled for the object, `false` otherwise. This method is only available for Apex classes and triggers saved using SalesforceAPI version 19.0 and later.

Signature

```
public Boolean isFeedEnabled()
```

Return Value

Type: Boolean

isMergeable()

Returns `true` if the object can be merged with other objects of its type by the current user, `false` otherwise. `true` is returned for leads, contacts, and accounts.

Signature

```
public Boolean isMergeable()
```

Return Value

Type: Boolean

isMruEnabled()

Returns `true` if Most Recently Used (MRU) list functionality is enabled for the object, `false` otherwise.

Signature

```
public Boolean isMruEnabled()
```

Return Value

Type: Boolean

`isQueryable()`

Returns `true` if the object can be queried by the current user, `false` otherwise

Signature

```
public Boolean isQueryable()
```

Return Value

Type: Boolean

`isSearchable()`

Returns `true` if the object can be searched by the current user, `false` otherwise.

Signature

```
public Boolean isSearchable()
```

Return Value

Type: Boolean

`isUndeletable()`

Returns `true` if the object can be undeleted by the current user, `false` otherwise.

Signature

```
public Boolean isUndeletable()
```

Return Value

Type: Boolean

`isUpdateable()`

Returns `true` if the object can be updated by the current user, `false` otherwise.

Signature

```
public Boolean isUpdateable()
```

Return Value

Type: Boolean

`toString()`

Returns a string that represents the SObject.

Signature

```
public String toString()
```

Return Value

Type: String

DescribeTabResult Class

Contains tab metadata information for a tab in a standard or custom app available in the Salesforce user interface.

Namespace

Schema

Usage

The `getTabs` method of the `Schema.DescribeTabSetResult` returns a list of `Schema.DescribeTabResult` objects that describe the tabs of one app.

The methods in the `Schema.DescribeTabResult` class can be called using their property counterparts. For each method starting with `get`, you can omit the `get` prefix and the ending parentheses `()` to call the property counterpart. For example, `tabResultObj.label` is equivalent to `tabResultObj.getLabel()`. Similarly, for each method starting with `is`, omit the `is` prefix and the ending parentheses `()`. For example, `tabResultObj.isCustom` is equivalent to `tabResultObj.custom`.

DescribeTabResult Methods

The following are methods for `DescribeTabResult`. All are instance methods.

IN THIS SECTION:

getColors()
Returns a list of color metadata information for all colors associated with this tab. Each color is associated with a theme and context.

getIconUrl()
Returns the URL for the main 32 x 32-pixel icon for a tab. This icon corresponds to the current theme (theme3) and appears next to the heading at the top of most pages.

getIcons()
Returns a list of icon metadata information for all icons associated with this tab. Each icon is associated with a theme and context.

getLabel()

Returns the display label of this tab.

getMiniIconUrl()

Returns the URL for the 16 x 16-pixel icon that represents a tab. This icon corresponds to the current theme (theme3) and appears in related lists and other locations.

getSobjectName()

Returns the name of the sObject that is primarily displayed on this tab (for tabs that display a particular SObject).

getUrl()

Returns a fully qualified URL for viewing this tab.

isCustom()

Returns `true` if this is a custom tab, or `false` if this is a standard tab.

getColors()

Returns a list of color metadata information for all colors associated with this tab. Each color is associated with a theme and context.

Signature

```
public List<Schema.DescribeColorResult> getColors()
```

Return Value

Type: List<Schema.DescribeColorResult>

getIconUrl()

Returns the URL for the main 32 x 32-pixel icon for a tab. This icon corresponds to the current theme (theme3) and appears next to the heading at the top of most pages.

Signature

```
public String getIconUrl()
```

Return Value

Type: String

getIcons()

Returns a list of icon metadata information for all icons associated with this tab. Each icon is associated with a theme and context.

Signature

```
public List<Schema.DescribeIconResult> getIcons()
```

Return Value

Type: List<Schema.DescribeIconResult>

getLabel()

Returns the display label of this tab.

Signature

```
public String getLabel()
```

Return Value

Type: String

getMiniIconUrl()

Returns the URL for the 16 x 16-pixel icon that represents a tab. This icon corresponds to the current theme (theme3) and appears in related lists and other locations.

Signature

```
public String getMiniIconUrl()
```

Return Value

Type: String

getSobjectName()

Returns the name of the sObject that is primarily displayed on this tab (for tabs that display a particular SObject).

Signature

```
public String getSobjectName()
```

Return Value

Type: String

getUrl()

Returns a fully qualified URL for viewing this tab.

Signature

```
public String getUrl()
```

Return Value

Type: String

`isCustom()`

Returns `true` if this is a custom tab, or `false` if this is a standard tab.

Signature

```
public Boolean isCustom()
```

Return Value

Type: Boolean

DescribeTabSetResult Class

Contains metadata information about a Salesforce Classic standard or custom app available in the Salesforce user interface.

Namespace

Schema

Usage

The `Schema.describeTabs` method returns a list of `Schema.DescribeTabSetResult` objects that describe Salesforce Classic standard and custom apps.

The methods in the `Schema.DescribeTabSetResult` class can be called using their property counterparts. For each method starting with `get`, you can omit the `get` prefix and the ending parentheses `()` to call the property counterpart. For example, `tabSetResultObj.label` is equivalent to `tabSetResultObj.getLabel()`. Similarly, for each method starting with `is`, omit the `is` prefix and the ending parentheses `()`. For example, `tabSetResultObj.isSelected` is equivalent to `tabSetResultObj.selected`.

Example

This example shows how to call the `Schema.describeTabs` method to get describe information for all available Salesforce Classic apps. This example iterates through each describe result and gets more metadata information for the Sales app.

```
// App we're interested to get more info about
String appName = 'Sales';

// Get tab set describes for each app
List<Schema.DescribeTabSetResult> tabSetDesc = Schema.describeTabs();

// Iterate through each tab set describe for each app and display the info
for(Schema.DescribeTabSetResult tsr : tabSetDesc) {
    // Get more information for the Sales app
    if (tsr.getLabel() == appName) {
        // Find out if the app is selected
        if (tsr.isSelected()) {
            System.debug('The ' + appName + ' app is selected. ');
        }
        // Get the app's Logo URL and namespace
        String logo = tsr.getLogoUrl();
```

```
        System.debug('Logo URL: ' + logo);
        String ns = tsr.getNamespace();
        if (ns == '') {
            System.debug('The ' + appName + ' app has no namespace defined.');
        }
        else {
            System.debug('Namespace: ' + ns);
        }
        // Get the number of tabs
        System.debug('The ' + appName + ' app has ' + tsr.getTabs().size() + ' tabs.');
    }
}

// Example debug statement output
// DEBUG|The Sales app is selected.
// DEBUG|Logo URL:
https://MyDomainName.my.salesforce.com/img/seasonLogos/2014_winter_aloha.png
// DEBUG|The Sales app has no namespace defined.
// DEBUG|The Sales app has 14 tabs.
```

DescribeTabSetResult Methods

The following are methods for `DescribeTabSetResult`. All are instance methods.

IN THIS SECTION:

getDescription()
Returns the display description for the standard or custom app.

getLabel()
Returns the display label for the standard or custom app.

getLogoUrl()
Returns a fully qualified URL to the logo image associated with the standard or custom app.

getNamespace()
Returns the developer namespace prefix of a Salesforce AppExchange managed package.

getTabs()
Returns metadata information about the standard or custom app's displayed tabs.

isSelected()
Returns `true` if this standard or custom app is the user's currently selected app in Salesforce Classic. Otherwise, returns `false`.

`getDescription()`

Returns the display description for the standard or custom app.

Signature

```
public String getDescription()
```

Return Value

Type: String

getLabel()

Returns the display label for the standard or custom app.

Signature

```
public String getLabel()
```

Return Value

Type: String

Usage

The display label changes when tabs are renamed in the Salesforce user interface. See the Salesforce online help for more information.

getLogoUrl()

Returns a fully qualified URL to the logo image associated with the standard or custom app.

Signature

```
public String getLogoUrl()
```

Return Value

Type: String

getNamespace()

Returns the developer namespace prefix of a Salesforce AppExchange managed package.

Signature

```
public String getNamespace()
```

Return Value

Type: String

Usage

This namespace prefix corresponds to the namespace prefix of the Developer Edition organization that was enabled to allow publishing a managed package. This method applies to a custom app containing a set of tabs and installed as part of a managed package.

`getTabs()`

Returns metadata information about the standard or custom app's displayed tabs.

Signature

```
public List<Schema.DescribeTabResult> getTabs()
```

Return Value

Type: List<Schema.DescribeTabResult>

`isSelected()`

Returns `true` if this standard or custom app is the user's currently selected app in Salesforce Classic. Otherwise, returns `false`.

Signature

```
public Boolean isSelected()
```

Return Value

Type: Boolean

DisplayType Enum

A `Schema.DisplayType` enum value is returned by the field describe result's `getType` method.

Namespace

Schema

Type Field Value	What the Field Object Contains
`ADDRESS`	Address values
`ANYTYPE`	Any value of the following types: `String`, `Picklist`, `Boolean`, `Integer`, `Double`, `Percent`, `ID`, `Date`, `DateTime`, `URL`, or `Email`.
`BASE64`	Base64-encoded arbitrary binary data (of type base64Binary)
`BOOLEAN`	Boolean (`true` or `false`) values
`COMBOBOX`	Comboboxes, which provide a set of enumerated values and allow the user to specify a value not in the list
`COMPLEXVALUE`	Complex Value Type (CVT)
`CURRENCY`	Currency values
`DATACATEGORYGROUPREFERENCE`	Reference to a data category group or a category unique name
`DATE`	Date values

Type Field Value	What the Field Object Contains
DATETIME	DateTime values
DOUBLE	Double values
EMAIL	Email addresses
ENCRYPTEDSTRING	Encrypted string
FLOATARRAY	Float array
ID	Primary key field for an object
INTEGER	Integer values
JSON	Json format
LOCATION	Location values, including latitude and longitude.
LONG	Long values
MULTIPICKLIST	Multi-select picklists, which provide a set of enumerated values from which multiple values can be selected
PERCENT	Percent values
PHONE	Phone numbers. Values can include alphabetic characters. Client applications are responsible for phone number formatting.
PICKLIST	Single-select picklists, which provide a set of enumerated values from which only one value can be selected
REFERENCE	Cross-references to a different object, analogous to a foreign key field
SOBJECT	An sObject variable represents a row of data and can only be declared in Apex using the SOAP API name of the object.
STRING	String values
TEXTAREA	String values that are displayed as multiline text fields
TIME	Time values
URL	URL values that are displayed as hyperlinks

Usage

For more information, see Field Types in the *Object Reference for Salesforce*. For more information about the methods shared by all enums, see Enum Methods.

FieldDescribeOptions Enum

A `Schema.FieldDescribeOptions` enum value is a parameter in the `SObjectType.getDescribe` method.

Usage

For more information about the method using this enum, see `getDescribe(options)`.

Enum Values

The following are the values of the `Schema.FieldDescribeOptions` enum.

Value	Description
`DEFAULT`	Compute context-specific, describe field results.
`FULL_DESCRIBE`	Compute all aspects of describe field results.

FieldSet Class

Contains methods for discovering and retrieving the details of field sets created on sObjects.

Namespace

Schema

Usage

Use the methods in the `Schema.FieldSet` class to discover the fields contained within a field set, and get details about the field set itself, such as the name, namespace, label, and so on. The following example shows how to get a collection of field set describe result objects for an sObject. The key of the returned Map is the field set name, and the value is the corresponding field set describe result.

```
Map<String, Schema.FieldSet> FsMap =
    Schema.SObjectType.Account.fieldSets.getMap();
```

Field sets are also available from sObject describe results. The following lines of code are equivalent to the prior sample:

```
Schema.DescribeSObjectResult d =
  Account.sObjectType.getDescribe();
Map<String, Schema.FieldSet> FsMap =
  d.fieldSets.getMap();
```

To work with an individual field set, you can access it via the map of field sets on an sObject or, when you know the name of the field set in advance, using an explicit reference to the field set. The following two lines of code retrieve the same field set:

```
Schema.FieldSet fs1 = Schema.SObjectType.Account.fieldSets.getMap().get('field_set_name');
Schema.FieldSet fs2 = Schema.SObjectType.Account.fieldSets.field_set_name;
```

Example: Displaying a Field Set on a Visualforce Page

This sample uses `Schema.FieldSet` and `Schema.FieldSetMember` methods to dynamically get all the fields in the Dimensions field set for the Merchandise custom object. The list of fields is then used to construct a SOQL query that ensures those fields are available for display. The Visualforce page uses the `MerchandiseDetails` class as its controller.

```
public class MerchandiseDetails {

    public Merchandise__c merch { get; set; }

    public MerchandiseDetails() {
        this.merch = getMerchandise();
    }

    public List<Schema.FieldSetMember> getFields() {
        return SObjectType.Merchandise__c.FieldSets.Dimensions.getFields();
    }

    private Merchandise__c getMerchandise() {
        String query = 'SELECT ';
        for(Schema.FieldSetMember f : this.getFields()) {
            query += f.getFieldPath() + ', ';
        }
        query += 'Id, Name FROM Merchandise__c LIMIT 1';
        return Database.query(query);
    }
}
```

The Visualforce page using the above controller is simple:

```
<apex:page controller="MerchandiseDetails">
    <apex:form >

      <apex:pageBlock title="Product Details">
          <apex:pageBlockSection title="Product">
              <apex:inputField value="{!merch.Name}"/>
          </apex:pageBlockSection>

          <apex:pageBlockSection title="Dimensions">
              <apex:repeat value="{!fields}" var="f">
                  <apex:inputField value="{!merch[f.fieldPath]}"
                      required="{!OR(f.required, f.dbrequired)}"/>
              </apex:repeat>
          </apex:pageBlockSection>

        </apex:pageBlock>

    </apex:form>
</apex:page>
```

One thing to note about the above markup is the expression used to determine if a field on the form should be indicated as being a required field. A field in a field set can be required by either the field set definition, or the field's own definition. The expression handles both cases.

FieldSet Methods

The following are methods for `FieldSet`. All are instance methods.

IN THIS SECTION:

getDescription()
Returns the field set's description.

getFields()
Returns a list of `Schema.FieldSetMember` objects for the fields making up the field set.

getLabel()
Returns the translation of the text label that is displayed next to the field in the Salesforce user interface.

getName()
Returns the field set's name.

getNamespace()
Returns the field set's namespace.

getSObjectType()
Returns the `Schema.sObjectType` of the sObject containing the field set definition.

`getDescription()`

Returns the field set's description.

Signature

```
public String getDescription()
```

Return Value

Type: `String`

Usage

Description is a required field for a field set, intended to describe the context and content of the field set. It's often intended for administrators who might be configuring a field set defined in a managed package, rather than for end users.

`getFields()`

Returns a list of `Schema.FieldSetMember` objects for the fields making up the field set.

Signature

```
public List<FieldSetMember> getFields()
```

Return Value

Type: List<Schema.FieldSetMember>

`getLabel()`

Returns the translation of the text label that is displayed next to the field in the Salesforce user interface.

Signature

```
public String getLabel()
```

Return Value

Type: `String`

`getName()`

Returns the field set's name.

Signature

```
public String getName()
```

Return Value

Type: `String`

`getNamespace()`

Returns the field set's namespace.

Signature

```
public String getNamespace()
```

Return Value

Type: `String`

Usage

The returned namespace is an empty string if your organization hasn't set a namespace, and the field set is defined in your organization. Otherwise, it's the namespace of your organization, or the namespace of the managed package containing the field set.

`getSObjectType()`

Returns the `Schema.sObjectType` of the sObject containing the field set definition.

Signature

```
public Schema.SObjectType getSObjectType()
```

Return Value

Type: `Schema.SObjectType`

FieldSetMember Class

Contains methods for accessing the metadata for field set member fields.

Namespace

Schema

Usage

Use the methods in the `Schema.FieldSetMember` class to get details about fields contained within a field set, such as the field label, type, a dynamic SOQL-ready field path, and so on. The following example shows how to get a collection of field set member describe result objects for a specific field set on an sObject:

```
List<Schema.FieldSetMember> fields =
    Schema.SObjectType.Account.fieldSets.getMap().get('field_set_name').getFields();
```

If you know the name of the field set in advance, you can access its fields more directly using an explicit reference to the field set:

```
List<Schema.FieldSetMember> fields =
    Schema.SObjectType.Account.fieldSets.field_set_name.getFields();
```

SEE ALSO:

FieldSet Class

FieldSetMember Methods

The following are methods for `FieldSetMember`. All are instance methods.

IN THIS SECTION:

getDBRequired()
Returns `true` if the field is required by the field's definition in its sObject, otherwise, `false`.

getFieldPath()
Returns a field path string in a format ready to be used in a dynamic SOQL query.

getLabel()
Returns the text label that's displayed next to the field in the Salesforce user interface.

getRequired()
Returns `true` if the field is required by the field set, otherwise, `false`.

getType()
Returns the field's Apex data type.

getSObjectField()
Returns the token for this field.

getDBRequired()

Returns `true` if the field is required by the field's definition in its sObject, otherwise, `false`.

Signature

```
public Boolean getDBRequired()
```

Return Value

Type: `Boolean`

getFieldPath()

Returns a field path string in a format ready to be used in a dynamic SOQL query.

Signature

```
public String getFieldPath()
```

Return Value

Type: `String`

Example

See Displaying a Field Set on a Visualforce Page for an example of how to use this method.

getLabel()

Returns the text label that's displayed next to the field in the Salesforce user interface.

Signature

```
public String getLabel()
```

Return Value

Type: `String`

getRequired()

Returns `true` if the field is required by the field set, otherwise, `false`.

Signature

```
public Boolean getRequired()
```

Return Value

Type: `Boolean`

`getType()`

Returns the field's Apex data type.

Signature

```
public Schema.DisplayType getType()
```

Return Value

Type: `Schema.DisplayType`

`getSObjectField()`

Returns the token for this field.

Signature

```
public Schema.sObjectField getSObjectField()
```

Return Value

Type: Schema.SObjectField

PicklistEntry Class

Represents a picklist entry.

Namespace

Schema

Usage

Picklist fields contain a list of one or more items from which a user chooses a single item. They display as drop-down lists in the Salesforce user interface. One of the items can be configured as the default item.

A `Schema.PicklistEntry` object is returned from the field describe result using the `getPicklistValues` method. For example:

```
Schema.DescribeFieldResult F = Account.Industry.getDescribe();
List<Schema.PicklistEntry> P = F.getPicklistValues();
```

PicklistEntry Methods

The following are methods for `PicklistEntry`. All are instance methods.

IN THIS SECTION:

getLabel()
Returns the display name of this item in the picklist.

getValue()
Returns the value of this item in the picklist.

isActive()
Returns `true` if this item must be displayed in the drop-down list for the picklist field in the user interface, `false` otherwise.

isDefaultValue()
Returns `true` if this item is the default value for the picklist, `false` otherwise. Only one item in a picklist can be designated as the default.

getLabel()

Returns the display name of this item in the picklist.

Signature

```
public String getLabel()
```

Return Value

Type: String

getValue()

Returns the value of this item in the picklist.

Signature

```
public String getValue()
```

Return Value

Type: String

isActive()

Returns `true` if this item must be displayed in the drop-down list for the picklist field in the user interface, `false` otherwise.

Signature

```
public Boolean isActive()
```

Return Value

Type: Boolean

isDefaultValue()

Returns `true` if this item is the default value for the picklist, `false` otherwise. Only one item in a picklist can be designated as the default.

Signature

```
public Boolean isDefaultValue()
```

Return Value

Type: Boolean

RecordTypeInfo Class

Contains methods for accessing record type information for an sObject with associated record types.

Namespace

Schema

Usage

A RecordTypeInfo object is returned from the sObject describe result using the `getRecordTypeInfos` method. For example:

```
Schema.DescribeSObjectResult R = Account.SObjectType.getDescribe();
List<Schema.RecordTypeInfo> RT = R.getRecordTypeInfos();
```

In addition to the `getRecordTypeInfos` method, you can use the `getRecordTypeInfosById` and the `getRecordTypeInfosByName` methods. These methods return maps that associate RecordTypeInfo with record IDs and record labels, respectively.

Example

The following example assumes at least one record type has been created for the Account object:

```
RecordType rt = [SELECT Id,Name FROM RecordType WHERE SobjectType='Account' LIMIT 1];
Schema.DescribeSObjectResult d = Schema.SObjectType.Account;
Map<Id,Schema.RecordTypeInfo> rtMapById = d.getRecordTypeInfosById();
Schema.RecordTypeInfo rtById =  rtMapById.get(rt.id);
Map<String,Schema.RecordTypeInfo> rtMapByName = d.getRecordTypeInfosByName();
Schema.RecordTypeInfo rtByName =  rtMapByName.get(rt.name);
System.assertEquals(rtById,rtByName);
```

RecordTypeInfo Methods

The following are methods for `RecordTypeInfo`. All are instance methods.

IN THIS SECTION:

getDeveloperName()
Returns the developer name for this record type.

getName()
Returns the UI label of this record type. The label can be translated into any language that Salesforce supports.

getRecordTypeId()
Returns the ID of this record type.

isActive()
Returns `true` if this record type is active, `false` otherwise.

isAvailable()
Returns `true` if this record type is available to the current user, `false` otherwise. Use this method to display a list of available record types to the user when he or she is creating a new record.

isDefaultRecordTypeMapping()
Returns `true` if this is the default record type for the user, `false` otherwise.

isMaster()
Returns `true` if this is the master record type and `false` otherwise. The master record type is the default record type that's used when a record has no custom record type associated with it.

getDeveloperName()

Returns the developer name for this record type.

Signature

```
public String getDeveloperName()
```

Return Value

Type: String

getName()

Returns the UI label of this record type. The label can be translated into any language that Salesforce supports.

Signature

```
public String getName()
```

Return Value

Type: String

getRecordTypeId()

Returns the ID of this record type.

Signature

```
public ID getRecordTypeId()
```

Return Value

Type: ID

isActive()

Returns `true` if this record type is active, `false` otherwise.

Signature

```
public Boolean isActive()
```

Return Value

Type: Boolean

isAvailable()

Returns `true` if this record type is available to the current user, `false` otherwise. Use this method to display a list of available record types to the user when he or she is creating a new record.

Signature

```
public Boolean isAvailable()
```

Return Value

Type: Boolean

isDefaultRecordTypeMapping()

Returns `true` if this is the default record type for the user, `false` otherwise.

Signature

```
public Boolean isDefaultRecordTypeMapping()
```

Return Value

Type: Boolean

isMaster()

Returns `true` if this is the master record type and `false` otherwise. The master record type is the default record type that's used when a record has no custom record type associated with it.

Signature

```
public Boolean isMaster()
```

Return Value

Type: Boolean

SOAPType Enum

A `Schema.SOAPType` enum value is returned by the field describe result `getSoapType` method.

Namespace

Schema

Type Field Value	What the Field Object Contains
`anytype`	Any value of the following types: `String`, `Boolean`, `Integer`, `Double`, `ID`, `Date` or `DateTime`.
`base64binary`	Base64-encoded arbitrary binary data (of type base64Binary)
`Boolean`	Boolean (`true` or `false`) values
`Date`	Date values
`DateTime`	DateTime values
`Double`	Double values
`ID`	Primary key field for an object
`Integer`	Integer values
`String`	String values
`Time`	Time values

Usage

To programmatically retrieve the list of valid SOAPType enum values, use this code sample.

```
system.debug(SoapType.values().size()); //Gets the number of supported values
for (SoapType st : SoapType.values()) system.debug(st);
```

For more information, see SOAPTypes in the *SOAP API Developer Guide*. For more information about the methods shared by all enums, see Enum Methods.

SObjectDescribeOptions Enum

A `Schema.SObjectDescribeOptions` enum value is a parameter in the `SObjectType.getDescribe` method.

Usage

For more information about the method using this enum, see `getDescribe(options)`.

Enum Values

The following are the values of the `Schema.SObjectDescribeOptions` enum.

Value	Description
DEFAULT	Either eager-load or lazy-load depending on the API version.
DEFERRED	Lazy-load child relationships; do not load all child relationships at the time of first invocation of the method.
FULL	Eager-load all elements of the describe, including child relationships, up-front at the time of method invocation.

See `getDescribe(options)`.

SObjectField Class

A `Schema.sObjectField` object is returned from the field describe result using the `getController` and `getSObjectField` methods.

Namespace

Schema

Example

```
Schema.DescribeFieldResult F = Account.Industry.getDescribe();
Schema.sObjectField T = F.getSObjectField();
```

sObjectField Methods

The following are instance methods for `sObjectField`.

IN THIS SECTION:

getDescribe()
Returns the describe field result for this field.

getDescribe(options)
Returns the describe field result for this field. This method also provides an option to get all the describe field results for an object.

`getDescribe()`

Returns the describe field result for this field.

Signature

```
public Schema.DescribeFieldResult getDescribe()
```

Return Value

Type: Schema.DescribeFieldResult

`getDescribe(options)`

Returns the describe field result for this field. This method also provides an option to get all the describe field results for an object.

Signature

```
public Schema.DescribeFieldResult getDescribe(Object options)
```

Parameters

`options`
Type: Object

Use this parameter to pass `FieldDescribeOptions.FULL_DESCRIBE` when a subset of system objects could have different results for picklist values based on the context they're invoked in. This parameter computes all aspects of describe field results.

For example, `AIConversationContext.PersonType` field is a picklist that contains a list of accessible object types.

Return Value

Type: Schema.DescribeFieldResult

SObjectType Class

A `Schema.sObjectType` object is returned from the field describe result using the `getReferenceTo` method, or from the sObject describe result using the `getSObjectType` method.

Namespace

Schema

Usage

```
Schema.DescribeFieldResult F = Account.Industry.getDescribe();
List<Schema.sObjectType> P = F.getReferenceTo();
```

SObjectType Methods

The following are methods for `SObjectType`. All are instance methods.

IN THIS SECTION:

getDescribe()
Returns the describe sObject result for this field.

getDescribe(options)
Returns the describe sObject result for this field; the parameter value determines whether all child relationships are loaded up-front, or not.

newSObject()
Constructs a new sObject of this type.

newSObject(id)

Constructs a new sObject of this type, with the specified ID.

newSObject(recordTypeId, loadDefaults)

Constructs a new sObject of this type, and optionally, of the specified record type ID and with default custom field values.

getDescribe()

Returns the describe sObject result for this field.

Signature

```
public Schema.DescribeSObjectResult getDescribe()
```

Return Value

Type: Schema.DescribeSObjectResult

getDescribe(options)

Returns the describe sObject result for this field; the parameter value determines whether all child relationships are loaded up-front, or not.

Signature

```
public Schema.DescribeSObjectResult getDescribe(Object options)
```

Parameters

options

Type: Object

The parameter values determine how the elements of the describe operation are loaded.

- Use `SObjectDescribeOptions.FULL` to eager-load all elements of the describe, including child relationships, up-front at the time of method invocation. This describe guarantees fully coherent results, even if the describe object is passed to another namespace, API version, or other Apex context that may have different results when generating describe attributes.
- Use `SObjectDescribeOptions.DEFERRED` to enable lazy initialization of describe attributes on first use. This means that all child relationships will not be loaded at the time of first invocation of the method.
- Use `SObjectDescribeOptions.DEFAULT` to default to either eager-load or lazy-load depending on the API version.

The type of describe operation, as determined by the parameter value is depicted in this table.

Table 2: Type of Load for SObjectType.getDescribe()

Parameter Value	API Version 43.0 and Earlier	API Version 44.0 and Later
Full	Eager	Eager
Deferred	Lazy	Lazy
Default	Lazy	Lazy

Return Value

Type: Schema.DescribeSObjectResult

`newSObject()`

Constructs a new sObject of this type.

Signature

```
public sObject newSObject()
```

Return Value

Type: sObject

Example

For an example, see Dynamic DML.

`newSObject(id)`

Constructs a new sObject of this type, with the specified ID.

Signature

```
public sObject newSObject(ID id)
```

Parameters

id
Type: ID

Return Value

Type: sObject

Usage

For the argument, pass the ID of an existing record in the database.

After you create a new sObject, the sObject returned has all fields set to `null`. You can set any updateable field to desired values and then update the record in the database. Only the fields you set new values for are updated and all other fields which are not system fields are preserved.

`newSObject(recordTypeId, loadDefaults)`

Constructs a new sObject of this type, and optionally, of the specified record type ID and with default custom field values.

Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Signature

```
public sObject newSObject(ID recordTypeId, Boolean loadDefaults)
```

Parameters

recordTypeId
Type: ID

Specifies the record type ID of the sObject to create. If no record type exists for this sObject, use `null`. If the sObject has record types and you specify `null`, the default record type is used.

loadDefaults
Type: Boolean

Specifies whether to populate custom fields with their predefined default values (`true`) or not (`false`).

Return Value

Type: sObject

Usage

- For required fields that have no default values, make sure to provide a value before inserting the new sObject. Otherwise, the insertion results in an error. An example is the Account Name field or a master-detail relationship field.
- Since picklists and multi-select picklists can have default values specified per record type, this method populates the default value corresponding to the record type specified.
- If fields have no predefined default values and the *loadDefaults* argument is `true`, this method creates the sObject with field values of `null`.
- If the *loadDefaults* argument is `false`, this method creates the sObject with field values of `null`.
- This method populates read-only custom fields of the new sObject with default values. You can then insert the new sObject with the read-only fields, even though these fields cannot be edited after they're inserted.
- If a custom field is marked as unique and also provides a default value, inserting more than one new sObject will cause a run-time exception because of duplicate field values.

To learn more about default field values, see "Default Field Values" in the Salesforce online help.

Note: You can also use this method to create a platform event with a prepopulated `EventUuid` field value for Apex publish callbacks. For more information, see Get the Result of Asynchronous Platform Event Publishing with Apex Publish Callbacks in the *Platform Events Developer Guide*.

Example: Creating New sObject with Default Values

This sample creates an account with any default values populated for its custom fields, if any, using the `newSObject` method. It also creates a second account for a specific record type. For both accounts, the sample sets the Name field, which is a required field that doesn't have a default value, before inserting the new accounts.

```
// Create an account with predefined default values
Account acct = (Account)Account.sObjectType.newSObject(null, true);
// Provide a value for Name
acct.Name = 'Acme';
// Insert new account
insert acct;
```

```
// This is for record type RT1 of Account
ID rtId = [SELECT Id FROM RecordType WHERE sObjectType='Account' AND Name='RT1'].Id;
Account acct2 = (Account)Account.sObjectType.newSObject(rtId, true);
// Provide a value for Name
acct2.Name = 'Acme2';
// Insert new account
insert acct2;
```

Search Namespace

The `Search` namespace provides classes for getting search results and suggestion results.

The following are the classes in the `Search` namespace.

IN THIS SECTION:

KnowledgeSuggestionFilter Class

Filter settings that narrow the results from a call to `System.Search.suggest(searchQuery, sObjectType, options)` when the SOSL search query contains a KnowledgeArticleVersion object.

QuestionSuggestionFilter Class

The `Search.QuestionSuggestionFilter` class filters results from a call to `System.Search.suggest(searchQuery, sObjectType, options)` when the SOSL `searchQuery` contains a `FeedItem` object.

SearchResult Class

A wrapper object that contains an sObject and search metadata.

SearchResults Class

Wraps the results returned by the `Search.find(String)` method.

SuggestionOption Class

Options that narrow record and article suggestion results returned from a call to `System.Search.suggest(String, String, Search.SuggestionOption)`.

SuggestionResult Class

A wrapper object that contains an sObject.

SuggestionResults Class

Wraps the results returned by the `Search.suggest(String, String, Search.SuggestionOption)` method.

SEE ALSO:

find(searchQuery)

suggest(searchQuery, sObjectType, suggestions)

KnowledgeSuggestionFilter Class

Filter settings that narrow the results from a call to `System.Search.suggest(searchQuery, sObjectType, options)` when the SOSL search query contains a KnowledgeArticleVersion object.

Namespace

Search

KnowledgeSuggestionFilter Methods

The following are methods for `KnowledgeSuggestionFilter`.

IN THIS SECTION:

addArticleType(articleType)
Adds a filter that narrows suggestion results to display the specified article type. This filter is optional.

addDataCategory(dataCategoryGroupName, dataCategoryName)
Adds a filter that narrows suggestion results to display articles in the specified data category. This filter is optional.

addTopic(topic)
Specifies the article topic to return. This filter is optional.

setChannel(channelName)
Sets a channel to narrow the suggestion results to articles in the specified channel. This filter is optional.

setDataCategories(dataCategoryFilters)
Adds filters that narrow suggestion results to display articles in the specified data categories. Use this method to set multiple data category group and name pairs in one call. This filter is optional.

setLanguage(localeCode)
Sets a language to narrow the suggestion results to display articles in that language. This filter value is required in calls to `System.Search.suggest(String, String, Search.SuggestionOption)`.

setPublishStatus(publishStatus)
Sets a publish status to narrow the suggestion results to display articles with that status. This filter value is required in calls to `System.Search.suggest(String, String, Search.SuggestionOption)`.

setValidationStatus(validationStatus)
Sets a validation status to narrow the suggestion results to display articles with that status. This filter is optional.

`addArticleType(articleType)`

Adds a filter that narrows suggestion results to display the specified article type. This filter is optional.

Signature

```
public void addArticleType(String articleType)
```

Parameters

`articleType`
Type: String

A three-character ID prefix indicating the desired article type.

Return Value

Type: void

Usage

To add more than 1 article type, call the method multiple times.

addDataCategory(dataCategoryGroupName, dataCategoryName)

Adds a filter that narrows suggestion results to display articles in the specified data category. This filter is optional.

Signature

```
public void addDataCategory(String dataCategoryGroupName, String dataCategoryName)
```

Parameters

dataCategoryGroupName
Type: String

The name of the data category group

dataCategoryName
Type: String

The name of the data category.

Return Value

Type: void

Usage

To set multiple data categories, call the method multiple times. The name of the data category group and name of the data category for desired articles, expressed as a mapping, for example,
`Search.KnowledgeSuggestionFilter.addDataCategory('Regions', 'Asia')`.

addTopic(topic)

Specifies the article topic to return. This filter is optional.

Signature

```
public void addTopic(String topic)
```

Parameters

addTopic
Type: String

The name of the article topic.

Return Value

Type: void

Usage

To add more than 1 article topic, call the method multiple times.

setChannel(channelName)

Sets a channel to narrow the suggestion results to articles in the specified channel. This filter is optional.

Signature

```
public void setChannel(String channelName)
```

Parameters

channelName
Type: String

The name of a channel. Valid values are:

- `AllChannels`–Visible in all channels the user has access to
- `App`–Visible in the internal Salesforce Knowledge application
- `Pkb`–Visible in the public knowledge base
- `Csp`–Visible in the Customer Portal
- `Prm`–Visible in the Partner Portal

If `channel` isn't specified, the default value is determined by the type of user.

- `Pkb` for a guest user
- `Csp` for a Customer Portal user
- `Prm` for a Partner Portal user
- `App` for any other type of user

If `channel` is specified, the specified value may not be the actual value requested, because of certain requirements.

- For guest, Customer Portal, and Partner Portal users, the specified value must match the default value for each user type. If the values don't match or `AllChannels` is specified, then `App` replaces the specified value.
- For all users other than guest, Customer Portal, and Partner Portal users:
  - If `Pkb`, `Csp`, `Prm`, or `App` are specified, then the specified value is used.
  - If `AllChannels` is specified, then `App` replaces the specified value.

Return Value

Type: void

setDataCategories(dataCategoryFilters)

Adds filters that narrow suggestion results to display articles in the specified data categories. Use this method to set multiple data category group and name pairs in one call. This filter is optional.

Signature

```
public void setDataCategories(Map dataCategoryFilters)
```

Parameters

dataCategoryFilters
Type: Map

A map of data category group and data category name pairs.

Return Value

Type: void

setLanguage(localeCode)

Sets a language to narrow the suggestion results to display articles in that language. This filter value is required in calls to `System.Search.suggest(String, String, Search.SuggestionOption)`.

Signature

```
public void setLanguage(String localeCode)
```

Parameters

localeCode
Type: String

A locale code. For example, `'en_US'` (English–United States), or `'es'` (Spanish).

Return Value

Type: void

SEE ALSO:

Supported Locales

setPublishStatus(publishStatus)

Sets a publish status to narrow the suggestion results to display articles with that status. This filter value is required in calls to `System.Search.suggest(String, String, Search.SuggestionOption)`.

Signature

```
public void setPublishStatus(String publishStatus)
```

Parameters

publishStatus
Type: String

A publish status. Valid values are:

- `Draft`–Articles aren't published in Salesforce Knowledge.
- `Online`–Articles are published in Salesforce Knowledge.
- `Archived`–Articles aren't published and are available in Archived Articles view.

`setValidationStatus(validationStatus)`

Sets a validation status to narrow the suggestion results to display articles with that status. This filter is optional.

Signature

```
public void setValidationStatus(String validationStatus)
```

Parameters

`validationStatus`
Type: String

An article validation status. These values are available in the `ValidationStatus` field on the KnowledgeArticleVersion object.

Return Value

Type: void

QuestionSuggestionFilter Class

The `Search.QuestionSuggestionFilter` class filters results from a call to `System.Search.suggest(searchQuery, sObjectType, options)` when the SOSL `searchQuery` contains a `FeedItem` object.

Namespace

Search

IN THIS SECTION:

QuestionSuggestionFilter Methods

QuestionSuggestionFilter Methods

The following are methods for `QuestionSuggestionFilter`.

IN THIS SECTION:

addGroupId(groupId)
Adds a filter to display questions associated with the single specified group whose ID is passed in as an argument. This filter is optional.

addNetworkId(networkId)
Adds a filter to display questions associated with the single specified network whose ID is passed in as an argument. This filter is optional.

addUserId(userId)

Adds a filter to display questions belonging to the single specified user whose ID is passed in as an argument. This filter is optional.

setGroupIds(groupIds)

Sets a new list of groups to replace the current list of groups where the group IDs are passed in as an argument. This filter is optional.

setNetworkIds(networkIds)

Sets a new list of networks to replace the current list of networks where the network IDs are passed in as an argument. This filter is optional.

setTopicId(topicId)

Sets a filter to display questions associated with the single specified topic whose ID is passed in as an argument. This filter is optional.

setUserIds(userIds)

Sets a new list of users to replace the current list of users where the users IDs are passed in as an argument. This filter is optional.

addGroupId(groupId)

Adds a filter to display questions associated with the single specified group whose ID is passed in as an argument. This filter is optional.

Signature

```
public void addGroupId(String groupId)
```

Parameters

groupId

Type: String

The ID for a group.

Return Value

Type: void

Usage

To add more than one group, call the method multiple times.

addNetworkId(networkId)

Adds a filter to display questions associated with the single specified network whose ID is passed in as an argument. This filter is optional.

Signature

```
public void addNetworkId(String networkId)
```

Parameters

networkId

Type: String

The ID of the Experience Cloud site about which you're retrieving this information.

Return Value

Type: void

Usage

To add more than one network, call the method multiple times.

addUserId(userId)

Adds a filter to display questions belonging to the single specified user whose ID is passed in as an argument. This filter is optional.

Signature

```
public void addUserId(String userId)
```

Parameters

userId
Type: String

The ID for the user.

Return Value

Type: void

Usage

To add more than one user, call the method multiple times.

setGroupIds(groupIds)

Sets a new list of groups to replace the current list of groups where the group IDs are passed in as an argument. This filter is optional.

Signature

```
public void setGroupIds(List<String> groupIds)
```

Parameters

groupIds
Type: List<String>

A list of group IDs.

Return Value

Type: void

setNetworkIds(networkIds)

Sets a new list of networks to replace the current list of networks where the network IDs are passed in as an argument. This filter is optional.

Signature

```
public void setNetworkIds(List<String> networkIds)
```

Parameters

networkIds
Type: List<String>

A list of network IDs.

Return Value

Type: void

setTopicId(topicId)

Sets a filter to display questions associated with the single specified topic whose ID is passed in as an argument. This filter is optional.

Signature

```
public void setTopicId(String topicId)
```

Parameters

topicId
Type: String

The ID for a topic.

Return Value

Type: void

setUserIds(userIds)

Sets a new list of users to replace the current list of users where the users IDs are passed in as an argument. This filter is optional.

Signature

```
public void setUserIds(List<String> userIds)
```

Parameters

userIds
Type: List<String>

A list of user IDs.

Return Value

Type: void

SearchResult Class

A wrapper object that contains an sObject and search metadata.

Namespace

Search

SearchResult Methods

The following are methods for `SearchResult`.

IN THIS SECTION:

getSObject()
Returns an sObject from a SearchResult object.

getSnippet(fieldName)
Returns a snippet from a Case, Feed, or Knowledge Article SearchResult object based on the specified field name.

getSnippet()
Returns a snippet from a SearchResult object based on the default field.

`getSObject()`

Returns an sObject from a SearchResult object.

Signature

```
public SObject getSObject()
```

Return Value

Type: SObject

SEE ALSO:

find(searchQuery)

Apex Developer Guide: Dynamic SOSL

`getSnippet(fieldName)`

Returns a snippet from a Case, Feed, or Knowledge Article SearchResult object based on the specified field name.

Signature

```
public String getSnippet(String fieldName)
```

Parameters

fieldName
Type: String

The field name to use for creating the snippet.

Valid values: `Case.Casenumber`, `FeedPost.Title`, `KnowledgeArticleVersion.Title`

Return Value

Type: String

SEE ALSO:

find(searchQuery)

Apex Developer Guide: Dynamic SOSL

`getSnippet()`

Returns a snippet from a SearchResult object based on the default field.

Signature

```
public String getSnippet()
```

Return Value

Type: String

SEE ALSO:

find(searchQuery)

Apex Developer Guide: Dynamic SOSL

SearchResults Class

Wraps the results returned by the `Search.find(String)` method.

Namespace

Search

SearchResults Methods

The following are methods for `SearchResults`.

IN THIS SECTION:

get(sObjectType)
Returns a list of `Search.SearchResult` objects that contain an sObject of the specified type.

`get(sObjectType)`

Returns a list of `Search.SearchResult` objects that contain an sObject of the specified type.

Signature

```
public List<Search.SearchResult> get(String sObjectType)
```

Parameters

`sObjectType`
Type: String

The name of an sObject in the dynamic SOSL query passed to the `Search.find(String)` method.

Return Value

Type: List<Search.SearchResult>

Usage

SOSL queries passed to the `Search.find(String)` method can return results for multiple objects. For example, the query `Search.find('FIND \'map\' IN ALL FIELDS RETURNING Account, Contact, Opportunity')` includes results for 3 objects. You can call `get(string)` to retrieve search results for 1 object at a time. For example, to get results for the Account object, call `Search.SearchResults.get('Account')`.

SEE ALSO:

find(searchQuery)

SearchResult Methods

Apex Developer Guide: Dynamic SOSL

SuggestionOption Class

Options that narrow record and article suggestion results returned from a call to `System.Search.suggest(String, String, Search.SuggestionOption)`.

Namespace

Search

SuggestionOption Methods

The following are methods for `SuggestionOption`.

IN THIS SECTION:

setFilter(knowledgeSuggestionFilter)
Set filters that narrow Salesforce Knowledge article results in a call to `System.Search.suggest(String, String, Search.SuggestionOption)`.

setLimit(limit)

The maximum number of record or article suggestions to retrieve.

setFilter(knowledgeSuggestionFilter)

Set filters that narrow Salesforce Knowledge article results in a call to `System.Search.suggest(String, String, Search.SuggestionOption)`.

Signature

```
public void setFilter(Search.KnowledegeSuggestionFilter knowledgeSuggestionFilter)
```

Parameters

knowledgeSuggestionFilter

Type: KnowledgeSuggestionFilter

An object containing filters that narrow the search results.

Return Value

Type: void

Usage

```
Search.KnowledgeSuggestionFilter filters = new Search.KnowledgeSuggestionFilter();
filters.setLanguage('en_US');
filters.setPublishStatus('Online');
filters.setChannel('app');

Search.SuggestionOption options = new Search.SuggestionOption();
options.setFilter(filters);

Search.SuggestionResults suggestionResults = Search.suggest('all', 'KnowledgeArticleVersion',
 options);

for (Search.SuggestionResult searchResult : suggestionResults.getSuggestionResults()) {

  KnowledgeArticleVersion article = (KnowledgeArticleVersion)searchResult.getSObject();

  System.debug(article.title);
}
```

setLimit(limit)

The maximum number of record or article suggestions to retrieve.

Signature

```
public void setLimit(Integer limit)
```

Parameters

limit
Type: Integer

The maximum number of record or article suggestions to retrieve.

Return Value

Type: void

Usage

By default, the `System.Search.suggest(String, String, Search.SuggestionOption)` method returns the 5 most relevant results. However, if your query is broad, it could match more than 5 results. If `Search.SuggestionResults.hasMoreResults()` returns `true`, there are more than 5 results. To retrieve them, call `setLimit(Integer)` to increase the number of suggestions results.

```
Search.SuggestionOption option = new Search.SuggestionOption();
option.setLimit(10);
Search.suggest('my query', 'mySObjectType', option);
```

SuggestionResult Class

A wrapper object that contains an sObject.

Namespace

Search

SuggestionResult Methods

The following are methods for `SuggestionResult`.

IN THIS SECTION:

getSObject()
Returns the sObject from a SuggestionResult object.

`getSObject()`

Returns the sObject from a SuggestionResult object.

Signature

```
public SObject getSObject()
```

Return Value

Type: SObject

SuggestionResults Class

Wraps the results returned by the `Search.suggest(String, String, Search.SuggestionOption)` method.

Namespace

Search

SuggestionResults Methods

The following are methods for `SuggestionResults`.

IN THIS SECTION:

getSuggestionResults()
Returns a list of SuggestionResult objects from the response to a call to `Search.suggest(String, String, Search.SuggestionOption)`.

hasMoreResults()
Indicates whether a call to `System.Search.suggest(String, String, Search.SuggestionOption)` has more results available than were returned.

`getSuggestionResults()`

Returns a list of SuggestionResult objects from the response to a call to `Search.suggest(String, String, Search.SuggestionOption)`.

Signature

```
public List<Search.SuggestionResult> getSuggestionResults()
```

Return Value

Type: List<SuggestionResult>

`hasMoreResults()`

Indicates whether a call to `System.Search.suggest(String, String, Search.SuggestionOption)` has more results available than were returned.

Signature

```
public Boolean hasMoreResults()
```

Return Value

Type: Boolean

Usage

If a limit isn't specified, 5 records are returned in calls to `System.Search.suggest(String, String, Search.SuggestionOption)`. If there are more suggested records than the limit specified, a call to `hasMoreResults()` returns `true`.

Sfc Namespace

The Sfc namespace contains classes used in Salesforce Files.

The following are the classes in the `Sfc` namespace.

IN THIS SECTION:

ContentDownloadContext Enum
This enum specifies the download context.

ContentDownloadHandler Class
Use ContentDownloadHandler to define a custom download handler that controls how content is downloaded.

ContentDownloadHandlerFactory Interface
Use this interface to provide a class factory that Salesforce can call to create instances of your custom ContentDownloadHandler.

ContentDownloadContext Enum

This enum specifies the download context.

Usage

If the operationContext is `CONTENT`, `CHATTER`, `DELIVERY`, `S1`, or `MOBILE`, it can be used in a shepherd servlet as a query parameter. It's possible for a user to change the query parameters. If a user enters a value other than `CONTENT`, `CHATTER`, `DELIVERY`, `S1`, or `MOBILE`, the value is treated as the default value `CONTENT`.

Users can't set query parameters to `REST_API`, `SOQL`, or `RETRIEVE`, so these values can be assumed to be accurate.

Enum Values

The Sfc.ContentDownloadContext enum value identifies the content download context. The enum value is provided as a query parameter in the file download servlet. The following are the values of the `Sfc.ContentDownloadContext` enum.

Value	Description
`CHATTER`	Download from Chatter.
`CONTENT`	Default value. Downloads from the Salesforce CRM Content product.
`DELIVERY`	Download of a content delivery.
`REST_API`	Download from the Connect API (`/connect/files/${fileId}/content` endpoint). Used in both Android and iOS apps.
`RETRIEVE`	Retrieve VersionData from SObject API.

Value	Description
`S1`	Download from Lightning Experience.
`SOQL`	Select VersionData from SOQL.

ContentDownloadHandler Class

Use ContentDownloadHandler to define a custom download handler that controls how content is downloaded.

Namespace

Sfc on page 3183

IN THIS SECTION:

ContentDownloadHandler Properties

ContentDownloadHandler Properties

The following are properties for `ContentDownloadHandler`.

IN THIS SECTION:

downloadErrorMessage
A customized error message explaining why the download isn't allowed.

isDownloadAllowed
Indicates whether or not download is allowed.

redirectUrl
The URL the user is redirected to when the download action isn't available, for applying Information Rights Management (IRM) control, virus scanning, or other behavior.

`downloadErrorMessage`

A customized error message explaining why the download isn't allowed.

Signature

```
public String downloadErrorMessage {get; set;}
```

Property Value

Type: String

This message is used if a `redirectUrl` is not provided. If the download is not allowed, Salesforce will throw a `ContentCustomizedDownloadException` exception that contains the `downloadErrorMessage`.

`isDownloadAllowed`

Indicates whether or not download is allowed.

Signature

```
public Boolean isDownloadAllowed {get; set;}
```

Property Value

Type: Boolean

`redirectUrl`

The URL the user is redirected to when the download action isn't available, for applying Information Rights Management (IRM) control, virus scanning, or other behavior.

Signature

```
public String redirectUrl {get; set;}
```

Property Value

Type: String

The URL must be a valid relative URL. For example, the redirect can be a custom Visualforce page such as "/apex/IRMControl". URLs with no path, such as "www.domain.com", results in an `InvalidParameterValueException`.

ContentDownloadHandlerFactory Interface

Use this interface to provide a class factory that Salesforce can call to create instances of your custom ContentDownloadHandler.

Namespace

Sfc on page 3183

Usage

ContentDownloadHandler getContentDownloadHandler(List<ID> ids, ContentDownloadContext context);

IN THIS SECTION:

ContentDownloadHandlerFactory Methods

ContentDownloadHandlerFactory Example Implementation

ContentDownloadHandlerFactory Methods

The following are methods for `ContentDownloadHandlerFactory`.

IN THIS SECTION:

getContentDownloadHandler(var1, var2)
Returns a ContentDownloadHandler for a given list of content IDs and a download context.

getContentDownloadHandler(var1, var2)

Returns a ContentDownloadHandler for a given list of content IDs and a download context.

Signature

```
public Sfc.ContentDownloadHandler getContentDownloadHandler(List<Id> var1,
Sfc.ContentDownloadContext var2)
```

Parameters

var1
Type: List<Id>

var2
Type: Sfc.ContentDownloadContext on page 3183

Return Value

Type: Sfc.ContentDownloadHandler on page 3184

ContentDownloadHandlerFactory Example Implementation

This example creates a class that implements the `Sfc.ContentDownloadHandlerFactory` interface and returns a download handler that blocks downloading content to mobile devices.

```
// Allow customization of the content Download experience
public class ContentDownloadHandlerFactoryImpl implements Sfc.ContentDownloadHandlerFactory
 {

  public Sfc.ContentDownloadHandler getContentDownloadHandler(List<ID> ids,
Sfc.ContentDownloadContext context) {
    Sfc.ContentDownloadHandler contentDownloadHandler = new Sfc.ContentDownloadHandler();


    if(context == Sfc.ContentDownloadContext.MOBILE) {
      contentDownloadHandler.isDownloadAllowed = false;
      contentDownloadHandler.downloadErrorMessage = 'Downloading a file from a mobile
device is not allowed.';
      return contentDownloadHandler;
    }
    contentDownloadHandler.isDownloadAllowed = true;
    return contentDownloadHandler;
  }
}
```

Sfdc_Checkout Namespace

The Sfdc_Checkout namespace provides an interface and classes for B2B Commerce apps in Salesforce.

The following are the classes in the `Sfdc_Checkout` namespace.

IN THIS SECTION:

AsyncCartProcessor Interface
Use this interface to implement asynchronous integrations in B2B Commerce.

B2BCheckoutController Class
Communicate with simple checkout Apex methods to work with data related to B2B Commerce checkout.

IntegrationInfo Class
Provides the values that B2B Commerce Checkout uses to map requests to responses, necessary metadata, and context.

IntegrationStatus Class
Supports synchronous execution of Apex integrations for B2B Commerce. The implementation must return the status of the execution.

IntegrationStatus.Status Enum
The IntegrationStatus.Status enum describes the status of the current integration.

AsyncCartProcessor Interface

Use this interface to implement asynchronous integrations in B2B Commerce.

Namespace

Sfdc_Checkout

IN THIS SECTION:

AsyncCartProcessor Methods

AsyncCartProcessor Example Implementation

AsyncCartProcessor Methods

The following are methods for `AsyncCartProcessor`.

IN THIS SECTION:

startCartProcessAsync(integrationInfo, cartId)
The startCartProcessAsync method is called asynchronously by the integration framework. Calling this method begins cart processing for Commerce checkout.

`startCartProcessAsync(integrationInfo, cartId)`

The startCartProcessAsync method is called asynchronously by the integration framework. Calling this method begins cart processing for Commerce checkout.

Signature

```
public sfdc_checkout.IntegrationStatus
startCartProcessAsync(sfdc_checkout.IntegrationInfo integrationInfo, Id cartId)
```

Parameters

integrationInfo
Type: IntegrationInfo

Provides values that B2B Commerce checkout APIs use to map requests to responses, necessary metadata, and context.

cartId
Type: Id

ID of the WebCart object.

Return Value

Type: IntegrationStatus

Status of the current integration. Possible values are `SUCCESS` and `FAILED`.

AsyncCartProcessor Example Implementation

This is an example implementation of the `sfdc_checkout.AsyncCartProcessor` interface.

```
global interface checkout_AsyncCartProcessor {

   //Integration for async processing
   IntegrationStatus startCartProcessAsync(
      IntegrationInfo integrationInfo,
      Id cartId);
}
```

AsyncCartProcessor is a base interface. There are four interfaces that extend it, including CartInventoryValidation, CartPriceCalculations, CartShippingCharges, and CartTaxCalculations. For more information about these interfaces, including code examples and test classes, see Checkout Integrations.

B2BCheckoutController Class

Communicate with simple checkout Apex methods to work with data related to B2B Commerce checkout.

Namespace

sfdc_checkout

Usage

You must specify the `sfdc_checkout` namespace when creating an instance of this class.

IN THIS SECTION:

B2BCheckoutController Methods

B2BCheckoutController Methods

The following are methods for `B2BCheckoutController`.

IN THIS SECTION:

licenseCompliance(cartId, orderId)
If you implement your own cart-to-order process without invoking the Cart to Order flow core action, you must invoke this method to correctly track your orders for GMV (Gross Merchandise Value) recognition.

`licenseCompliance(cartId, orderId)`

If you implement your own cart-to-order process without invoking the Cart to Order flow core action, you must invoke this method to correctly track your orders for GMV (Gross Merchandise Value) recognition.

Signature

```
public static void licenseCompliance(String cartId, String orderId)
```

Parameters

`cartId`
Type: String

The `cartId` of a web cart from which an order is created.

`orderId`
Type: String

The `orderId` of the order you created from the cart.

Return Value

Type: Void

IntegrationInfo Class

Provides the values that B2B Commerce Checkout uses to map requests to responses, necessary metadata, and context.

Namespace

sfdc_checkout on page 3187

Usage

This class provides information about a B2B Commerce integration. An instance of this class is passed as a parameter into the integration interface.

IN THIS SECTION:

IntegrationInfo Properties

IntegrationInfo Properties

The following are properties for `IntegrationInfo`.

IN THIS SECTION:

integrationId
The unique ID of a B2B Commerce integration.

jobId
The ID of the job, specific to the Salesforce Background Operation framework.

siteLanguage
Site language to be used by third party services.

integrationId

The unique ID of a B2B Commerce integration.

Signature

```
public String integrationId {get; set;}
```

Property Value

Type: String

jobId

The ID of the job, specific to the Salesforce Background Operation framework.

Signature

```
public String jobId {get; set;}
```

Property Value

Type: String

siteLanguage

Site language to be used by third party services.

Signature

```
public String siteLanguage {get; set;}
```

Property Value

Type: String

IntegrationStatus Class

Supports synchronous execution of Apex integrations for B2B Commerce. The implementation must return the status of the execution.

Namespace

sfdc_checkout

Usage

You must specify the `sfdc_checkout` namespace when creating an instance of this class.

IN THIS SECTION:

IntegrationStatus Properties

IntegrationStatus Properties

The following are properties for `IntegrationStatus`.

IN THIS SECTION:

status
Indicates the status of the integration process and whether or not it completed successfully.

`status`

Indicates the status of the integration process and whether or not it completed successfully.

Signature

```
public sfdc_checkout.IntegrationStatus.Status status {get; set;}
```

Property Value

Type: sfdc_checkout.IntegrationStatus.Status on page 3191

IntegrationStatus.Status Enum

The IntegrationStatus.Status enum describes the status of the current integration.

Enum Values

The following are the values of the `sfdc_checkout.IntegrationStatus.Status` enum.

Value	Description
`FAILED`	Indicates transient, unknown error, managed by the implementor. The buyer can retry this action.

Value	Description
SUCCESS	Indicates the integration executed successfully.

Sfdc_Enablement Namespace

The `sfdc_enablement` namespace provides classes for creating custom learning items to implement custom exercise types in Enablement programs. Lightning web components are used to render the custom exercises on Program Builder.

The following are the classes in the `sfdc_enablement` namespace.

IN THIS SECTION:

LearningEvaluation Class
Contains methods to retrieve and update details that are required to evaluate a learning item.

LearningEvaluationResult Class
Represents a user's progress and progress status of a custom exercise in an Enablement program.

LearningItemEvaluationHandler Class
Contains methods to customize the evaluation process of a learning item.

LearningItemProgressStatus Enum
Represents the status of a user's progress for a learning item in an Enablement program.

LearningItemSerializeDeserializer Class
Serializes and deserializes the content associated with a custom exercise when migrating an Enablement program from one org to another.

LearningEvaluation Class

Contains methods to retrieve and update details that are required to evaluate a learning item.

Namespace

sfdc_enablement

Usage

Pass this class as input to the sfdc_enablement.LearningEvaluationResult class.

Example

See example code in sfdc_enablement.LearningItemEvaluationHandler on page 3196.

IN THIS SECTION:

LearningEvaluation Methods

LearningEvaluation Methods

The following are methods for `LearningEvaluation`.

IN THIS SECTION:

getDetails()
Retrieves the details associated with the learning evaluation instance.

getLearningItemId()
Retrieves the record ID of the learning item that's associated with this learning evaluation instance.

setDetails(details)
Sets or updates the details of the learning item record for this learning evaluation instance.

setLearningItemId(learningItemId)
Sets or updates the learning item record ID for this learning evaluation instance.

`getDetails()`

Retrieves the details associated with the learning evaluation instance.

Signature

```
public Map<String,Object> getDetails()
```

Return Value

Type: Map on page 3624<String,Object on page 3691>

`getLearningItemId()`

Retrieves the record ID of the learning item that's associated with this learning evaluation instance.

Signature

```
public String getLearningItemId()
```

Return Value

Type: String

`setDetails(details)`

Sets or updates the details of the learning item record for this learning evaluation instance.

Signature

```
public void setDetails(Map<String,Object> details)
```

Parameters

details
Type: Map<String,Object>

The details of the learning item record that you get by calling evaluateLearningItem API.

Return Value

Type: void

setLearningItemId(learningItemId)

Sets or updates the learning item record ID for this learning evaluation instance.

Signature

```
public void setLearningItemId(String learningItemId)
```

Parameters

learningItemId
Type: String

Return Value

Type: void

LearningEvaluationResult Class

Represents a user's progress and progress status of a custom exercise in an Enablement program.

Namespace

sfdc_enablement

Usage

To calculate the user's progress through an exercise as a percentage and return the progress status, use the `sfdc_enablement.LearningEvaluationResult` class inside the sfdc_enablement.LearningItemEvaluationHandler. In your custom code, set the percentages to correspond to these sfdc_enablement.LearningItemProgressStatus on page 3198 enum values.

- `NotStarted` is equal to 0.00
- `InProgress` is from 0.01 through 99.99
- `Completed` is equal to 100.00

Example

See example code in sfdc_enablement.LearningItemEvaluationHandler on page 3196.

IN THIS SECTION:

LearningEvaluationResult Methods

LearningEvaluationResult Methods

The following are methods for `LearningEvaluationResult`.

IN THIS SECTION:

getLearningItemProgress()
Returns the progress percentage of the learning item.

getLearningItemProgressStatus()
Retrieves the progress status of the learning item.

setLearningItemProgress(learningItemProgress)
Sets the progress percentage of the learning item.

setLearningItemProgressStatus(learningItemProgressStatus)
Sets the progress status of the learning item.

`getLearningItemProgress()`

Returns the progress percentage of the learning item.

Signature

```
public Double getLearningItemProgress()
```

Return Value

Type: Double

The progress percentage is formatted to two decimal places.

`getLearningItemProgressStatus()`

Retrieves the progress status of the learning item.

Signature

```
public sfdc_enablement.LearningItemProgressStatus getLearningItemProgressStatus()
```

Return Value

Type: sfdc_enablement.LearningItemProgressStatus on page 3198

`setLearningItemProgress(learningItemProgress)`

Sets the progress percentage of the learning item.

Signature

```
public void setLearningItemProgress(Double learningItemProgress)
```

Parameters

learningItemProgress
Type: Double

The progress in percentage formatted to two decimal places.

Return Value

Type: void

setLearningItemProgressStatus(learningItemProgressStatus)

Sets the progress status of the learning item.

Signature

```
public void setLearningItemProgressStatus(sfdc_enablement.LearningItemProgressStatus
learningItemProgressStatus)
```

Parameters

learningItemProgressStatus
Type: Sfdc_enablement.LearningItemProgressStatus on page 3198

Return Value

Type: void

LearningItemEvaluationHandler Class

Contains methods to customize the evaluation process of a learning item.

Namespace

sfdc_enablement

Usage

Extend this class and implement your custom progress evaluation method. Then link this class to a LearningItemType metadata record by passing the Apex class name to the `ApexEvaluationHandler` field.

Example

This code updates a user's progress when they take a custom screen flow exercise in an Enablement program. The code updates the progress by checking the number of screens the user has navigated, calculating the progress percentage, and returning the progress status. See Track a User's Progress in a Custom Exercise from *Salesforce Developer Guide*: Sales Programs and Partner Tracks with Enablement.

```
global class ScreenFlowEvaluationHandler extends
sfdc_enablement.LearningItemEvaluationHandler {
    global override sfdc_enablement.LearningEvaluationResult
evaluate(sfdc_enablement.LearningEvaluation learningEvaluation) {
        sfdc_enablement.LearningEvaluationResult result = new
sfdc_enablement.LearningEvaluationResult();
        Double percentage = 100.0d;

        Map<String, Object> details = learningEvaluation.getDetails();

        String currentScreen = (String) details.get('currentScreen');

        String allScreensString = (String) details.get('allScreens');
        List<String> allScreens = allScreensString.split(',');

        String status = (String) details.get('status');
        if (status == 'FINISHED') {
            percentage = 100;
        } else {
            Integer index = 0;
            for (Integer i = 0; i < allScreens.size(); i++) {
                if (allScreens.get(i).equals(currentScreen)) {
                    index = i + 1;
                    break;
                }
            }

            if (index == allScreens.size()) {
                percentage = 99.0d;
            } else {
                percentage = (Double.valueOf(index) / Double.valueOf(allScreens.size()))
* 100.0d;
            }
        }

        result.setLearningItemProgress(percentage);

        if (percentage == 100.0d) {

result.setLearningItemProgressStatus(sfdc_enablement.LearningItemProgressStatus.Completed);

        } else if (percentage == 0.0d) {

result.setLearningItemProgressStatus(sfdc_enablement.LearningItemProgressStatus.NotStarted);

        } else {

result.setLearningItemProgressStatus(sfdc_enablement.LearningItemProgressStatus.InProgress);
```

```
        }

        return result;
    }
}
```

IN THIS SECTION:

LearningItemEvaluationHandler Methods

LearningItemEvaluationHandler Methods

The following are methods for `LearningItemEvaluationHandler`.

IN THIS SECTION:

evaluate(learningEvaluation)
Contains the custom logic for evaluating a learning item.

`evaluate(learningEvaluation)`

Contains the custom logic for evaluating a learning item.

Signature

```
public Sfdc_enablement.LearningEvaluationResult
evaluate(Sfdc_enablement.LearningEvaluation learningEvaluation)
```

Parameters

`learningEvaluation`
Type: Sfdc_enablement.LearningEvaluation on page 3192

The details of the learning item record to be evaluated.

Return Value

Type: Sfdc_enablement.LearningEvaluationResult on page 3194

The result of the evaluation, including progress and status details.

LearningItemProgressStatus Enum

Represents the status of a user's progress for a learning item in an Enablement program.

Usage

To set the progress status in the sfdc_enablement.LearningEvaluationResult on page 3194 class, use this enum.

Enum Values

The following are the values for the `sfdc_enablement.LearningItemProgressStatus` enum.

Value	Description
`NotStarted`	The user hasn't started the custom exercise.
`InProgress`	The user's custom exercise is in progress.
`Completed`	The user completed the custom exercise.

LearningItemSerializeDeserializer Class

Serializes and deserializes the content associated with a custom exercise when migrating an Enablement program from one org to another.

Namespace

sfdc_enablement

Usage

The class contains methods to serialize and deserialize custom exercise content between orgs when an Enablement program that includes a custom exercise is migrated from one org to another through change sets or packaging.

Extend the `sfdc_enablement.LearningItemSerializeDeserializer` Apex abstract class and add the class name to the `ApexSerializerDeserializer` field of the LearningItemType metadata record. If you don't add the class name to the LearningItemType metadata record, the `customContent` property for the custom exercise is empty in the destination org and no corresponding LearningItem record is created for the exercise's EnblProgramTaskDefinition record.

The serialize on page 3201 method serializes the custom content of the learning item from the source org. This method is called when you retrieve custom content from the source org.

The deserialize on page 3201 method is called during the deployment of a program. This method takes the serialized custom content, recreates the custom object record in the target org, and returns a new learning item record ID.

Example

The sample code serializes and deserializes the custom content for a given learning item of a custom screen flow exercise in an Enablement program. For this example to work, make sure the screen flow exists in the target org.

```
global class ScreenFlowSerializerDeserializer extends
Sfdc_enablement.LearningItemSerializeDeserializer {
    // The serialize method returns the serialized output of the
    // learning item's custom content.
    global override String serialize(String learningItemId) {
        // Get the screen flow record ID associated with the learning item.
        LearningItem learningItem = [SELECT ScreenFlow_Field__c from LearningItem where
Id =: learningItemId LIMIT 1];
        String screenFlowRecordId = learningItem.ScreenFlow_Field__c;

        // Get the flow version ID associated with that screen flow.
```

```
        ScreenFlow_Object__c screenFlowRecord = [SELECT FlowVersionId__c from
ScreenFlow_Object__c where Id =: screenFlowRecordId LIMIT 1];
        String flowVersionId = screenFlowRecord.FlowVersionId__c;

        // Query the flow definition associated with that flow version.
        // Get the information you need to recreate the custom object
        // record in the destination org.
        // In this example, we're only getting the API name of the
        // flow version.
        FlowDefinitionView flowDefinitionView = [SELECT ApiName from FlowDefinitionView
where ActiveVersionId =: flowVersionId LIMIT 1];

        // Return the serialized string.
        // In this example, we're only returning the API name of the flow
        // definition in the string.
        return flowDefinitionView.ApiName;
    }

    // The deserialize method deserializes the string containing the custom
    // content. In the method, you recreate the custom object record
    // for the destination org and populate it with the custom content.
    // Then insert the record in the destination org and return the new
    // custom object record ID.
    global override String deserialize(String serializedOutput) {
        // Find the flow active version ID of the same screen flow in the
        // destination org.
        FlowDefinitionView flowDefinitionView = [SELECT ActiveVersionId from
FlowDefinitionView where ApiName =: serializedOutput LIMIT 1];
        String flowActiveVersionId = flowDefinitionView.ActiveVersionId;

        // Create the screen flow custom object record using the
        // information you passed to the string in the serialize method.
        // In this example, we only passed the API name of the screen flow
        // to the string.
        ScreenFlow_Object__c screenFlowRecord = new ScreenFlow_Object__c();
        screenFlowRecord.Name = serializedOutput;
        screenFlowRecord.FlowVersionId__c = flowActiveVersionId;

        // Insert the custom object record into the destination org.
        insert screenFlowRecord;

        // Return the new screen flow record ID for the new learning item
        // in the destination org.
        return screenFlowRecord.Id;
    }
}
```

IN THIS SECTION:

LearningItemSerializeDeserializer Methods

LearningItemSerializeDeserializer Methods

The following are methods for `LearningItemSerializeDeserializer`.

IN THIS SECTION:

deserialize(serializedOutput)

Deserializes the provided custom content string and returns the record ID of the learning item.

serialize(learningItemId)

Serializes the custom content associated with the specified learning item. The serialized string represents the metadata of the custom content and is used to recreate the custom content in the target Salesforce org during deployment.

`deserialize(serializedOutput)`

Deserializes the provided custom content string and returns the record ID of the learning item.

Signature

```
public String deserialize(String serializedOutput)
```

Parameters

`serializedOutput`
Type: String

The serialized information of custom content associated with a learning item The serialize(learningItemId) on page 3201 method returns this information as a string that is less than or equal to 250 characters.

Return Value

Type: String

The ID of the learning item created for the target org.

`serialize(learningItemId)`

Serializes the custom content associated with the specified learning item. The serialized string represents the metadata of the custom content and is used to recreate the custom content in the target Salesforce org during deployment.

Signature

```
public String serialize(String learningItemId)
```

Parameters

`learningItemId`
Type: String

The ID of the learning item associated with the custom content to be serialized.

Return Value

Type: String

The serialized information of the custom content of the specified learning item. The format is a string that's less than or equal to 250 characters long.

sfdc_surveys Namespace

The `sfdc_surveys` namespace provides an interface for shortening survey invitations.

The following are the classes in the `sfdc_surveys` namespace.

IN THIS SECTION:

SurveyInvitationLinkShortener Interface
Use this interface to provide a class factory that Salesforce can call to create instances of your custom `SurveyInvitationLinkShortener`.

Example Implementation to Associate SurveySubjects with SurveyInvitation and SurveyResponses
If no survey responses are populated, create a custom code to associate SurveySubjects with SurveyInvitation and SurveyResponses.

SurveyInvitationLinkShortener Interface

Use this interface to provide a class factory that Salesforce can call to create instances of your custom `SurveyInvitationLinkShortener`.

Namespace

sfdc_surveys

Usage

Implement an instance of the `SurveyInvitationLinkShortener` interface to shorten the survey invitation that can be distributed as short URLs over customer engaged channels, such as SMS, WhatsApp, or Facebook Messenger.

Special access rules

To implement this interface, you must have the Salesforce Feedback Management license enabled in your Salesforce organization.

IN THIS SECTION:

SurveyInvitationLinkShortener Methods

SurveyInvitationLinkShortener Example Implementation

SurveyInvitationLinkShortener Methods

The following are methods for `SurveyInvitationLinkShortener`.

IN THIS SECTION:

getShortenedURL(var1)
Returns a shortened URL for a given survey invitation.

getShortenedURL(var1)

Returns a shortened URL for a given survey invitation.

Signature

```
public String getShortenedURL(String var1)
```

Parameters

var1
Type: String

Return Value

Type: String

SurveyInvitationLinkShortener Example Implementation

This is an example implementation of the `sfdc_surveys.SurveyInvitationLinkShortener` interface.

This sample code uses Named Credentials for authentication. For more information on Named Credentials, see Named Credentials as Callout Endpoints.

```
public class SurveyInvitationLinkShortenerImpl implements
sfdc_surveys.SurveyInvitationLinkShortener {
   public String getShortenedURL(String invitationURL) {
      return shortenUrlUsingBitlyService(invitationURL);
   }
   public String shortenUrlUsingBitlyService(String invitationURL) {
      HttpRequest request = new HttpRequest();
      request.setEndpoint('callout:bitly/v4/shorten');
      request.setMethod('POST');
      request.setHeader('Authorization', 'Bearer {!$Credential.Password}');
      request.setHeader('Accept', 'application/json');
      request.setHeader('Content-Type', 'application/json');
      request.setBody(JSON.serialize(new Map<String, Object>{
      'group_guid' => '{!$Credential.UserName}',
      'long_url' => invitationURL
      }));

      Http http = new Http();
      HttpResponse res = http.send(request);

      Object result = JSON.deserializeUntyped(res.getBody());
      if (result instanceof Map<String, Object>) {
         Map<String, Object> resultMap = (Map<String, Object>) result;
```

```
         Object shortenedLinkVal = resultMap.get('link');
         if(shortenedLinkVal != null && shortenedLinkVal instanceof String) {
            return (String) shortenedLinkVal;
         }
      }
      return invitationURL;
   }
}
```

Example Implementation to Associate SurveySubjects with SurveyInvitation and SurveyResponses

If no survey responses are populated, create a custom code to associate SurveySubjects with SurveyInvitation and SurveyResponses.

This example shows how to associate SurveySubjects with SurveyInvitation and SurveyResponses.

```
public class CreateEntriesInSurveyInvitationRespRL {
    // Utility to create SurveyInvitation and SurveySubject record
    public static void addEntry(String associatedRecordId, String surveyId, String
participantId) {
        String invitationId = createSurveyInvitation(surveyId, participantId);

        createSurveySubject(invitationId, associatedRecordId);
    }

    // Create an unauthenticated invitation by setting the surveyId and participantId
    private static String createSurveyInvitation(String surveyId, String participantId) {

        SurveyInvitation surveyInv = new SurveyInvitation();
        surveyInv.Name = 'SurveyInvitationForCase'; // add your survey invitation name
here
        surveyInv.ParticipantId = participantId;
        surveyInv.CommunityId = '0DBRM0000004n4y'; //add your community id here
        surveyInv.OptionsAllowGuestUserResponse = true;
        surveyInv.SurveyId = surveyId;

        // Insert the SurveyInvitation Record
        insert surveyInv;

        return surveyInv.Id;
    }

    // Associate the above invitation to the required record (eg: Case, Opportunity...)
   private static void createSurveySubject(String invitationId, String associatedRecordId)
 {
        SurveySubject subj = new SurveySubject();
        subj.Name = 'Sur_Subject_for_invitation';
       subj.ParentId = invitationId; // similary you can use survey response id to associate
 survey subject to a response record.
        subj.SubjectId = associatedRecordId;

        // Insert the SurveySubject Record
        insert subj;
```

```
    }
}



//Use this trigger to create a survey subject record associated to
//the Survey Response record

trigger SurveyResponseForCaseTrigger on SurveyResponse (after insert) {

    System.debug('Inside Survey response trigger ');
     for(SurveyResponse sr: Trigger.New)
    {
      SurveySubject subj = new SurveySubject();
        subj.Name = 'Sur_Subject_for_response';
        subj.ParentId = sr.id; //Associating survey response id

       //Get the associatedRecordId recordId (like Case, Opportunity etc) using the
SurveyInvitation Id and
       //assigning it to SubjectId, assuming we inserted SurveySubject record for the
associated invitation
       //using the previous code

       List<SurveySubject> SurSubj=[select subjectid from SurveySubject where parentid =
:sr.invitationId];
        for(SurveySubject sub:SurSubj){
            String ids=String.valueOf(sub.subjectid).substring(0,3);
            if('500'.equals(ids)){

                subj.SubjectId =sub.subjectid;
        // Insert the SurveySubject Record
            insert subj;
                break;

            }

}
```

Site Namespace

The `Site` namespace provides an interface for rewriting Sites URLs.

The following is the interface in the `Site` namespace.

IN THIS SECTION:

UrlRewriter Interface

Enables rewriting Sites URLs.

Site Exceptions

The `Site` namespace contains an exception class.

UrlRewriter Interface

Enables rewriting Sites URLs.

Namespace

Site

Usage

Sites provides built-in logic that helps you display user-friendly URLs and links to site visitors. Create rules to rewrite URL requests typed into the address bar, launched from bookmarks, or linked from external websites. You can also create rules to rewrite the URLs for links within site pages. URL rewriting not only makes URLs more descriptive and intuitive for users, it allows search engines to better index your site pages.

For example, let's say that you have a blog site. Without URL rewriting, a blog entry's URL might look like this:
`https://myblog.my.salesforce-sites.com/posts?id=003D000000Q0PcN`

To rewrite URLs for a site, create an Apex class that maps the original URLs to user-friendly URLs, and then add the Apex class to your site.

UrlRewriter Methods

The following are methods for `UrlRewriter`. All are instance methods.

IN THIS SECTION:

generateUrlFor(salesforceUrls)
Maps a list of Salesforce URLs to a list of user-friendly URLs.

mapRequestUrl(userFriendlyUrl)
Maps a user-friendly URL to a Salesforce URL.

`generateUrlFor(salesforceUrls)`

Maps a list of Salesforce URLs to a list of user-friendly URLs.

Signature

```
public System.PageReference[] generateUrlFor(System.PageReference[] salesforceUrls)
```

Parameters

`salesforceUrls`
Type: System.PageReference[]

Return Value

Type: System.PageReference[]

Usage

You can use `List<PageReference>` instead of `PageReference[]`, if you prefer.

Important: The size and order of the input list of Salesforce URLs must exactly correspond to the size and order of the generated list of user-friendly URLs. The `generateUrlFor` method maps input URLs to output URLs based on the order in the lists.

`mapRequestUrl(userFriendlyUrl)`

Maps a user-friendly URL to a Salesforce URL.

Signature

```
public System.PageReference mapRequestUrl(System.PageReference userFriendlyUrl)
```

Parameters

`userFriendlyUrl`
Type: System.PageReference

Return Value

Type: System.PageReference

Site Exceptions

The `Site` namespace contains an exception class.

All exception classes support built-in methods for returning the error message and exception type. See Exception Class and Built-In Exceptions.

The `Site` namespace contains this exception:

Exception	Description	Methods
`Site.ExternalUserCreateException`	Unable to create external user	Use the `String getMessage()` to get the error message and write it to debug log. Use `List<String> getDisplayMessages()` to get a list of errors displayed to the end user. This exception can't be subclassed or thrown in code.

Slack Namespace

The `Slack` Namespace provides tools designed to accelerate and ease the process of developing Slack apps on the Salesforce platform.

The following are the classes in the `Slack` namespace.

App Class

Action Class

AppClient

AppRequest Classes
Apps Classes
Auth Classes
BotClient Class
BotsInfo Classes
Call Classes
Channel Class
Chat Classes
Conversation Class
Dnd Classes
Emoji CLasses
Event Classes
Field Class
File Classes
Latest Classes
Message Classes
MigrationExchange Classes
Modals Class
Options Classes
Paging Class
Pin Classes
Purpose Class
Reaction Classes
Reminder Classes
RequestContext Classes
ResponseMetadata Classes
RunnableHandler Interface
Search Classes
Shortcut Classes
SlackCommand Classes
Star Classes
Team Classes
TestHarness Classes
Topic Class
User Classes
UserClient Class
Usergroup Classes

UserMapping Service Class

Views Classes

Workflow Classes

Support Namespace

The `Support` namespace provides an interface used for Case Feed.

The following is the interface in the `Support` namespace.

IN THIS SECTION:

EmailTemplateSelector Interface

The `Support.EmailTemplateSelector` interface enables providing default email templates in Case Feed. With default email templates, specified email templates are preloaded for cases based on criteria such as case origin or subject.

MilestoneTriggerTimeCalculator Interface

The `Support.MilestoneTriggerTimeCalculator` interface calculates the time trigger for a milestone.

EmailTemplateSelector Interface

The `Support.EmailTemplateSelector` interface enables providing default email templates in Case Feed. With default email templates, specified email templates are preloaded for cases based on criteria such as case origin or subject.

`Support.EmailTemplateSelector` works only in Salesforce Classic, not in Lightning Experience. Lightning Experience users can specify default values for emails using the `QuickActionDefaultsHandler` interface.

Namespace

Support

To specify default templates, you must create a class that implements `Support.EmailTemplateSelector`.

When you implement this interface, provide an empty parameterless constructor.

IN THIS SECTION:

EmailTemplateSelector Methods

EmailTemplateSelector Example Implementation

EmailTemplateSelector Methods

The following are methods for `EmailTemplateSelector`.

IN THIS SECTION:

getDefaultTemplateId(caseId)

Returns the ID of the email template to preload for the case currently being viewed in the case feed using the specified case ID.

`getDefaultTemplateId(caseId)`

Returns the ID of the email template to preload for the case currently being viewed in the case feed using the specified case ID.

Signature

```
public ID getDefaultTemplateId(ID caseId)
```

Parameters

`caseId`
Type: ID

Return Value

Type: ID

EmailTemplateSelector Example Implementation

This is an example implementation of the `Support.EmailTemplateSelector` interface.

The `getDefaultEmailTemplateId` method implementation retrieves the subject and description of the case corresponding to the specified case ID. Next, it selects an email template based on the case subject and returns the email template ID.

```
global class MyCaseTemplateChooser implements Support.EmailTemplateSelector {
    // Empty constructor
    global MyCaseTemplateChooser() {    }

    // The main interface method
    global ID getDefaultEmailTemplateId(ID caseId) {
        // Select the case we're interested in, choosing any fields that are relevant to
our decision
        Case c = [SELECT Subject, Description FROM Case WHERE Id=:caseId];

        EmailTemplate et;

        if (c.subject.contains('LX-1150')) {
            et = [SELECT id FROM EmailTemplate WHERE DeveloperName = 'LX1150_template'];
        } else if(c.subject.contains('LX-1220')) {
            et = [SELECT id FROM EmailTemplate WHERE DeveloperName = 'LX1220_template'];
        }

        // Return the ID of the template selected
        return et.id;
    }
}
```

The following example tests the above code:

```
@isTest
private class MyCaseTemplateChooserTest {

    static testMethod void testChooseTemplate() {
```

```
        MyCaseTemplateChooser chooser = new MyCaseTemplateChooser();

        // Create a simulated case to test with
        Case c = new Case();
        c.Subject = 'I\'m having trouble with my LX-1150';
        Database.insert(c);

        // Make sure the proper template is chosen for this subject
        Id actualTemplateId = chooser.getDefaultEmailTemplateId(c.Id);
        EmailTemplate expectedTemplate =
          [SELECT id FROM EmailTemplate WHERE DeveloperName = 'LX1150_template'];
        Id expectedTemplateId = expectedTemplate.Id;
        System.assertEquals(actualTemplateId, expectedTemplateId);

        // Change the case properties to match a different template
        c.Subject = 'My LX1220 is overheating';
        Database.update(c);

        // Make sure the correct template is chosen in this case
       actualTemplateId = chooser.getDefaultEmailTemplateId(c.Id);
        expectedTemplate =
          [SELECT id FROM EmailTemplate WHERE DeveloperName = 'LX1220_template'];
        expectedTemplateId = expectedTemplate.Id;
        System.assertEquals(actualTemplateId, expectedTemplateId);

    }
}
```

MilestoneTriggerTimeCalculator Interface

The `Support.MilestoneTriggerTimeCalculator` interface calculates the time trigger for a milestone.

Namespace

Support

Implement the `Support.MilestoneTriggerTimeCalculator` interface to calculate a dynamic time trigger for a milestone based on the milestone type, the properties of the case, and case-related objects. To implement the `Support.MilestoneTriggerTimeCalculator` interface, you must first declare a class with the `implements` keyword as follows:

```
global class Employee implements Support.MilestoneTriggerTimeCalculator {
```

Next, your class must provide an implementation for the following method:

```
global Integer calculateMilestoneTriggerTime(String caseId, String milestoneTypeId)
```

The implemented method must be declared as `global` or `public`.

IN THIS SECTION:

- MilestoneTriggerTimeCalculator Methods
- MilestoneTriggerTimeCalculator Example Implementation

MilestoneTriggerTimeCalculator Methods

The following are instance methods for `MilestoneTriggerTimeCalculator`.

IN THIS SECTION:

calculateMilestoneTriggerTime(caseId, milestoneTypeId)
Calculates the milestone trigger time based on the specified case and milestone type and returns the time in minutes.

`calculateMilestoneTriggerTime(caseId, milestoneTypeId)`

Calculates the milestone trigger time based on the specified case and milestone type and returns the time in minutes.

Syntax

```
public Integer calculateMilestoneTriggerTime(String caseId, String milestoneTypeId)
```

Parameters

caseId
Type: String

ID of the case the milestone is applied to.

milestoneTypeId
Type: String

ID of the milestone type.

Return Value

Type: Integer

The calculated trigger time in minutes.

MilestoneTriggerTimeCalculator Example Implementation

This sample class demonstrates the implementation of the`Support.MilestoneTriggerTimeCalculator` interface. In this sample, the case's priority and the milestone `m1` determine that the time trigger is 18 minutes.

```
global class myMilestoneTimeCalculator implements Support.MilestoneTriggerTimeCalculator
{
     global Integer calculateMilestoneTriggerTime(String caseId, String milestoneTypeId){

        Case c = [SELECT Priority FROM Case WHERE Id=:caseId];
        MilestoneType mt = [SELECT Name FROM MilestoneType WHERE Id=:milestoneTypeId];
        if (c.Priority != null && c.Priority.equals('High')){
              if (mt.Name != null && mt.Name.equals('m1')) { return 7;}
              else { return 5; }
        }
        else {
            return 18;
        }
```

```
    }
}
```

This test class can be used to test the implementation of `Support.MilestoneTriggerTimeCalculator`.

```
@isTest
private class MilestoneTimeCalculatorTest {
    static testMethod void testMilestoneTimeCalculator() {

        // Select an existing milestone type to test with
        MilestoneType[] mtLst = [SELECT Id, Name FROM MilestoneType LIMIT 1];
        if(mtLst.size() == 0) { return; }
        MilestoneType mt = mtLst[0];

        // Create case data.
        // Typically, the milestone type is related to the case,
        // but for simplicity, the case is created separately for this test.
        Case c = new Case(priority = 'High');
        insert c;

        myMilestoneTimeCalculator calculator = new myMilestoneTimeCalculator();
       Integer actualTriggerTime = calculator.calculateMilestoneTriggerTime(c.Id, mt.Id);


        if(mt.name != null && mt.Name.equals('m1')) {
            System.assertEquals(actualTriggerTime, 7);
        }
        else {
            System.assertEquals(actualTriggerTime, 5);
        }

        c.priority = 'Low';
        update c;
        actualTriggerTime = calculator.calculateMilestoneTriggerTime(c.Id, mt.Id);
        System.assertEquals(actualTriggerTime, 18);
    }
}
```

System Namespace

The `System` namespace provides classes and methods for core Apex functionality.

The following are the classes in the `System` namespace.

IN THIS SECTION:

AccessLevel Class

Defines the different modes, such as system or user mode, that Apex database operations execute in.

AccessType Enum

Specifies the access check type for the fields of an sObject.

Address Class

Contains methods for accessing the component fields of address compound fields.

Answers Class

Represents zone answers.

ApexPages Class

Use `ApexPages` to add and check for messages associated with the current page, as well as to reference the current page.

Approval Class

Contains methods for processing approval requests and setting approval-process locks and unlocks on records.

Assert Class

Contains methods to assert various conditions with test methods, such as whether two values are the same, a condition is true, or a variable is null.

AsyncInfo Class

Provides methods to get the current stack depth, maximum stack depth, and the minimum queueable delay for Queueable transactions, and to determine if maximum stack depth is set.

AsyncOptions Class

Contains maximum stack depths for queueable transactions and the minimum queueable delay in minutes. Passed as parameter to the `System.enqueueJob()` method to define a unique queueable job signature, the maximum stack depth for queueable transactions and the minimum queueable delay in minutes.

Blob Class

Contains methods for the Blob primitive data type.

Boolean Class

Contains methods for the Boolean primitive data type.

BusinessHours Class

Use the `BusinessHours` methods to set the business hours at which your customer support team operates.

CallbackStatus Enum

Specifies the status of asynchronous requests to an external system.

Callable Interface

Enables developers to use a common interface to build loosely coupled integrations between Apex classes or triggers, even for code in separate packages. Agreeing upon a common interface enables developers from different companies or different departments to build upon one another's solutions. Implement this interface to enable the broader community, which might have different solutions than the ones you had in mind, to extend your code's functionality.

Cases Class

Use the `Cases` class to interact with case records.

Collator Class

Contains methods to get locale-specific instances that can be used for comparisons and sorting. Use the `getInstance()` method to obtain the Collator instance for a given locale and pass the Collator as the Comparator parameter to the `list.sort()` method.

Comparable Interface

Adds sorting support for Lists that contain non-primitive types, that is, Lists of user-defined types. Your implementation must explicitly handle null inputs in the `compareTo()` method to avoid a null pointer exception.

Comparator Interface

Implement different sort orders with the Comparator interface's `compare()` method, and pass the Comparator as a parameter to `List.sort()`. Your implementation must explicitly handle null inputs in the `compare()` method to avoid a null pointer exception.

Continuation Class

Use the `Continuation` class to make callouts asynchronously to a SOAP or REST Web service.

Cookie Class

The `Cookie` class lets you access cookies for your Salesforce site using Apex.

Crypto Class

Provides methods for creating digests, message authentication codes, and signatures, as well as encrypting and decrypting information.

Custom Metadata Type Methods

Custom metadata types are customizable, deployable, packageable, and upgradeable application metadata. All custom metadata is exposed in the application cache, which allows access without repeated queries to the database. The metadata is then available for formula fields, validation rules, flows, Apex, and SOAP API. All methods are static.

Custom Settings Methods

Custom settings are similar to custom objects and enable application developers to create custom sets of data, as well as create and associate custom data for an organization, profile, or specific user. All custom settings data is exposed in the application cache, which enables efficient access without the cost of repeated queries to the database. This data is then available for formula fields, validation rules, flows, Apex, and the SOAP API.

Database Class

Contains methods for creating and manipulating data.

Date Class

Contains methods for the Date primitive data type.

Datetime Class

Contains methods for the Datetime primitive data type.

Decimal Class

Contains methods for the Decimal primitive data type.

Domain Class

Represents an existing domain hosted by Salesforce that serves the org or its content. Contains methods to obtain information about these domains, such as the domain type, My Domain name, and sandbox name.

DomainCreator Class

Use the DomainCreator class to return a hostname specific to the org. For example, get the org's Visualforce hostname. Values are returned as a hostname, such as ***`MyDomainName`***`.lightning.force.com`.

DomainParser Class

Use the DomainParser class to parse a domain that Salesforce hosts for the org and extract information about the domain.

DomainType Enum

Specifies the domain type for a System.Domain.

Double Class

Contains methods for the Double primitive data type.

EmailMessages Class

Use the methods in the `EmailMessages` class to interact with emails and email threading.

EncodingUtil Class

Use the methods in the `EncodingUtil` class to encode and decode URL strings, and convert strings to hexadecimal format.

Enum Methods

An enum is an abstract data type with values that each take on exactly one of a finite set of identifiers that you specify. Apex provides built-in enums, such as `LoggingLevel`, and you can define your own enum.

EventBus Class

Contains methods for publishing platform events.

Exception Class and Built-In Exceptions

An exception denotes an error that disrupts the normal flow of code execution. You can use Apex built-in exceptions or create custom exceptions. All exceptions have common methods.

ExternalServiceTest Class

Provides methods to test an external service's asynchronous callouts, enables sending a mock request, asserts the expected request payload, then triggers the mocked external service's asynchronous callback response.

FlexQueue Class

Contains methods that reorder batch jobs in the Apex flex queue.

FeatureManagement Class

Use the methods in the `System.FeatureManagement` class to check and modify the values of feature parameters, and to show or hide custom objects and custom permissions in your subscribers' orgs.

Formula Class

Contains methods to get a builder for creating a formula instance and to update all formula fields on the input SObjects..

FormulaRecalcFieldError Class

The return type of the `FormulaRecalcResult.getErrors` method.

FormulaRecalcResult Class

The return type of the `Formula.recalculateFormulas` method.

Http Class

Use the `Http` class to initiate an HTTP request and response.

HttpCalloutMock Interface

Enables sending fake responses when testing HTTP callouts.

HttpRequest Class

Use the `HttpRequest` class to programmatically create HTTP requests like GET, POST, PATCH, PUT, and DELETE.

HttpResponse Class

Use the `HttpResponse` class to handle the HTTP response returned by the `Http` class.

Id Class

Contains methods for the ID primitive data type.

Ideas Class

Represents zone ideas.

InstallHandler Interface

Enables custom code to run after a managed package installation or upgrade.

Integer Class

Contains methods for the Integer primitive data type.

JSON Class

Contains methods for serializing Apex objects into JSON format and deserializing JSON content that was serialized using the `serialize` method in this class.

JSONGenerator Class

Contains methods used to serialize objects into JSON content using the standard JSON encoding.

JSONParser Class

Represents a parser for JSON-encoded content.

JSONToken Enum

Contains all token values used for parsing JSON content.

Label Class

Provides methods to retrieve a custom label or to check if translation exists for a label in a specific language and namespace. Label names are dynamically resolved at run time, overriding the user's current language if a translation exists for the requested language. You can't access labels that are protected in a different namespace.

Limits Class

Contains methods that return limit information for specific resources.

List Class

Contains methods for the List collection type.

Location Class

Contains methods for accessing the component fields of geolocation compound fields.

LoggingLevel Enum

Specifies the logging level for the `System.debug` method.

Long Class

Contains methods for the Long primitive data type.

Map Class

Contains methods for the Map collection type.

Matcher Class

Matchers use Patterns to perform match operations on a character string.

Math Class

Contains methods for mathematical operations.

Messaging Class

Contains messaging methods used when sending a single or mass email.

MultiStaticResourceCalloutMock Class

Utility class used to specify a fake response using multiple resources for testing HTTP callouts.

Network Class

Manage Experience Cloud sites.

Object Class

Contains methods that are implemented by all Apex types.

OrgLimit Class

Contains methods that provide the name, maximum value, and current value of an org limit.

OrgLimits Class

Contains methods that provide a list or map of all OrgLimit instances for Salesforce your org, such as SOAP API requests, Bulk API requests, and Streaming API limits.

PageReference Class

A PageReference is a reference to an instantiation of a page. Among other attributes, PageReferences consist of a URL and a set of query parameter names and values.

Packaging Class

Contains a method for obtaining information about managed and unlocked packages.

Pattern Class

Represents a compiled representation of a regular expression.

Queueable Interface

Enables the asynchronous execution of Apex jobs that can be monitored.

QueueableContext Interface

Represents the parameter type of the `execute()` method in a class that implements the `Queueable` interface and contains the job ID. This interface is implemented internally by Apex.

QueueableDuplicateSignature Class

Used in the `AsyncOptions` class to store the queueable job signature in the `DuplicateSignature` property.

QueueableDuplicateSignature.Builder Class

Build a unique signature for your queueable job using this inner builder class. The `build()` class method builds a `QueueableDuplicateSignature` object, with input from the `addId()`, `addInteger()`, and `addString()` methods. Use the `DuplicateSignature` property in the `AsyncOptions` class to store the queueable job signature. Enqueue your job by using the `System.enqueueJob()` with the `AsyncOptions` parameter.

QuickAction Class

Use Apex to request and process actions on objects that allow custom fields, on objects that appear in a Chatter feed, or on objects that are available globally.

Quiddity Enum

Specifies a Quiddity value used by the methods in the System.Request class

RemoteObjectController

Use `RemoteObjectController` to access the standard Visualforce Remote Objects operations in your Remote Objects override methods.

Request Class

Contains methods to obtain the request ID and Quiddity value of the current Salesforce request.

ResetPasswordResult Class

Represents the result of a password reset.

RestContext Class

Contains the `RestRequest` and `RestResponse` objects.

RestRequest Class

Use the `System.RestRequest` class to access and pass request data in a RESTful Apex method.

RestResponse Class

Represents an object used to pass data from an Apex RESTful Web service method to an HTTP response.

SandboxPostCopy Interface

To make your sandbox environment business ready, automate data manipulation or business logic tasks. Extend this interface and add methods to perform post-copy tasks, then specify the class during sandbox creation.

Schedulable Interface

The class that implements this interface can be scheduled to run at different intervals.

SchedulableContext Interface

Represents the parameter type of a method in a class that implements the `Schedulable` interface and contains the scheduled job ID. This interface is implemented internally by Apex.

Schema Class

Contains methods for obtaining schema describe information.

Search Class

Use the methods of the Search class to perform dynamic SOSL queries.

Security Class

Contains methods to securely implement Apex applications.

SelectOption Class

A `SelectOption` object specifies one of the possible values for a Visualforce `selectCheckboxes`, `selectList`, or `selectRadio` component.

Set Class

Represents a collection of unique elements with no duplicate values.

Site Class

Use the `Site` Class to manage your sites. Change, reset, validate, and check the expiration of passwords. Create site users, person accounts, and portal users. Get the admin email and ID. Get various URLs, the path prefix, the ID, the template, and the type of the site. Log in to the site.

SObject Class

Contains methods for the sObject data type.

SObjectAccessDecision Class

Contains the results of a call to the Security.stripInaccessible method and methods to retrieve those results.

SoqlStubProvider Class

Contains a method to create a mock test class for handling SOQL query responses for Data Cloud data model objects (DMOs).

StaticResourceCalloutMock Class

Utility class used to specify a fake response for testing HTTP callouts.

String Class

Contains methods for the String primitive data type.

StubProvider Interface

`StubProvider` is a callback interface that you can use as part of the Apex stub API to implement a mocking framework. Use this interface with the `Test.createStub()` method to create stubbed Apex objects for testing.

System Class

Contains methods for system operations, such as writing debug messages and scheduling jobs.

Test Class

Contains methods related to Apex tests.

Time Class

Contains methods for the Time primitive data type.

TimeZone Class

Represents a time zone. Contains methods for creating a new time zone and obtaining time zone properties, such as the time zone ID, offset, and display name.

Trigger Class

Use the `Trigger` class to access run-time context information in a trigger, such as the type of trigger or the list of sObject records that the trigger operates on.

TriggerOperation Enum

System.TriggerOperation enum values are associated with trigger events.

Type Class

Contains methods for getting the Apex type that corresponds to an Apex class and for instantiating new types.

UninstallHandler Interface

Enables custom code to run after a managed package is uninstalled.

URL Class

Represents a uniform resource locator (URL) and provides access to parts of the URL. Enables access to the base URL used to access your Salesforce org.

UserInfo Class

Contains methods for obtaining information about the context user.

UserManagement Class

Contains methods to manage end users, for example, to register their verification methods, verify their identity, or remove their personal information.

UUID Class

Contains methods to randomly generate a version 4 universally unique identifier (UUID), compare UUIDs, and convert UUID instance to a string.

Version Class

Use the Version methods to get the version of a first-generation managed package, and to compare package versions.

WebServiceCallout Class

Enables making callouts to SOAP operations on an external Web service. This class is used in the Apex stub class that is auto-generated from a WSDL.

WebServiceMock Interface

Enables sending fake responses when testing Web service callouts of a class auto-generated from a WSDL.

XmlStreamReader Class

The `XmlStreamReader` class provides methods for forward, read-only access to XML data. You can pull data from XML or skip unwanted events. You can parse nested XML content that's up to 50 nodes deep.

XmlStreamWriter Class

The `XmlStreamWriter` class provides methods for writing XML data.

AccessLevel Class

Defines the different modes, such as system or user mode, that Apex database operations execute in.

Namespace

System

Usage

By default, Apex code runs in system mode, which means that it runs with substantially elevated permissions over the user running the code. In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the current user's object permissions, field-level security, and sharing rules are enforced.

Many of the DML methods of the `System.Database` and `System.Search` classes include an `accessLevel` parameter to specify the execution mode.

Example

If the user running this Apex code doesn't have write access to the Account object, the `Database.insert()` method returns an error.

```
List<Account> toInsert = new List<Account>{new Account(Name = 'Exciting New Account')};

List<Database.SaveResult> sr = Database.insert(toInsert, AccessLevel.USER_MODE);
```

In contrast, this example shows the method running in system mode. The success of the insert doesn't depend on whether the user running the Apex code has create access to the Account object.

```
List<Account> toInsert = new List<Account>{new Account(Name = 'Exciting New Account')};

List<Database.SaveResult> sr = Database.insert(toInsert, AccessLevel.SYSTEM_MODE);
```

IN THIS SECTION:

AccessLevel Methods

AccessLevel Properties

AccessLevel Methods

The following are methods for `AccessLevel`.

IN THIS SECTION:

withPermissionSetId(permissionSetId)(Developer Preview)

Supports database and search operations to be run with permissions specified in a permission set. Apex enforces field-level security (FLS) and object permissions as per the specified permission set, in addition to the running user's permissions.

`withPermissionSetId(permissionSetId)(Developer Preview)`

Supports database and search operations to be run with permissions specified in a permission set. Apex enforces field-level security (FLS) and object permissions as per the specified permission set, in addition to the running user's permissions.

Note: Feature is available as a developer preview. Feature isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in a production environment. You can provide feedback and suggestions for the "Permission Sets with User Mode" feature in the Trailblazer Community.

This feature is available in scratch orgs where the `ApexUserModeWithPermset` feature is enabled. If the feature isn't enabled, Apex code with this feature can be compiled but not executed.

Signature

```
public System.AccessLevel withPermissionSetId(String permissionSetId)
```

Parameters

permissionSetId
Type: String

Permissions in the specified permission set are enforced while running user-mode DML operations, in addition to the running user's permissions.

Return Value

Type: Access Level Class

Example: This example runs the `AccessLevel.withPermissionSetId()` method with the specified permission set and inserts a custom object.

```
@isTest
public with sharing class ElevateUserModeOperations_Test {
    @isTest
    static void objectCreatePermViaPermissionSet() {
        Profile p = [SELECT Id FROM Profile WHERE Name='Minimum Access - Salesforce'];

        User u = new User(Alias = 'standt', Email='standarduser@testorg.com',
            EmailEncodingKey='UTF-8', LastName='Testing', LanguageLocaleKey='en_US',
            LocaleSidKey='en_US', ProfileId = p.Id,
            TimeZoneSidKey='America/Los_Angeles',
            UserName='standarduser' + DateTime.now().getTime() + '@testorg.com');

        System.runAs(u) {
            try {
                Database.insert(new Account(name='foo'), AccessLevel.User_mode);
                Assert.fail();
            } catch (SecurityException ex) {
                Assert.isTrue(ex.getMessage().contains('Account'));
            }
           //Get ID of previously created permission set named 'AllowCreateToAccount'

            Id permissionSetId = [Select Id from PermissionSet
                where Name = 'AllowCreateToAccount' limit 1].Id;

            Database.insert(new Account(name='foo'),
AccessLevel.User_mode.withPermissionSetId(permissionSetId));

            // The elevated access level in not persisted to subsequent operations
            try {
                Database.insert(new Account(name='foo2'), AccessLevel.User_mode);
                Assert.fail();
            } catch (SecurityException ex) {
```

```
                Assert.isTrue(ex.getMessage().contains('Account'));
            }

        }
    }
}
```

AccessLevel Properties

The following are properties for `AccessLevel`.

IN THIS SECTION:

SYSTEM_MODE
Execution mode in which the the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords.

USER_MODE
Execution mode in which the object permissions, field-level security, and sharing rules of the current user are enforced.

SYSTEM_MODE

Execution mode in which the the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords.

Signature

```
public System.AccessLevel SYSTEM_MODE {get;}
```

Property Value

Type: System.AccessLevel

USER_MODE

Execution mode in which the object permissions, field-level security, and sharing rules of the current user are enforced.

Signature

```
public System.AccessLevel USER_MODE {get;}
```

Property Value

Type: System.AccessLevel

AccessType Enum

Specifies the access check type for the fields of an sObject.

Usage

Use these enum values for the `accessCheckType` parameter of the stripInaccessible method.

Enum Values

The following are the values of the `System.AccessType` enum.

Value	Description
`CREATABLE`	Check the fields of an sObject for create access.
`READABLE`	Check the fields of an sObject for read access.
`UPDATABLE`	Check the fields of an sObject for update access.
`UPSERTABLE`	Check the fields of an sObject for both insert and update access.

Address Class

Contains methods for accessing the component fields of address compound fields.

Namespace

System

Usage

Each of these methods is also equivalent to a read-only property. For each getter method, you can access the property using dot notation. For example, `myAddress.getCity()` is equivalent to `myAddress.city`.

You can't use dot notation to access compound fields' subfields directly on the parent field. Instead, assign the parent field to a variable of type `Address`, and then access its components. For example, to access the `City` field in `myAccount.BillingAddress`, do the following:

```
Address addr = myAccount.BillingAddress;
String acctCity = addr.City;
```

Important: "Address" in Salesforce can also refer to the Address standard object. When referencing the Address object in your Apex code, always use `Schema.Address` instead of `Address` to prevent confusion with the standard Address compound field. If referencing both the Address object and the Address standard field in the same snippet, you can differentiate between the two by using `System.Address` for the field and `Schema.Address` for the object.

Example

```
// Select and access Address fields.
// Call the getDistance() method in different ways.
Account[] records = [SELECT id, BillingAddress FROM Account LIMIT 10];
for(Account acct : records) {
   Address addr = acct.BillingAddress;
```

```
    Double lat = addr.latitude;
    Double lon = addr.longitude;
    Location loc1 = Location.newInstance(30.1944,-97.6682);
    Double apexDist1 = addr.getDistance(loc1, 'mi');
    Double apexDist2 = loc1.getDistance(addr, 'mi');
    System.assertEquals(apexDist1, apexDist2);
    Double apexDist3 = Location.getDistance(addr, loc1, 'mi');
    System.assertEquals(apexDist2, apexDist3);
}
```

IN THIS SECTION:

Address Methods

Address Methods

The following are methods for `Address`.

IN THIS SECTION:

getCity()
Returns the city field of this address.

getCountry()
Returns the text-only country/territory name component of this address.

getCountryCode()
Returns the country/territory code of this address if state and country/territory picklists are enabled in your organization. Otherwise, returns `null`.

getDistance(toLocation, unit)
Returns the distance from this location to the specified location using the specified unit.

getGeocodeAccuracy()
When using geolocation data for a given address, this method gives you relative location information based on latitude and longitude values. For example, you can find out if the latitude and longitude values point to the middle of the street, instead of the exact address.

getLatitude()
Returns the latitude field of this address.

getLongitude()
Returns the longitude field of this address.

getPostalCode()
Returns the postal code of this address.

getState()
Returns the text-only state name component of this address.

getStateCode()
Returns the state code of this address if state and country/territory picklists are enabled in your organization. Otherwise, returns `null`.

getStreet()
Returns the street field of this address.

getCity()

Returns the city field of this address.

Signature

```
public String getCity()
```

Return Value

Type: String

getCountry()

Returns the text-only country/territory name component of this address.

Signature

```
public String getCountry()
```

Return Value

Type: String

getCountryCode()

Returns the country/territory code of this address if state and country/territory picklists are enabled in your organization. Otherwise, returns `null`.

Signature

```
public String getCountryCode()
```

Return Value

Type: String

getDistance(toLocation, unit)

Returns the distance from this location to the specified location using the specified unit.

Signature

```
public Double getDistance(Location toLocation, String unit)
```

Parameters

toLocation
Type: Location

The `Location` to which you want to calculate the distance from the current `Location`.

unit
Type: String

The distance unit you want to use: `mi` or `km`.

Return Value

Type: Double

getGeocodeAccuracy()

When using geolocation data for a given address, this method gives you relative location information based on latitude and longitude values. For example, you can find out if the latitude and longitude values point to the middle of the street, instead of the exact address.

Signature

```
public String getGeocodeAccuracy()
```

Return Value

Type: String

The `getGeocodeAccuracy()` return value tells you more about the location at a latitude and longitude for a given address. For example, `Zip` means the latitude and longitude point to the center of the zip code area, in case a match for an exact street address can't be found.

Accuracy Value	Description
`Address`	In the same building
`NearAddress`	Near the address
`Block`	Midway point of the block
`Street`	Midway point of the street
`ExtendedZip`	Center of the extended zip code area
`Zip`	Center of the zip code area
`Neighborhood`	Center of the neighborhood
`City`	Center of the city
`County`	Center of the county
`State`	Center of the state
`Unknown`	No match for the address was found

Geocodes are added only for some standard addresses.

- `Billing Address` on accounts
- `Shipping Address` on accounts
- `Mailing Address` on contacts
- `Address` on leads

Person accounts are not supported.

Note: For `getGeocodeAccuracy()` to work, set up and activate the geocode data integration rules for the related address fields.

`getLatitude()`

Returns the latitude field of this address.

Signature

```
public Double getLatitude()
```

Return Value

Type: Double

`getLongitude()`

Returns the longitude field of this address.

Signature

```
public Double getLongitude()
```

Return Value

Type: Double

`getPostalCode()`

Returns the postal code of this address.

Signature

```
public String getPostalCode()
```

Return Value

Type: String

`getState()`

Returns the text-only state name component of this address.

Signature

```
public String getState()
```

Return Value

Type: String

`getStateCode()`

Returns the state code of this address if state and country/territory picklists are enabled in your organization. Otherwise, returns `null`.

Signature

```
public String getStateCode()
```

Return Value

Type: String

`getStreet()`

Returns the street field of this address.

Signature

```
public String getStreet()
```

Return Value

Type: String

Answers Class

Represents zone answers.

Namespace

System

Usage

Answers is a feature that enables users to ask questions and have zone members post replies. Members can then vote on the helpfulness of each reply, and the person who asked the question can mark one reply as the best answer.

For more information on answers, see "Answers Overview" in the Salesforce online help.

Example

The following example finds questions in an internal zone that have similar titles as a new question:

```
public class FindSimilarQuestionController {

   public static void test() {
   // Instantiate a new question
   Question question = new Question ();

   // Specify a title for the new question
   question.title = 'How much vacation time do full-time employees get?';

   // Specify the communityID (INTERNAL_COMMUNITY) in which to find similar questions.
   Community community = [ SELECT Id FROM Community WHERE Name = 'INTERNAL_COMMUNITY' ];

   question.communityId = community.id;

   ID[] results = Answers.findSimilar(question);
   }
}
```

The following example marks a reply as the best reply:

```
ID questionId = [SELECT Id FROM Question WHERE Title = 'Testing setBestReplyId' LIMIT
1].Id;
ID replyID = [SELECT Id FROM Reply WHERE QuestionId = :questionId LIMIT 1].Id;
Answers.setBestReply(questionId,replyId);
```

Answers Methods

The following are methods for `Answers`. All methods are static.

IN THIS SECTION:

findSimilar(yourQuestion)

Returns a list of similar questions based on the title of the specified question.

setBestReply(questionId, replyId)

Sets the specified reply for the specified question as the best reply. Because a question can have multiple replies, setting the best reply helps users quickly identify the reply that contains the most helpful information.

`findSimilar(yourQuestion)`

Returns a list of similar questions based on the title of the specified question.

Signature

```
public static ID[] findSimilar(Question yourQuestion)
```

Parameters

yourQuestion
Type: Question

Return Value

Type: ID[]

Usage

Each `findSimilar` call counts against the SOSL statements governor limit allowed for the process.

`setBestReply(questionId, replyId)`

Sets the specified reply for the specified question as the best reply. Because a question can have multiple replies, setting the best reply helps users quickly identify the reply that contains the most helpful information.

Signature

```
public static Void setBestReply(String questionId, String replyId)
```

Parameters

questionId
Type: String

replyId
Type: String

Return Value

Type: Void

ApexPages Class

Use `ApexPages` to add and check for messages associated with the current page, as well as to reference the current page.

Namespace

System

Usage

In addition, `ApexPages` is used as a namespace for the PageReference Class and the Message Class.

ApexPages Methods

The following are methods for `ApexPages`. All are instance methods.

IN THIS SECTION:

addMessage(message)
Add a message to the current page context.

addMessages(exceptionThrown)
Adds a list of messages to the current page context based on a thrown exception.

currentPage()
Returns the current page's PageReference.

getMessages()
Returns a list of the messages associated with the current context.

hasMessages()
Returns `true` if there are messages associated with the current context, `false` otherwise.

hasMessages(severity)
Returns `true` if messages of the specified severity exist, `false` otherwise.

addMessage(message)

Add a message to the current page context.

Signature

```
public Void addMessage(ApexPages.Message message)
```

Parameters

message
Type: ApexPages.Message

Return Value

Type: Void

addMessages(exceptionThrown)

Adds a list of messages to the current page context based on a thrown exception.

Signature

```
public Void addMessages(Exception exceptionThrown)
```

Parameters

exceptionThrown
Type: Exception

Return Value

Type: Void

currentPage()

Returns the current page's PageReference.

Signature

```
public System.PageReference currentPage()
```

Return Value

Type: System.PageReference

Example

This code segment returns the id parameter of the current page.

```
public MyController() {
    account = [
        SELECT Id, Name, Site
        FROM Account
        WHERE Id =
            :ApexPages.currentPage().
             getParameters().
             get('id')
    ];
}
```

getMessages()

Returns a list of the messages associated with the current context.

Signature

```
public ApexPages.Message[] getMessages()
```

Return Value

Type: ApexPages.Message[]

hasMessages()

Returns `true` if there are messages associated with the current context, `false` otherwise.

Signature

```
public Boolean hasMessages()
```

Return Value

Type: Boolean

`hasMessages(severity)`

Returns `true` if messages of the specified severity exist, `false` otherwise.

Signature

```
public Boolean hasMessages(ApexPages.Severity severity)
```

Parameters

sev
Type: ApexPages.Severity

Return Value

Type: Boolean

Approval Class

Contains methods for processing approval requests and setting approval-process locks and unlocks on records.

Namespace

System

Usage

Salesforce admins can edit locked records. Depending on your approval process configuration settings, an assigned approver can also edit locked records. Locks and unlocks that are set programmatically use the same record editability settings as other approval-process locks and unlocks.

Record locks and unlocks are treated as DML. They're blocked before a callout, they count toward your DML limits, and if a failure occurs, they're rolled back along with the rest of your transaction. To change this rollback behavior, use an `allOrNone` parameter.

Approval is also used as a namespace for the `ProcessRequest` and `ProcessResult` classes.

SEE ALSO:

Approval Process Considerations

Approval Methods

The following are methods for `Approval`. All methods are static.

IN THIS SECTION:

isLocked(id)
Returns `true` if the record with the ID `id` is locked, or `false` if it's not.

isLocked(ids)
Returns a map of record IDs and their lock statuses. If the record is locked the status is `true`. If the record is not locked the status is `false`.

isLocked(sobject)

Returns `true` if the `sobject` record is locked, or `false` if it's not.

isLocked(sobjects)

Returns a map of record IDs to lock statuses. If the record is locked the status is `true`. If the record is not locked the status is `false`.

lock(recordId)

Locks an object, and returns the lock results.

lock(recordIds)

Locks a set of objects, and returns the lock results, including failures.

lock(recordToLock)

Locks an object, and returns the lock results.

lock(recordsToLock)

Locks a set of objects, and returns the lock results, including failures.

lock(recordId, allOrNothing)

Locks an object, with the option for partial success, and returns the lock result.

lock(recordIds, allOrNothing)

Locks a set of objects, with the option for partial success. It returns the lock results, including failures.

lock(recordToLock, allOrNothing)

Locks an object, with the option for partial success, and returns the lock result.

lock(recordsToLock, allOrNothing)

Locks a set of objects, with the option for partial success. It returns the lock results, including failures.

process(approvalRequest)

Submits a new approval request and approves or rejects existing approval requests.

process(approvalRequest, allOrNone)

Submits a new approval request and approves or rejects existing approval requests.

process(approvalRequests)

Submits a list of new approval requests, and approves or rejects existing approval requests.

process(approvalRequests, allOrNone)

Submits a list of new approval requests, and approves or rejects existing approval requests.

unlock(recordId)

Unlocks an object, and returns the unlock results.

unlock(recordIds)

Unlocks a set of objects, and returns the unlock results, including failures.

unlock(recordToUnlock)

Unlocks an object, and returns the unlock results.

unlock(recordsToUnlock)

Unlocks a set of objects, and returns the unlock results, including failures.

unlock(recordId, allOrNothing)

Unlocks an object, with the option for partial success, and returns the unlock result.

unlock(recordIds, allOrNothing)

Unlocks a set of objects, with the option for partial success. It returns the unlock results, including failures.

unlock(recordToUnlock, allOrNothing)

Unlocks an object, with the option for partial success, and returns the unlock result.

unlock(recordsToUnlock, allOrNothing)

Unlocks a set of objects, with the option for partial success. It returns the unlock results, including failures.

isLocked(id)

Returns `true` if the record with the ID `id` is locked, or `false` if it's not.

Signature

```
public static Boolean isLocked(Id id)
```

Parameters

id

Type: Id

The ID of the record whose lock or unlock status is in question.

Return Value

Type: Boolean

isLocked(ids)

Returns a map of record IDs and their lock statuses. If the record is locked the status is `true`. If the record is not locked the status is `false`.

Signature

```
public static Map<Id,Boolean> isLocked(List<Id> ids)
```

Parameters

ids

Type: List<Id>

The IDs of the records whose lock or unlock statuses are in question.

Return Value

Type: Map<Id,Boolean>

isLocked(sobject)

Returns `true` if the `sobject` record is locked, or `false` if it's not.

Signature

```
public static Boolean isLocked(SObject sobject)
```

Parameters

sobject
Type: SObject

The record whose lock or unlock status is in question.

Return Value

Type: Boolean

isLocked(sobjects)

Returns a map of record IDs to lock statuses. If the record is locked the status is `true`. If the record is not locked the status is `false`.

Signature

```
public static Map<Id,Boolean> isLocked(List<SObject> sobjects)
```

Parameters

sobjects
Type: List<SObject>

The records whose lock or unlock statuses are in question.

Return Value

Type: Map<Id,Boolean>

lock(recordId)

Locks an object, and returns the lock results.

Signature

```
public static Approval.LockResult lock(Id recordId)
```

Parameters

recordId
Type: Id

ID of the object to lock.

Return Value

Type: Approval.LockResult

lock(recordIds)

Locks a set of objects, and returns the lock results, including failures.

Signature

```
public static List<Approval.LockResult> lock(List<Id> ids)
```

Parameters

ids
Type: List<Id>

IDs of the objects to lock.

Return Value

Type: List<Approval.LockResult>

lock(recordToLock)

Locks an object, and returns the lock results.

Signature

```
public static Approval.LockResult lock(SObject recordToLock)
```

Parameters

recordToLock
Type: SObject

Return Value

Type: Approval.LockResult

lock(recordsToLock)

Locks a set of objects, and returns the lock results, including failures.

Signature

```
public static List<Approval.LockResult> lock(List<SObject> recordsToLock)
```

Parameters

recordsToLock
Type: List<SObject>

Return Value

Type: List<Approval.LockResult>

lock(recordId, allOrNothing)

Locks an object, with the option for partial success, and returns the lock result.

Signature

```
public static Approval.LockResult lock(Id recordId, Boolean allOrNothing)
```

Parameters

recordId
Type: Id

ID of the object to lock.

allOrNothing
Type: Boolean

Specifies whether this operation allows partial success. If you specify `false` and a record fails, the remainder of the DML operation can still succeed. This method returns a result object that you can use to verify which records succeeded, which failed, and why.

Return Value

Type: Approval.LockResult

lock(recordIds, allOrNothing)

Locks a set of objects, with the option for partial success. It returns the lock results, including failures.

Signature

```
public static List<Approval.LockResult> lock(List<Id> recordIds, Boolean allOrNothing)
```

Parameters

recordIds
Type: List<Id>

IDs of the objects to lock.

allOrNothing
Type: Boolean

Specifies whether this operation allows partial success. If you specify `false` and a record fails, the remainder of the DML operation can still succeed. This method returns a result object that you can use to verify which records succeeded, which failed, and why.

Return Value

Type: List<Approval.LockResult>

lock(recordToLock, allOrNothing)

Locks an object, with the option for partial success, and returns the lock result.

Signature

```
public static Approval.LockResult lock(SObject recordToLock, Boolean allOrNothing)
```

Parameters

recordToLock
Type: SObject

allOrNothing
Type: Boolean

Specifies whether this operation allows partial success. If you specify `false` and a record fails, the remainder of the DML operation can still succeed. This method returns a result object that you can use to verify which records succeeded, which failed, and why.

Return Value

Type: Approval.LockResult

lock(recordsToLock, allOrNothing)

Locks a set of objects, with the option for partial success. It returns the lock results, including failures.

Signature

```
public static List<Approval.LockResult> lock(List<SObject> recordsToLock, Boolean
allOrNothing)
```

Parameters

recordsToLock
Type: List<SObject>

allOrNothing
Type: Boolean

Specifies whether this operation allows partial success. If you specify `false` and a record fails, the remainder of the DML operation can still succeed. This method returns a result object that you can use to verify which records succeeded, which failed, and why.

Return Value

Type: List<Approval.LockResult>

process(approvalRequest)

Submits a new approval request and approves or rejects existing approval requests.

Signature

```
public static Approval.ProcessResult process(Approval.ProcessRequest approvalRequest)
```

Parameters

approvalRequest
Type: Approval.ProcessRequest

Return Value

Type: Approval.ProcessResult

Example

```
// Insert an account

Account a = new Account(Name='Test',
                    annualRevenue=100.0);

insert a;

// Create an approval request for the account
Approval.ProcessSubmitRequest req1 =
      new Approval.ProcessSubmitRequest();
req1.setObjectId(a.id);

// Submit the approval request for the account
Approval.ProcessResult result =
                  Approval.process(req1);
```

process(approvalRequest, allOrNone)

Submits a new approval request and approves or rejects existing approval requests.

Signature

```
public static Approval.ProcessResult process(Approval.ProcessRequest approvalRequest,
Boolean allOrNone)
```

Parameters

approvalRequest
Approval.ProcessRequest

allOrNone
Type: Boolean

The optional *allOrNone* parameter specifies whether the operation allows for partial success. If you specify `false` for this parameter and an approval fails, the remainder of the approval processes can still succeed.

Return Value

Approval.ProcessResult

process(approvalRequests)

Submits a list of new approval requests, and approves or rejects existing approval requests.

Signature

```
public static Approval.ProcessResult [] process(Approval.ProcessRequest[]
approvalRequests)
```

Parameters

approvalRequests
Approval.ProcessRequest []

Return Value

Approval.ProcessResult []

process(approvalRequests, allOrNone)

Submits a list of new approval requests, and approves or rejects existing approval requests.

Signature

```
public static Approval.ProcessResult [] process(Approval.ProcessRequest[]
approvalRequests, Boolean allOrNone)
```

Parameters

approvalRequests
Approval.ProcessRequest []

allOrNone
Type: Boolean

The optional *allOrNone* parameter specifies whether the operation allows for partial success. If you specify `false` for this parameter and an approval fails, the remainder of the approval processes can still succeed.

Return Value

Approval.ProcessResult []

unlock(recordId)

Unlocks an object, and returns the unlock results.

Signature

```
public static Approval.UnlockResult unlock(Id recordId)
```

Parameters

recordId
Type: Id

ID of the object to unlock.

Return Value

Type: Approval.UnlockResult

`unlock(recordIds)`

Unlocks a set of objects, and returns the unlock results, including failures.

Signature

```
public static List<Approval.UnlockResult> unlock(List<Id> recordIds)
```

Parameters

recordIds
Type: List<Id>

IDs of the objects to unlock.

Return Value

Type: List<Approval.UnlockResult>

`unlock(recordToUnlock)`

Unlocks an object, and returns the unlock results.

Signature

```
public static Approval.UnlockResult unlock(SObject recordToUnlock)
```

Parameters

recordToUnlock
Type: SObject

Return Value

Type: Approval.UnlockResult

`unlock(recordsToUnlock)`

Unlocks a set of objects, and returns the unlock results, including failures.

Signature

```
public static List<Approval.UnlockResult> unlock(List<SObject> recordsToUnlock)
```

Parameters

recordsToUnlock
Type: List<SObject>

Return Value

Type: List<Approval.UnlockResult>

`unlock(recordId, allOrNothing)`

Unlocks an object, with the option for partial success, and returns the unlock result.

Signature

```
public static Approval.UnlockResult unlock(Id recordId, Boolean allOrNothing)
```

Parameters

recordId
Type: Id

ID of the object to lock.

allOrNothing
Type: Boolean

Specifies whether this operation allows partial success. If you specify `false` and a record fails, the remainder of the DML operation can still succeed. This method returns a result object that you can use to verify which records succeeded, which failed, and why.

Return Value

Type: Approval.UnlockResult

`unlock(recordIds, allOrNothing)`

Unlocks a set of objects, with the option for partial success. It returns the unlock results, including failures.

Signature

```
public static List<Approval.UnlockResult> unlock(List<Id> recordIds, Boolean
allOrNothing)
```

Parameters

recordIds
Type: List<Id>

IDs of the objects to unlock.

allOrNothing
Type: Boolean

Specifies whether this operation allows partial success. If you specify `false` and a record fails, the remainder of the DML operation can still succeed. This method returns a result object that you can use to verify which records succeeded, which failed, and why.

Return Value

Type: List<Approval.UnlockResult>

unlock(recordToUnlock, allOrNothing)

Unlocks an object, with the option for partial success, and returns the unlock result.

Signature

```
public static Approval.UnlockResult unlock(SObject recordToUnlock, Boolean allOrNothing)
```

Parameters

recordToUnlock
Type: SObject

allOrNothing
Type: Boolean

Specifies whether this operation allows partial success. If you specify `false` and a record fails, the remainder of the DML operation can still succeed. This method returns a result object that you can use to verify which records succeeded, which failed, and why.

Return Value

Type: Approval.UnlockResult

unlock(recordsToUnlock, allOrNothing)

Unlocks a set of objects, with the option for partial success. It returns the unlock results, including failures.

Signature

```
public static List<Approval.UnlockResult> unlock(List<SObject> recordsToUnlock, Boolean
allOrNothing)
```

Parameters

recordsToUnlock
Type: List<SObject>

allOrNothing
Type: Boolean

Specifies whether this operation allows partial success. If you specify `false` and a record fails, the remainder of the DML operation can still succeed. This method returns a result object that you can use to verify which records succeeded, which failed, and why.

Return Value

Type: List<Approval.UnlockResult>

Assert Class

Contains methods to assert various conditions with test methods, such as whether two values are the same, a condition is true, or a variable is null.

Namespace

System

Assert Methods

The following are methods for `Assert`.

IN THIS SECTION:

areEqual(expected, actual, msg)
Asserts that the first two arguments are the same.

areEqual(expected, actual)
Asserts that the two arguments are the same.

areNotEqual(notExpected, actual, msg)
Asserts that the first two arguments aren't the same.

areNotEqual(notExpected, actual)
Asserts that the two arguments aren't the same.

fail(msg)
Immediately return a fatal error that causes code execution to halt.

fail()
Immediately return a fatal error that causes code execution to halt.

isFalse(condition, msg)
Asserts that the specified condition is `false`.

isFalse(condition)
Asserts that the specified condition is `false`.

isInstanceOfType(instance, expectedType, msg)
Asserts that the instance is of the specified type.

isInstanceOfType(instance, expectedType)
Asserts that the instance is of the specified type.

isNotInstanceOfType(instance, notExpectedType, msg)
Asserts that the instance isn't of the specified type.

isNotInstanceOfType(instance, notExpectedType)
Asserts that the instance isn't of the specified type.

isNotNull(value, msg)
Asserts that the value isn't null.

isNotNull(value)
Asserts that the value isn't null.

isNull(value, msg)
Asserts that the value is null.

isNull(value)
Asserts that the value is null.

isTrue(condition, msg)

Asserts that the specified condition is `true`.

isTrue(condition)

Asserts that the specified condition is `true`.

areEqual(expected, actual, msg)

Asserts that the first two arguments are the same.

Signature

```
public static void areEqual(Object expected, Object actual, String msg)
```

Parameters

`expected`

Type: Object

Expected value.

`actual`

Type: Object

Actual value.

`msg`

Type: String

(Optional) Custom message returned as part of the error message.

Return Value

Type: void

Usage

If the first two arguments aren't the same, a fatal error is returned that causes code execution to halt.

You can't catch an assertion failure using a try/catch block even though it's logged as an exception.

Example

```
String sub = 'abcde'.substring(2);
Assert.areEqual('cde', sub, 'Expected characters after first two'); // Succeeds
```

areEqual(expected, actual)

Asserts that the two arguments are the same.

Signature

```
public static void areEqual(Object expected, Object actual)
```

Parameters

expected
Type: Object

Expected value.

actual
Type: Object

Actual value.

Return Value

Type: void

Usage

If the two arguments aren't the same, a fatal error is returned that causes code execution to halt.

You can't catch an assertion failure using a try/catch block even though it's logged as an exception.

Example

```
String sub = 'abcde'.substring(2);
Assert.areEqual('cde', sub); // Succeeds
```

areNotEqual(notExpected, actual, msg)

Asserts that the first two arguments aren't the same.

Signature

```
public static void areNotEqual(Object notExpected, Object actual, String msg)
```

Parameters

notExpected
Type: Object

Value that's not expected.

actual
Type: Object

Actual value.

msg
Type: String

(Optional) Custom message returned as part of the error message.

Return Value

Type: void

Usage

If the first two arguments are the same, a fatal error is returned that causes code execution to halt.

You can't catch an assertion failure using a try/catch block even though it's logged as an exception.

Example

```
String sub = 'abcde'.substring(2);
Assert.areNotEqual('xyz', sub, 'Characters not expected after first two'); // Succeeds
```

areNotEqual(notExpected, actual)

Asserts that the two arguments aren't the same.

Signature

```
public static void areNotEqual(Object notExpected, Object actual)
```

Parameters

notExpected
Type: Object

Value that's not expected.

actual
Type: Object

Actual value.

Return Value

Type: void

Usage

If the two arguments are the same, a fatal error is returned that causes code execution to halt.

You can't catch an assertion failure using a try/catch block even though it's logged as an exception.

Example

```
String sub = 'abcde'.substring(2);
Assert.areNotEqual('xyz', sub); // Succeeds
```

fail(msg)

Immediately return a fatal error that causes code execution to halt.

Signature

```
public static void fail(String msg)
```

Parameters

msg
Type: String

(Optional) Custom message returned as part of the error message.

Return Value

Type: void

Usage

Commonly used in a try/catch block test case where an exception is expected to be thrown. You can't, however, catch the assertion failure in the try/catch block even though it's logged as an exception.

Example

```
// test case where exception is expected
try {
    SomeClass.methodUnderTest();
    Assert.fail('DmlException Expected');
} catch (DmlException ex) {
    // Add assertions here about the expected exception
}
```

fail()

Immediately return a fatal error that causes code execution to halt.

Signature

```
public static void fail()
```

Return Value

Type: void

Usage

Commonly used in a try/catch block test case where an exception is expected to be thrown. You can't, however, catch the assertion failure in the try/catch block even though it's logged as an exception.

Example

```
// test case where exception is expected
try {
    SomeClass.methodUnderTest();
    Assert.fail();
} catch (DmlException ex) {
    // Add assertions here about the expected exception
}
```

isFalse(condition, msg)

Asserts that the specified condition is `false`.

Signature

```
public static void isFalse(Boolean condition, String msg)
```

Parameters

`condition`
Type: Boolean

Condition you're checking to determine if it's `false`.

`msg`
Type: String

(Optional) Custom message returned as part of the error message.

Return Value

Type: void

Usage

If the condition is `true`, a fatal error is returned that causes code execution to halt.

You can't catch an assertion failure using a try/catch block even though it's logged as an exception.

Example

```
Boolean containsCode = 'Salesforce'.contains('code');
Assert.isFalse(containsCode, 'No code'); // Assertion succeeds
```

isFalse(condition)

Asserts that the specified condition is `false`.

Signature

```
public static void isFalse(Boolean condition)
```

Parameters

`condition`
Type: Boolean

Condition you're checking to determine if it's `false`.

Return Value

Type: void

Usage

If the condition is `true`, a fatal error is returned that causes code execution to halt.

You can't catch an assertion failure using a try/catch block even though it's logged as an exception.

Example

```
Boolean containsCode = 'Salesforce'.contains('code');
Assert.isFalse(containsCode); // Assertion succeeds
```

isInstanceOfType(instance, expectedType, msg)

Asserts that the instance is of the specified type.

Signature

```
public static void isInstanceOfType(Object instance, System.Type expectedType, String msg)
```

Parameters

instance
Type: Object

Instance whose type you're checking.

expectedType
Type: System.Type on page 3969

Expected type.

msg
Type: String

(Optional) Custom message returned as part of the error message.

Return Value

Type: void

Usage

If the instance isn't of the specified type, a fatal error is returned that causes code execution to halt.

You can't catch an assertion failure using a try/catch block even though it's logged as an exception.

Example

```
Account o = new Account();
Assert.isInstanceOfType(o, Account.class); // Succeeds
```

isInstanceOfType(instance, expectedType)

Asserts that the instance is of the specified type.

Signature

```
public static void isInstanceOfType(Object instance, System.Type expectedType)
```

Parameters

instance
Type: Object

Instance whose type you're checking.

expectedType
Type: System.Type on page 3969

Expected type.

Return Value

Type: void

Usage

If the instance isn't of the specified type, a fatal error is returned that causes code execution to halt.

You can't catch an assertion failure using a try/catch block even though it's logged as an exception.

Example

```
Account o = new Account();
Assert.isInstanceOfType(o, Account.class); // Succeeds
```

```
Account o = new Account();
Assert.isInstanceOfType(o, Account.class, 'Expected type.'); // Succeeds
```

isNotInstanceOfType(instance, notExpectedType, msg)

Asserts that the instance isn't of the specified type.

Signature

```
public static void isNotInstanceOfType(Object instance, System.Type notExpectedType,
String msg)
```

Parameters

instance
Type: Object

Instance whose type you're checking.

notExpectedType
Type: System.Type on page 3969

Type that's not expected.

msg
Type: String

(Optional) Custom message returned as part of the error message.

Return Value

Type: void

Usage

If the instance is of the specified type, a fatal error is returned that causes code execution to halt.

You can't catch an assertion failure using a try/catch block even though it's logged as an exception.

Example

```
Contact con = new Contact();
Assert.isNotInstanceOfType(con, Account.class, 'Not expected type'); // Succeeds
```

isNotInstanceOfType(instance, notExpectedType)

Asserts that the instance isn't of the specified type.

Signature

```
public static void isNotInstanceOfType(Object instance, System.Type notExpectedType)
```

Parameters

instance
Type: Object

Instance whose type you're checking.

notExpectedType
Type: System.Type on page 3969

Type that's not expected.

Return Value

Type: void

Usage

If the instance is of the specified type, a fatal error is returned that causes code execution to halt.

You can't catch an assertion failure using a try/catch block even though it's logged as an exception.

Example

```
Contact con = new Contact();
Assert.isNotInstanceOfType(con, Account.class); // Succeeds
```

isNotNull(value, msg)

Asserts that the value isn't null.

Signature

```
public static void isNotNull(Object value, String msg)
```

Parameters

value
Type: Object

Value you're checking to determine if it's not null.

msg
Type: String

(Optional) Custom message returned as part of the error message.

Return Value

Type: void

Usage

If the value is null, a fatal error is returned that causes code execution to halt.

You can't catch an assertion failure using a try/catch block even though it's logged as an exception.

Example

```
String myString = 'value';
Assert.isNotNull(myString, 'myString should not be null'); // Succeeds
```

isNotNull(value)

Asserts that the value isn't null.

Signature

```
public static void isNotNull(Object value)
```

Parameters

value
Type: Object

Value you're checking to determine if it's not null.

Return Value

Type: void

Usage

If the value is null, a fatal error is returned that causes code execution to halt.

You can't catch an assertion failure using a try/catch block even though it's logged as an exception.

Example

```
String myString = 'value';
Assert.isNotNull(myString); // Succeeds
```

isNull(value, msg)

Asserts that the value is null.

Signature

```
public static void isNull(Object value, String msg)
```

Parameters

value
Type: Object

Value you're checking to determine if it's null.

msg
Type: String

(Optional) Custom message returned as part of the error message.

Return Value

Type: void

Usage

If the value isn't null, a fatal error is returned that causes code execution to halt.

You can't catch an assertion failure using a try/catch block even though it's logged as an exception.

Example

```
String myString = null;
Assert.isNull(myString, 'String should be null'); // Succeeds
```

isNull(value)

Asserts that the value is null.

Signature

```
public static void isNull(Object value)
```

Parameters

value
Type: Object

Value you're checking to determine if it's null.

Return Value

Type: void

Usage

If the value isn't null, a fatal error is returned that causes code execution to halt.

You can't catch an assertion failure using a try/catch block even though it's logged as an exception.

Example

```
String myString = null;
Assert.isNull(myString); // Succeeds
```

isTrue(condition, msg)

Asserts that the specified condition is `true`.

Signature

```
public static void isTrue(Boolean condition, String msg)
```

Parameters

condition
Type: Boolean

Condition you're checking to determine if it's `true`.

msg
Type: String

(Optional) Custom message returned as part of the error message.

Return Value

Type: void

Usage

If the specified condition is `false`, a fatal error is returned that causes code execution to halt.

You can't catch an assertion failure using a try/catch block even though it's logged as an exception.

Example

```
Boolean containsForce = 'Salesforce'.contains('force');
Assert.isTrue(containsForce, 'Contains force'); // Assertion succeeds
```

isTrue(condition)

Asserts that the specified condition is `true`.

Signature

```
public static void isTrue(Boolean condition)
```

Parameters

`condition`
Type: Boolean

Condition you're checking to determine if it's `true`.

Return Value

Type: void

Usage

If the specified condition is `false`, a fatal error is returned that causes code execution to halt.

You can't catch an assertion failure using a try/catch block even though it's logged as an exception.

Example

```
Boolean containsForce = 'Salesforce'.contains('force');
Assert.isTrue(containsForce); // Assertion succeeds
```

AsyncInfo Class

Provides methods to get the current stack depth, maximum stack depth, and the minimum queueable delay for Queueable transactions, and to determine if maximum stack depth is set.

Namespace

System

IN THIS SECTION:

AsyncInfo Methods

AsyncInfo Methods

The following are methods for `AsyncInfo`.

IN THIS SECTION:

getCurrentQueueableStackDepth()
Get the current queueable stack depth for queueable transactions.

getMaximumQueueableStackDepth()
Get the maximum queueable stack depth for queueable transactions.

getMinimumQueueableDelayInMinutes()
Get the minimum queueable delay for queueable transactions (in minutes).

hasMaxStackDepth()
Determine if maximum stack depth is set for your queueable requests.

`getCurrentQueueableStackDepth()`

Get the current queueable stack depth for queueable transactions.

Signature

```
public static Integer getCurrentQueueableStackDepth()
```

Return Value

Type: Integer

`getMaximumQueueableStackDepth()`

Get the maximum queueable stack depth for queueable transactions.

Signature

```
public static Integer getMaximumQueueableStackDepth()
```

Return Value

Type: Integer

`getMinimumQueueableDelayInMinutes()`

Get the minimum queueable delay for queueable transactions (in minutes).

Signature

```
public static Integer getMinimumQueueableDelayInMinutes()
```

Return Value

Type: Integer

Returns null if no delay is defined.

`hasMaxStackDepth()`

Determine if maximum stack depth is set for your queueable requests.

Signature

```
public static Boolean hasMaxStackDepth()
```

Return Value

Type: Boolean

AsyncOptions Class

Contains maximum stack depths for queueable transactions and the minimum queueable delay in minutes. Passed as parameter to the `System.enqueueJob()` method to define a unique queueable job signature, the maximum stack depth for queueable transactions and the minimum queueable delay in minutes.

Namespace

System

IN THIS SECTION:

AsyncOptions Properties

AsyncOptions Properties

The following are properties for `AsyncOptions`.

IN THIS SECTION:

DuplicateSignature

A unique signature for a Queueable job.

MaximumQueueableStackDepth

Maximum stack depth for queueable transactions.

MinimumQueueableDelayInMinutes

Minimum queueable delay for queueable transactions.

`DuplicateSignature`

A unique signature for a Queueable job.

Signature

```
public System.QueueableDuplicateSignature DuplicateSignature {get; set;}
```

Property Value

Type: QueueableDuplicateSignature Class

`MaximumQueueableStackDepth`

Maximum stack depth for queueable transactions.

Signature

```
public Integer MaximumQueueableStackDepth {get; set;}
```

Property Value

Type: Integer

`MinimumQueueableDelayInMinutes`

Minimum queueable delay for queueable transactions.

Signature

```
public Integer MinimumQueueableDelayInMinutes {get; set;}
```

Property Value

Type: Integer

Blob Class

Contains methods for the Blob primitive data type.

Namespace

System

Usage

Salesforce supports Blob manipulation only with Apex class methods that are supplied by Salesforce. For more information on Blobs, see Primitive Data Types.

Blob Methods

The following are methods for `Blob`.

IN THIS SECTION:

size()
Returns the number of bytes in the Blob.

toPdf(stringToConvert)
Creates a binary object out of the given string, encoding it as a PDF file.

toString()
Casts the Blob into a String.

valueOf(stringToBlob)
Casts the specified String to a Blob.

size()

Returns the number of bytes in the Blob.

Signature

```
public Integer size()
```

Return Value

Type: Integer

Example

```
String myString = 'StringToBlob';
Blob myBlob = Blob.valueof(myString);
Integer size = myBlob.size();
```

toPdf(stringToConvert)

Creates a binary object out of the given string, encoding it as a PDF file.

Signature

```
public static Blob toPdf(String stringToConvert)
```

Parameters

stringToConvert
Type: String

Note: Referencing a static resource throws an InvalidParameterValue exception.

Return Value

Type: Blob

Usage

`Blob.toPDF(stringToConvert)` doesn't support web fonts, so you can't specify a font for the string.

`Blob.toPDF(stringToConvert)` also doesn't support multibyte characters, such as Japanese and accented international characters. To render multibyte characters in a PDF file, consider adding the string to a Visualforce page, and then rendering the page as a PDF file. See Render a Visualforce Page as a PDF File.

Example

```
String pdfContent = 'This is a test string';
Account a = new account(name = 'test');
insert a;
Attachment attachmentPDF = new Attachment();
attachmentPdf.parentId = a.id;
attachmentPdf.name = a.name + '.pdf';
attachmentPdf.body = Blob.toPDF(pdfContent);
insert attachmentPDF;
```

`toString()`

Casts the Blob into a String.

Signature

```
public String toString()
```

Return Value

Type: String

Example

```
String myString = 'StringToBlob';
Blob myBlob = Blob.valueof(myString);
System.assertEquals('StringToBlob', myBlob.toString());
```

`valueOf(stringToBlob)`

Casts the specified String to a Blob.

Signature

```
public static Blob valueOf(String stringToBlob)
```

Parameters

`stringToBlob`
Type: String

Return Value

Type: Blob

Example

```
String myString = 'StringToBlob';
Blob myBlob = Blob.valueof(myString);
```

Boolean Class

Contains methods for the Boolean primitive data type.

Namespace

System

Boolean Methods

The following are methods for `Boolean`. All methods are static.

IN THIS SECTION:

valueOf(stringToBoolean)
Converts the specified string to a Boolean value and returns `true` if the specified string value is `true`. Otherwise, returns `false`.

valueOf(fieldValue)
Converts the specified object to a Boolean value. Use this method to convert a history tracking field value or an object that represents a Boolean value.

`valueOf(stringToBoolean)`

Converts the specified string to a Boolean value and returns `true` if the specified string value is `true`. Otherwise, returns `false`.

Signature

```
public static Boolean valueOf(String stringToBoolean)
```

Parameters

`stringToBoolean`
Type: String

Return Value

Type: Boolean

Usage

If the specified argument is null, this method throws an exception.

Example

```
Boolean b = Boolean.valueOf('true');
System.assertEquals(true, b);
```

`valueOf(fieldValue)`

Converts the specified object to a Boolean value. Use this method to convert a history tracking field value or an object that represents a Boolean value.

Signature

```
public static Boolean valueOf(Object fieldValue)
```

Parameters

`fieldValue`
Type: Object

Return Value

Type: Boolean

Usage

Use this method with the `OldValue` or `NewValue` fields of history sObjects, such as `AccountHistory`, when the field type corresponds to a Boolean type, like a checkbox field.

Example

```
List<AccountHistory> ahlist =
     [SELECT Field,OldValue,NewValue FROM AccountHistory];
for(AccountHistory ah : ahlist) {
   System.debug('Field: ' + ah.Field);
   if (ah.field == 'IsPlatinum__c') {
      Boolean oldValue = Boolean.valueOf(ah.OldValue);
      Boolean newValue = Boolean.valueOf(ah.NewValue);
   }
}
```

BusinessHours Class

Use the `BusinessHours` methods to set the business hours at which your customer support team operates.

Namespace

System

BusinessHours Methods

The following are methods for `BusinessHours`. All methods are static.

IN THIS SECTION:

add(businessHoursId, startDate, intervalMilliseconds)
Adds an interval of time from a start Datetime traversing business hours only. Returns the result Datetime in the local time zone.

addGmt(businessHoursId, startDate, intervalMilliseconds)
Adds an interval of milliseconds from a start Datetime traversing business hours only. Returns the result Datetime in GMT.

diff(businessHoursId, startDate, endDate)
Returns the difference in milliseconds between a start and end Datetime based on a specific set of business hours.

isWithin(businessHoursId, targetDate)
Returns `true` if the specified target date occurs within business hours. Holidays are included in the calculation.

nextStartDate(businessHoursId, targetDate)
Starting from the specified target date, returns the next date when business hours are open. If the specified target date falls within business hours, this target date is returned.

add(businessHoursId, startDate, intervalMilliseconds)

Adds an interval of time from a start Datetime traversing business hours only. Returns the result Datetime in the local time zone.

Signature

```
public static Datetime add(String businessHoursId, Datetime startDate, Long
intervalMilliseconds)
```

Parameters

businessHoursId
Type: String

startDate
Type: Datetime

intervalMilliseconds
Type: Long

Interval value should be provided in milliseconds, however time precision smaller than one minute is ignored.

Return Value

Type: Datetime

addGmt(businessHoursId, startDate, intervalMilliseconds)

Adds an interval of milliseconds from a start Datetime traversing business hours only. Returns the result Datetime in GMT.

Signature

```
public static Datetime addGmt(String businessHoursId, Datetime startDate, Long
intervalMilliseconds)
```

Parameters

businessHoursId
Type: String

startDate
Type: Datetime

intervalMilliseconds
Type: Long

Return Value

Type: Datetime

diff(businessHoursId, startDate, endDate)

Returns the difference in milliseconds between a start and end Datetime based on a specific set of business hours.

Signature

```
public static Long diff(String businessHoursId, Datetime startDate, Datetime endDate)
```

Parameters

businessHoursId
Type: String

startDate
Type: Datetime

endDate
Type: Datetime

Return Value

Type: Long

isWithin(businessHoursId, targetDate)

Returns `true` if the specified target date occurs within business hours. Holidays are included in the calculation.

Signature

```
public static Boolean isWithin(String businessHoursId, Datetime targetDate)
```

Parameters

businessHoursId
Type: String

The business hours ID.

targetDate
Type: Datetime

The date to verify.

Return Value

Type: Boolean

Example

The following example finds whether a given time is within the default business hours.

```
// Get the default business hours
BusinessHours bh = [SELECT Id FROM BusinessHours WHERE IsDefault=true];

// Create Datetime on May 28, 2013 at 1:06:08 AM in the local timezone.
Datetime targetTime = Datetime.newInstance(2013, 5, 28, 1, 6, 8);

// Find whether the time is within the default business hours
Boolean isWithin= BusinessHours.isWithin(bh.id, targetTime);
```

nextStartDate(businessHoursId, targetDate)

Starting from the specified target date, returns the next date when business hours are open. If the specified target date falls within business hours, this target date is returned.

Signature

```
public static Datetime nextStartDate(String businessHoursId, Datetime targetDate)
```

Parameters

businessHoursId
Type: String

The business hours ID.

targetDate
Type: Datetime

The date used as a start date to obtain the next date.

Return Value

Type: Datetime

Example

The following example finds the next date starting from the target date when business hours reopens. If the target date is within the given business hours, the target date is returned. The returned time is in the local time zone.

```
// Get the default business hours
BusinessHours bh = [SELECT Id FROM BusinessHours WHERE IsDefault=true];

// Create Datetime on May 28, 2013 at 1:06:08 AM in the local timezone.
Datetime targetTime = Datetime.newInstance(2013, 5, 28, 1, 6, 8);
// Starting from the targetTime, find the next date when business hours reopens. Return
the target time.

// if it is within the business hours. The returned time will be in the local time zone
Datetime nextStart = BusinessHours.nextStartDate(bh.id, targetTime);
```

CallbackStatus Enum

Specifies the status of asynchronous requests to an external system.

Enum Values

The following are the values of the `System.CallbackStatus` enum.

Value	Description
CANCELLED	The asynchronous operation has been cancelled.
COMPLETED	The asynchronous operation has been completed.
PENDING	The asynchronous operation is in progress.
TIMED_OUT	The asynchronous operation has timed out.

Callable Interface

Enables developers to use a common interface to build loosely coupled integrations between Apex classes or triggers, even for code in separate packages. Agreeing upon a common interface enables developers from different companies or different departments to build upon one another's solutions. Implement this interface to enable the broader community, which might have different solutions than the ones you had in mind, to extend your code's functionality.

Note: This interface is not an analog of the Java Callable interface, which is used for asynchronous invocation. Don't confuse the two.

Namespace

System

Usage

To implement the `Callable` interface, you need to write only one method: `call(String action, Map<String, Object> args)`.

In code that utilizes or tests an implementation of `Callable`, cast an instance of your type to `Callable`.

This interface is not intended to replace defining more specific interfaces. Rather, the `Callable` interface allows integrations in which code from different classes or packages can use common base types.

IN THIS SECTION:

Callable Methods

Callable Example Implementation

Callable Methods

The following are methods for `Callable`.

IN THIS SECTION:

call(action, args)
Provides functionality that other classes or packages can utilize and build upon.

`call(action, args)`

Provides functionality that other classes or packages can utilize and build upon.

Signature

```
public Object call(String action, Map<String,Object> args)
```

Parameters

`action`
Type: String

The behavior for the method to exhibit.

`args`
Type: Map on page 3624<String,Object>

Arguments to be used by the specified action.

Return Value

Type: Object

The result of the method invocation.

Callable Example Implementation

This class is an example implementation of the `System.Callable` interface.

```
public class Extension implements Callable {

   // Actual method
   String concatStrings(String stringValue) {
     return stringValue + stringValue;
   }

   // Actual method
   Decimal multiplyNumbers(Decimal decimalValue) {
     return decimalValue * decimalValue;
   }

   // Dispatch actual methods
   public Object call(String action, Map<String, Object> args) {
     switch on action {
       when 'concatStrings' {
         return this.concatStrings((String)args.get('stringValue'));
       }
       when 'multiplyNumbers' {
         return this.multiplyNumbers((Decimal)args.get('decimalValue'));
       }
       when else {
        throw new ExtensionMalformedCallException('Method not implemented');
       }
     }
   }

   public class ExtensionMalformedCallException extends Exception {}
}
```

The following test code illustrates how calling code utilizes the interface to call a method.

```
@IsTest
private with sharing class ExtensionCaller {

   @IsTest
   private static void givenConfiguredExtensionWhenCalledThenValidResult() {

      // Given
      String extensionClass = 'Extension'; // Typically set via configuration
      Decimal decimalTestValue = 10;

      // When
      Callable extension =
         (Callable) Type.forName(extensionClass).newInstance();
      Decimal result = (Decimal)
         extension.call('multiplyNumbers', new Map<String, Object> {
            'decimalValue' => decimalTestValue
         });

      // Then
```

```
        System.assertEquals(100, result);
    }
}
```

SEE ALSO:

Apex Developer Guide: Classes and Casting

Cases Class

Use the `Cases` class to interact with case records.

Namespace

System

Cases Methods

The following are static methods for `Cases`.

IN THIS SECTION:

generateThreadingMessageId(caseId)
Returns an RFC 2822-compliant message identifier that contains information used to match the email and its replies to a case.

getCaseIdFromEmailHeaders(headers)
Returns the case ID corresponding to the specified email header information, or returns null if none is found.

getCaseIdFromEmailThreadId(emailThreadId)
Returns the case ID corresponding to the specified email thread ID. **(Deprecated. Use getCaseIdFromEmailHeaders and EmailMessages.getRecordIdFromEmail instead.)**

`generateThreadingMessageId(caseId)`

Returns an RFC 2822-compliant message identifier that contains information used to match the email and its replies to a case.

Signature

```
public static String generateThreadingMessageId(Id caseId)
```

Parameters

`caseId`
Type: Id

The case SObject ID to which replies to this email should be attached.

Return Value

Type: String

Usage

Use the returned message identifier when sending case-related emails in Apex. The returned message identifier can be used in Message-ID or References headers. However, because Salesforce doesn't let users specify the Message-ID, we set this identifier in the References header. When users reply to the sent email, replies should be attached to the specified case.

Example

In this sample, we create an email with a message identifier so that the email and any responses can be associated with the related case.

```
//Get your Case ID. Here we use a dummy ID
ID caseId = Id.valueOf('500xx000000bpkTAAQ');
//Create a SingleEmailMessage object
Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
//Set recipients and other fields
email.setToAddresses(new String[] {'test@salesforce.com'});
email.setPlainTextBody('Test Email Notification');
//.......... more fields ..........
//Get the threading message identifier
String messageId = Cases.generateThreadingMessageId(caseId);
//Insert the message identifier into the References header
email.setReferences(messageId);
//Send out the email
Messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
```

`getCaseIdFromEmailHeaders(headers)`

Returns the case ID corresponding to the specified email header information, or returns null if none is found.

Signature

```
public static Id getCaseIdFromEmailHeaders(List<Messaging.InboundEmail.Header> headers)
```

Parameters

`headers`
Type: List<Messaging.InboundEmail.Header>

Return Value

Type: Id

Usage

To optimize finding a match between email threads and cases in your custom code, we recommend that you use this method and `EmailMessages.getRecordIdFromEmail` to implement a combination of token- and header-based threading.

If you are transitioning from Ref ID threading, we recommend that you replace `Cases.getCaseIdFromEmailThreadId` with a combination of `Cases.getCaseIdFromEmailHeaders` and `EmailMessages.getRecordIdFromEmail`. If you choose to implement header-based threading only, replace `Cases.getCaseIdFromEmailThreadId` with `Cases.getCaseIdFromEmailHeaders`.

The *`headers`* argument is used to find the matching Case Id using values for the `In-Reply-To` and `References` headers based on RFC 2822. If Email-to-Case can't find any emails with a matching `In-Reply-To` or `References` header, it also checks

the incoming email for an Outlook-specific header called `Thread-Index`. The first 22 bytes of this header uniquely identify the thread. If Email-to-Case detects a `Thread-Index` header on the incoming mail, it looks for matching information in the ClientThreadIdentifier field in EmailMessage records. If a match is found, the customer's reply email is linked to the related case.

Typically this method is used in Email Services so that you can provide your own handling of inbound emails using Apex code.

Example

If you implement header-based threading in your Email Services currently, we recommend that you use Lightning threading, which combines token-based threading and header-based threading. For header-based threading to continue to work, store emails as EmailMessage records with the MessagedIdentifier field set properly. With Lightning threading, you can use threading tokens as the primary threading method and rely on header-based threading as a fallback, or vice versa.

In this example, we rely on threading tokens and use header-based threading as a fallback.

```
global class AttachEmailMessageToCaseExample implements Messaging.InboundEmailHandler {
    global Messaging.InboundEmailResult handleInboundEmail(Messaging.inboundEmail email,

                Messaging.InboundEnvelope env) {

        // Create an InboundEmailResult object for returning the result of the
        // Apex Email Service.
        Messaging.InboundEmailResult result = new Messaging.InboundEmailResult();

        // Try to find the Case ID using threading tokens in email attributes.
       Id caseId = EmailMessages.getRecordIdFromEmail(email.subject, email.plainTextBody,
 email.htmlBody);

        // If we haven't found the Case ID, try finding it using headers.
        if (caseId == null) {
            caseId = Cases.getCaseIdFromEmailHeaders(email.headers);
        }

        // If a Case isn't found, create a new Case record.
        if (caseId == null) {
            Case c = new Case(Subject = email.subject);
            insert c;
            System.debug('New Case Object: ' + c);
            caseId = c.Id;
        }

        // Process recipients
        String toAddresses;
        if (email.toAddresses != null) {
            toAddresses = String.join(email.toAddresses, '; ');
        }

        // To store an EmailMessage for threading, you need at minimum
        // the Status, the MessageIdentifier, and the ParentId fields.
        EmailMessage em = new EmailMessage(
            Status = '0',
            MessageIdentifier = email.messageId,
            ParentId = caseId,
            // Other important fields.
            FromAddress = email.fromAddress,
```

```
            FromName = email.fromName,
            ToAddress = toAddresses,
            TextBody = email.plainTextBody,
            HtmlBody = email.htmlBody,
            Subject = email.subject
            // Other fields you wish to add.
        );

        // Insert the new EmailMessage.
        insert em;
        System.debug('New EmailMessage Object: ' + em );

    // Set the result to true. No need to send an email back to the user
    // with an error message.
    result.success = true;

    // Return the result for the Apex Email Service.
    return result;
  }
}
```

`getCaseIdFromEmailThreadId(emailThreadId)`

Returns the case ID corresponding to the specified email thread ID. **(Deprecated. Use getCaseIdFromEmailHeaders and EmailMessages.getRecordIdFromEmail instead.)**

Signature

```
public static ID getCaseIdFromEmailThreadId(String emailThreadId)
```

Parameters

`emailThreadId`
Type: String

Return Value

Type: ID

Usage

The argument for emailThreadId, also known as Ref ID, has the format `!00Dxx01gEW.!500xx0Yktl`. This format was introduced in the Winter '24 release. The previous format, `_00Dxx1gEW._500xxYktl`, is supported for backward compatibility, but emails sent from the Winter '24 release onward use the new format. Other formats that include `ref:` or `[ref:` aren't supported by this method.

Collator Class

Contains methods to get locale-specific instances that can be used for comparisons and sorting. Use the `getInstance()` method to obtain the Collator instance for a given locale and pass the Collator as the Comparator parameter to the `list.sort()` method.

Namespace

System

Usage

Because locale-sensitive sorting can produce different results depending on the user running the code, avoid using it in triggers or in code that expects a particular sort order.

Example

This example performs a default list sort and then uses Collator to sort based on the user locale.

```
@IsTest
    static void userLocaleSort() {

        string userLocale = 'fr_FR';

        User u = new User(Alias = 'standt', Email='standarduser@testorg.com',
        EmailEncodingKey='UTF-8', LastName='Testing', LanguageLocaleKey='en_US',
        LocaleSidKey=userLocale, TimeZoneSidKey='America/Los_Angeles',
        ProfileId = [SELECT Id FROM Profile WHERE Name='Standard User'].Id,
        UserName='standarduser' + DateTime.now().getTime() + '@testorg.com');

        System.runAs(u) {

            List<String> shoppingList = new List<String> {
                'épaule désosé Agneau',
                'Juice',
                'à la mélasse Galette 5 kg',
                'Bread',
                'Grocery'
            };

            // Default sort
            shoppingList.sort();
            Assert.areEqual('Bread', shoppingList[0]);

            // Sort based on user Locale
            Collator myCollator = Collator.getInstance();
            shoppingList.sort(myCollator);
            Assert.areEqual('à la mélasse Galette 5 kg', shoppingList[0]);
            Assert.areEqual('Bread', shoppingList[1]);
            Assert.areEqual('épaule désosé Agneau', shoppingList[2]);
            Assert.areEqual('Grocery', shoppingList[3]);
            Assert.areEqual('Juice', shoppingList[4]);
        }
    }
```

IN THIS SECTION:

Collator Methods

Collator Methods

The following are methods for `Collator`.

IN THIS SECTION:

compare(source, target)
Perform string comparisons for a given locale.

getInstance()
Gets the Collator instance for the current user's locale.

`compare(source, target)`

Perform string comparisons for a given locale.

Signature

```
public Integer compare(String source, String target)
```

Parameters

source
Type: String

target
Type: String

Return Value

Type: Integer

`getInstance()`

Gets the Collator instance for the current user's locale.

Signature

```
public static System.Collator getInstance()
```

Return Value

Type: Collator Class

Comparable Interface

Adds sorting support for Lists that contain non-primitive types, that is, Lists of user-defined types. Your implementation must explicitly handle null inputs in the `compareTo()` method to avoid a null pointer exception.

Namespace

System

Usage

To add List sorting support for your Apex class, you must implement the `Comparable` interface with its `compareTo` method in your class.

To implement the `Comparable` interface, you must first declare a class with the `implements` keyword as follows:

```
public class Employee implements Comparable {
```

Next, your class must provide an implementation for the following method:

```
public Integer compareTo(Object compareTo) {
    // Your code here
}
```

The implemented method must be declared as `global` or `public`.

IN THIS SECTION:

- Comparable Methods
- Comparable Example Implementation

SEE ALSO:

- List Class

Comparable Methods

The following are methods for `Comparable`.

IN THIS SECTION:

- compareTo(objectToCompareTo)
  Returns an Integer value that is the result of the comparison.

compareTo(objectToCompareTo)

Returns an Integer value that is the result of the comparison.

Signature

```
public Integer compareTo(Object objectToCompareTo)
```

Parameters

objectToCompareTo
Type: Object

Return Value

Type: Integer

Usage

The implementation of this method returns the following values:

- 0 if this instance and *objectToCompareTo* are equal
- > 0 if this instance is greater than *objectToCompareTo*
- < 0 if this instance is less than *objectToCompareTo*

If this object instance and *objectToCompareTo* are incompatible, a `System.TypeException` is thrown.

Comparable Example Implementation

This example implements the `Comparable` interface. The `compareTo` method in this example compares the employee of this class instance with the employee passed in the argument. The method returns an Integer value based on the comparison of the employee IDs.

```
public class Employee implements Comparable {

    public Long id;
    public String name;
    public String phone;

    // Constructor
    public Employee(Long i, String n, String p) {
        id = i;
        name = n;
        phone = p;
    }

    // Implement the compareTo() method
    public Integer compareTo(Object compareTo) {
        Employee compareToEmp = (Employee)compareTo;
        if (id == compareToEmp.id) return 0;
        if (id > compareToEmp.id) return 1;
        return -1;
    }
}
```

This example tests the sort order of a list of `Employee` objects.

```
@isTest
private class EmployeeSortingTest {
    @isTest
    static void test1() {
        List<Employee> empList = new List<Employee>();
        empList.add(new Employee(101,'Joe Smith', '4155551212'));
        empList.add(new Employee(101,'J. Smith', '4155551212'));
        empList.add(new Employee(25,'Caragh Smith', '4155551000'));
        empList.add(new Employee(105,'Mario Ruiz', '4155551099'));

        // Sort using the custom compareTo() method
```

```
        empList.sort();

        // Write list contents to the debug log
        System.debug(empList);

        // Verify list sort order.
        Assert.areEqual('Caragh Smith', empList[0].Name);
        Assert.areEqual('Joe Smith', empList[1].Name);
        Assert.areEqual('J. Smith', empList[2].Name);
        Assert.areEqual('Mario Ruiz', empList[3].Name);
    }
}
```

Comparator Interface

Implement different sort orders with the Comparator interface's `compare()` method, and pass the Comparator as a parameter to `List.sort()`. Your implementation must explicitly handle null inputs in the `compare()` method to avoid a null pointer exception.

Namespace

System

IN THIS SECTION:

Comparator Methods

Comparator Example Implementation
Use the Comparator interface to impose different kinds of sorting.

Comparator Methods

The following are methods for `Comparator`.

IN THIS SECTION:

compare(var1, var2)
Compares the two arguments and returns a negative integer, zero, or a positive integer depending on whether the first argument is less than, equal to, or greater than the second argument.

`compare(var1, var2)`

Compares the two arguments and returns a negative integer, zero, or a positive integer depending on whether the first argument is less than, equal to, or greater than the second argument.

Signature

```
public Integer compare(T var1, T var2)
```

Parameters

var1

Type: T

T - The type determined by the parameterized type of the Comparator. For example, if the class implements `Comparator<Account>` then *var1* and *var2* are of type Account .

var2

Type: T

T - The type determined by the parameterized type of the Comparator. For example, if the class implements `Comparator<Account>` then *var1* and *var2* are of type Account .

Return Value

Type: Integer

Comparator Example Implementation

Use the Comparator interface to impose different kinds of sorting.

This example implements two different ways of sorting employees.

```
public class Employee {

    private Long id;
    private String name;
    private Integer yearJoined;

    // Constructor
    public Employee(Long i, String n, Integer y) {
        id = i;
        name = n;
        yearJoined = y;
    }

    public String getName() { return name; }
    public Integer getYear() { return yearJoined; }
}
```

```
// Class to compare Employees by name
    public class NameCompare implements Comparator<Employee> {
        public Integer compare(Employee e1, Employee e2) {
            if(e1?.getName() == null && e2?.getName() == null) {
                return 0;
            } else if(e1?.getName() == null) {
                return -1;
            } else if(e2?.getName() == null) {
                return 1;
            }
            return e1.getName().compareTo(e2.getName());
            }
        }

    // Class to compare Employees by year joined
```

```
    public class YearCompare implements Comparator<Employee> {
        public Integer compare(Employee e1, Employee e2) {
            // Guard against null operands for '<' or '>' operators because
            // they will always return false and produce inconsistent sorting
            Integer result;
            if(e1?.getYear() == null && e2?.getYear() == null) {
                result = 0;
            } else if(e1?.getYear() == null) {
                  result = -1;
            } else if(e2?.getYear() == null) {
                  result = 1;
            } else if (e1.getYear() < e2.getYear()) {
                  result = -1;
            } else if (e1.getYear() > e2.getYear()) {
                  result = 1;
            } else {
                  result = 0;
            }
        return result;
        }
    }
```

The following example tests the implementation:

```
@isTest
private class EmployeeSortingTest {
    @isTest
    static void sortWithComparators() {
        List<Employee> empList = new List<Employee>();
        empList.add(new Employee(101,'Joe Smith', 2020));
        empList.add(new Employee(102,'J. Smith', 2020));
        empList.add(new Employee(25,'Caragh Smith', 2021));
        empList.add(new Employee(105,'Mario Ruiz', 2019));
        // Sort by name
        NameCompare nameCompare = new NameCompare();
        empList.sort(nameCompare);
        // Expected order: Caragh Smith, J. Smith, Joe Smith, Mario Ruiz
        Assert.areEqual('Caragh Smith', empList.get(0).getName());

        // Sort by year joined
        YearCompare yearCompare = new YearCompare();
        empList.sort(yearCompare);
        // Expected order: Mario Ruiz, J. Smith, Joe Smith, Caragh Smith
        Assert.areEqual('Mario Ruiz', empList.get(0).getName());
    }
}
```

Continuation Class

Use the `Continuation` class to make callouts asynchronously to a SOAP or REST Web service.

Namespace

System

Example

For a code example, see Make Long-Running Callouts from a Visualforce Page.

IN THIS SECTION:

Continuation Constructors
Continuation Properties
Continuation Methods

Continuation Constructors

The following are constructors for `Continuation`.

IN THIS SECTION:

Continuation(timeout)
Creates an instance of the `Continuation` class by using the specified timeout in seconds. The timeout maximum is 120 seconds.

`Continuation(timeout)`

Creates an instance of the `Continuation` class by using the specified timeout in seconds. The timeout maximum is 120 seconds.

Signature

```
public Continuation(Integer timeout)
```

Parameters

`timeout`
Type: Integer

The timeout for this continuation in seconds.

Continuation Properties

The following are properties for `Continuation`.

IN THIS SECTION:

continuationMethod
The name of the callback method that is called after the callout response returns.
timeout
The timeout of the continuation in seconds. Maximum: 120 seconds.

state
Data that is stored in this continuation and that can be retrieved after the callout is finished and the callback method is invoked.

`continuationMethod`

The name of the callback method that is called after the callout response returns.

Signature

```
public String continuationMethod {get; set;}
```

Property Value

Type: String

Usage

Note: If the `continuationMethod` property is not set for a Continuation, the same action method that made the asynchronous callout is called again when the callout response returns.

`timeout`

The timeout of the continuation in seconds. Maximum: 120 seconds.

Signature

```
public Integer timeout {get; set;}
```

Property Value

Type: Integer

`state`

Data that is stored in this continuation and that can be retrieved after the callout is finished and the callback method is invoked.

Signature

```
public Object state {get; set;}
```

Property Value

Type: Object

Example

This example shows how to save state information for a continuation in a controller.

```
// Declare inner class to hold state info
private class StateInfo {
    String msg { get; set; }
```

```
    List<String> urls  { get; set; }
    StateInfo(String msg, List<String> urls) {
        this.msg = msg;
        this.urls = urls;
    }
}

// Then in the action method, set state for the continuation
continuationInstance.state = new StateInfo('Some state data', urls);
```

Continuation Methods

The following are methods for `Continuation`.

IN THIS SECTION:

addHttpRequest(request)
Adds the HTTP request for the callout that is associated with this continuation.

getRequests()
Returns all labels and requests that are associated with this continuation as key-value pairs.

getResponse(requestLabel)
Returns the response for the request that corresponds to the specified label.

addHttpRequest(request)

Adds the HTTP request for the callout that is associated with this continuation.

Signature

```
public String addHttpRequest(System.HttpRequest request)
```

Parameters

request
Type: HttpRequest

The HTTP request to be sent to the external service by this continuation.

Return Value

Type: String

A unique label that identifies the HTTP request that is associated with this continuation. This label is used in the map that getRequests() returns to identify individual requests in a continuation.

Usage

You can add up tothree requests to a continuation.

Note: The timeout that is set in each passed-in request is ignored. Only the global timeout maximum of 120 seconds applies for a continuation.

`getRequests()`

Returns all labels and requests that are associated with this continuation as key-value pairs.

Signature

```
public Map<String,System.HttpRequest> getRequests()
```

Return Value

Type: Map<String,HttpRequest>

A map of all requests that are associated with this continuation. The map key is the request label, and the map value is the corresponding HTTP request.

`getResponse(requestLabel)`

Returns the response for the request that corresponds to the specified label.

Signature

```
public static HttpResponse getResponse(String requestLabel)
```

Parameters

`requestLabel`
Type: String

The request label to get the response for.

Return Value

Type: HttpResponse

Usage

The status code is returned in the HttpResponse object and can be obtained by calling `getStatusCode()` on the response. A status code of `200` indicates that the request was successful. Other status code values indicate the type of problem that was encountered.

Sample of Error Status Codes

When a problem occurs with the response, some possible status code values are:

- `2000`: The timeout was reached, and the server didn't get a chance to respond.
- `2001`: There was a connection failure.
- `2002`: Exceptions occurred.
- `2003`: The response hasn't arrived (which also means that the Apex asynchronous callout framework hasn't resumed).
- `2004`: The response size is too large (greater than 1 MB).

Cookie Class

The `Cookie` class lets you access cookies for your Salesforce site using Apex.

Namespace

System

Usage

Use the `setCookies` method of the PageReference Class to attach cookies to a page.

Important:

- Cookie names and values set in Apex are URL encoded, that is, characters such as @ are replaced with a percent sign and their hexadecimal representation.
- The `setCookies` method adds the prefix "`apex__`" to the cookie names.
- Setting a cookie's value to `null` sends a cookie with an empty string value instead of setting an expired attribute.
- After you create a cookie, the properties of the cookie can't be changed.
- Be careful when storing sensitive information in cookies. Pages are cached regardless of a cookie value. If you use a cookie value to generate dynamic content, you should disable page caching. For more information, see Configure Site Caching in Salesforce Help.

Consider the following limitations when using the `Cookie` class:

- The `Cookie` class can only be accessed using Apex that is saved using the Salesforce API version 19 and above.
- The maximum number of cookies that can be set per Salesforce Sites domain depends on your browser. Newer browsers have higher limits than older ones.
- Cookies must be less than 4K, including name and attributes.
- The maximum header size of a Visualforce page, including cookies, is 8,192 bytes.

For more information on sites, see "Salesforce Sites" in the Salesforce online help.

Example

The following example creates a class, `CookieController`, which is used with a Visualforce page (see markup below) to update a counter each time a user displays a page. The number of times a user goes to the page is stored in a cookie.

```
// A Visualforce controller class that creates a cookie
// used to keep track of how often a user displays a page
public class CookieController {

    public CookieController() {
        Cookie counter = ApexPages.currentPage().getCookies().get('counter');

        // If this is the first time the user is accessing the page,
        // create a new cookie with name 'counter', an initial value of '1',
        // path 'null', maxAge '-1', and isSecure 'true'.
        if (counter == null) {
            counter = new Cookie('counter','1',null,-1,true);
        } else {
        // If this isn't the first time the user is accessing the page
        // create a new cookie, incrementing the value of the original count by 1
            Integer count = Integer.valueOf(counter.getValue());
            counter = new Cookie('counter', String.valueOf(count+1),null,-1,true);
        }
```

```
        // Set the new cookie for the page
        ApexPages.currentPage().setCookies(new Cookie[]{counter});
    }

    // This method is used by the Visualforce action {!count} to display the current
    // value of the number of times a user had displayed a page.
    // This value is stored in the cookie.
    public String getCount() {
        Cookie counter = ApexPages.currentPage().getCookies().get('counter');
        if(counter == null) {
            return '0';
        }
        return counter.getValue();
    }
}
```

```
// Test class for the Visualforce controller
@isTest
private class CookieControllerTest {
  // Test method for verifying the positive test case
  static testMethod void testCounter() {
    //first page view
    CookieController controller = new CookieController();
    System.assert(controller.getCount() == '1');

    //second page view
    controller = new CookieController();
    System.assert(controller.getCount() == '2');
  }
}
```

The following is the Visualforce page that uses the `CookieController` Apex controller above. The action `{!count}` calls the `getCount` method in the controller above.

```
<apex:page controller="CookieController">
You have seen this page {!count} times
</apex:page>
```

IN THIS SECTION:

Cookie Constructors

Cookie Methods

Cookie Constructors

The following are constructors for `Cookie`.

IN THIS SECTION:

Cookie(name, value, path, maxAge, isSecure)

Creates a new instance of the `Cookie` class using the specified name, value, path, age, and the secure setting.

Cookie(name, value, path, maxAge, isSecure, SameSite)

Creates a new instance of the `Cookie` class using the specified name, value, path, and age, and settings for security and cross-domain behavior.

Cookie(name, value, path, maxAge, isSecure, SameSite, isHttpOnly)

Creates a new instance of the `Cookie` class using the specified name, value, path, age, and settings for security, cross-domain behavior, and JavaScript access.

Cookie(name, value, path, maxAge, isSecure)

Creates a new instance of the `Cookie` class using the specified name, value, path, age, and the secure setting.

Signature

```
public Cookie(String name, String value, String path, Integer maxAge, Boolean isSecure)
```

Parameters

name
Type: String

The cookie name. It can't be `null`.

value
Type: String

The cookie data, such as session ID.

path
Type: String

The path from where you can retrieve the cookie.

maxAge
Type: Integer

A number representing how long a cookie is valid for in seconds. If set to less than zero, a session cookie is issued. If set to zero, the cookie is deleted.

isSecure
Type: Boolean

A value indicating whether the cookie can only be accessed through HTTPS (`true`) or not (`false`).

Cookie(name, value, path, maxAge, isSecure, SameSite)

Creates a new instance of the `Cookie` class using the specified name, value, path, and age, and settings for security and cross-domain behavior.

Note: Google Chrome 80 introduces a new default cookie attribute setting of `SameSite`, which is set to `Lax`. Previously, the `SameSite` cookie attribute defaulted to the value of `None`. When `SameSite` is set to `None`, cookies must be tagged with the `isSecure` attribute indicating that they require an encrypted HTTPS connection.

Signature

```
public Cookie(String name, String value, String path, Integer maxAge, Boolean isSecure,
String SameSite)
```

Parameters

name
Type: String

The cookie name. It can't be `null`.

value
Type: String

The cookie data, such as session ID.

path
Type: String

The path from where you can retrieve the cookie.

maxAge
Type: Integer

A number representing how long a cookie is valid for in seconds. If set to less than zero, a session cookie is issued. If set to zero, the cookie is deleted.

isSecure
Type: Boolean

A value indicating whether the cookie can only be accessed through HTTPS (`true`) or not (`false`).

SameSite
Type: String

The `SameSite` attribute on a cookie controls its cross-domain behavior. The valid values are `None`, `Lax`, and `Strict`. After the Chrome 80 release, a cookie with a `SameSite` value of `None` must also be marked secure by setting a value of `None; Secure`.

SEE ALSO:

Salesforce Spring '20 Release Notes: Prepare for Google Chrome's Changes in SameSite Cookie Behavior That Can Break Salesforce Integrations

Chrome Platform Status: Reject insecure SameSite=None cookies

Cookie(name, value, path, maxAge, isSecure, SameSite, isHttpOnly)

Creates a new instance of the `Cookie` class using the specified name, value, path, age, and settings for security, cross-domain behavior, and JavaScript access.

Signature

```
public Cookie(String name, String value, String path, Integer maxAge, Boolean isSecure,
String SameSite, Boolean isHttpOnly)
```

Parameters

name
Type: String

The cookie name. It can't be `null`.

value
Type: String

The cookie data, such as session ID.

path
Type: String

The path from where you can retrieve the cookie.

maxAge
Type: Integer

A number representing how long a cookie is valid for in seconds. If set to less than zero, a session cookie is issued. If set to zero, the cookie is deleted.

isSecure
Type: Boolean

A value indicating whether the cookie can only be accessed through HTTPS (`true`) or not (`false`).

SameSite
Type: String

The `SameSite` attribute on a cookie controls its cross-domain behavior. The valid values are `None`, `Lax`, and `Strict`. After the Chrome 80 release, a cookie with a `SameSite` value of `None` must also be marked secure by setting a value of `None; Secure`.

isHttpOnly
Type: Boolean

A value indicating whether the HttpOnly attribute for the cookie is set (`true`) or not (`false`). If `true`, client-side JavaScript can't access the cookie.

SEE ALSO:

MDN Web Docs: Set-Cookie HTTP Response Header

Cookie Methods

The following are methods for `Cookie`. All are instance methods.

IN THIS SECTION:

getDomain()
Returns the name of the server making the request.

getMaxAge()
Returns a number representing how long the cookie is valid for, in seconds. If set to `< 0`, a session cookie is issued. If set to `0`, the cookie is deleted.

getName()
Returns the name of the cookie. Can't be `null`.

getPath()
Returns the path from which you can retrieve the cookie. If `null` or blank, the location is set to root, or "/".

getSameSite()
Returns the value for the `SameSite` attribute of the cookie.

getValue()
Returns the data captured in the cookie, such as Session ID.

isSecure()
Returns `true` if the cookie can only be accessed through HTTPS, otherwise returns `false`.

isHttpOnly()
Returns `true` if client-side JavaScript is forbidden from accessing the cookie; otherwise returns `false`.

`getDomain()`

Returns the name of the server making the request.

Signature

```
public String getDomain()
```

Return Value

Type: String

`getMaxAge()`

Returns a number representing how long the cookie is valid for, in seconds. If set to `< 0`, a session cookie is issued. If set to `0`, the cookie is deleted.

Signature

```
public Integer getMaxAge()
```

Return Value

Type: Integer

`getName()`

Returns the name of the cookie. Can't be `null`.

Signature

```
public String getName()
```

Return Value

Type: String

`getPath()`

Returns the path from which you can retrieve the cookie. If `null` or blank, the location is set to root, or "/".

Signature

```
public String getPath()
```

Return Value

Type: String

`getSameSite()`

Returns the value for the `SameSite` attribute of the cookie.

Signature

```
public String getSameSite()
```

Return Value

Type: String

SEE ALSO:

web.dev: SameSite Cookies Explained

`getValue()`

Returns the data captured in the cookie, such as Session ID.

Signature

```
public String getValue()
```

Return Value

Type: String

`isSecure()`

Returns `true` if the cookie can only be accessed through HTTPS, otherwise returns `false`.

Signature

```
public Boolean isSecure()
```

Return Value

Type: Boolean

`isHttpOnly()`

Returns `true` if client-side JavaScript is forbidden from accessing the cookie; otherwise returns `false`.

Signature

```
public Boolean isHttpOnly()
```

Return Value

Type: Boolean

SEE ALSO:

MDN Web Docs: Set-Cookie HTTP Response Header

Crypto Class

Provides methods for creating digests, message authentication codes, and signatures, as well as encrypting and decrypting information.

Namespace

System

Usage

The methods in the `Crypto` class can be used for securing content in Lightning Platform, or for integrating with external services such as Google or Amazon WebServices (AWS).

Each method in this class supports a unique set of AES encryption algorithms, depending on its purpose. To confirm which algorithms are available for the action you want to do, check each method.

Encrypt and Decrypt Exceptions

The following exceptions can be thrown for these methods:

- `decrypt`
- `encrypt`
- `decryptWithManagedIV`
- `encryptWithManagedIV`

Exception	Message	Description
`InvalidParameterValue`	Unable to parse the initialization vector from encrypted data.	Thrown if you're using managed initialization vectors, and the cipher text is less than 16 bytes.
	Invalid algorithm *`algoName`*. Must be one of `AES128`, `AES192`, or `AES256`.	Thrown if the algorithm name isn't one of the valid values.

Exception	Message	Description
	Invalid private key. Must be *size* bytes.	Thrown if the size of the private key doesn't match the specified algorithm.
	Invalid initialization vector. Must be 16 bytes.	Thrown if the initialization vector isn't 16 bytes.
	Invalid data. Input data is *size* bytes, which exceeds the limit of 1,048,576 bytes.	Thrown if the data is greater than 1 MB. For decryption, 1,048,608 bytes are allowed for the initialization vector header, plus any additional padding the encryption added to align to block size.
`NullPointerException`	*Argument* can't be null.	Thrown if one of the required method arguments is null.
`SecurityException`	Given final block isn't properly padded.	Thrown if the data isn't properly block-aligned or similar issues occur during encryption or decryption.
`SecurityException`	*Message Varies*	Thrown if something goes wrong during either encryption or decryption.

These exceptions are a subset of the exceptions that can be thrown from the System namespace. Refer to Exception Class and Built-In Exceptions

The `Crypto` class uses AES / CBC / PKCS7 padding, which is vulnerable to a Padding Oracle attack. You can protect against a Padding Oracle attack using the Encrypt-then-MAC method. In this method, you encrypt the cipher text and MAC separately:

- For encryption, first encrypt the data with AES using one encryption key. Then, with a different encryption key, use the generateMac(algorithmName, input, privateKey) method to generate a message authentication code (MAC) for the cipher text. Append the MAC to the cipher text before sending it to its recipient.
- For decryption, start by checking the authenticity and integrity of the cipher text by using the verifyHMac(algorithmName, data, privateKey, macToVerify) method. If either the authenticity or integrity check fails, throw an exception and don't decrypt the cipher text. The decryption of the cipher text must only happen in a second step, after the message authenticity and integrity has been verified.

Other Errors

Under rare conditions you may encounter the `invalid_client` error from the JSON Web Tokens (JWT) service.

Error	Message	Description
`invalid_client`	The actual text varies, but describes the inability to validate the client credentials.	The JWT public certificate in the Salesforce Connected Application doesn't appear to match the known private key.

For Shield Platform Encryption, this error can happen when you use a custom JWT implementation that uses the P11363 format, and you also want to use the `ECDSA-SHA256` algorithm. The solution is to specify the `ECDSA-SHA256-PLAIN` algorithm instead. The `ECDSA-SHA256-PLAIN` is available to the several `sign()` and `verify()` methods.

For example, in order to comply with your program requirements, you sign your token using the Elliptic Curve Digital Signature Algorithm (ECDSA) with the P-256 curve. This algorithm is in the P11363 format, so when you try to use `Crypto.verify()` using the `ECDSA SHA256`, you receive a response containing `invalid_client`. You change `ECDSA-SHA256` to `ECDSA-SHA256-PLAIN` and the error is resolved.

SEE ALSO:

Encrypt-then-MAC (EtM)

ISO/IEC 19772:2020 - Information Security Authenticated Encryption

Exception Class and Built-In Exceptions

Crypto Methods

The following are methods for `Crypto`. All methods are static.

IN THIS SECTION:

decrypt(algorithmName, privateKey, initializationVector, cipherText)

Decrypts the *`cipherText`* blob using the specified algorithm, private key, and initialization vector. Use this method to decrypt blobs encrypted using a third-party application or the `encrypt` method.

decryptWithManagedIV(algorithmName, privateKey, IVAndCipherText)

Decrypts the *`IVAndCipherText`* blob using the specified algorithm and private key. Use this method to decrypt blobs encrypted using a third-party application or the `encryptWithManagedIV` method.

encrypt(algorithmName, privateKey, initializationVector, clearText)

Encrypts the *`clearText`* blob using the specified algorithm, private key, and initialization vector. Use this method when you want to specify your own initialization vector.

encryptWithManagedIV(algorithmName, privateKey, clearText)

Encrypts the *`clearText`* blob using the specified algorithm and private key. Use this method when you want Salesforce to generate the initialization vector.

generateAesKey(size)

Generates an Advanced Encryption Standard (AES) key.

generateDigest(algorithmName, input)

Computes a secure, one-way hash digest using the specified algorithm on the supplied *`input`* blob.

generateMac(algorithmName, input, privateKey)

Computes a message authentication code (MAC) for the *`input`* blob value using the private key and the specified algorithm.

getRandomInteger()

Returns a random integer value.

getRandomLong()

Returns a random long value.

sign(algorithmName, input, privateKey)

Computes a unique digital signature for the *`input`* blob value, using the specified algorithm and the supplied private key.

signWithCertificate(algorithmName, input, certDevName)

Computes a unique digital signature for the input blob value, using the specified algorithm and the supplied certificate and key pair.

signXML(algorithmName, node, idAttributeName, certDevName)

Envelops the signature into an XML document.

signXML(algorithmName, node, idAttributeName, certDevName, refChild)

Inserts the signature envelope before the specified child node.

verify(algorithmName, data, signature, publicKey)

Verifies the digital signature for the *data* blob using the specified algorithm and the supplied public key. Use this method to verify a blob signed by a digital signature created using a third-party application or the sign method.

verify(algorithmName, data, signature, certDevName)

Verifies the digital signature for the *data* blob using the specified algorithm and the public key associated with *certDevName*. Use this method to verify a blob signed by a digital signature created using a third-party application or the `signWithCertificate` method.

verifyHMac(algorithmName, data, privateKey, macToVerify)

Verifies the HMAC signature for the *data* blob using the specified algorithm, input data, private key, and the mac. Use this method to verify a blob signed by a digital signature created using a third-party application or the sign method.

decrypt(algorithmName, privateKey, initializationVector, cipherText)

Decrypts the *cipherText* blob using the specified algorithm, private key, and initialization vector. Use this method to decrypt blobs encrypted using a third-party application or the `encrypt` method.

Signature

```
public static Blob decrypt(String algorithmName, Blob privateKey, Blob
initializationVector, Blob cipherText)
```

Parameters

algorithmName

Type: String

Must be one of these industry-standard Advanced Encryption Standard (AES) algorithms with different size keys. The algorithms use cipher block chaining (CBC) and PKCS7 padding.

- `AES128`
- `AES192`
- `AES256`

privateKey

Type: Blob

Private key text. The length of *privateKey* must match the size required by *algorithmName*: 128 bits, 192 bits, or 256 bits, which is 16 bytes, 24 bytes, or 32 bytes, respectively. You can use a third-party application or the `generateAesKey` method to generate this key.

initializationVector

Type: Blob

Any 128 bit (16 byte) string to provide the initial state to this method. The initialization vector must be 128 bits (16 bytes.)

cipherText

Type: Blob

The content to decrypt.

Return Value

Type: Blob

Contains the decrypted contents of *cipherText*.

Example

You can use your preferred Salesforce development environment to test this function. Create the following Apex class:

```
public class TestDecrypt {

    public void testDecrypt(){
        // 16-byte string
        Blob exampleIv = Blob.valueOf('Example of IV123');
        Blob key = Crypto.generateAesKey(128);
        Blob data = Blob.valueOf('Data to be encrypted');
        Blob encrypted = Crypto.encrypt('AES128', key, exampleIv, data);

        Blob decrypted = Crypto.decrypt('AES128', key, exampleIv, encrypted);
        String decryptedString = decrypted.toString();
        System.debug('Decrypted Value: ' + decryptedString);
       Assert.areEqual('Data to be encrypted', decryptedString, 'Error: not equal!');

        return;
    }
}
```

To invoke this method, run the following:

```
TestDecrypt td = new TestDecrypt();
td.testDecrypt();
```

decryptWithManagedIV(algorithmName, privateKey, IVAndCipherText)

Decrypts the *IVAndCipherText* blob using the specified algorithm and private key. Use this method to decrypt blobs encrypted using a third-party application or the `encryptWithManagedIV` method.

Signature

```
public static Blob decryptWithManagedIV(String algorithmName, Blob privateKey, Blob
IVAndCipherText)
```

Parameters

algorithmName

Type: String

Must be one of these industry-standard Advanced Encryption Standard (AES) algorithms with different size keys. The algorithms use cipher block chaining (CBC) and PKCS7 padding.

- `AES128`

- `AES192`
- `AES256`

`privateKey`
Type: Blob

Private key text. The length of *`privateKey`* must match the size required by *`algorithmName`*: 128 bits, 192 bits, or 256 bits, which is 16 bytes, 24 bytes, or 32 bytes, respectively. You can use a third-party application or the `generateAesKey` method to generate this key for you.

`IVAndCipherText`
Type: Blob

A concatenation of the initialization vector and the encrypted text you want to decrypt. The initialization vector must be 128 bits (16 bytes), and be the first 16 bytes of *`IVAndCipherText`*.

Return Value

Type: Blob

Contains the decrypted contents of *`IVAndCipherText`*.

Example

You can use your preferred Salesforce development environment to test this function. Create the following Apex class:

```
    public class TestDecryptWithManagedIV {

        public void testDecryptWithManagedIV(){
            String algorithmName = 'AES128';
            // 16-byte IV
            String exampleIV = 'Example of IV24';
            Blob key = Crypto.generateAesKey(128);
            Blob data = Blob.valueOf(exampleIV + 'Data to be encrypted');
            Blob encrypted = Crypto.encryptWithManagedIV(algorithmName, key, data);
            Blob decrypted = Crypto.decryptWithManagedIV(algorithmName, key, encrypted);
            String decryptedString = decrypted.toString();
            Assert.areEqual(exampleIV + 'Data to be encrypted', decryptedString, 'Error:
the strings are not equal!');

        }
    }
```

To invoke this method, run the following:

```
TestDecryptWithManagedIV tdiv = new TestDecryptWithManagedIV();
tdiv.testDecryptWithManagedIV();
```

`encrypt(algorithmName, privateKey, initializationVector, clearText)`

Encrypts the *`clearText`* blob using the specified algorithm, private key, and initialization vector. Use this method when you want to specify your own initialization vector.

Signature

```
public static Blob encrypt(String algorithmName, Blob privateKey, Blob
initializationVector, Blob clearText)
```

Parameters

algorithmName
Type: String

Algorithm for encrypting *clearText*.

Must be one of these industry-standard Advanced Encryption Standard (AES) algorithms with different size keys. The algorithms use cipher block chaining (CBC) and PKCS7 padding.

- AES128
- AES192
- AES256

privateKey
Type: Blob

Private key text. The length of *privateKey* must match the size required by *algorithmName*: 128 bits, 192 bits, or 256 bits, which is 16 bytes, 24 bytes, or 32 bytes, respectively. You can use a third-party application or the generateAesKey method to generate this key for you.

initializationVector
Type: Blob

128-bit initialization vector. The initialization vector must be 128 bits (16 bytes).

clearText
Type: Blob

The content you want to encrypt.

Return Value

Type: Blob

Contains the encrypted contents of *clearText*.

Example

You can use your preferred Salesforce development environment to test this function. Create the following Apex class:

```
public class TestEncrypt {

    public void testEncrypt(){
        Blob exampleIv = Blob.valueOf('Example of IV123');
        Blob key = Crypto.generateAesKey(128);
        Blob data = Blob.valueOf('Encryption Example Text.');
        Blob encrypted = Crypto.encrypt('AES128', key, exampleIv, data);

        Blob decrypted = Crypto.decrypt('AES128', key, exampleIv, encrypted);
        String decryptedString = decrypted.toString();
       Assert.areEqual('Encryption Example Text.', decryptedString, 'Error: the values
```

```
 are not equal!');
            return;
        }
    }
```

To invoke this method, run the following:

```
TestEncrypt te = new TestEncrypt();
te.testEncrypt();
```

encryptWithManagedIV(algorithmName, privateKey, clearText)

Encrypts the *clearText* blob using the specified algorithm and private key. Use this method when you want Salesforce to generate the initialization vector.

Signature

```
public static Blob encryptWithManagedIV(String algorithmName, Blob privateKey, Blob
clearText)
```

Parameters

algorithmName
Type: String

The algorithm for encrypting *clearText*. Must be one of these industry-standard Advanced Encryption Standard (AES) algorithms with different size keys. The algorithms use cipher block chaining (CBC) and PKCS7 padding.

- `AES128`
- `AES192`
- `AES256`

privateKey
Type: Blob

Private key text. The length of *privateKey* must match the size required by *algorithmName*: 128 bits, 192 bits, or 256 bits, which is 16 bytes, 24 bytes, or 32 bytes, respectively. You can use a third-party application or the `generateAesKey` method to generate this key for you.

clearText
Type: Blob

The content you want to encrypt.

Return Value

Type: Blob

Contains the encrypted contents of *clearText*.

The initialization vector is stored as the first 128 bits (16 bytes) of the encrypted blob. Use either third-party applications or the `decryptWithManagedIV` method to decrypt blobs encrypted with this method. Use the `encrypt` method if you want to generate your own initialization vector.

Example

You can use your preferred Salesforce development environment to test this function. Create the following Apex class:

```
    public class TestEncryptWithManagedIV {

        public void testEncryptWithManagedIV(){
            String algorithmName = 'AES128';
            // 16-byte IV
            String exampleIV = 'Example of IV24';
            Blob key = Crypto.generateAesKey(128);
            Blob data = Blob.valueOf(exampleIV + 'Data to be encrypted');
            Blob encrypted = Crypto.encryptWithManagedIV(algorithmName, key, data);
            Blob decrypted = Crypto.decryptWithManagedIV(algorithmName, key, encrypted);
            String decryptedString = decrypted.toString();
            Assert.areEqual(exampleIV + 'Data to be encrypted', decryptedString, 'Error:
the strings are not equal!');

        }
    }
```

To invoke this method, run the following:

```
TestEncryptWithManagedIV teiv = new TestEncryptWithManagedIV();
teiv.testEncryptWithManagedIV();
```

`generateAesKey(size)`

Generates an Advanced Encryption Standard (AES) key.

Signature

```
public static Blob generateAesKey(Integer size)
```

Parameters

size

Type: Integer

The key's size in bits. Valid values are:

- 128
- 192
- 256

Return Value

Type: Blob

Contains the generated AES key.

Example

You can use your preferred Salesforce development environment to test this function. Create the following Apex class:

```
public class TestGenerateAESKey {

    public void testGenerateAESKey() {
      Blob key = Crypto.generateAesKey(128);
      System.debug('Generated AES Key: ');
      String strKey = EncodingUtil.base64Encode(key);
      System.debug(strKey);

    }
}
```

To invoke this method, run the following:

```
TestGenerateAESKey tgaes = new TestGenerateAESKey();
tgaes.testGenerateAESKey();
```

generateDigest(algorithmName, input)

Computes a secure, one-way hash digest using the specified algorithm on the supplied *input* blob.

Signature

```
public static Blob generateDigest(String algorithmName, Blob input)
```

Parameters

algorithmName
Type: String

The algorithm you want to use to generate the digest. Valid values for *algorithmName* are:

- MD5
- SHA1
- SHA3-256
- SHA3-384
- SHA3-512
- SHA-256
- SHA-512

input
Type: Blob

The content for which you want to generate the digest.

Return Value

Type: Blob

Contains the generated digest.

Example

You can use your preferred Salesforce development environment to test this function. Create the following Apex class:

```
    public class TestGenerateDigest {

        public void testGenerateDigest(){
            Blob targetBlob = Blob.valueOf('ExampleMD5String');
            Blob hash = Crypto.generateDigest('MD5', targetBlob);
            String result = EncodingUtil.base64Encode(hash);
            System.debug('Value: ' + result);
        }
    }
```

To invoke this method, run the following:

```
TestGenerateDigest tgd = new TestGenerateDigest();
tgd.testGenerateDigest();
```

generateMac(algorithmName, input, privateKey)

Computes a message authentication code (MAC) for the *input* blob value using the private key and the specified algorithm.

Signature

```
public static Blob generateMac(String algorithmName, Blob input, Blob privateKey)
```

Parameters

algorithmName
Type: String

These are valid values for *algorithmName*.

- hmacMD5
- hmacSHA1
- hmacSHA256
- hmacSHA512

input
Type: Blob

The content for which you want to generate the MAC.

privateKey
Type: Blob

The key to use to generate the MAC. You may supply a private key that has been encoded using Base64 encoding. However if you do, then you must also supply the Base64-encoded private key when verifying the MAC using the `verifyHMac` method. The value of *privateKey* can't exceed 4 KB.

Return Value

Type: Blob

The message authentication code.

Example

You can use your preferred Salesforce development environment to test this function. Create the following Apex class:

```
public class TestGenerateMAC {

    public void testGenerateMAC() {
        String salt = String.valueOf(Crypto.getRandomInteger());
        String key = 'key';
        Blob data = crypto.generateMac('HmacSHA256',
                                       Blob.valueOf(salt),
                                       Blob.valueOf(key));
        System.debug('Generated MAC: ');
        System.debug(EncodingUtil.base64Encode(data));
    }
}
```

To invoke this method, run the following:

```
TestGenerateMAC tgm = new TestGenerateMAC();
tgm.testGenerateMAC();
```

`getRandomInteger()`

Returns a random integer value.

Signature

```
public static Integer getRandomInteger()
```

Return Value

Type: Integer

Returns a random 4-byte integer. Salesforce invokes the `java.security.SecureRandom` api to generate this number. For information on how the number is generated, see java.security.SecureRandom.

Example

You can use your preferred Salesforce development environment to exercise this function. Create the following Apex class:

```
public class TestGetRandomInteger {

    public void testGetRandomInteger() {
        Integer i1 = Crypto.getRandomInteger();
        Integer i2 = Crypto.getRandomInteger();
        System.debug('Integer 1: ' + i1);
        System.debug('Integer 2: ' + i2);
        Assert.areNotEqual(i1, i2, 'Sorry, those aren't random!');
        //This is just an example. This is not a true test of randomness
    }
}
```

To invoke this method, run the following:

```
TestGetRandomInteger tri = new TestGetRandomInteger();
tri.testGetRandomInteger();
```

SEE ALSO:

java.security.SecureRandom

getRandomLong()

Returns a random long value.

Signature

```
public static Long getRandomLong()
```

Return Value

Type: Long

Returns a random 8-byte long. Salesforce invokes the `java.security.SecureRandom` api to generate this number. For information on how the number is generated, see java.security.SecureRandom.

Example

You can use your preferred Salesforce development environment to exercise this function. Create the following Apex class:

```
    public class TestGetRandomLong {

        public void testGetRandomLong() {
            Long L1 = Crypto.getRandomLong();
            Long L2 = Crypto.getRandomLong();
            System.debug('Long 1: ' + L1);
            System.debug('Long 2: ' + L2);
            Assert.areNotEqual(L1, L2, 'Sorry, not random!');
            //This is just an example. This is not a true test of randomness
        }
    }
```

To invoke this method, run the following:

```
TestGetRandomLong trl = new TestGetRandomLong();
trl.testGetRandomLong();
```

SEE ALSO:

java.security.SecureRandom

sign(algorithmName, input, privateKey)

Computes a unique digital signature for the *input* blob value, using the specified algorithm and the supplied private key.

Signature

```
public static Blob sign(String algorithmName, Blob input, Blob privateKey)
```

Parameters

algorithmName
Type: String

These are valid values for *algorithmName*.

- `RSA`: the same as `RSA-SHA1`.
- `RSA-SHA1`: an RSA signature (with an asymmetric key pair) of an SHA1 hash.
- `RSA-SHA256`: an RSA signature of an SHA256 hash.
- `RSA-SHA384`: an RSA signature of an SHA384 hash.
- `RSA-SHA512`: an RSA signature of an SHA512 hash.
- `ECDSA-SHA256`: an ECDSA signature of an SHA256 hash.
- `ECDSA-SHA256-PLAIN`: an ECDSA signature of an SHA256 hash (P1363 format) .
- `ECDSA-SHA384`: an ECDSA signature of an SHA384 hash.
- `ECDSA-SHA512`: an ECDSA signature of an SHA512 hash.

input
Type: Blob

The data to sign.

privateKey
Type: Blob

The key to use for signing. The value of *privateKey* must be decoded using the `EncodingUtilbase64Decode` method, and should be in RSA's PKCS #8 (1.2) Private-Key Information Syntax Standard form. The value can't exceed 4 KB.

Return Value

Type: Blob

The new digital signature.

Example

You can use your preferred Salesforce development environment to test this function. To run it correctly, you need a PKCS8 private key. At your terminal, use `openssl` to create one. First, create the key. Then convert it to PKCS8:

```
$ openssl genrsa -out myprivatekey.pem 1024
$ openssl pkey -in myprivatekey.pem -out myprivatekey.pkcs8.pem
```

After you create the PKCS8 compatible key, you decode just the key portion of the text (without the BEGIN PRIVATE KEY or END PRIVATE KEY lines) for the *privateKey* parameter.

```
    public class TestSign {

        public void testSign() {
                Blob input = Blob.valueOf('Some text.');
                String algorithmName = 'RSA';
                String rawKey = '<text value of your pkcs8 private key>';
```

```
                //no BEGIN PRIVATE KEY or END PRIVATE KEY header/footer !
                Blob privateKey = EncodingUtil.base64Decode(rawKey);
                System.debug(privateKey);
                Blob signedKey = Crypto.sign(algorithmName, input, privateKey);

        }
    }
```

To invoke this method, run the following:

```
TestSign ts = new TestSign();
ts.testSign();
```

signWithCertificate(algorithmName, input, certDevName)

Computes a unique digital signature for the input blob value, using the specified algorithm and the supplied certificate and key pair.

Signature

```
public static Blob signWithCertificate(String algorithmName, Blob input, String
certDevName)
```

Parameters

algorithmName
Type: String

These are valid values for *algorithmName*.

- `RSA`: the same as `RSA-SHA1`.
- `RSA-SHA1`: an RSA signature (with an asymmetric key pair) of an SHA1 hash.
- `RSA-SHA256`: an RSA signature of an SHA256 hash.
- `RSA-SHA384`: an RSA signature of an SHA384 hash.
- `RSA-SHA512`: an RSA signature of an SHA512 hash.
- `ECDSA-SHA256`: an ECDSA signature of an SHA256 hash.
- `ECDSA-SHA256-PLAIN`: an ECDSA signature of an SHA256 hash (P1363 format) .
- `ECDSA-SHA384`: an ECDSA signature of an SHA384 hash.
- `ECDSA-SHA512`: an ECDSA signature of an SHA512 hash.

input
Type: Blob

The data to sign.

certDevName
Type: String

The value listed in the `Unique Name` field for a certificate stored in the Salesforce org's Certificate and Key Management page to use for signing.

To access the Certificate and Key Management page from Setup, enter *Certificate and Key Management* in the **Quick Find** box, then select **Certificate and Key Management**.

Return Value

Type: Blob

The signed content.

Example

You can use your preferred Salesforce development environment to test this function. Create the following Apex class. For the *TestCertName* variable, use the unique name value for a self-signed or CA certificate that you have created in the org in which you run this test.

```
    public class TestSignWithCert {

        public void testSignWithCert() {

            String algorithmName = 'RSA';
            Blob input = Blob.valueOf('Test Sign With Certificate.');
            String TestCertName = 'your-cert-unique-name';
          Blob signedKey = Crypto.signWithCertificate(algorithmName, input, TestCertName);

        }
    }
```

To invoke the method, run the following:

```
TestSignWithCert tswc = new TestSignWithCert();
tswc.testSignWithCert();
```

signXML(algorithmName, node, idAttributeName, certDevName)

Envelops the signature into an XML document.

Signature

```
public Void signXML(String algorithmName, Dom.XmlNode node, String idAttributeName,
String certDevName)
```

Parameters

algorithmName
Type: String

These are valid values for *algorithmName*.

- RSA: the same as RSA-SHA1.
- RSA-SHA1: an RSA signature (with an asymmetric key pair) of an SHA1 hash.
- RSA-SHA256: an RSA signature of an SHA256 hash.
- RSA-SHA384: an RSA signature of an SHA384 hash.
- RSA-SHA512: an RSA signature of an SHA512 hash.
- ECDSA-SHA256: an ECDSA signature of an SHA256 hash.
- ECDSA-SHA256-PLAIN: an ECDSA signature of an SHA256 hash (P1363 format) .
- ECDSA-SHA384: an ECDSA signature of an SHA384 hash.

- `ECDSA-SHA512`: an ECDSA signature of an SHA512 hash.

`node`
Type: Dom.XmlNode

The XML node to sign and insert the signature into.

`idAttributeName`
Type: String

The full name (including the namespace) of the attribute on the node (XmlNode) to use as the reference ID. If `null`, this method uses the `ID` attribute on the node. If there's no `ID` attribute, Salesforce generates a new ID and adds it to the node.

`certDevName`
Type: String

The unique name for a certificate stored in the Salesforce org's Certificate and Key Management page to use for signing.

To access the Certificate and Key Management page from Setup, enter *`Certificate and Key Management`* in the **Quick Find** box, then select **Certificate and Key Management**.

Return Value

Type: void

This method doesn't return a value. The signature envelope is inserted within *`node`*.

Example

You can use your preferred Salesforce development environment to test this function. Create the following Apex class. For the *`testCertName`* variable, use the unique name value for a self-signed or CA certificate that you have created in the org in which you run this test.

```
public class TestSignXML {
  public void testSignXML() {
    String algorithmName = 'RSA';
    String testCertName = 'your-cert-unique-name';
    Dom.Document doc = new Dom.Document();
    String docToLoad = '<?xml version=\"1.0\"?>\n' +
    '<customers>\n' +
    '  <customer id="2">\n' +
    '    <name>Company One</name>\n' +
    '  </customer>\n' +
    '</customers>';
    doc.load(docToLoad);

    System.Crypto.signXML(algorithmName, doc.getRootElement(), null, testCertName);

    //dump the content of the signed XML document to the debug log
    System.Debug(doc.toXmlString());
  }
}
```

To invoke this method, run the following:

```
TestSignXML tswxml = new TestSignXML();
tswxml.testSignXML();
```

signXML(algorithmName, node, idAttributeName, certDevName, refChild)

Inserts the signature envelope before the specified child node.

Signature

```
public static void signXml(String algorithmName, Dom.XmlNode node, String
idAttributeName, String certDevName, Dom.XmlNode refChild)
```

Parameters

algorithmName
Type: String

These are valid values for *algorithmName*.

- `RSA`: the same as `RSA-SHA1`.
- `RSA-SHA1`: an RSA signature (with an asymmetric key pair) of an SHA1 hash.
- `RSA-SHA256`: an RSA signature of an SHA256 hash.
- `RSA-SHA384`: an RSA signature of an SHA384 hash.
- `RSA-SHA512`: an RSA signature of an SHA512 hash.
- `ECDSA-SHA256`: an ECDSA signature of an SHA256 hash.
- `ECDSA-SHA256-PLAIN`: an ECDSA signature of an SHA256 hash (P1363 format) .
- `ECDSA-SHA384`: an ECDSA signature of an SHA384 hash.
- `ECDSA-SHA512`: an ECDSA signature of an SHA512 hash.

node
Type: Dom.XmlNode

The XML node to sign and insert the signature into.

idAttributeName
Type: String

The full name (including the namespace) of the attribute on the node (XmlNode) to use as the reference ID. If `null`, this method uses the `ID` attribute on the node. If there's no `ID` attribute, Salesforce generates a new ID and adds it to the node.

certDevName
Type: String

The unique name for a certificate stored in the Salesforce org's Certificate and Key Management page to use for signing.

To access the Certificate and Key Management page from Setup, enter *Certificate and Key Management* in the **Quick Find** box, then select **Certificate and Key Management**.

refChild
Dom.XmlNode

The XML node before which to insert the signature. If *refChild* is `null`, the signature is added at the end.

Return Value

Type: Void

This method doesn't return a value. The signature envelope is inserted within *node*.

Example

You can use your preferred Salesforce development environment to test this function. Create the following Apex class. For the *testCertName* variable, use the unique name value for a self-signed or CA certificate that you have created in the org in which you run this test.

```
    public class TestSignXML_2 {
        public void testSignXML_2() {
            String algorithmName = 'RSA';
            String testCertName = 'your-cert-unique-name';
            Dom.Document doc = new Dom.Document();
            String docToLoad = '<?xml version="1.0"?>\n' +
            '<customers>\n' +
            '  <customer id="2">\n' +
            '    <name>Company One</name>\n' +
            '  </customer>\n' +
            '</customers>';
            doc.load(docToLoad);
            Dom.XmlNode rootNode = doc.getRootElement();
            Dom.XmlNode commentNode = rootNode.addCommentNode('SomeComment');

           System.Crypto.signXML(algorithmName, doc.getRootElement(), null, testCertName,
 commentNode);

            //send the content of the signed XML document to the debug log
            System.Debug(doc.toXmlString());
        }

    }
```

To invoke this method, run the following:

```
TestSignXML_2 tswxml2 = new TestSignXML_2();
tswxml2.testSignXML_2();
```

verify(algorithmName, data, signature, publicKey)

Verifies the digital signature for the *data* blob using the specified algorithm and the supplied public key. Use this method to verify a blob signed by a digital signature created using a third-party application or the sign method.

Signature

```
public static Boolean verify(String algorithmName, Blob data, Blob signature, Blob
publicKey)
```

Parameters

algorithmName
Type: String

These are valid values for *algorithmName*.

- `RSA`: the same as `RSA-SHA1`.
- `RSA-SHA1`: an RSA signature (with an asymmetric key pair) of an SHA1 hash.

- `RSA-SHA256`: an RSA signature of an SHA256 hash.
- `RSA-SHA384`: an RSA signature of an SHA384 hash.
- `RSA-SHA512`: an RSA signature of an SHA512 hash.
- `ECDSA-SHA256`: an ECDSA signature of an SHA256 hash.
- `ECDSA-SHA256-PLAIN`: an ECDSA signature of an SHA256 hash (P1363 format) .
- `ECDSA-SHA384`: an ECDSA signature of an SHA384 hash.
- `ECDSA-SHA512`: an ECDSA signature of an SHA512 hash.

`data`
Type: Blob

The data to sign.

`signature`
Type:

Blob

The RSA or EDSA-compliant signature.

`publicKey`
Type: Blob

The value of *`publicKey`* must be decoded using the `EncodingUtilbase64Decode` method, and be in X.509 standard format.

Return Value

Type: Boolean

`true` if and only if the signature is successfully verified.

Example

You can use your preferred Salesforce development environment to test this function. To run it correctly, you must:

- generate an X.509 private key and public certificate
- convert the private key to PKCS8
- extract the public key from the public certificate

You provide the private PKCS8 key to the `sign` method, and the extracted public key to the `verify` method (along with the signature you generate with `sign`.

At your terminal, use `openssl` to create the X.509 key pair:

```
$ openssl req -x509 -newkey rsa:2048 -keyout myPriv509.key -out myPub509.cert -days 365
```

Convert the private key to PKCS8:

```
openssl pkey -in myPriv509.key -out myPriv509pkcs8.pem
```

Extract the public key from `myPub509.cert`:

```
openssl x509 -in myPub509.cert -inform pem -pubkey -out myPub509.pem
```

After you create the `myPub509.pem` key, you decode just the key portions of the text (without the BEGIN PRIVATE KEY or END PRIVATE KEY lines) for both the *`privateKey`* and *`publicKey`* parameters.

```
    public class TestVerify {
        public void testVerify() {

            String algorithmName = 'RSA';
            Blob input = Blob.valueOf('Here is some text.');

            //contents of myPriv509pkcs8.pem
            String myPriv509pkcs8 = 'contents of myPriv509pkcs8.pem';

            Blob privateKey = EncodingUtil.base64Decode(myPriv509pkcs8);

            Blob signature = Crypto.sign(algorithmName, input, privateKey);

            //contents of myPub509.pem
            String publicKeyTxt64 = 'contents of myPub509.pem';

            Blob publicKey = EncodingUtil.base64Decode(publicKeyTxt64);

            Boolean verified = false;
            verified = Crypto.verify(algorithmName, input, signature, publicKey);

            Assert.areEqual(true, verified);

        }
    }
```

To invoke, run the following:

```
TestVerify tv = new TestVerify();
tv.testVerify();
```

SEE ALSO:

X.509 Standard

verify(algorithmName, data, signature, certDevName)

Verifies the digital signature for the *`data`* blob using the specified algorithm and the public key associated with *`certDevName`*. Use this method to verify a blob signed by a digital signature created using a third-party application or the `signWithCertificate` method.

Signature

```
public static Boolean verify(String algorithmName, Blob data, Blob signature, String
certDevName)
```

Parameters

`algorithmName`
Type: String

These are valid values for *algorithmName*.

- `RSA`: the same as `RSA-SHA1`.
- `RSA-SHA1`: an RSA signature (with an asymmetric key pair) of an SHA1 hash.
- `RSA-SHA256`: an RSA signature of an SHA256 hash.
- `RSA-SHA384`: an RSA signature of an SHA384 hash.
- `RSA-SHA512`: an RSA signature of an SHA512 hash.
- `ECDSA-SHA256`: an ECDSA signature of an SHA256 hash.
- `ECDSA-SHA256-PLAIN`: an ECDSA signature of an SHA256 hash (P1363 format) .
- `ECDSA-SHA384`: an ECDSA signature of an SHA384 hash.
- `ECDSA-SHA512`: an ECDSA signature of an SHA512 hash.

data

Type: Blob

The data to sign.

signature

Type:

Blob

The RSA or ECDSA signature.

certDevName

Type: String

The value listed in the `Unique Name` field for a certificate stored in the Salesforce organization's Certificate and Key Management page to use for signing.

To access the Certificate and Key Management page from Setup, enter *Certificate and Key Management* in the **Quick Find** box, then select **Certificate and Key Management**.

Return Value

Type: Boolean

Returns `true` if the signature is successfully verified.

Example

You can use your preferred Salesforce development environment to test this function. Create the following Apex class. For the *TestCertName* variable, use the unique name value for a self-signed or CA certificate that you have created in the org in which you run this test.

```
    public class TestVerify_2 {
       public void testVerify_2() {

            String algorithmName = 'RSA';
            Blob input = Blob.valueOf('Test Sign With Certificate.');
            String TestCertName = 'your-cert-unique-name';
          Blob signedKey = Crypto.signWithCertificate(algorithmName, input, TestCertName);


            Boolean verified = false;
```

```
            verified = Crypto.verify(algorithmName, input, signedKey, TestCertName);
            Assert.areEqual(true, verified);
        }
    }
```

To invoke this method, run the following:

```
TestVerify_2 tv_2 = new TestVerify_2();
tv_2.testVerify_2();
```

verifyHMac(algorithmName, data, privateKey, macToVerify)

Verifies the HMAC signature for the *data* blob using the specified algorithm, input data, private key, and the mac. Use this method to verify a blob signed by a digital signature created using a third-party application or the sign method.

Signature

```
public static Boolean verifyHMac(String algorithmName, Blob data, Blob privateKey, Blob
macToVerify)
```

Parameters

algorithmName
Type: String

These are valid values for *algorithmName*.

- hmacMD5
- hmacSHA1
- hmacSHA256
- hmacSHA512

data
Type: Blob

The data to sign.

privateKey
Type: Blob

If the private key used to generate the MAC was Base64 encoded, then the value of *privateKey* must also be Base64 encoded. The value cannot exceed 4 KB.

hmacToVerify
Type: Blob

The value of the mac must be verified against the provided *privateKey*, *data*, and *algorithmName*.

Return Value

Type: Boolean

The verification status of the data to verify.

Example

You can use your preferred Salesforce development environment to test this function. Create the following Apex class:

```
    public class TestVerifyMAC {

        public void testVerifyMAC() {
            String salt = String.valueOf(Crypto.getRandomInteger());
            String key = 'key';
            Blob data = crypto.generateMac('HmacSHA256',
                Blob.valueOf(salt),
                Blob.valueOf(key));
            System.debug('Generated MAC: ');
            System.debug(EncodingUtil.base64Encode(data));

            Boolean verified = false;

          verified = Crypto.verifyHMac('HmacSHA256', Blob.valueOf(salt), Blob.valueOf(key),
 data);
            Assert.areEqual(true, verified);
        }
    }
```

To invoke this method, run the following:

```
TestVerifyMAC tvm = new TestVerifyMAC();
tvm.testVerifyMAC();
```

Custom Metadata Type Methods

Custom metadata types are customizable, deployable, packageable, and upgradeable application metadata. All custom metadata is exposed in the application cache, which allows access without repeated queries to the database. The metadata is then available for formula fields, validation rules, flows, Apex, and SOAP API. All methods are static.

Usage

Custom metadata types methods are instance type methods and are called by and operate on a specific instance of a custom metadata type.

Custom Metadata Types Example

The following example uses the `getAll()` method. The custom metadata type named Games has a field called *GameType__c*. This example determines if the field value of the first record is equal to the string *PC*.

```
List<Games__mdt> mcs = Games__mdt.getAll().values();
boolean textField = null;
if (mcs[0].GameType__c == 'PC') {
   textField = true;
}
system.assertEquals(textField, true);
```

IN THIS SECTION:

getAll()
Returns a map containing custom metadata records for the specific custom metadata type. The map keys are the record DeveloperNames and the map values are the record sObjects.

getInstance(recordId)
Returns a single custom metadata type record sObject for a specified record ID. Returns null if no record matches the parameter.

getInstance(developerName)
Returns a single custom metadata type record sObject for a specified developerName field of the custom metadata type object. Returns null if no record matches the parameter.

getInstance(qualifiedApiName)
Returns a single custom metadata type record sObject for a qualified API name. Returns null if no record matches the parameter.

getAll()

Returns a map containing custom metadata records for the specific custom metadata type. The map keys are the record DeveloperNames and the map values are the record sObjects.

Signature

```
public Map<String, CustomMetadataType__mdt> getAll()
```

Return Value

Type: `Map<String, CustomMetadataType__mdt>`

Usage

If no records are defined for the type, this method returns an empty map. To iterate over the list of custom metadata type record sObjects, use `getAll().values()`. Only the first 255 characters are returned for any field in a custom metadata type record, so longer text fields get truncated. If you want all the field data from a custom metadata type record, use a SOQL query.

Example

This sample returns a map of all the records for a custom metadata type named `Games__mdt`.

```
Map<String, Games__mdt> mcs = Games__mdt.getAll();
```

getInstance(recordId)

Returns a single custom metadata type record sObject for a specified record ID. Returns null if no record matches the parameter.

Signature

```
public CustomMetadataType__mdt getInstance(recordId)
```

Parameters

recordId
Type: String

Return Value

Type: `CustomMetadataType__mdt`

Usage

Use this method to explicitly retrieve custom metadata type information at the user level. Only the first 255 characters of any field in a custom metadata type record are returned. Therefore, fields such as long text fields can be truncated. If you want all the field data from a custom metadata type record, use a SOQL query.

Example

This sample returns a single record sObject for the custom metadata type named `Games_mdt` with *recordID* specified as `m00000000000001`.

```
Games__mdt mc = Games__mdt.getInstance('m00000000000001');
```

getInstance(developerName)

Returns a single custom metadata type record sObject for a specified developerName field of the custom metadata type object. Returns null if no record matches the parameter.

Signature

```
public CustomMetadataType__mdt getInstance(String developerName)
```

Parameters

developerName
Type: String

Return Value

Type: `CustomMetadataType__mdt`

Usage

Use this method to return a single custom metadata type record for the specified *developerName*. The *developerName* is the unique name of the custom metadata type object in the API. Only the first 255 characters of any field in a custom metadata type record are returned. Therefore, fields such as long text fields can be truncated. If you want all the field data from a custom metadata type record, use a SOQL query.

Example

Returns a single record sObject for the custom metadata type named `Games_mdt` with *`developerName`* specified as `FirstRecord`.

```
Games__mdt mc = Games__mdt.getInstance('FirstRecord');
```

getInstance(qualifiedApiName)

Returns a single custom metadata type record sObject for a qualified API name. Returns null if no record matches the parameter.

Signature

```
public CustomMetadataType__mdt getInstance(String qualifiedApiName)
```

Parameters

`qualifiedApiName`
Type: String

Return Value

Type: `CustomMetadataType__mdt`

Usage

Use this method to return a single custom metadata type record for the specified *`qualifiedApiName`*. The *`qualifiedApiName`* is a concatenation of the namespace prefix and developerName, and has this format: *`namespacePrefix__developerName`*. The developerName is the unique name of the custom metadata type object in the API. Only the first 255 characters of any field in a custom metadata type record are returned. Therefore, fields such as long text fields can be truncated. If you want all the field data from a custom metadata type record, use a SOQL query.

Example

This sample returns a single record sObject for the custom metadata type named `Games_mdt` with *`qualifiedApiName`* specified as `MyNamespace__FirstRecord`.

```
Games__mdt mc = Games__mdt.getInstance('MyNamespace__FirstRecord');
```

Custom Settings Methods

Custom settings are similar to custom objects and enable application developers to create custom sets of data, as well as create and associate custom data for an organization, profile, or specific user. All custom settings data is exposed in the application cache, which enables efficient access without the cost of repeated queries to the database. This data is then available for formula fields, validation rules, flows, Apex, and the SOAP API.

Usage

Custom settings methods are all instance methods, that is, they are called by and operate on a specific instance of a custom setting. There are two types of custom settings: hierarchy and list. There are two types of methods: methods that work with list custom settings, and methods that work with hierarchy custom settings.

Note: All custom settings data is exposed in the application cache, which enables efficient access without the cost of repeated queries to the database. However, querying custom settings data using Standard Object Query Language (SOQL) doesn't use the application cache and is similar to querying a custom object. To benefit from caching, use other methods for accessing custom settings data such as the Apex Custom Settings methods.

For more information on creating custom settings in the Salesforce user interface, see "Create Custom Settings" in the Salesforce online help.

Custom Setting Examples

The following example uses a list custom setting called `Games`. The `Games` setting has a field called `GameType`. This example determines if the value of the first data set is equal to the string `PC`.

```
List<Games__C> mcs = Games__c.getall().values();
boolean textField = null;
if (mcs[0].GameType__c == 'PC') {
  textField = true;
}
system.assertEquals(textField, true);
```

The following example uses a custom setting called `Foundation_Countries`. This example demonstrates that the `getValues` and `getInstance` methods return identical values.

```
Foundation_Countries__c myCS1 = Foundation_Countries__c.getValues('United States');
String myCCVal = myCS1.Country_code__c;
Foundation_Countries__c myCS2 = Foundation_Countries__c.getInstance('United States');
String myCCInst = myCS2.Country_code__c;
system.assertEquals(myCCinst, myCCVal);
```

Hierarchy Custom Setting Examples

In the following example, the hierarchy custom setting GamesSupport has a field called `Corporate_number`. The code returns the value for the profile specified with `pid`.

```
GamesSupport__c mhc = GamesSupport__c.getInstance(pid);
string mPhone = mhc.Corporate_number__c;
```

The example is identical if you choose to use the `getValues` method.

The following example shows how to use hierarchy custom settings methods. For `getInstance`, the example shows how field values that aren't set for a specific user or profile are returned from fields defined at the next lowest level in the hierarchy. The example also shows how to use `getOrgDefaults`.

Finally, the example demonstrates how `getValues` returns fields in the custom setting record only for the specific user or profile, and doesn't merge values from other levels of the hierarchy. Instead, `getValues` returns `null` for any fields that aren't set. This example uses a hierarchy custom setting called Hierarchy. Hierarchy has two fields: `OverrideMe` and `DontOverrideMe`. In addition, a user named Robert has a System Administrator profile. The organization, profile, and user settings for this example are as follows:

Organization settings

`OverrideMe`: Hello

`DontOverrideMe`: World

Profile settings

`OverrideMe`: Goodbye

`DontOverrideMe` is not set.

User settings

`OverrideMe`: Fluffy

`DontOverrideMe` is not set.

The following example demonstrates the result of the `getInstance` method when Robert calls it in his organization:

```
Hierarchy__c CS = Hierarchy__c.getInstance();
System.Assert(CS.OverrideMe__c == 'Fluffy');
System.assert(CS.DontOverrideMe__c == 'World');
```

If Robert passes his user ID specified by `RobertId` to `getInstance`, the results are the same. The identical results are because the lowest level of data in the custom setting is specified at the user level.

```
Hierarchy__c CS = Hierarchy__c.getInstance(RobertId);
System.Assert(CS.OverrideMe__c == 'Fluffy');
System.assert(CS.DontOverrideMe__c == 'World');
```

If Robert passes the System Administrator profile ID specified by `SysAdminID` to `getInstance`, the result is different. The data specified for the profile is returned:

```
Hierarchy__c CS = Hierarchy__c.getInstance(SysAdminID);
System.Assert(CS.OverrideMe__c == 'Goodbye');
System.assert(CS.DontOverrideMe__c == 'World');
```

When Robert tries to return the data set for the organization using `getOrgDefaults`, the result is:

```
Hierarchy__c CS = Hierarchy__c.getOrgDefaults();
System.Assert(CS.OverrideMe__c == 'Hello');
System.assert(CS.DontOverrideMe__c == 'World');
```

By using the `getValues` method, Robert can get the hierarchy custom setting values specific to his user and profile settings. For example, if Robert passes his user ID `RobertId` to `getValues`, the result is:

```
Hierarchy__c CS = Hierarchy__c.getValues(RobertId);
System.Assert(CS.OverrideMe__c == 'Fluffy');
// Note how this value is null, because you are returning
// data specific for the user
System.assert(CS.DontOverrideMe__c == null);
```

If Robert passes his System Administrator profile ID `SysAdminID` to `getValues`, the result is:

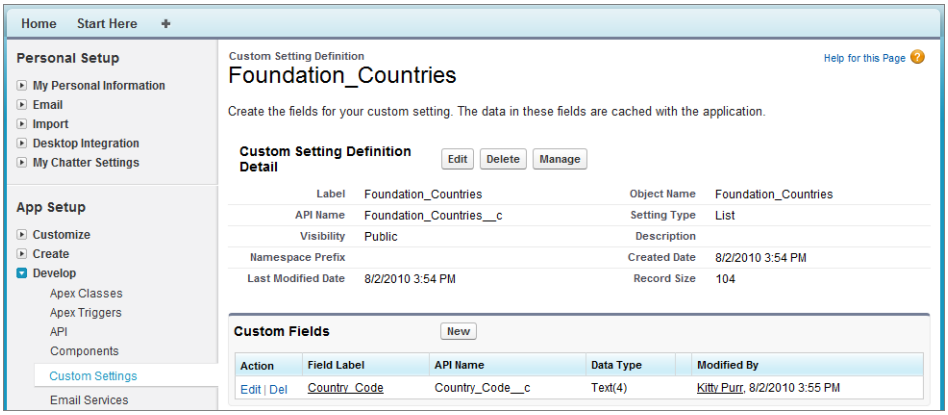
```
Hierarchy__c CS = Hierarchy__c.getValues(SysAdminID);
System.Assert(CS.OverrideMe__c == 'Goodbye');
// Note how this value is null, because you are returning
// data specific for the profile
System.assert(CS.DontOverrideMe__c == null);
```

Country and State Code Custom Settings Example

This example illustrates using two custom setting objects for storing related information, and a Visualforce page to display the data in a set of related picklists.

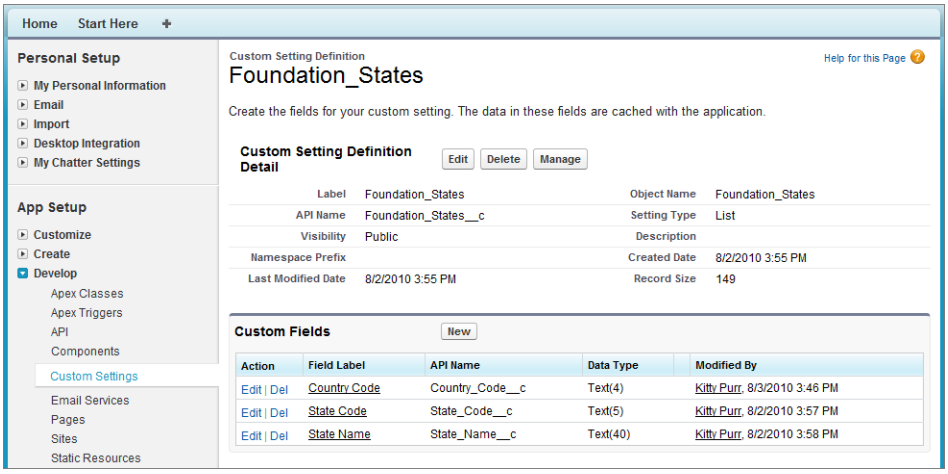
In the following example, country and state codes are stored in two different custom settings: Foundation_Countries and Foundation_States.

The Foundation_Countries custom setting is a list type custom setting and has a single field, `Country_Code`.

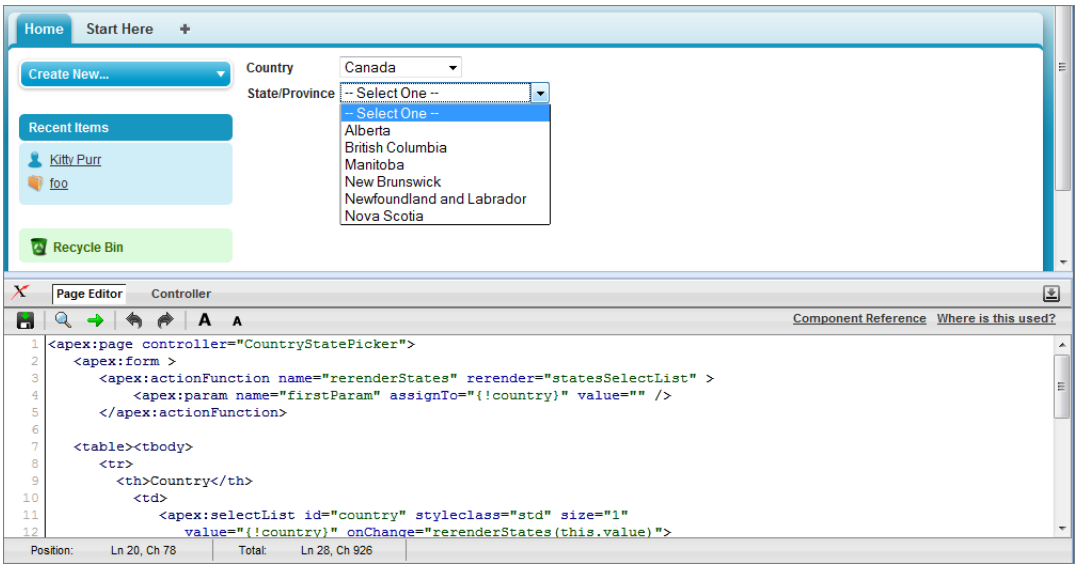


The Foundation_States custom setting is also a List type of custom setting and has the following fields:

- `Country Code`
- `State Code`
- `State Name`



The Visualforce page shows two picklists: one for country and one for state.




```
<apex:page controller="CountryStatePicker">
   <apex:form >
      <apex:actionFunction name="rerenderStates" rerender="statesSelectList" >
          <apex:param name="firstParam" assignTo="{!country}" value="" />
      </apex:actionFunction>

   <table><tbody>
      <tr>
        <th>Country</th>
          <td>
             <apex:selectList id="country" styleclass="std" size="1"
                value="{!country}" onChange="rerenderStates(this.value)">
                    <apex:selectOptions value="{!countriesSelectList}"/>
             </apex:selectList>
          </td>
      </tr>
      <tr id="state_input">
        <th>State/Province</th>
          <td>
            <apex:selectList id="statesSelectList" styleclass="std" size="1"
                 value="{!state}">
                   <apex:selectOptions value="{!statesSelectList}"/>
            </apex:selectList>
          </td>
      </tr>
   </tbody></table>
   </apex:form>
</apex:page>
```

The Apex controller `CountryStatePicker` finds the values entered into the custom settings, then returns them to the Visualforce page.

```
public with sharing class CountryStatePicker {

// Variables to store country and state selected by user
```

```
public String state { get; set; }
public String country {get; set;}

// Generates country dropdown from country settings
public List<SelectOption> getCountriesSelectList() {
    List<SelectOption> options = new List<SelectOption>();
    options.add(new SelectOption('', '-- Select One --'));

    // Find all the countries in the custom setting
   Map<String, Foundation_Countries__c> countries = Foundation_Countries__c.getAll();


    // Sort them by name
    List<String> countryNames = new List<String>();
    countryNames.addAll(countries.keySet());
    countryNames.sort();

    // Create the Select Options.
    for (String countryName : countryNames) {
        Foundation_Countries__c country = countries.get(countryName);
        options.add(new SelectOption(country.country_code__c, country.Name));
    }
    return options;
}

// To generate the states picklist based on the country selected by user.
public List<SelectOption> getStatesSelectList() {
    List<SelectOption> options = new List<SelectOption>();
    // Find all the states we have in custom settings.
    Map<String, Foundation_States__c> allstates = Foundation_States__c.getAll();

    // Filter states that belong to the selected country
   Map<String, Foundation_States__c> states = new Map<String, Foundation_States__c>();

    for(Foundation_States__c state : allstates.values()) {
        if (state.country_code__c == this.country) {
            states.put(state.name, state);
        }
    }

    // Sort the states based on their names
    List<String> stateNames = new List<String>();
    stateNames.addAll(states.keySet());
    stateNames.sort();

    // Generate the Select Options based on the final sorted list
    for (String stateName : stateNames) {
        Foundation_States__c state = states.get(stateName);
        options.add(new SelectOption(state.state_code__c, state.state_name__c));
    }

    // If no states are found, just say not required in the dropdown.
    if (options.size() > 0) {
        options.add(0, new SelectOption('', '-- Select One --'));
```

```
        } else {
            options.add(new SelectOption('', 'Not Required'));
        }
        return options;
    }
}
```

IN THIS SECTION:

List Custom Setting Methods

Hierarchy Custom Setting Methods

SEE ALSO:

Apex Developer Guide: Custom Settings

List Custom Setting Methods

The following are instance methods for list custom settings.

IN THIS SECTION:

getAll()
Returns a map of the data sets defined for the custom setting.

getInstance(dataSetName)
Returns the custom setting data set record for the specified data set name. This method returns the exact same object as `getValues(dataSetName)`.

getValues(dataSetName)
Returns the custom setting data set record for the specified data set name. This method returns the exact same object as `getInstance(dataSetName)`.

`getAll()`

Returns a map of the data sets defined for the custom setting.

Signature

```
public Map<String, CustomSetting__c> getAll()
```

Return Value

Type: Map<String, CustomSetting__c>

Usage

If no data set is defined, this method returns an empty map.

Note: For Apex saved using Salesforce API version 20.0 or earlier, the data set names, which are the keys in the returned map, are converted to lower case. For Apex saved using Salesforce API version 21.0 and later, the case of the data set names in the returned map keys is not changed and the original case is preserved.

`getInstance(dataSetName)`

Returns the custom setting data set record for the specified data set name. This method returns the exact same object as `getValues(dataSetName)`.

Signature

```
public CustomSetting__c getInstance(String dataSetName)
```

Parameters

dataSetName
Type: String

Return Value

Type: CustomSetting__c

Usage

If no data is defined for the specified data set, this method returns `null`.

`getValues(dataSetName)`

Returns the custom setting data set record for the specified data set name. This method returns the exact same object as `getInstance(dataSetName)`.

Signature

```
public CustomSetting__c getValues(String dataSetName)
```

Parameters

dataSetName
Type: String

Return Value

Type: CustomSetting__c

Usage

If no data is defined for the specified data set, this method returns `null`.

Hierarchy Custom Setting Methods

The following are instance methods for hierarchy custom settings.

Note:

- In API version 41.0 and below, each method in an Apex test class, including `testSetup` methods, are able to insert hierarchy custom setting values. This behavior is true even when the methods have the same `SetupOwnerId` value as a hierarchy custom setting record inserted in a different test method.
- In API version 42.0 and later, if a hierarchy custom setting is inserted in a `testSetup` method, inserting a hierarchy custom setting record with the same `SetupOwnerId` in a test method throws a `DUPLICATE_VALUE` exception.

IN THIS SECTION:

getInstance()

Returns a custom setting data set record for the current user. The fields returned in the custom setting record are merged based on the lowest level fields that are defined in the hierarchy.

getInstance(userId)

Returns the custom setting data set record for the specified user ID. The lowest level custom setting record and fields are returned. Use this when you want to explicitly retrieve data for the custom setting at the user level.

getInstance(profileId)

Returns the custom setting data set record for the specified profile ID. The lowest level custom setting record and fields are returned. Use this when you want to explicitly retrieve data for the custom setting at the profile level.

getOrgDefaults()

Returns the custom setting data set record for the organization.

getValues(userId)

Returns the custom setting data set record for the specified user ID.

getValues(profileId)

Returns the custom setting data set for the specified profile ID.

`getInstance()`

Returns a custom setting data set record for the current user. The fields returned in the custom setting record are merged based on the lowest level fields that are defined in the hierarchy.

Signature

```
public CustomSetting__c getInstance()
```

Return Value

Type: CustomSetting__c

Usage

If no custom setting data is defined for the user, this method returns a new custom setting object. The new custom setting object contains an ID set to `null` and merged fields from higher in the hierarchy. You can add this new custom setting record for the user by using `insert` or `upsert`. If no custom setting data is defined in the hierarchy, the returned custom setting has empty fields, except for the `SetupOwnerId` field which contains the user ID.

Note: For Apex saved using Salesforce API version 21.0 or earlier, this method returns the custom setting data set record with fields merged from field values defined at the lowest hierarchy level, starting with the user. Also, if no custom setting data is defined in the hierarchy, this method returns `null`.

This method is equivalent to a method call to `getInstance(User_Id)` for the current user.

Example

- Custom setting data set defined for the user: If you have a custom setting data set defined for the user "Uriel Jones," for the profile "System Administrator," and for the organization as a whole, and the user running the code is Uriel Jones, this method returns the custom setting record defined for Uriel Jones.
- Merged fields: If you have a custom setting data set with fields A and B for the user "Uriel Jones" and for the profile "System Administrator," and field A is defined for Uriel Jones, field B is `null` but is defined for the System Adminitrator profile, this method returns the custom setting record for Uriel Jones with field A for Uriel Jones and field B from the System Administrator profile.
- No custom setting data set record defined for the user: If the current user is "Barbara Mahonie," who also shares the "System Administrator" profile, but no data is defined for Barbara as a user, this method returns a new custom setting record with the ID set to `null` and with fields merged based on the fields defined in the lowest level in the hierarchy.

`getInstance(userId)`

Returns the custom setting data set record for the specified user ID. The lowest level custom setting record and fields are returned. Use this when you want to explicitly retrieve data for the custom setting at the user level.

Signature

```
public CustomSetting__c getInstance(ID userId)
```

Parameters

`userId`
Type: ID

Return Value

Type: CustomSetting__c

Usage

If no custom setting data is defined for the user, this method returns a new custom setting object. The new custom setting object contains an ID set to `null` and merged fields from higher in the hierarchy. You can add this new custom setting record for the user by using `insert` or `upsert`. If no custom setting data is defined in the hierarchy, the returned custom setting has empty fields, except for the `SetupOwnerId` field which contains the user ID.

Note: For Apex saved using Salesforce API version 21.0 or earlier, this method returns the custom setting data set record with fields merged from field values defined at the lowest hierarchy level, starting with the user. Also, if no custom setting data is defined in the hierarchy, this method returns `null`.

`getInstance(profileId)`

Returns the custom setting data set record for the specified profile ID. The lowest level custom setting record and fields are returned. Use this when you want to explicitly retrieve data for the custom setting at the profile level.

Signature

```
public CustomSetting__c getInstance(ID profileId)
```

Parameters

profileId
Type: ID

Return Value

Type: CustomSetting__c

Usage

If no custom setting data is defined for the profile, this method returns a new custom setting record. The new custom setting object contains an ID set to `null` and with merged fields from your organization's default values. You can add this new custom setting for the profile by using `insert` or `upsert`. If no custom setting data is defined in the hierarchy, the returned custom setting has empty fields, except for the `SetupOwnerId` field which contains the profile ID.

Note: For Apex saved using SalesforceAPI version 21.0 or earlier, this method returns the custom setting data set record with fields merged from field values defined at the lowest hierarchy level, starting with the profile. Also, if no custom setting data is defined in the hierarchy, this method returns `null`.

`getOrgDefaults()`

Returns the custom setting data set record for the organization.

Signature

```
public CustomSetting__c getOrgDefaults()
```

Return Value

Type: CustomSetting__c

Usage

If no custom setting data is defined for the organization, this method returns an empty custom setting object.

Note: For Apex saved using Salesforce API version 21.0 or earlier, this method returns `null` if no custom setting data is defined for the organization.

`getValues(userId)`

Returns the custom setting data set record for the specified user ID.

Signature

```
public CustomSetting__c getValues(ID userId)
```

Parameters

userId
Type: ID

Return Value

Type: CustomSetting__c

Usage

Use this if you only want the subset of custom setting data that has been defined at the user level. For example, suppose you have a custom setting field that has been assigned a value of "alpha" at the organizational level, but has no value assigned at the user or profile level. Using `getValues(UserId)` returns `null` for this custom setting field.

`getValues(profileId)`

Returns the custom setting data set for the specified profile ID.

Signature

```
public CustomSetting__c getValues(ID profileId)
```

Parameters

profileId
Type: ID

Return Value

Type: CustomSetting__c

Usage

Use this if you only want the subset of custom setting data that has been defined at the profile level. For example, suppose you have a custom setting field that has been assigned a value of "alpha" at the organizational level, but has no value assigned at the user or profile level. Using `getValues(ProfileId)` returns `null` for this custom setting field.

Database Class

Contains methods for creating and manipulating data.

Namespace

System

Usage

Some Database methods also exist as DML statements.

SEE ALSO:

Apex DML Operations

Database Methods

The following are methods for `Database`. All methods are static.

IN THIS SECTION:

convertLead(leadToConvert, allOrNone)
Converts a lead into an account and contact, as well as (optionally) an opportunity.

convertLead(leadsToConvert, allOrNone)
Converts a list of LeadConvert objects into accounts and contacts, as well as (optionally) opportunities.

convertLead(leadToConvert, dmlOptions)
Converts a lead into an account and contact, as well as (optionally) an opportunity.

convertLead(leadsToConvert, dmlOptions)
Converts a list of LeadConvert objects into accounts and contacts, as well as (optionally) opportunities.

convertLead(leadToConvert, allOrNone, accessLevel)
Converts a lead into an account and contact, as well as (optionally) an opportunity.

convertLead(leadsToConvert, allOrNone, accessLevel)
Converts a list of LeadConvert objects into accounts and contacts, as well as (optionally) opportunities.

convertLead(leadToConvert, dmlOptions, accessLevel)
Converts a lead into an account and contact, as well as (optionally) an opportunity.

convertLead(leadsToConvert, dmlOptions, accessLevel)
Converts a list of LeadConvert objects into accounts and contacts, as well as (optionally) opportunities.

countQuery(query)
Returns the number of records that a dynamic SOQL query would return when executed.

countQuery(query, accessLevel)
Returns the number of records that a dynamic SOQL query would return when executed.

countQueryWithBinds(query, bindMap, accessLevel)
Returns the number of records that a dynamic SOQL query would return when executed. Bind variables in the query are resolved from the *`bindMap`* Map parameter directly with the key, rather than from Apex code variables.

delete(recordToDelete, allOrNone)
Deletes an existing sObject record, such as an individual account or contact, from your organization's data.

delete(recordsToDelete, allOrNone)
Deletes a list of existing sObject records, such as individual accounts or contacts, from your organization's data.

delete(recordID, allOrNone)
Deletes existing sObject records, such as individual accounts or contacts, from your organization's data.

delete(recordIDs, allOrNone)

Deletes a list of existing sObject records, such as individual accounts or contacts, from your organization's data.

delete(recordToDelete, allOrNone, accessLevel)

Deletes an existing sObject record, such as an individual account or contact, from your organization's data.

delete(recordsToDelete, allOrNone, accessLevel)

Deletes a list of existing sObject records, such as individual accounts or contacts, from your organization's data.

delete(recordID, allOrNone, accessLevel)

Deletes existing sObject records, such as individual accounts or contacts, from your organization's data.

delete(recordIDs, allOrNone, accessLevel)

Deletes a list of existing sObject records, such as individual accounts or contacts, from your organization's data.

deleteAsync(sobjects, callback)

Initiates requests to delete the external data that corresponds to the specified external object records. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source. Allows referencing a callback class whose `processDelete` method is called for each record after deletion.

deleteAsync(sobject, callback)

Initiates a request to delete the external data that corresponds to the specified external object record. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source. Allows referencing a callback class whose `processDelete` method is called after deletion.

deleteAsync(sobjects)

Initiates requests to delete the external data that corresponds to the specified external object records. The requests are executed asynchronously, as background operations, and are sent to the external systems that are defined by the external objects' associated external data sources.

deleteAsync(sobject)

Initiates a request to delete the external data that corresponds to the specified external object record. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source.

deleteAsync(sobjects, callback, accessLevel)

Initiates requests to delete the external data that corresponds to the specified external object records. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source. Allows referencing a callback class whose `processDelete` method is called for each record after deletion.

deleteAsync(sobject, callback, accessLevel)

Initiates a request to delete the external data that corresponds to the specified external object record. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source. Allows referencing a callback class whose `processDelete` method is called after deletion.

deleteAsync(sobjects, accessLevel)

Initiates requests to delete the external data that corresponds to the specified external object records. The requests are executed asynchronously, as background operations, and are sent to the external systems that are defined by the external objects' associated external data sources.

deleteAsync(sobject, accessLevel)

Initiates a request to delete the external data that corresponds to the specified external object record. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source.

deleteImmediate(sobjects)

Initiates requests to delete the external data that corresponds to the specified external object records. The requests are executed synchronously and are sent to the external systems that are defined by the external objects' associated external data sources. If the Apex transaction contains pending changes, the synchronous operations can't be completed and throw exceptions.

deleteImmediate(sobject)

Initiates a request to delete the external data that corresponds to the specified external object record. The request is executed synchronously and is sent to the external system that's defined by the external object's associated external data source. If the Apex transaction contains pending changes, the synchronous operation can't be completed and throws an exception.

deleteImmediate(sobjects, accessLevel)

Initiates requests to delete the external data that corresponds to the specified external object records. The requests are executed synchronously and are sent to the external systems that are defined by the external objects' associated external data sources. If the Apex transaction contains pending changes, the synchronous operations can't be completed and throw exceptions.

deleteImmediate(sobject, accessLevel)

Initiates a request to delete the external data that corresponds to the specified external object record. The request is executed synchronously and is sent to the external system that's defined by the external object's associated external data source. If the Apex transaction contains pending changes, the synchronous operation can't be completed and throws an exception.

emptyRecycleBin(recordIds)

Permanently deletes the specified records from the Recycle Bin.

emptyRecycleBin(obj)

Permanently deletes the specified sObject from the Recycle Bin.

emptyRecycleBin(listOfSObjects)

Permanently deletes the specified sObjects from the Recycle Bin.

executeBatch(batchClassObject)

Submits a batch Apex job for execution corresponding to the specified class.

executeBatch(batchClassObject, scope)

Submits a batch Apex job for execution using the specified class and scope.

getAsyncDeleteResult(deleteResult)

Retrieves the status of an asynchronous delete operation that's identified by a `Database.DeleteResult` object.

getAsyncDeleteResult(asyncLocator)

Retrieves the result of an asynchronous delete operation based on the result's unique identifier.

getAsyncLocator(result)

Returns the `asyncLocator` associated with the result of a specified asynchronous insert, update, or delete operation.

getAsyncSaveResult(saveResult)

Returns the status of an asynchronous insert or update operation that's identified by a `Database.SaveResult` object.

getAsyncSaveResult(asyncLocator)

Returns the status of an asynchronous insert or update operation based on the unique identifier associated with each modification.

getCursor(query)

Creates a cursor when the specified SOQL query is executed.

getCursor(query, accessLevel)

Creates a cursor when the specified SOQL query is executed.

getCursorWithBinds(query, bindMap, accessLevel)

Creates a cursor when the specified SOQL query is executed.

getDeleted(sObjectType, startDate, endDate)

Returns the list of individual records that have been deleted for an sObject type within the specified start and end dates and times and that are still in the Recycle Bin.

getQueryLocator(staticSoqlQueryResult)

Creates a QueryLocator object used in batch Apex or Visualforce.

getQueryLocator(query)

Creates a QueryLocator object used in batch Apex or Visualforce.

getQueryLocator(staticSoqlQueryResult, accessLevel)

Creates a QueryLocator object used in batch Apex or Visualforce.

getQueryLocator(query, accessLevel)

Creates a QueryLocator object used in batch Apex or Visualforce.

getQueryLocatorWithBinds(query, bindMap, accessLevel)

Creates a QueryLocator object used in batch Apex or Visualforce. Bind variables in the query are resolved from the *`bindMap`* Map parameter directly with the key, rather than from Apex code variables.

getUpdated(sobjectType, startDate, endDate)

Returns the list of individual records that have been updated for an sObject type within the specified start and end dates and times.

insert(recordToInsert, allOrNone)

Adds an sObject, such as an individual account or contact, to your organization's data.

insert(recordsToInsert, allOrNone)

Adds one or more sObjects, such as individual accounts or contacts, to your organization’s data.

insert(recordToInsert, dmlOptions)

Adds an sObject, such as an individual account or contact, to your organization's data.

insert(recordsToInsert, dmlOptions)

Adds one or more sObjects, such as individual accounts or contacts, to your organization's data.

insert(recordToInsert, allOrNone, accessLevel)

Adds an sObject, such as an individual account or contact, to your organization's data.

insert(recordsToInsert, allOrNone, accessLevel)

Adds one or more sObjects, such as individual accounts or contacts, to your organization’s data.

insert(recordToInsert, dmlOptions, accessLevel)

Adds an sObject, such as an individual account or contact, to your organization's data.

insert(recordsToInsert, dmlOptions, accessLevel)

Adds one or more sObjects, such as individual accounts or contacts, to your organization's data.

insertAsync(sobjects, callback)

Initiates requests to add external object data to the relevant external systems. The requests are executed asynchronously, as background operations, and are sent to the external systems that are defined by the external objects' associated external data sources. Allows referencing a callback class whose `processSave` method is called for each record after the remote operations are completed.

insertAsync(sobject, callback)

Initiates a request to add external object data to the relevant external system. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source. Allows referencing a callback class whose `processSave` method is called after the remote operation is completed.

insertAsync(sobjects)

Initiates requests to add external object data to the relevant external systems. The requests are executed asynchronously, as background operations, and are sent to the external systems that are defined by the external objects' associated external data sources.

insertAsync(sobject)

Initiates a request to add external object data to the relevant external system. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source.

insertAsync(sobjects, callback, accessLevel)

Initiates requests to add external object data to the relevant external systems. The requests are executed asynchronously, as background operations, and are sent to the external systems that are defined by the external objects' associated external data sources. Allows referencing a callback class whose `processSave` method is called for each record after the remote operations are completed.

insertAsync(sobject, callback, accessLevel)

Initiates a request to add external object data to the relevant external system. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source. Allows referencing a callback class whose `processSave` method is called after the remote operation is completed.

insertAsync(sobjects, accessLevel)

Initiates requests to add external object data to the relevant external systems. The requests are executed asynchronously, as background operations, and are sent to the external systems that are defined by the external objects' associated external data sources.

insertAsync(sobject, accessLevel)

Initiates a request to add external object data to the relevant external system. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source.

insertImmediate(sobjects)

Initiates requests to add external object data to the relevant external systems. The requests are executed synchronously and are sent to the external systems that are defined by the external objects' associated external data sources. If the Apex transaction contains pending changes, the synchronous operations can't be completed and throw exceptions.

insertImmediate(sobject)

Initiates a request to add external object data to the relevant external system. The request is executed synchronously and is sent to the external system that's defined by the external object's associated external data source. If the Apex transaction contains pending changes, the synchronous operation can't be completed and throws an exception.

insertImmediate(sobjects, accessLevel)

Initiates requests to add external object data to the relevant external systems. The requests are executed synchronously and are sent to the external systems that are defined by the external objects' associated external data sources. If the Apex transaction contains pending changes, the synchronous operations can't be completed and throw exceptions.

insertImmediate(sobject, accessLevel)

Initiates a request to add external object data to the relevant external system. The request is executed synchronously and is sent to the external system that's defined by the external object's associated external data source. If the Apex transaction contains pending changes, the synchronous operation can't be completed and throws an exception.

merge(mergeToRecord, duplicateId)

Merges the duplicate record into the `mergeToRecord` sObject record of the same type, deleting the duplicate, and reparenting any related records. Merges only accounts, contacts, or leads.

merge(mergeToRecord, duplicateRecord)

Merges the duplicate sObject record into the `mergeToRecord` sObject record of the same type, deleting the duplicate, and reparenting any related records.

merge(mergeToRecord, duplicateIds)

Merges up to two records of the same sObject type into the `mergeToRecord` sObject record, deleting the others, and reparenting any related records.

merge(mergeToRecord, duplicateRecords)

Merges up to two records of the same object type into the `mergeToRecord` sObject record, deleting the others, and reparenting any related records.

merge(mergeToRecord, duplicateId, allOrNone)

Merges the duplicate record into the `mergeToRecord` sObject record of the same type, optionally returning any errors, deleting the duplicate, and reparenting any related records. Merges only accounts, contacts, or leads.

merge(mergeToRecord, duplicateRecord, allOrNone)

Merges the duplicate sObject record into the `mergeToRecord` sObject of the same type, optionally returning any errors, deleting the duplicate, and reparenting any related records.

merge(mergeToRecord, duplicateIds, allOrNone)

Merges up to two records of the same sObject type into the `mergeToRecord` sObject record, optionally returning any errors, deleting the duplicates, and reparenting any related records.

merge(mergeToRecord, duplicateRecords, allOrNone)

Merges up to two records of the same object type into the `mergeToRecord` sObject record, optionally returning any errors, deleting the duplicates, and reparenting any related records.

merge(mergeToRecord, duplicateId, accessLevel)

Merges the duplicate record into the `mergeToRecord` sObject record of the same type, deleting the duplicate, and reparenting any related records. Merges only accounts, contacts, or leads.

merge(mergeToRecord, duplicateRecord, accessLevel)

Merges the specified duplicate sObject record into the `mergeToRecord` sObject of the same type, deleting the duplicate, and reparenting any related records.

merge(mergeToRecord, duplicateIds, accessLevel)

Merges up to two records of the same sObject type into the `mergeToRecord` sObject record, deleting the others, and reparenting any related records.

merge(mergeToRecord, duplicateRecords, accessLevel)

Merges up to two records of the same object type into the `mergeToRecord` sObject record, deleting the others, and reparenting any related records.

merge(mergeToRecord, duplicateId, allOrNone, accessLevel)

Merges the duplicate record into the `mergeToRecord` sObject record of the same type, optionally returning any errors, deleting the duplicate, and reparenting any related records. Merges only accounts, contacts, or leads.

merge(mergeToRecord, duplicateRecord, allOrNone, accessLevel)

Merges the duplicate sObject record into the `mergeToRecord` sObject record of the same type, optionally returning any errors, deleting the duplicate, and reparenting any related records.

merge(mergeToRecord, duplicateIds, allOrNone, accessLevel)

Merges up to two records of the same sObject type into the `mergeToRecord` sObject record, optionally returning any errors, deleting the duplicates, and reparenting any related records.

merge(mergeToRecord, duplicateRecords, allOrNone, accessLevel)

Merges up to two records of the same object type into the `mergeToRecord` sObject record, optionally returning any errors, deleting the duplicates, and reparenting any related records.

query(queryString)

Creates a dynamic SOQL query at runtime.

query(queryString, accessLevel)

Creates a dynamic SOQL query at runtime.

queryWithBinds(queryString, bindMap, accessLevel)

Creates a dynamic SOQL query at runtime. Bind variables in the query are resolved from the *`bindMap`* Map parameter directly with the key, rather than from Apex code variables.

releaseSavepoint(databaseSavepoint)

Releases a given savepoint. All savepoints that are subsequent to the given one are also released.

rollback(databaseSavepoint)

Restores the database to the state specified by the savepoint variable. Any emails submitted since the last savepoint are also rolled back and not sent.

setSavepoint()

Returns a savepoint variable that can be stored as a local variable, then used with the `rollback` method to restore the database to that point.

undelete(recordToUndelete, allOrNone)

Restores an existing sObject record, such as an individual account or contact, from your organization's Recycle Bin.

undelete(recordsToUndelete, allOrNone)

Restores one or more existing sObject records, such as individual accounts or contacts, from your organization's Recycle Bin.

undelete(recordID, allOrNone)

Restores an existing sObject record, such as an individual account or contact, from your organization's Recycle Bin.

undelete(recordIDs, allOrNone)

Restores one or more existing sObject records, such as individual accounts or contacts, from your organization's Recycle Bin.

undelete(recordToUndelete, allOrNone, accessLevel)

Restores an existing sObject record, such as an individual account or contact, from your organization's Recycle Bin.

undelete(recordsToUndelete, allOrNone, accessLevel)

Restores one or more existing sObject records, such as individual accounts or contacts, from your organization's Recycle Bin.

undelete(recordID, allOrNone, accessLevel)

Restores an existing sObject record, such as an individual account or contact, from your organization's Recycle Bin.

undelete(recordIDs, allOrNone, accessLevel)

Restores one or more existing sObject records, such as individual accounts or contacts, from your organization's Recycle Bin.

update(recordToUpdate, allOrNone)

Modifies an existing sObject record, such as an individual account or contact, in your organization's data.

update(recordsToUpdate, allOrNone)

Modifies one or more existing sObject records, such as individual accounts or contacts, in your organization's data.

update(recordToUpdate, dmlOptions)

Modifies an existing sObject record, such as an individual account or contact, in your organization's data.

update(recordsToUpdate, dmlOptions)

Modifies one or more existing sObject records, such as individual accounts or contacts, in your organization's data.

update(recordToUpdate, allOrNone, accessLevel)

Modifies an existing sObject record, such as an individual account or contact, in your organization's data.

update(recordsToUpdate, allOrNone, accessLevel)

Modifies one or more existing sObject records, such as individual accounts or contacts, in your organization's data.

update(recordToUpdate, dmlOptions, accessLevel)

Modifies an existing sObject record, such as an individual account or contact, in your organization's data.

update(recordsToUpdate, dmlOptions, accessLevel)

Modifies one or more existing sObject records, such as individual accounts or contacts, in your organization's data.

upsert(recordToUpsert, externalIdField, allOrNone)

Creates a new sObject record or updates an existing sObject record within a single statement, using a specified field to determine the presence of existing objects, or the ID field if no field is specified.

upsert(recordsToUpsert, externalIdField, allOrNone)

Creates new sObject records or updates existing sObject records within a single statement, using a specified field to determine the presence of existing objects, or the ID field if no field is specified.

upsert(recordToUpsert, externalIdField, allOrNone, accessLevel)

Creates a new sObject record or updates an existing sObject record within a single statement, using a specified field to determine the presence of existing objects, or the ID field if no field is specified.

upsert(recordsToUpsert, externalIdField, allOrNone, accessLevel)

Creates new sObject records or updates existing sObject records within a single statement, using a specified field to determine the presence of existing objects, or the ID field if no field is specified.

updateAsync(sobjects, callback)

Initiates requests to update external object data on the relevant external systems. The requests are executed asynchronously, as background operations, and are sent to the external systems that are defined by the external objects' associated external data sources. Allows referencing a callback class whose `processSave` method is called for each record after the remote operations are completed.

updateAsync(sobject, callback)

Initiates a request to update external object data on the relevant external system. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source. Allows referencing a callback class whose `processSave` method is called after the remote operation is completed.

updateAsync(sobjects)

Initiates requests to update external object data on the relevant external systems. The requests are executed asynchronously, as background operations, and are sent to the external systems that are defined by the external objects' associated external data sources.

updateAsync(sobject)

Initiates a request to update external object data on the relevant external system. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source.

updateAsync(sobjects, callback, accessLevel)

Initiates requests to update external object data on the relevant external systems. The requests are executed asynchronously, as background operations, and are sent to the external systems that are defined by the external objects' associated external data sources. Allows referencing a callback class whose `processSave` method is called for each record after the remote operations are completed.

updateAsync(sobject, callback, accessLevel)

Initiates a request to update external object data on the relevant external system. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source. Allows referencing a callback class whose `processSave` method is called after the remote operation is completed.

updateAsync(sobjects, accessLevel)

Initiates requests to update external object data on the relevant external systems. The requests are executed asynchronously, as background operations, and are sent to the external systems that are defined by the external objects' associated external data sources.

updateAsync(sobject, accessLevel)

Initiates a request to update external object data on the relevant external system. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source.

updateImmediate(sobjects)

Initiates requests to update external object data on the relevant external systems. The requests are executed synchronously and are sent to the external systems that are defined by the external objects' associated external data sources. If the Apex transaction contains pending changes, the synchronous operations can't be completed and throw exceptions.

updateImmediate(sobject)

Initiates a request to update external object data on the relevant external system. The request is executed synchronously and is sent to the external system that's defined by the external object's associated external data source. If the Apex transaction contains pending changes, the synchronous operation can't be completed and throws an exception.

updateImmediate(sobjects, accessLevel)

Initiates requests to update external object data on the relevant external systems. The requests are executed synchronously and are sent to the external systems that are defined by the external objects' associated external data sources. If the Apex transaction contains pending changes, the synchronous operations can't be completed and throw exceptions.

updateImmediate(sobject, accessLevel)

Initiates a request to update external object data on the relevant external system. The request is executed synchronously and is sent to the external system that's defined by the external object's associated external data source. If the Apex transaction contains pending changes, the synchronous operation can't be completed and throws an exception.

`convertLead(leadToConvert, allOrNone)`

Converts a lead into an account and contact, as well as (optionally) an opportunity.

Signature

```
public static Database.LeadConvertResult convertLead(Database.LeadConvert leadToConvert,
Boolean allOrNone)
```

Parameters

`leadToConvert`
Type: Database.LeadConvert

`allOrNone`
Type: Boolean

(Optional) The *`allOrNone`* parameter specifies whether the operation allows partial success. If *`allOrNone`* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *`allOrNone`* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

Return Value

Type: Database.LeadConvertResult

Usage

We recommend passing a maximum of 100 `LeadConvert` objects to the `convertLead` method. Including more than 100 objects per call can result in Apex governor limit errors.

Each executed `convertLead` method counts against the governor limit for DML statements.

convertLead(leadsToConvert, allOrNone)

Converts a list of LeadConvert objects into accounts and contacts, as well as (optionally) opportunities.

Signature

```
public static Database.LeadConvertResult[] convertLead(Database.LeadConvert[]
leadsToConvert, Boolean allOrNone)
```

Parameters

leadsToConvert
Type: Database.LeadConvert[]

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

Return Value

Type: Database.LeadConvertResult[]

Usage

We recommend passing a maximum of 100 `LeadConvert` objects to the `convertLead` method. Including more than 100 objects per call can result in Apex governor limit errors.

Each executed `convertLead` method counts against the governor limit for DML statements.

convertLead(leadToConvert, dmlOptions)

Converts a lead into an account and contact, as well as (optionally) an opportunity.

Signature

```
public static Database.LeadConvertResult convertLead(Database.LeadConvert leadToConvert,
Database.DMLOptions dmlOptions)
```

Parameters

leadToConvert
Type: Database.LeadConvert

dmlOptions
Type: Database.DMLOptions

The optional *dmlOptions* parameter specifies additional data for the transaction, such as assignment rule information or rollback behavior when errors occur during record insertions.

Return Value

Type: Database.LeadConvertResult

Usage

We recommend passing a maximum of 100 `LeadConvert` objects to the `convertLead` method. Including more than 100 objects per call can result in Apex governor limit errors.

Each executed `convertLead` method counts against the governor limit for DML statements.

convertLead(leadsToConvert, dmlOptions)

Converts a list of LeadConvert objects into accounts and contacts, as well as (optionally) opportunities.

Signature

```
public static List<Database.LeadConvertResult> convertLead(List<Database.LeadConvert>
leadsToConvert, Database.DMLOptions dmlOptions)
```

Parameters

leadsToConvert
Type: List<Database.LeadConvert>

dmlOptions
Type: Database.DMLOptions

The optional *dmlOptions* parameter specifies additional data for the transaction, such as assignment rule information or rollback behavior when errors occur during record insertions.

Return Value

Type: List<Database.LeadConvertResult>

Usage

We recommend passing a maximum of 100 `LeadConvert` objects to the `convertLead` method. Including more than 100 objects per call can result in Apex governor limit errors.

Each executed `convertLead` method counts against the governor limit for DML statements.

convertLead(leadToConvert, allOrNone, accessLevel)

Converts a lead into an account and contact, as well as (optionally) an opportunity.

Signature

```
public static Database.LeadConvertResult convertLead(Database.LeadConvert leadToConvert,
Boolean allOrNone, System.AccessLevel accessLevel)
```

Parameters

leadToConvert
Type: Database.LeadConvert

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.LeadConvertResult

Usage

We recommend passing a maximum of 100 `LeadConvert` objects to the `convertLead` method. Including more than 100 objects per call can result in Apex governor limit errors.

Each executed `convertLead` method counts against the governor limit for DML statements.

convertLead(leadsToConvert, allOrNone, accessLevel)

Converts a list of LeadConvert objects into accounts and contacts, as well as (optionally) opportunities.

Signature

```
public static List<Database.LeadConvertResult> convertLead(List<Database.LeadConvert>
leadsToConvert, Boolean allOrNone, System.AccessLevel accessLevel)
```

Parameters

leadsToConvert
Type: List<Database.LeadConvert>

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: List<Database.LeadConvertResult>

Usage

We recommend passing a maximum of 100 `LeadConvert` objects to the `convertLead` method. Including more than 100 objects per call can result in Apex governor limit errors.

Each executed `convertLead` method counts against the governor limit for DML statements.

convertLead(leadToConvert, dmlOptions, accessLevel)

Converts a lead into an account and contact, as well as (optionally) an opportunity.

Signature

```
public static Database.LeadConvertResult convertLead(Database.LeadConvert leadToConvert,
Database.DMLOptions dmlOptions, System.AccessLevel accessLevel)
```

Parameters

leadToConvert
Type: Database.LeadConvert

dmlOptions
Type: Database.DMLOptions

The optional *dmlOptions* parameter specifies additional data for the transaction, such as assignment rule information or rollback behavior when errors occur during record insertions.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.LeadConvertResult

Usage

We recommend passing a maximum of 100 `LeadConvert` objects to the `convertLead` method. Including more than 100 objects per call can result in Apex governor limit errors.

Each executed `convertLead` method counts against the governor limit for DML statements.

convertLead(leadsToConvert, dmlOptions, accessLevel)

Converts a list of LeadConvert objects into accounts and contacts, as well as (optionally) opportunities.

Signature

```
public static List<Database.LeadConvertResult> convertLead(List<Database.LeadConvert>
leadsToConvert, Database.DMLOptions dmlOptions, System.AccessLevel accessLevel)
```

Parameters

leadsToConvert
Type: List<Database.LeadConvert>

dmlOptions
Type: Database.DMLOptions

The optional *dmlOptions* parameter specifies additional data for the transaction, such as assignment rule information or rollback behavior when errors occur during record insertions.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: List<Database.LeadConvertResult>

Usage

We recommend passing a maximum of 100 `LeadConvert` objects to the `convertLead` method. Including more than 100 objects per call can result in Apex governor limit errors.

Each executed `convertLead` method counts against the governor limit for DML statements.

countQuery(query)

Returns the number of records that a dynamic SOQL query would return when executed.

Signature

```
public static Integer countQuery(String query)
```

Parameters

query
Type: String

Return Value

Type: Integer

Usage

For more information, see Dynamic SOQL.

Each executed `countQuery` method counts against the governor limit for SOQL queries.

Example

```
String QueryString =
    'SELECT count() FROM Account';
Integer i =
    Database.countQuery(QueryString);
```

countQuery(query, accessLevel)

Returns the number of records that a dynamic SOQL query would return when executed.

Signature

```
public static Integer countQuery(String query, System.AccessLevel accessLevel)
```

Parameters

query
Type: String

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Integer

Usage

For more information, see Dynamic SOQL.

Each executed `countQuery` method counts against the governor limit for SOQL queries.

countQueryWithBinds(query, bindMap, accessLevel)

Returns the number of records that a dynamic SOQL query would return when executed. Bind variables in the query are resolved from the *bindMap* Map parameter directly with the key, rather than from Apex code variables.

Signature

```
public static Integer countQueryWithBinds(String query, Map<String, Object> bindMap,
System.AccessLevel accessLevel)
```

Parameters

query
Type: String

SOQL query that includes Apex bind variables preceded by a colon. All bind variables must have a key in the *bindMap* Map.

bindMap
Type: Map<String, Object>

A map that contains keys for each bind variable specified in the SOQL *queryString* and its value. The keys can't be null or duplicates, and the values can't be null or empty strings.

accessLevel
Type: System.AccessLevel

The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced.

Return Value

Type: Integer

Usage

For more information, see Dynamic SOQL.

Each executed `countQueryWithBinds` method counts against the governor limit for SOQL queries.

Example

In this example, the SOQL query uses a bind variable for an Account name. Its value (`Acme Inc.`) is passed in to the method using the *nameBind* Map. The `accountName` variable isn't (and doesn't have to be) in scope when the query is executed within the method.

```
public static Integer simpleBindingSoqlQuery(Map<String, Object> bindParams) {
    String queryString =
        'SELECT count() ' +
        'FROM Account ' +
        'WHERE name = :name';
    return Database.countQueryWithBinds(
        queryString,
        bindParams,
```

```
        AccessLevel.USER_MODE
    );
}

String accountName = 'Acme Inc.';
Map<String, Object> nameBind = new Map<String, Object>{'name' => accountName};
Integer acctCount = simpleBindingSoqlQuery(nameBind);
System.debug(acctCount);
```

delete(recordToDelete, allOrNone)

Deletes an existing sObject record, such as an individual account or contact, from your organization's data.

Signature

```
public static Database.DeleteResult delete(SObject recordToDelete, Boolean allOrNone)
```

Parameters

recordToDelete
Type: sObject

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

Return Value

Type: Database.DeleteResult

Usage

`delete` is analogous to the `delete()` statement in the SOAP API.

Each executed `delete` method counts against the governor limit for DML statements.

delete(recordsToDelete, allOrNone)

Deletes a list of existing sObject records, such as individual accounts or contacts, from your organization's data.

Signature

```
public static Database.DeleteResult[] delete(SObject[] recordsToDelete, Boolean
allOrNone)
```

Parameters

recordsToDelete
Type: sObject[]

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

Return Value

Type: Database.DeleteResult[]

Usage

`delete` is analogous to the `delete()` statement in the SOAP API.

Each executed `delete` method counts against the governor limit for DML statements.

Example

The following example deletes an account named 'DotCom':

```
Account[] doomedAccts = [SELECT Id, Name FROM Account WHERE Name = 'DotCom'];
Database.DeleteResult[] DR_Dels = Database.delete(doomedAccts);
```

delete(recordID, allOrNone)

Deletes existing sObject records, such as individual accounts or contacts, from your organization's data.

Signature

```
public static Database.DeleteResult delete(ID recordID, Boolean allOrNone)
```

Parameters

recordID
Type: ID

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

Return Value

Type: Database.DeleteResult

Usage

`delete` is analogous to the `delete()` statement in the SOAP API.

Each executed `delete` method counts against the governor limit for DML statements.

`delete(recordIDs, allOrNone)`

Deletes a list of existing sObject records, such as individual accounts or contacts, from your organization's data.

Signature

```
public static Database.DeleteResult[] delete(ID[] recordIDs, Boolean allOrNone)
```

Parameters

`recordIDs`
Type: ID[]

`allOrNone`
Type: Boolean

(Optional) The *`allOrNone`* parameter specifies whether the operation allows partial success. If *`allOrNone`* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *`allOrNone`* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

Return Value

Type: Database.DeleteResult[]

Usage

`delete` is analogous to the `delete()` statement in the SOAP API.

Each executed `delete` method counts against the governor limit for DML statements.

`delete(recordToDelete, allOrNone, accessLevel)`

Deletes an existing sObject record, such as an individual account or contact, from your organization's data.

Signature

```
public static Database.DeleteResult delete(SObject recordToDelete, Boolean allOrNone,
System.AccessLevel accessLevel)
```

Parameters

`recordToDelete`
Type: sObject

`allOrNone`
Type: Boolean

(Optional) The *`allOrNone`* parameter specifies whether the operation allows partial success. If *`allOrNone`* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *`allOrNone`* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.DeleteResult

Usage

`delete` is analogous to the `delete()` statement in the SOAP API.

Each executed `delete` method counts against the governor limit for DML statements.

delete(recordsToDelete, allOrNone, accessLevel)

Deletes a list of existing sObject records, such as individual accounts or contacts, from your organization's data.

Signature

```
public static List<Database.DeleteResult> delete(List<SObject> recordsToDelete, Boolean
allOrNone, System.AccessLevel accessLevel)
```

Parameters

recordsToDelete
Type: List<sObject>

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: List<Database.DeleteResult>

Usage

`delete` is analogous to the `delete()` statement in the SOAP API.

Each executed `delete` method counts against the governor limit for DML statements.

delete(recordID, allOrNone, accessLevel)

Deletes existing sObject records, such as individual accounts or contacts, from your organization's data.

Signature

```
public static Database.DeleteResult delete(Id recordID, Boolean allOrNone,
System.AccessLevel accessLevel)
```

Parameters

recordID
Type: ID

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.DeleteResult

Usage

`delete` is analogous to the `delete()` statement in the SOAP API.

Each executed `delete` method counts against the governor limit for DML statements.

delete(recordIDs, allOrNone, accessLevel)

Deletes a list of existing sObject records, such as individual accounts or contacts, from your organization's data.

Signature

```
public static List<Database.DeleteResult> delete(List<Id> recordIDs, Boolean allOrNone,
System.AccessLevel accessLevel)
```

Parameters

recordIDs
Type: List<ID>

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: List<Database.DeleteResult>

Usage

`delete` is analogous to the `delete()` statement in the SOAP API.

Each executed `delete` method counts against the governor limit for DML statements.

deleteAsync(sobjects, callback)

Initiates requests to delete the external data that corresponds to the specified external object records. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source. Allows referencing a callback class whose `processDelete` method is called for each record after deletion.

Signature

```
public static List<Database.DeleteResult> deleteAsync(List<SObject> sobjects,
DataSource.AsyncDeleteCallback callback)
```

Parameters

sobjects
Type: List<SObject>

List of external object records to delete.

callback
Type: DataSource.AsyncDeleteCallback

The callback that contains the state in the originating context and an action (the `processDelete` method) that is executed after the insert operation is completed. Use the action callback to update org data according to the operation's results. The callback object must extend `DataSource.AsyncDeleteCallback`.

Return Value

Type: List<Database.DeleteResult>

Status results for the delete operation. Each result corresponds to a record processed by this asynchronous operation and is associated with a unique identifier (`asyncLocator`). The `asyncLocator` value is included in the errors array of the result. You can retrieve this identifier with `Database.getAsyncLocator()`. Retrieve the final result with `Database.getAsyncDeleteResult()`.

`deleteAsync(sobject, callback)`

Initiates a request to delete the external data that corresponds to the specified external object record. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source. Allows referencing a callback class whose `processDelete` method is called after deletion.

Signature

```
public static Database.DeleteResult deleteAsync(SObject sobject,
DataSource.AsyncDeleteCallback callback)
```

Parameters

sobject
Type: SObject

The external object record to delete.

callback
Type: DataSource.AsyncDeleteCallback

The callback that contains the state in the originating context and an action (the `processDelete` method) that is executed after the insert operation is completed. Use the action callback to update org data according to the operation's results. The callback object must extend `DataSource.AsyncDeleteCallback`.

Return Value

Type: Database.DeleteResult

Status result for the delete operation. The result corresponds to the record processed by this asynchronous operation and is associated with a unique identifier (`asyncLocator`). The `asyncLocator` value is included in the errors array of the result. You can retrieve this identifier with `Database.getAsyncLocator()`. Retrieve the final result with `Database.getAsyncDeleteResult()`.

`deleteAsync(sobjects)`

Initiates requests to delete the external data that corresponds to the specified external object records. The requests are executed asynchronously, as background operations, and are sent to the external systems that are defined by the external objects' associated external data sources.

Signature

```
public static List<Database.DeleteResult> deleteAsync(List<SObject> sobjects)
```

Parameters

sobjects
Type: List<SObject>

List of external object records to delete.

Return Value

Type: List<Database.DeleteResult>

Status results for the delete operation. Each result corresponds to a record processed by this asynchronous operation and is associated with a unique identifier (`asyncLocator`). The `asyncLocator` value is included in the errors array of the result. You can retrieve this identifier with `Database.getAsyncLocator()`. Retrieve the final result with `Database.getAsyncDeleteResult()`.

`deleteAsync(sobject)`

Initiates a request to delete the external data that corresponds to the specified external object record. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source.

Signature

```
public static Database.DeleteResult deleteAsync(SObject sobject)
```

Parameters

sobject
Type: SObject

The external object record to delete.

Return Value

Type: Database.DeleteResult

Status result for the delete operation. The result corresponds to the record processed by this asynchronous operation and is associated with a unique identifier (`asyncLocator`). The `asyncLocator` value is included in the errors array of the result. You can retrieve this identifier with `Database.getAsyncLocator()`. Retrieve the final result with `Database.getAsyncDeleteResult()`.

deleteAsync(sobjects, callback, accessLevel)

Initiates requests to delete the external data that corresponds to the specified external object records. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source. Allows referencing a callback class whose `processDelete` method is called for each record after deletion.

Signature

```
public static List<Database.DeleteResult> deleteAsync(List<SObject> sobjects,
DataSource.AsyncDeleteCallback callback, System.AccessLevel accessLevel)
```

Parameters

sobjects
Type: List<SObject>

List of external object records to delete.

callback
Type: DataSource.AsyncDeleteCallback

The callback that contains the state in the originating context and an action (the `processDelete` method) that is executed after the insert operation is completed. The execution is in system mode regardless of the `accessLevel` parameter. Use the action callback to update org data according to the operation's results. The callback object must extend `DataSource.AsyncDeleteCallback`.

accessLevel
Type: System.AccessLevel

(Optional) The *`accessLevel`* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: List<Database.DeleteResult>

Status results for the delete operation. Each result corresponds to a record processed by this asynchronous operation and is associated with a unique identifier (`asyncLocator`). The `asyncLocator` value is included in the errors array of the result. You can retrieve this identifier with `Database.getAsyncLocator()`. Retrieve the final result with `Database.getAsyncDeleteResult()`.

deleteAsync(sobject, callback, accessLevel)

Initiates a request to delete the external data that corresponds to the specified external object record. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source. Allows referencing a callback class whose `processDelete` method is called after deletion.

Signature

```
public static Database.DeleteResult deleteAsync(SObject sobject,
DataSource.AsyncDeleteCallback callback, System.AccessLevel accessLevel)
```

Parameters

sobject
Type: SObject

The external object record to delete.

callback
Type: DataSource.AsyncDeleteCallback

The callback that contains the state in the originating context and an action (the `processDelete` method) that is executed after the insert operation is completed. The execution is in system mode regardless of the `accessLevel` parameter. Use the action callback to update org data according to the operation's results. The callback object must extend `DataSource.AsyncDeleteCallback`.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.DeleteResult

Status result for the delete operation. The result corresponds to the record processed by this asynchronous operation and is associated with a unique identifier (`asyncLocator`). The `asyncLocator` value is included in the errors array of the result. You can retrieve this identifier with `Database.getAsyncLocator()`. Retrieve the final result with `Database.getAsyncDeleteResult()`.

deleteAsync(sobjects, accessLevel)

Initiates requests to delete the external data that corresponds to the specified external object records. The requests are executed asynchronously, as background operations, and are sent to the external systems that are defined by the external objects' associated external data sources.

Signature

```
public static List<Database.DeleteResult> deleteAsync(List<SObject> sobjects,
System.AccessLevel accessLevel)
```

Parameters

sobjects
Type: List<SObject>

List of external object records to delete.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: List<Database.DeleteResult>

Status results for the delete operation. Each result corresponds to a record processed by this asynchronous operation and is associated with a unique identifier (`asyncLocator`). The `asyncLocator` value is included in the errors array of the result. You can retrieve this identifier with `Database.getAsyncLocator()`. Retrieve the final result with `Database.getAsyncDeleteResult()`.

deleteAsync(sobject, accessLevel)

Initiates a request to delete the external data that corresponds to the specified external object record. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source.

Signature

```
public static Database.DeleteResult deleteAsync(SObject sobject, System.AccessLevel
accessLevel)
```

Parameters

sobject
Type: SObject

The external object record to delete.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.DeleteResult

Status result for the delete operation. The result corresponds to the record processed by this asynchronous operation and is associated with a unique identifier (`asyncLocator`). The `asyncLocator` value is included in the errors array of the result. You can retrieve this identifier with `Database.getAsyncLocator()`. Retrieve the final result with `Database.getAsyncDeleteResult()`.

deleteImmediate(sobjects)

Initiates requests to delete the external data that corresponds to the specified external object records. The requests are executed synchronously and are sent to the external systems that are defined by the external objects' associated external data sources. If the Apex transaction contains pending changes, the synchronous operations can't be completed and throw exceptions.

Signature

```
public static List<Database.DeleteResult> deleteImmediate(List<SObject> sobjects)
```

Parameters

sobjects
Type: List<SObject>

List of external object records to delete.

Return Value

Type: List<Database.DeleteResult>

Status results for the delete operation.

Usage

The batch limit for big objects using `deleteImmediate()` is 50,000 records at once.

deleteImmediate(sobject)

Initiates a request to delete the external data that corresponds to the specified external object record. The request is executed synchronously and is sent to the external system that's defined by the external object's associated external data source. If the Apex transaction contains pending changes, the synchronous operation can't be completed and throws an exception.

Signature

```
public static Database.DeleteResult deleteImmediate(SObject sobject)
```

Parameters

sobject
Type: SObject

The external object record to delete.

Return Value

Type: Database.DeleteResult

Status result for the delete operation.

deleteImmediate(sobjects, accessLevel)

Initiates requests to delete the external data that corresponds to the specified external object records. The requests are executed synchronously and are sent to the external systems that are defined by the external objects' associated external data sources. If the Apex transaction contains pending changes, the synchronous operations can't be completed and throw exceptions.

Signature

```
public static List<Database.DeleteResult> deleteImmediate(List<SObject> sobjects,
System.AccessLevel accessLevel)
```

Parameters

sobjects
Type: List<SObject>

List of external object records to delete.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: List<Database.DeleteResult>

Status results for the delete operation.

Usage

The batch limit for big objects using `deleteImmediate()` is 50,000 records at once.

deleteImmediate(sobject, accessLevel)

Initiates a request to delete the external data that corresponds to the specified external object record. The request is executed synchronously and is sent to the external system that's defined by the external object's associated external data source. If the Apex transaction contains pending changes, the synchronous operation can't be completed and throws an exception.

Signature

```
public static Database.DeleteResult deleteImmediate(SObject sobject, System.AccessLevel accessLevel)
```

Parameters

sobject
Type: SObject

The external object record to delete.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.DeleteResult

Status result for the delete operation.

emptyRecycleBin(recordIds)

Permanently deletes the specified records from the Recycle Bin.

Signature

```
public static Database.EmptyRecycleBinResult[] emptyRecycleBin(ID [] recordIds)
```

Parameters

recordIds
Type: ID[]

Return Value

Type: Database.EmptyRecycleBinResult[]

Usage

Note the following:

- After records are deleted using this method, they cannot be undeleted.
- Only 10,000 records can be specified for deletion.
- Logged in users can delete any record that they can query in their Recycle Bin, or the recycle bins of any subordinates. If logged in users have "Modify All Data" permission, they can query and delete records from any Recycle Bin in the organization.
- Cascade delete record IDs should not be included in the list of IDs; otherwise an error occurs. For example, if an account record is deleted, all related contacts, opportunities, contracts, and so on are also deleted. Only include the Id of the top-level account. All related records are automatically removed.
- Deleted items are added to the number of items processed by a DML statement, and the method call is added to the total number of DML statements issued. Each executed `emptyRecycleBin` method counts against the governor limit for DML statements.

`emptyRecycleBin(obj)`

Permanently deletes the specified sObject from the Recycle Bin.

Signature

```
public static Database.EmptyRecycleBinResult emptyRecycleBin(sObject obj)
```

Parameters

obj
Type: sObject

Return Value

Type: Database.EmptyRecycleBinResult

Usage

Note the following:

- After an sObject is deleted using this method, it cannot be undeleted.
- Only 10,000 sObjects can be specified for deletion.
- The logged-in user can delete any sObject (that can be queried) in their Recycle Bin, or the recycle bins of any subordinates. If the logged-in user has "Modify All Data" permission, they can query and delete sObjects from any Recycle Bin in the organization.
- Do not include an sObject that was deleted due to a cascade delete; otherwise an error occurs. For example, if an account is deleted, all related contacts, opportunities, contracts, and so on are also deleted. Only include sObjects of the top-level account. All related sObjects are automatically removed.

`emptyRecycleBin(listOfSObjects)`

Permanently deletes the specified sObjects from the Recycle Bin.

Signature

```
public static Database.EmptyRecycleBinResult[] emptyRecycleBin(sObject[] listOfSObjects)
```

Parameters

listOfSObjects
Type: sObject[]

Return Value

Type: Database.EmptyRecycleBinResult[]

Usage

Note the following:

- After an sObject is deleted using this method, it cannot be undeleted.
- Only 10,000 sObjects can be specified for deletion.
- The logged-in user can delete any sObject (that can be queried) in their Recycle Bin, or the recycle bins of any subordinates. If the logged-in user has "Modify All Data" permission, they can query and delete sObjects from any Recycle Bin in the organization.
- Do not include an sObject that was deleted due to a cascade delete; otherwise an error occurs. For example, if an account is deleted, all related contacts, opportunities, contracts, and so on are also deleted. Only include sObjects of the top-level account. All related sObjects are automatically removed.

executeBatch(batchClassObject)

Submits a batch Apex job for execution corresponding to the specified class.

Signature

```
public static ID executeBatch(Object batchClassObject)
```

Parameters

batchClassObject
Type: Object

An instance of a class that implements the Database.Batchable interface.

Return Value

Type: ID

The ID of the new batch job (AsyncApexJob).

Usage

When calling this method, Salesforce chunks the records returned by the `start` method of the batch class into batches of 200, and then passes each batch to the `execute` method. Apex governor limits are reset for each execution of `execute`.

For more information, see Using Batch Apex.

Versioned Behavior Changes

If the `executeBatch` call fails to acquire an Apex flex queue lock:

- In API version 52.0 and later, the call throws a `System.AsyncException`.
- In API version 51.0 and earlier, the call returns an empty ID, "000000000000000", instead of throwing an exception.

executeBatch(batchClassObject, scope)

Submits a batch Apex job for execution using the specified class and scope.

Signature

```
public static ID executeBatch(Object batchClassObject, Integer scope)
```

Parameters

batchClassObject
Type: Object

An instance of a class that implements the Database.Batchable interface.

scope
Type: Integer

Number of records to be passed into the `execute` method for batch processing.

Return Value

Type: ID

The ID of the new batch job (AsyncApexJob).

Usage

The value for *scope* must be greater than 0.

If the `start` method of the batch class returns a `Database.QueryLocator`, the scope parameter of `Database.executeBatch` can have a maximum value of 2,000. If set to a higher value, Salesforce chunks the records returned by the `QueryLocator` into smaller batches of up to 200 records. If the `start` method of the batch class returns an iterable, the scope parameter value has no upper limit; however, if you use a very high number, you could run into other limits.

Apex governor limits are reset for each execution of `execute`.

For more information, see Using Batch Apex.

Versioned Behavior Changes

If the `executeBatch` call fails to acquire an Apex flex queue lock:

- In API version 52.0 and later, the call throws a `System.AsyncException`.
- In API version 51.0 and earlier, the call returns an empty ID, "000000000000000", instead of throwing an exception.

getAsyncDeleteResult(deleteResult)

Retrieves the status of an asynchronous delete operation that's identified by a `Database.DeleteResult` object.

Signature

```
public static Database.DeleteResult getAsyncDeleteResult(Database.DeleteResult
deleteResult)
```

Parameters

deleteResult
Type: Database.DeleteResult

The result record for the delete operation being retrieved.

Return Value

Type: Database.DeleteResult

The result of a completed asynchronous delete of a record or records.

getAsyncDeleteResult(asyncLocator)

Retrieves the result of an asynchronous delete operation based on the result's unique identifier.

Signature

```
public static Database.DeleteResult getAsyncDeleteResult(String asyncLocator)
```

Parameters

asyncLocator
Type: String

The unique identifier associated with the result of an asynchronous operation.

Return Value

Type: Database.DeleteResult

The result of a completed asynchronous delete of a record or records.

getAsyncLocator(result)

Returns the `asyncLocator` associated with the result of a specified asynchronous insert, update, or delete operation.

Signature

```
public static String getAsyncLocator(Object result)
```

Parameters

result
Type: Object

The saved result of an asynchronous insert, update, or delete operation. The result object can be of type `Database.SaveResult` or `Database.DeleteResult`.

Return Value

Type: String

The unique identifier associated with the result of the specified operation.

getAsyncSaveResult(saveResult)

Returns the status of an asynchronous insert or update operation that's identified by a `Database.SaveResult` object.

Signature

```
public static Database.SaveResult getAsyncSaveResult(Database.SaveResult saveResult)
```

Parameters

saveResult
Type: Database.SaveResult

The result record for the insert or update operation being retrieved.

Return Value

Database.SaveResult

The result of a completed asynchronous operation on a record or records.

getAsyncSaveResult(asyncLocator)

Returns the status of an asynchronous insert or update operation based on the unique identifier associated with each modification.

Signature

```
public static Database.SaveResult getAsyncSaveResult(String asyncLocator)
```

Parameters

asyncLocator
Type: String

The unique identifier associated with the result of an asynchronous operation.

Return Value

Database.SaveResult

The result of a completed asynchronous operation on a record or records.

getCursor(query)

Creates a cursor when the specified SOQL query is executed.

Signature

```
public static Database.Cursor getCursor(String query)
```

Parameters

query
Type: String

The SOQL query to be run.

Return Value

Type: Database.Cursor

getCursor(query, accessLevel)

Creates a cursor when the specified SOQL query is executed.

Signature

```
public static Database.Cursor getCursor(String query, Object accessLevel)
```

Parameters

query
Type: String

The SOQL query to be run.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.Cursor

getCursorWithBinds(query, bindMap, accessLevel)

Creates a cursor when the specified SOQL query is executed.

Signature

```
public static Database.Cursor getCursorWithBinds(String query, Map bindMap, Object
accessLevel)
```

Parameters

query
Type: String

The SOQL query to be run.

bindMap
Type: Map

A map that contains placeholder keys for each bind variable specified in the SOQL query string and its value.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.Cursor

getDeleted(sObjectType, startDate, endDate)

Returns the list of individual records that have been deleted for an sObject type within the specified start and end dates and times and that are still in the Recycle Bin.

Signature

```
public static Database.GetDeletedResult getDeleted(String sObjectType, Datetime
startDate, Datetime endDate)
```

Parameters

sObjectType
Type: String

The *sObjectType* argument is the sObject type name for which to get the deleted records, such as account or merchandise__c.

startDate
Type: Datetime

Start date and time of the deleted records time window.

endDate
Type: Datetime

End date and time of the deleted records time window.

Return Value

Type: Database.GetDeletedResult

Usage

Because the Recycle Bin holds records up to 15 days, results are returned for no more than 15 days previous to the day the call is executed (or earlier if an administrator has purged the Recycle Bin).

Example

```
Database.GetDeletedResult r =
 Database.getDeleted(
  'Merchandise__c',
  Datetime.now().addHours(-1),
  Datetime.now());
```

getQueryLocator(staticSoqlQueryResult)

Creates a QueryLocator object used in batch Apex or Visualforce.

Signature

```
public static Database. QueryLocator getQueryLocator(sObject [] staticSoqlQueryResult)
```

Parameters

staticSoqlQueryResult
Type: sObject []

The *staticSoqlQueryResult* parameter must be a static, inline SOQL query.

Return Value

Type: Database.QueryLocator

Usage

You can't use `getQueryLocator` with any query that contains an aggregate function.

Each executed `getQueryLocator` method counts against the governor limit of 10,000 total records retrieved and the total number of SOQL queries issued.

For more information, see Understanding Apex Managed Sharing, and IdeaStandardSetController Class.

getQueryLocator(query)

Creates a QueryLocator object used in batch Apex or Visualforce.

Signature

```
public static Database.QueryLocator getQueryLocator(String query)
```

Parameters

query
Type: String

Return Value

Type: Database.QueryLocator

Usage

You can't use `getQueryLocator` with any query that contains an aggregate function.

Each executed `getQueryLocator` method counts against the governor limit of 10,000 total records retrieved and the total number of SOQL queries issued.

For more information, see Understanding Apex Managed Sharing, and StandardSetController Class.

getQueryLocator(staticSoqlQueryResult, accessLevel)

Creates a QueryLocator object used in batch Apex or Visualforce.

Signature

```
public static Database.QueryLocator getQueryLocator(sObject [] staticSoqlQueryResult,
System.AccessLevel accessLevel)
```

Parameters

`staticSoqlQueryResult`
Type: sObject []

The *`staticSoqlQueryResult`* parameter must be a static, inline SOQL query.

`accessLevel`
Type: System.AccessLevel

(Optional) The *`accessLevel`* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.QueryLocator

Usage

The access level is evaluated only when the `QueryLocator` is created. A `QueryLocator` can be long lived, such as when used in a batch. We don't reevaluate the object and field-level security with each iteration of the `QueryLocator`. As a result, if you specify user mode, and then change the security settings after the `QueryLocator` is created, the new settings aren't enforced.

You can't use `getQueryLocator` with any query that contains an aggregate function.

Each executed `getQueryLocator` method counts against the governor limit of 10,000 total records retrieved and the total number of SOQL queries issued.

For more information, see Understanding Apex Managed Sharing, and IdeaStandardSetController Class.

getQueryLocator(query, accessLevel)

Creates a QueryLocator object used in batch Apex or Visualforce.

Signature

```
public static Database.QueryLocator getQueryLocator(String query, System.AccessLevel
accessLevel)
```

Parameters

query
Type: String

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.QueryLocator

Usage

The access level is evaluated only when the `QueryLocator` is created. A `QueryLocator` can be long lived, such as when used in a batch. We don't reevaluate the object and field-level security with each iteration of the `QueryLocator`. As a result, if you specify user mode, and then change the security settings after the `QueryLocator` is created, the new settings aren't enforced.

You can't use `getQueryLocator` with any query that contains an aggregate function.

Each executed `getQueryLocator` method counts against the governor limit of 10,000 total records retrieved and the total number of SOQL queries issued.

For more information, see Understanding Apex Managed Sharing, and StandardSetController Class.

getQueryLocatorWithBinds(query, bindMap, accessLevel)

Creates a QueryLocator object used in batch Apex or Visualforce. Bind variables in the query are resolved from the *bindMap* Map parameter directly with the key, rather than from Apex code variables.

Signature

```
public static Database.QueryLocator getQueryLocatorWithBinds(String query, Map<String,
Object> bindMap, System.AccessLevel accessLevel)
```

Parameters

query
Type: String

SOQL query that includes Apex bind variables preceded by a colon. All bind variables must have a key in the *bindMap* Map.

bindMap
Type: Map<String, Object>

A map that contains keys for each bind variable specified in the SOQL *queryString* and its value. The keys can't be null or duplicates, and the values can't be null or empty strings.

accessLevel
Type: System.AccessLevel

The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced.

Return Value

Type: Database.QueryLocator

Usage

The access level is evaluated only when the `QueryLocator` is created. A `QueryLocator` can be long lived, such as when used in a batch. We don't reevaluate the object and field-level security with each iteration of the `QueryLocator`. As a result, if you specify user mode, and then change the security settings after the `QueryLocator` is created, the new settings aren't enforced.

You can't use `getQueryLocatorWithBinds` with any query that contains an aggregate function.

Each executed `getQueryLocatorWithBinds` method counts against the governor limit for the total number of records retrieved by Database.getQueryLocator(10,000) and the total number of SOQL queries issued. See Per Transaction Apex Limits.

For more information, see Understanding Apex Managed Sharing, and StandardSetController Class.

Example

In this example, the SOQL query uses a bind variable for an Account name. Its value (`Acme Corporation`) is passed in using the *acctBinds* Map.

```
public class TestBatch implements Database.Batchable<sObject>{

   private Map<String, Object> acctBinds = new Map<String, Object>{'acctName' => 'Acme
Corporation'};

   private String query = 'Select Id From Account where name = :acctName';

   public Database.QueryLocator start(Database.BatchableContext BC){
      return Database.getQueryLocatorWithBinds(query, acctBinds, AccessLevel.USER_MODE);
   }

   public void execute(Database.BatchableContext BC, List<sObject> scope){
   }

   public void finish(Database.BatchableContext BC){
   }
}
```

getUpdated(sobjectType, startDate, endDate)

Returns the list of individual records that have been updated for an sObject type within the specified start and end dates and times.

Signature

```
public static Database.GetUpdatedResult getUpdated(String sobjectType, Datetime startDate, Datetime endDate)
```

Parameters

sobjectType
Type: String

The *sObjectType* argument is the sObject type name for which to get the updated records, such as account or merchandise__c.

startDate
Type: Datetime

The *startDate* argument is the start date and time of the updated records time window.

endDate
Type: Datetime

The *endDate* argument is the end date and time of the updated records time window.

Return Value

Type: Database.GetUpdatedResult

Usage

The date range for the returned results is no more than 30 days previous to the day the call is executed.

Example

```
Database.GetUpdatedResult r =
 Database.getUpdated(
  'Merchandise__c',
  Datetime.now().addHours(-1),
  Datetime.now());
```

insert(recordToInsert, allOrNone)

Adds an sObject, such as an individual account or contact, to your organization's data.

Signature

```
public static Database.SaveResult insert(sObject recordToInsert, Boolean allOrNone)
```

Parameters

recordToInsert
Type: sObject

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

Return Value

Type: Database.SaveResult

Usage

`insert` is analogous to the INSERT statement in SQL.

Apex classes and triggers saved (compiled) using API version 15.0 and higher produce a runtime error if you assign a String value that is too long for the field.

Each executed `insert` method counts against the governor limit for DML statements.

`insert(recordsToInsert, allOrNone)`

Adds one or more sObjects, such as individual accounts or contacts, to your organization's data.

Signature

```
public static Database.SaveResult[] insert(sObject[] recordsToInsert, Boolean allOrNone)
```

Parameters

recordsToInsert
Type: sObject []

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

If *allOrNone* is set to `false` and a before-trigger assigns an invalid value to a field, the partial set of valid records isn't inserted.

Return Value

Type: Database.SaveResult[]

Usage

`insert` is analogous to the INSERT statement in SQL.

Apex classes and triggers saved (compiled) using API version 15.0 and higher produce a runtime error if you assign a String value that is too long for the field.

Each executed `insert` method counts against the governor limit for DML statements.

Example

Example:

The following example inserts two accounts:

```
Account a = new Account(name = 'Acme1');
Database.SaveResult[] lsr = Database.insert(
    new Account[]{a, new Account(Name = 'Acme2')},
    false);
```

insert(recordToInsert, dmlOptions)

Adds an sObject, such as an individual account or contact, to your organization's data.

Signature

```
public static Database.SaveResult insert(sObject recordToInsert, Database.DMLOptions
dmlOptions)
```

Parameters

recordToInsert
Type: sObject

dmlOptions
Type: Database.DMLOptions

The optional *dmlOptions* parameter specifies additional data for the transaction, such as assignment rule information or rollback behavior when errors occur during record insertions.

Return Value

Type: Database.SaveResult

Usage

`insert` is analogous to the INSERT statement in SQL.

Apex classes and triggers saved (compiled) using API version 15.0 and higher produce a runtime error if you assign a String value that is too long for the field.

Each executed `insert` method counts against the governor limit for DML statements.

insert(recordsToInsert, dmlOptions)

Adds one or more sObjects, such as individual accounts or contacts, to your organization's data.

Signature

```
public static Database.SaveResult insert(sObject[] recordsToInsert, Database.DMLOptions
dmlOptions)
```

Parameters

recordsToInsert
Type: sObject[]

dmlOptions
Type: Database.DMLOptions

The optional *dmlOptions* parameter specifies additional data for the transaction, such as assignment rule information or rollback behavior when errors occur during record insertions.

Return Value

Type: Database.SaveResult[]

Usage

`insert` is analogous to the INSERT statement in SQL.

Apex classes and triggers saved (compiled) using API version 15.0 and higher produce a runtime error if you assign a String value that is too long for the field.

Each executed `insert` method counts against the governor limit for DML statements.

insert(recordToInsert, allOrNone, accessLevel)

Adds an sObject, such as an individual account or contact, to your organization's data.

Signature

```
public static Database.SaveResult insert(SObject recordToInsert, Boolean allOrNone,
System.AccessLevel accessLevel)
```

Parameters

recordToInsert
Type: sObject

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.SaveResult

Usage

If you use the `accessLevel` parameter to specify that the method runs in user mode, we report all encountered inaccessible fields. The way to retrieve the names of these inaccessible fields depends on the value of this method's `allOrNone` parameter, or the equivalent `DmlOptions.optAllOrNone` property. If you specify that:

- `allOrNone=true` or `DmlOptions.optAllOrNone=true`: Catch the `DMLException` and use the `DMLException.getDMLFieldNames()` method to retrieve the list of inaccessible fields. See Exception Class and Built-In Exceptions for more information.
- `allOrNone=false` or `DmlOptions.optAllOrNone=false`: For each failing record, we update the `Database.Error` object that results from the DML operation. Use the `Error.getFields()` method to retrieve the list of inaccessible fields. See the Error Class methods for more information.

`insert` is analogous to the INSERT statement in SQL.

Apex classes and triggers saved (compiled) using API version 15.0 and higher produce a runtime error if you assign a String value that is too long for the field.

Each executed `insert` method counts against the governor limit for DML statements.

`insert(recordsToInsert, allOrNone, accessLevel)`

Adds one or more sObjects, such as individual accounts or contacts, to your organization's data.

Signature

```
public static List<Database.SaveResult> insert(List<SObject> recordsToInsert, Boolean
allOrNone, System.AccessLevel accessLevel)
```

Parameters

`recordsToInsert`
Type: List<sObject>

`allOrNone`
Type: Boolean

(Optional) The *`allOrNone`* parameter specifies whether the operation allows partial success. If *`allOrNone`* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *`allOrNone`* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

If *`allOrNone`* is set to `false` and a before-trigger assigns an invalid value to a field, the partial set of valid records isn't inserted.

`accessLevel`
Type: System.AccessLevel

(Optional) The *`accessLevel`* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: List<Database.SaveResult>

Usage

If you use the `accessLevel` parameter to specify that the method runs in user mode, we report all encountered inaccessible fields. The way to retrieve the names of these inaccessible fields depends on the value of this method's `allOrNone` parameter, or the equivalent `DmlOptions.optAllOrNone` property. If you specify that:

- `allOrNone=true` or `DmlOptions.optAllOrNone=true`: Catch the `DMLException` and use the `DMLException.getDMLFieldNames()` method to retrieve the list of inaccessible fields. See Exception Class and Built-In Exceptions for more information.
- `allOrNone=false` or `DmlOptions.optAllOrNone=false`: For each failing record, we update the `Database.Error` object that results from the DML operation. Use the `Error.getFields()` method to retrieve the list of inaccessible fields. See the Error Class methods for more information.

`insert` is analogous to the INSERT statement in SQL.

Apex classes and triggers saved (compiled) using API version 15.0 and higher produce a runtime error if you assign a String value that is too long for the field.

Each executed `insert` method counts against the governor limit for DML statements.

`insert(recordToInsert, dmlOptions, accessLevel)`

Adds an sObject, such as an individual account or contact, to your organization's data.

Signature

```
public static Database.SaveResult insert(SObject recordToInsert, Database.DMLOptions
dmlOptions, System.AccessLevel accessLevel)
```

Parameters

`recordToInsert`
Type: sObject

`dmlOptions`
Type: Database.DMLOptions

The optional *`dmlOptions`* parameter specifies additional data for the transaction, such as assignment rule information or rollback behavior when errors occur during record insertions.

`accessLevel`
Type: System.AccessLevel

(Optional) The *`accessLevel`* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.SaveResult

Usage

`insert` is analogous to the INSERT statement in SQL.

Apex classes and triggers saved (compiled) using API version 15.0 and higher produce a runtime error if you assign a String value that is too long for the field.

Each executed `insert` method counts against the governor limit for DML statements.

insert(recordsToInsert, dmlOptions, accessLevel)

Adds one or more sObjects, such as individual accounts or contacts, to your organization's data.

Signature

```
public static List<Database.SaveResult> insert(List<SObject> recordsToInsert,
Database.DMLOptions dmlOptions, System.AccessLevel accessLevel)
```

Parameters

recordsToInsert
Type: List<sObject>

dmlOptions
Type: Database.DMLOptions

The optional *dmlOptions* parameter specifies additional data for the transaction, such as assignment rule information or rollback behavior when errors occur during record insertions.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: List<Database.SaveResult>

Usage

`insert` is analogous to the INSERT statement in SQL.

Apex classes and triggers saved (compiled) using API version 15.0 and higher produce a runtime error if you assign a String value that is too long for the field.

Each executed `insert` method counts against the governor limit for DML statements.

insertAsync(sobjects, callback)

Initiates requests to add external object data to the relevant external systems. The requests are executed asynchronously, as background operations, and are sent to the external systems that are defined by the external objects' associated external data sources. Allows referencing a callback class whose `processSave` method is called for each record after the remote operations are completed.

Signature

```
public static List<Database.SaveResult> insertAsync(List<SObject> sobjects,
DataSource.AsyncSaveCallback callback)
```

Parameters

sobjects
Type: List<SObject>

List of external object records to insert.

callback
Type: DataSource.AsyncSaveCallback

The callback object that contains the state in the originating context and an action (the `processSave` method) that executes after the insert operation is completed. Use the action callback to update org data according to the operation's results. The callback object must extend `DataSource.AsyncSaveCallback`.

Return Value

Type: List<Database.SaveResult>

Status results for the insert operation. Each result corresponds to a record processed by this asynchronous operation and is associated with a unique identifier (`asyncLocator`). The `asyncLocator` value is included in the errors array of the result. You can retrieve this identifier with `Database.getAsyncLocator()`. Retrieve the final result with `Database.getAsyncSaveResult()`.

Usage

`Database.insertAsync()` methods can't be executed in the context of a portal user, even when the portal user is a community member. To add external object records via Apex, use `Database.insertImmediate()` methods.

insertAsync(sobject, callback)

Initiates a request to add external object data to the relevant external system. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source. Allows referencing a callback class whose `processSave` method is called after the remote operation is completed.

Signature

```
public static Database.SaveResult insertAsync(SObject sobject,
DataSource.AsyncSaveCallback callback)
```

Parameters

sobject
Type: SObject

The external object record to insert.

callback
Type: DataSource.AsyncSaveCallback

The callback object that contains the state in the originating context and an action (the `processSave` method) that executes after the insert operation is completed. Use the action callback to update org data according to the operation's results. The callback object must extend `DataSource.AsyncSaveCallback`.

Return Value

Type: Database.SaveResult

Status result for the insert operation. The result corresponds to the record processed by this asynchronous operation and is associated with a unique identifier (`asyncLocator`). The `asyncLocator` value is included in the errors array of the result. You can retrieve this identifier with `Database.getAsyncLocator()`. Retrieve the final result with `Database.getAsyncSaveResult()`.

Usage

`Database.insertAsync()` methods can't be executed in the context of a portal user, even when the portal user is a community member. To add external object records via Apex, use `Database.insertImmediate()` methods.

`insertAsync(sobjects)`

Initiates requests to add external object data to the relevant external systems. The requests are executed asynchronously, as background operations, and are sent to the external systems that are defined by the external objects' associated external data sources.

Signature

```
public static List<Database.SaveResult> insertAsync(List<SObject> sobjects)
```

Parameters

sobjects
Type: List<SObject>

List of external object records to insert.

Return Value

Type: List<Database.SaveResult>

Status results for the insert operation. Each result corresponds to a record processed by this asynchronous operation and is associated with a unique identifier (`asyncLocator`). The `asyncLocator` value is included in the errors array of the result. You can retrieve this identifier with `Database.getAsyncLocator()`. Retrieve the final result with `Database.getAsyncSaveResult()`.

Usage

`Database.insertAsync()` methods can't be executed in the context of a portal user, even when the portal user is a community member. To add external object records via Apex, use `Database.insertImmediate()` methods.

`insertAsync(sobject)`

Initiates a request to add external object data to the relevant external system. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source.

Signature

```
public static Database.SaveResult insertAsync(SObject sobject)
```

Parameters

sobject
Type: SObject

The external object record to insert.

Return Value

Type: Database.SaveResult

Status result for the insert operation. The result corresponds to the record processed by this asynchronous operation and is associated with a unique identifier (`asyncLocator`). The `asyncLocator` value is included in the errors array of the result. You can retrieve this identifier with `Database.getAsyncLocator()`. Retrieve the final result with `Database.getAsyncSaveResult()`.

Usage

`Database.insertAsync()` methods can't be executed in the context of a portal user, even when the portal user is a community member. To add external object records via Apex, use `Database.insertImmediate()` methods.

insertAsync(sobjects, callback, accessLevel)

Initiates requests to add external object data to the relevant external systems. The requests are executed asynchronously, as background operations, and are sent to the external systems that are defined by the external objects' associated external data sources. Allows referencing a callback class whose `processSave` method is called for each record after the remote operations are completed.

Signature>

```
public static List<Database.SaveResult> insertAsync(List<SObject> sobjects,
DataSource.AsyncSaveCallback callback, System.AccessLevel accessLevel)
```

Parameters>

sobjects
Type: List<SObject>

List of external object records to insert.

callback
Type: DataSource.AsyncSaveCallback

The callback object that contains the state in the originating context and an action (the `processSave` method) that executes after the insert operation is completed. The execution is in system mode regardless of the `accessLevel` parameter. Use the action callback to update org data according to the operation's results. The callback object must extend `DataSource.AsyncSaveCallback`.

accessLevel
Type: System.AccessLevel

(Optional) The *`accessLevel`* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are

ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value>

Type: List<Database.SaveResult>

Status results for the insert operation. Each result corresponds to a record processed by this asynchronous operation and is associated with a unique identifier (`asyncLocator`). The `asyncLocator` value is included in the errors array of the result. You can retrieve this identifier with `Database.getAsyncLocator()`. Retrieve the final result with `Database.getAsyncSaveResult()`.

Usage>

`Database.insertAsync()` methods can't be executed in the context of a portal user, even when the portal user is a community member. To add external object records via Apex, use `Database.insertImmediate()` methods.

insertAsync(sobject, callback, accessLevel)

Initiates a request to add external object data to the relevant external system. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source. Allows referencing a callback class whose `processSave` method is called after the remote operation is completed.

Signature

```
public static Database.SaveResult insertAsync(SObject sobject,
DataSource.AsyncSaveCallback callback, System.AccessLevel accessLevel)
```

Parameters

`sobject`
Type: SObject

The external object record to insert.

`callback`
Type: DataSource.AsyncSaveCallback

The callback object that contains the state in the originating context and an action (the `processSave` method) that executes after the insert operation is completed. The execution is in system mode regardless of the `accessLevel` parameter. Use the action callback to update org data according to the operation's results. The callback object must extend `DataSource.AsyncSaveCallback`.

`accessLevel`
Type: System.AccessLevel

(Optional) The *`accessLevel`* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.SaveResult

Status result for the insert operation. The result corresponds to the record processed by this asynchronous operation and is associated with a unique identifier (`asyncLocator`). The `asyncLocator` value is included in the errors array of the result. You can retrieve this identifier with `Database.getAsyncLocator()`. Retrieve the final result with `Database.getAsyncSaveResult()`.

Usage

`Database.insertAsync()` methods can't be executed in the context of a portal user, even when the portal user is a community member. To add external object records via Apex, use `Database.insertImmediate()` methods.

insertAsync(sobjects, accessLevel)

Initiates requests to add external object data to the relevant external systems. The requests are executed asynchronously, as background operations, and are sent to the external systems that are defined by the external objects' associated external data sources.

Signature

```
public static List<Database.SaveResult> insertAsync(List<SObject> sobjects,
System.AccessLevel accessLevel)
```

Parameters

`sobjects`
Type: List<SObject>

List of external object records to insert.

`accessLevel`
Type: System.AccessLevel

(Optional) The *`accessLevel`* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: List<Database.SaveResult>

Status results for the insert operation. Each result corresponds to a record processed by this asynchronous operation and is associated with a unique identifier (`asyncLocator`). The `asyncLocator` value is included in the errors array of the result. You can retrieve this identifier with `Database.getAsyncLocator()`. Retrieve the final result with `Database.getAsyncSaveResult()`.

Usage

`Database.insertAsync()` methods can't be executed in the context of a portal user, even when the portal user is a community member. To add external object records via Apex, use `Database.insertImmediate()` methods.

insertAsync(sobject, accessLevel)

Initiates a request to add external object data to the relevant external system. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source.

Signature

```
public static Database.SaveResult insertAsync(SObject sobject, System.AccessLevel
accessLevel)
```

Parameters

sobject
Type: SObject

The external object record to insert.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.SaveResult

Status result for the insert operation. The result corresponds to the record processed by this asynchronous operation and is associated with a unique identifier (`asyncLocator`). The `asyncLocator` value is included in the errors array of the result. You can retrieve this identifier with `Database.getAsyncLocator()`. Retrieve the final result with `Database.getAsyncSaveResult()`.

Usage

`Database.insertAsync()` methods can't be executed in the context of a portal user, even when the portal user is a community member. To add external object records via Apex, use `Database.insertImmediate()` methods.

insertImmediate(sobjects)

Initiates requests to add external object data to the relevant external systems. The requests are executed synchronously and are sent to the external systems that are defined by the external objects' associated external data sources. If the Apex transaction contains pending changes, the synchronous operations can't be completed and throw exceptions.

Signature

```
public static List<Database.SaveResult> insertImmediate(List<SObject> sobjects)
```

Parameters

sobjects
Type: List<SObject>

List of external object records to insert.

Return Value

Type: List<Database.SaveResult>

Status results for the insert operation.

Usage

The operation allows partial success. If one or more record inserts fail, the method doesn't throw an exception and the remainder of the DML operation can still succeed. The returned `SaveResult` objects indicate whether the operation was successful. If it wasn't successful, the objects also return the error code and description.

`insertImmediate(sobject)`

Initiates a request to add external object data to the relevant external system. The request is executed synchronously and is sent to the external system that's defined by the external object's associated external data source. If the Apex transaction contains pending changes, the synchronous operation can't be completed and throws an exception.

Signature

```
public static Database.SaveResult insertImmediate(SObject sobject)
```

Parameters

sobject
Type: SObject

The external object record to insert.

Return Value

Type: Database.SaveResult

Status result for the insert operation.

Usage

If a record insert fails, the method doesn't throw an exception. The returned `SaveResult` object indicates whether the operation was successful. If it wasn't successful, the object returns the error code and description.

`insertImmediate(sobjects, accessLevel)`

Initiates requests to add external object data to the relevant external systems. The requests are executed synchronously and are sent to the external systems that are defined by the external objects' associated external data sources. If the Apex transaction contains pending changes, the synchronous operations can't be completed and throw exceptions.

Signature

```
public static List<Database.SaveResult> insertImmediate(List<SObject> sobjects,
System.AccessLevel accessLevel)
```

Parameters

sobjects
Type: List<SObject>

List of external object records to insert.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: List<Database.SaveResult>

Status results for the insert operation.

Usage

The operation allows partial success. If one or more record inserts fail, the method doesn't throw an exception and the remainder of the DML operation can still succeed. The returned `SaveResult` objects indicate whether the operation was successful. If it wasn't successful, the objects also return the error code and description.

`insertImmediate(sobject, accessLevel)`

Initiates a request to add external object data to the relevant external system. The request is executed synchronously and is sent to the external system that's defined by the external object's associated external data source. If the Apex transaction contains pending changes, the synchronous operation can't be completed and throws an exception.

Signature

```
public static Database.SaveResult insertImmediate(SObject sobject, System.AccessLevel
accessLevel)
```

Parameters

sobject
Type: SObject

The external object record to insert.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.SaveResult

Status result for the insert operation.

Usage

If a record update fails, the method doesn't throw an exception. The returned `SaveResult` object indicates whether the operation was successful. If it failed, the object returns the error code and description.

`merge(mergeToRecord, duplicateId)`

Merges the duplicate record into the `mergeToRecord` sObject record of the same type, deleting the duplicate, and reparenting any related records. Merges only accounts, contacts, or leads.

Signature

```
public static Database.MergeResult merge(sObject mergeToRecord, Id duplicateId)
```

Parameters

`mergeToRecord`
Type: sObject

The sObject record that the duplicate record is merged into.

`duplicateId`
Type: ID

The ID of the record to merge with the mergeToRecord. This record must be of the same sObject type as the mergeToRecord.

Return Value

Type: Database.MergeResult

Usage

Each executed `merge` method counts against the governor limit for DML statements.

`merge(mergeToRecord, duplicateRecord)`

Merges the duplicate sObject record into the `mergeToRecord` sObject record of the same type, deleting the duplicate, and reparenting any related records.

Signature

```
public static Database.MergeResult merge(sObject mergeToRecord, sObject duplicateRecord)
```

Parameters

`mergeToRecord`
Type: sObject

The sObject record that the duplicate record is merged into.

`duplicateRecord`
Type: sObject

The sObject record to merge with the mergeToRecord. This sObject must be of the same type as the mergeToRecord.

Return Value

Type: Database.MergeResult

Usage

Each executed `merge` method counts against the governor limit for DML statements.

`merge(mergeToRecord, duplicateIds)`

Merges up to two records of the same sObject type into the `mergeToRecord` sObject record, deleting the others, and reparenting any related records.

Signature

```
public static List<Database.MergeResult> merge(sObject mergeToRecord, List<Id>
duplicateIds)
```

Parameters

`mergeToRecord`
Type: SObject

The sObject record that the other records are merged into.

`duplicateIds`
Type: List<Id>

A list of IDs of up to two records to merge with the mergeToRecord. These records must be of the same sObject type as the mergeToRecord.

Return Value

Type: List<Database.MergeResult>

Usage

Each executed `merge` method counts against the governor limit for DML statements.

`merge(mergeToRecord, duplicateRecords)`

Merges up to two records of the same object type into the `mergeToRecord` sObject record, deleting the others, and reparenting any related records.

Signature

```
public static List<Database.MergeResult> merge(sObject mergeToRecord, List<SObject>
duplicateRecords)
```

Parameters

`mergeToRecord`
Type: SObject

The sObject record that the other sObjects are merged into.

`duplicateRecords`
Type: List<SObject>

A list of up to two sObject records to merge with the mergeToRecord. These sObjects must be of the same type as the mergeToRecord.

Return Value

Type: List<Database.MergeResult>

Usage

Each executed `merge` method counts against the governor limit for DML statements.

merge(mergeToRecord, duplicateId, allOrNone)

Merges the duplicate record into the `mergeToRecord` sObject record of the same type, optionally returning any errors, deleting the duplicate, and reparenting any related records. Merges only accounts, contacts, or leads.

Signature

```
public static Database.MergeResult merge(sObject mergeToRecord, Id duplicateId, Boolean allOrNone)
```

Parameters

mergeToRecord
Type: sObject

The sObject record that the duplicate record is merged into.

duplicate
Type: ID

The ID of the record to merge with the mergeToRecord. This record must be of the same sObject type as the mergeToRecord.

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

Return Value

Type: Database.MergeResult

Usage

Each executed `merge` method counts against the governor limit for DML statements.

merge(mergeToRecord, duplicateRecord, allOrNone)

Merges the duplicate sObject record into the `mergeToRecord` sObject of the same type, optionally returning any errors, deleting the duplicate, and reparenting any related records.

Signature

```
public static Database.MergeResult merge(sObject mergeToRecord, sObject duplicateRecord,
Boolean allOrNone)
```

Parameters

mergeToRecord
Type: sObject

The sObject record that the duplicate record is merged into.

duplicateRecord
Type: sObject

The sObject record to merge with the mergeToRecord. This sObject must be of the same type as the mergeToRecord.

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

Return Value

Type: Database.MergeResult

Usage

Each executed `merge` method counts against the governor limit for DML statements.

merge(mergeToRecord, duplicateIds, allOrNone)

Merges up to two records of the same sObject type into the `mergeToRecord` sObject record, optionally returning any errors, deleting the duplicates, and reparenting any related records.

Signature

```
public static List<Database.MergeResult> merge(sObject mergeToRecord, List<Id>
duplicateIds, Boolean allOrNone)
```

Parameters

mergeToRecord
Type: SObject

The sObject record that the other records are merged into.

duplicateIds
Type: List<Id>

A list of IDs of up to two records to merge with the mergeToRecord. These records must be of the same sObject type as the mergeToRecord.

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

Return Value

Type: List<Database.MergeResult>

Usage

Each executed `merge` method counts against the governor limit for DML statements.

merge(mergeToRecord, duplicateRecords, allOrNone)

Merges up to two records of the same object type into the `mergeToRecord` sObject record, optionally returning any errors, deleting the duplicates, and reparenting any related records.

Signature

```
public static List<Database.MergeResult> merge(sObject mergeToRecord, List<SObject>
duplicateRecords, Boolean allOrNone)
```

Parameters

mergeToRecord
Type: sObject

The sObject record that the other sObjects are merged into.

duplicateRecords
Type: List<SObject>

A list of up to two sObject records to merge with the mergeToRecord. These sObjects must be of the same type as the mergeToRecord.

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

Return Value

Type: List<Database.MergeResult>

Usage

Each executed `merge` method counts against the governor limit for DML statements.

merge(mergeToRecord, duplicateId, accessLevel)

Merges the duplicate record into the `mergeToRecord` sObject record of the same type, deleting the duplicate, and reparenting any related records. Merges only accounts, contacts, or leads.

Signature

```
public static Database.MergeResult merge(SObject mergeToRecord, Id duplicateId,
System.AccessLevel accessLevel)
```

Parameters

mergeToRecord
Type: sObject

The sObject record that the duplicate record is merged into.

duplicateId
Type: ID

The ID of the record to merge with the mergeToRecord. This record must be of the same sObject type as the mergeToRecord.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.MergeResult

Usage

Each executed `merge` method counts against the governor limit for DML statements.

merge(mergeToRecord, duplicateRecord, accessLevel)

Merges the specified duplicate sObject record into the `mergeToRecord` sObject of the same type, deleting the duplicate, and reparenting any related records.

Signature

```
public static Database.MergeResult merge(SObject mergeToRecord, SObject duplicateRecord,
System.AccessLevel accessLevel)
```

Parameters

mergeToRecord
Type: sObject

The sObject record that the duplicate record is merged into.

duplicateRecord
Type: sObject

The sObject record to merge with the mergeToRecord. This sObject must be of the same type as the mergeToRecord.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.MergeResult

Usage

Each executed `merge` method counts against the governor limit for DML statements.

merge(mergeToRecord, duplicateIds, accessLevel)

Merges up to two records of the same sObject type into the `mergeToRecord` sObject record, deleting the others, and reparenting any related records.

Signature

```
public static List<Database.MergeResult> merge(SObject mergeToRecord, List<Id>
duplicateIds, System.AccessLevel accessLevel)
```

Parameters

mergeToRecord
Type: SObject

The sObject record that the other records are merged into.

duplicateIds
Type: List<Id>

A list of IDs of up to two records to merge with the mergeToRecord. These records must be of the same sObject type as the mergeToRecord.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: List<Database.MergeResult>

Usage

Each executed `merge` method counts against the governor limit for DML statements.

merge(mergeToRecord, duplicateRecords, accessLevel)

Merges up to two records of the same object type into the `mergeToRecord` sObject record, deleting the others, and reparenting any related records.

Signature

```
public static List<Database.MergeResult> merge(SObject mergeToRecord, List<SObject>
duplicateRecords, System.AccessLevel accessLevel)
```

Parameters

mergeToRecord
Type: SObject

The sObject that the other sObject records are merged into.

duplicateRecords
Type: List<SObject>

A list of up to two sObject records to merge with the mergeToRecord. These sObjects must be of the same type as the mergeToRecord.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: List<Database.MergeResult>

Usage

Each executed `merge` method counts against the governor limit for DML statements.

merge(mergeToRecord, duplicateId, allOrNone, accessLevel)

Merges the duplicate record into the `mergeToRecord` sObject record of the same type, optionally returning any errors, deleting the duplicate, and reparenting any related records. Merges only accounts, contacts, or leads.

Signature

```
public static Database.MergeResult merge(SObject mergeToRecord, Id duplicateId, Boolean
allOrNone, System.AccessLevel accessLevel)
```

Parameters

mergeToRecord
Type: sObject

The sObject record that the duplicate record is merged into.

duplicateId
Type: ID

The ID of the record to merge with the mergeToRecord. This record must be of the same sObject type as the mergeToRecord.

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.MergeResult

Usage

If you use the `accessLevel` parameter to specify that the method runs in user mode, we report all encountered inaccessible fields. The way to retrieve the names of these inaccessible fields depends on the value of this method's `allOrNone` parameter, or the equivalent `DmlOptions.optAllOrNone` property. If you specify that:

- `allOrNone=true` or `DmlOptions.optAllOrNone=true`: Catch the `DMLException` and use the `DMLException.getDMLFieldNames()` method to retrieve the list of inaccessible fields. See Exception Class and Built-In Exceptions for more information.
- `allOrNone=false` or `DmlOptions.optAllOrNone=false`: For each failing record, we update the `Database.Error` object that results from the DML operation. Use the `Error.getFields()` method to retrieve the list of inaccessible fields. See the Error Class methods for more information.

Each executed `merge` method counts against the governor limit for DML statements.

merge(mergeToRecord, duplicateRecord, allOrNone, accessLevel)

Merges the duplicate sObject record into the `mergeToRecord` sObject record of the same type, optionally returning any errors, deleting the duplicate, and reparenting any related records.

Signature

```
public static Database.MergeResult merge(SObject mergeToRecord, SObject duplicateRecord,
Boolean allOrNone, System.AccessLevel accessLevel)
```

Parameters

mergeToRecord
Type: sObject

The sObject record that the duplicate record is merged into.

duplicateRecord
Type: sObject

The sObject record to merge with the mergeToRecord. This sObject must be of the same type as the mergeToRecord.

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.MergeResult

Usage

If you use the `accessLevel` parameter to specify that the method runs in user mode, we report all encountered inaccessible fields. The way to retrieve the names of these inaccessible fields depends on the value of this method's `allOrNone` parameter, or the equivalent `DmlOptions.optAllOrNone` property. If you specify that:

- `allOrNone=true` or `DmlOptions.optAllOrNone=true`: Catch the `DMLException` and use the `DMLException.getDMLFieldNames()` method to retrieve the list of inaccessible fields. See Exception Class and Built-In Exceptions for more information.
- `allOrNone=false` or `DmlOptions.optAllOrNone=false`: For each failing record, we update the `Database.Error` object that results from the DML operation. Use the `Error.getFields()` method to retrieve the list of inaccessible fields. See the Error Class methods for more information.

Each executed `merge` method counts against the governor limit for DML statements.

`merge(mergeToRecord, duplicateIds, allOrNone, accessLevel)`

Merges up to two records of the same sObject type into the `mergeToRecord` sObject record, optionally returning any errors, deleting the duplicates, and reparenting any related records.

Signature

```
public static List<Database.MergeResult> merge(SObject mergeToRecord, List<Id>
duplicateIds, Boolean allOrNone, System.AccessLevel accessLevel)
```

Parameters

mergeToRecord
Type: SObject

The sObject record that the other records are merged into.

duplicateIds
Type: List<Id>

A list of IDs of up to two records to merge with the mergeToRecord. These records must be of the same sObject type as the mergeToRecord.

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: List<Database.MergeResult>

Usage

If you use the `accessLevel` parameter to specify that the method runs in user mode, we report all encountered inaccessible fields. The way to retrieve the names of these inaccessible fields depends on the value of this method's `allOrNone` parameter, or the equivalent `DmlOptions.optAllOrNone` property. If you specify that:

- `allOrNone=true` or `DmlOptions.optAllOrNone=true`: Catch the `DMLException` and use the `DMLException.getDMLFieldNames()` method to retrieve the list of inaccessible fields. See Exception Class and Built-In Exceptions for more information.
- `allOrNone=false` or `DmlOptions.optAllOrNone=false`: For each failing record, we update the `Database.Error` object that results from the DML operation. Use the `Error.getFields()` method to retrieve the list of inaccessible fields. See the Error Class methods for more information.

Each executed `merge` method counts against the governor limit for DML statements.

`merge(mergeToRecord, duplicateRecords, allOrNone, accessLevel)`

Merges up to two records of the same object type into the `mergeToRecord` sObject record, optionally returning any errors, deleting the duplicates, and reparenting any related records.

Signature

```
public static List<Database.MergeResult> merge(SObject mergeToRecord, List<SObject>
duplicateRecords, Boolean allOrNone, System.AccessLevel accessLevel)
```

Parameters

mergeToRecord
Type: sObject

The sObject record that the other sObjects are merged into.

duplicateRecords
Type: List<SObject>

A list of up to two sObject records to merge with the mergeToRecord. These sObjects must be of the same type as the mergeToRecord.

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: List<Database.MergeResult>

Usage

If you use the `accessLevel` parameter to specify that the method runs in user mode, we report all encountered inaccessible fields. The way to retrieve the names of these inaccessible fields depends on the value of this method's `allOrNone` parameter, or the equivalent `DmlOptions.optAllOrNone` property. If you specify that:

- `allOrNone=true` or `DmlOptions.optAllOrNone=true`: Catch the `DMLException` and use the `DMLException.getDMLFieldNames()` method to retrieve the list of inaccessible fields. See Exception Class and Built-In Exceptions for more information.
- `allOrNone=false` or `DmlOptions.optAllOrNone=false`: For each failing record, we update the `Database.Error` object that results from the DML operation. Use the `Error.getFields()` method to retrieve the list of inaccessible fields. See the Error Class methods for more information.

Each executed `merge` method counts against the governor limit for DML statements.

`query(queryString)`

Creates a dynamic SOQL query at runtime.

Signature

```
public static List<SObject> query(String queryString)
```

Parameters

queryString
Type: String

Return Value

Type: List on page 3603<sObject>

Usage

This method can be used wherever a static SOQL query can be used, such as in regular assignment statements and `for` loops. Unlike inline SOQL, fields in bind variables aren't supported. For more information, see Dynamic SOQL.

`Database.query()` calls containing an inner query for a related child object may not return the entire result set based on the size and complexity of the records requested. Instead, use `Database.getQueryLocator()` in conjunction with Apex Batch. Alternatively, you can use the same SOQL query with SOAP API to be able to access all the resulting records.

Each executed `query` method counts against the governor limit for SOQL queries.

query(queryString, accessLevel)

Creates a dynamic SOQL query at runtime.

Signature

```
public static List<SObject> query(String queryString, System.AccessLevel accessLevel)
```

Parameters

queryString
Type: String

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: List on page 3603<sObject>

Usage

This method can be used wherever a static SOQL query can be used, such as in regular assignment statements and `for` loops. Unlike inline SOQL, fields in bind variables aren't supported. For more information, see Dynamic SOQL.

`Database.query()` calls containing an inner query for a related child object may not return the entire result set based on the size and complexity of the records requested. Instead, use `Database.getQueryLocator()` in conjunction with Apex Batch. Alternatively, you can use the same SOQL query with SOAP API to be able to access all the resulting records.

Each executed `query` method counts against the governor limit for SOQL queries.

`queryWithBinds(queryString, bindMap, accessLevel)`

Creates a dynamic SOQL query at runtime. Bind variables in the query are resolved from the *`bindMap`* Map parameter directly with the key, rather than from Apex code variables.

Signature

```
public static List<SObject> queryWithBinds(String queryString, Map<String, Object>
bindMap, System.AccessLevel accessLevel)
```

Parameters

`queryString`
Type: String

SOQL query that includes Apex bind variables or expressions preceded by a colon. All bind variables must have a key in the *`bindMap`* Map.

`bindMap`
Type: Map<String, Object>

A map that contains keys for each bind variable specified in the SOQL *`queryString`* and its value. The keys can't be null or duplicates, and the values can't be null or empty strings.

`accessLevel`
Type: System.AccessLevel

The *`accessLevel`* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced.

Return Value

Type: List on page 3603<sObject>

Usage

This method can be used wherever a static SOQL query can be used, such as in regular assignment statements and `for` loops.

For more information, see Dynamic SOQL.

Each executed `queryWithBinds` method counts against the governor limit for SOQL queries.

Example

In this example, the SOQL query uses a bind variable for an Account name. Its value (`Acme Inc.`) is passed in to the method using the *`nameBind`* Map. The `accountName` variable isn't (and doesn't have to be) in scope when the query is executed within the method.

```
public static List<Account> simpleBindingSoqlQuery(Map<String, Object> bindParams) {
    String queryString =
        'SELECT Id, Name ' +
        'FROM Account ' +
        'WHERE name = :name';
    return Database.queryWithBinds(
        queryString,
        bindParams,
        AccessLevel.USER_MODE
    );
}

String accountName = 'Acme Inc.';
Map<String, Object> nameBind = new Map<String, Object>{'name' => accountName};
List<Account> accounts = simpleBindingSoqlQuery(nameBind);
System.debug(accounts);
```

releaseSavepoint(databaseSavepoint)

Releases a given savepoint. All savepoints that are subsequent to the given one are also released.

Signature

```
public static void releaseSavepoint(System.Savepoint databaseSavepoint)
```

Parameters

`databaseSavepoint`
Type: System.Savepoint

Return Value

Type: void

Versioned Behavior Changes

For Apex tests with API version 60.0 or later, all savepoints are released when `Test.startTest()` and `Test.stopTest()` are called. If any savepoints are reset, a `SAVEPOINT_RESET` event is logged.

Before API version 60.0, making a callout after creating savepoints throws a `CalloutException` regardless of whether there was uncommitted DML or the changes were rolled back to a savepoint. Also, before API version 60.0, both `Database.rollback(databaseSavepoint)` and `Database.setSavepoint()` calls incremented the DML row usage limit.

rollback(databaseSavepoint)

Restores the database to the state specified by the savepoint variable. Any emails submitted since the last savepoint are also rolled back and not sent.

Signature

```
public static Void rollback(System.Savepoint databaseSavepoint)
```

Parameters

databaseSavepoint
Type: System.Savepoint

Return Value

Type: Void

Usage

Note the following:

- Static variables aren't reverted during a rollback. If you try to run the trigger again, the static variables retain the values from the first run.
- Each rollback counts against the governor limit for DML statements. You receive a runtime error if you try to roll back the database additional times.
- The ID on an sObject inserted after setting a savepoint isn't cleared after a rollback. Create an sObject to insert after a rollback. Attempting to insert the sObject using the variable created before the rollback fails because the sObject variable has an ID. Updating or upserting the sObject using the same variable also fails because the sObject isn't in the database and, thus, can't be updated.

For an example, see Transaction Control.

Versioned Behavior Changes

For Apex tests with API version 60.0 or later, all savepoints are released when `Test.startTest()` and `Test.stopTest()` are called. If any savepoints are reset, a `SAVEPOINT_RESET` event is logged.

Before API version 60.0, making a callout after creating savepoints throws a `CalloutException` regardless of whether there was uncommitted DML or the changes were rolled back to a savepoint. Also, before API version 60.0, both `Database.rollback(Savepoint)` and `Database.setSavepoint()` calls incremented the DML row usage limit.

setSavepoint()

Returns a savepoint variable that can be stored as a local variable, then used with the `rollback` method to restore the database to that point.

Signature

```
public static System.Savepoint setSavepoint()
```

Return Value

Type: System.Savepoint

Usage

Note the following:

- If you set more than one savepoint, then roll back to a savepoint that isn't the last savepoint you generated, the later savepoint variables become invalid. For example, if you generated savepoint `SP1` first, savepoint `SP2` after that, and then you rolled back to `SP1`, the variable `SP2` would no longer be valid. You receive a runtime error if you try to use it.
- References to savepoints can't cross trigger invocations because each trigger invocation is a new trigger context. If you declare a savepoint as a static variable then try to use it across trigger contexts, you receive a run-time error.
- Each savepoint you set counts against the governor limit for DML statements.

For an example, see Transaction Control.

Versioned Behavior Changes

For Apex tests with API version 60.0 or later, all savepoints are released when `Test.startTest()` and `Test.stopTest()` are called. If any savepoints are reset, a `SAVEPOINT_RESET` event is logged.

Before API version 60.0, making a callout after creating savepoints throws a `CalloutException` regardless of whether there was uncommitted DML or the changes were rolled back to a savepoint. Also, before API version 60.0, both `Database.rollback(Savepoint)` and `Database.setSavepoint()` calls incremented the DML row usage limit.

`undelete(recordToUndelete, allOrNone)`

Restores an existing sObject record, such as an individual account or contact, from your organization's Recycle Bin.

Signature

```
public static Database.UndeleteResult undelete(sObject recordToUndelete, Boolean allOrNone)
```

Parameters

`recordToUndelete`
Type: sObject

`allOrNone`
Type: Boolean

(Optional) The *`allOrNone`* parameter specifies whether the operation allows partial success. If *`allOrNone`* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *`allOrNone`* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

Return Value

Type: Database.UndeleteResult

Usage

`undelete` is analogous to the UNDELETE statement in SQL.

Each executed `undelete` method counts against the governor limit for DML statements.

undelete(recordsToUndelete, allOrNone)

Restores one or more existing sObject records, such as individual accounts or contacts, from your organization's Recycle Bin.

Signature

```
public static Database.UndeleteResult[] undelete(sObject[] recordsToUndelete, Boolean allOrNone)
```

Parameters

recordsToUndelete
Type: sObject []

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

Return Value

Type: Database.UndeleteResult[]

Usage

`undelete` is analogous to the UNDELETE statement in SQL.

Each executed `undelete` method counts against the governor limit for DML statements.

Example

The following example restores all accounts named 'Universal Containers'. The `ALL ROWS` keyword queries all rows for both top-level and aggregate relationships, including deleted records and archived activities.

```
Account[] savedAccts = [SELECT Id, Name FROM Account
                                                    WHERE Name = 'Universal
 Containers' ALL ROWS];
Database.UndeleteResult[] UDR_Dels = Database.undelete(savedAccts);
```

undelete(recordID, allOrNone)

Restores an existing sObject record, such as an individual account or contact, from your organization's Recycle Bin.

Signature

```
public static Database.UndeleteResult undelete(ID recordID, Boolean allOrNone)
```

Parameters

recordID
Type: ID

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

Return Value

Type: Database.UndeleteResult

Usage

`undelete` is analogous to the UNDELETE statement in SQL.

Each executed `undelete` method counts against the governor limit for DML statements.

undelete(recordIDs, allOrNone)

Restores one or more existing sObject records, such as individual accounts or contacts, from your organization's Recycle Bin.

Signature

```
public static Database.UndeleteResult[] undelete(ID[] recordIDs, Boolean allOrNone)
```

Parameters

RecordIDs
Type: ID[]

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

Return Value

Type: Database.UndeleteResult[]

Usage

`undelete` is analogous to the UNDELETE statement in SQL.

Each executed `undelete` method counts against the governor limit for DML statements.

undelete(recordToUndelete, allOrNone, accessLevel)

Restores an existing sObject record, such as an individual account or contact, from your organization's Recycle Bin.

Signature

```
public static Database.UndeleteResult undelete(SObject recordToUndelete, Boolean
allOrNone, System.AccessLevel accessLevel)
```

Parameters

recordToUndelete
Type: SObject

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.UndeleteResult

Usage

`undelete` is analogous to the UNDELETE statement in SQL.

Each executed `undelete` method counts against the governor limit for DML statements.

undelete(recordsToUndelete, allOrNone, accessLevel)

Restores one or more existing sObject records, such as individual accounts or contacts, from your organization's Recycle Bin.

Signature

```
public static List<Database.UndeleteResult> undelete(List<SObject> recordsToUndelete,
Boolean allOrNone, System.AccessLevel accessLevel)
```

Parameters

recordsToUndelete
Type: List<sObject>

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: List<Database.UndeleteResult>

Usage

`undelete` is analogous to the UNDELETE statement in SQL.

Each executed `undelete` method counts against the governor limit for DML statements.

undelete(recordID, allOrNone, accessLevel)

Restores an existing sObject record, such as an individual account or contact, from your organization's Recycle Bin.

Signature

```
public static Database.UndeleteResult undelete(Id recordID, Boolean allOrNone,
System.AccessLevel accessLevel)
```

Parameters

recordID
Type: Id

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.UndeleteResult

Usage

`undelete` is analogous to the UNDELETE statement in SQL.

Each executed `undelete` method counts against the governor limit for DML statements.

undelete(recordIDs, allOrNone, accessLevel)

Restores one or more existing sObject records, such as individual accounts or contacts, from your organization's Recycle Bin.

Signature

```
public static List<Database.UndeleteResult> undelete(List<Id> recordIDs, Boolean
allOrNone, System.AccessLevel accessLevel)
```

Parameters

recordIDs
Type: List<ID>

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: List<Database.UndeleteResult>

Usage

`undelete` is analogous to the UNDELETE statement in SQL.

Each executed `undelete` method counts against the governor limit for DML statements.

update(recordToUpdate, allOrNone)

Modifies an existing sObject record, such as an individual account or contact, in your organization's data.

Signature

```
public static Database.SaveResult update(sObject recordToUpdate, Boolean allOrNone)
```

Parameters

recordToUpdate
Type: sObject

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

Return Value

Type: Database.SaveResult

Usage

`update` is analogous to the UPDATE statement in SQL.

Apex classes and triggers saved (compiled) using API version 15.0 and higher produce a runtime error if you assign a String value that is too long for the field.

Each executed `update` method counts against the governor limit for DML statements.

Example

The following example updates the `BillingCity` field on a single account.

```
Account a = new Account(Name='SFDC');
insert(a);

Account myAcct =
  [SELECT Id, Name, BillingCity
   FROM Account WHERE Id = :a.Id];
myAcct.BillingCity = 'San Francisco';

Database.SaveResult SR =
  Database.update(myAcct);
```

update(recordsToUpdate, allOrNone)

Modifies one or more existing sObject records, such as individual accounts or contacts, in your organization's data.

Signature

```
public static Database.SaveResult[] update(sObject[] recordsToUpdate, Boolean allOrNone)
```

Parameters

recordsToUpdate
Type: sObject []

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

Return Value

Type: Database.SaveResult[]

Usage

`update` is analogous to the UPDATE statement in SQL.

Each executed `update` method counts against the governor limit for DML statements.

update(recordToUpdate, dmlOptions)

Modifies an existing sObject record, such as an individual account or contact, in your organization's data.

Signature

```
public static Database.SaveResult update(sObject recordToUpdate, Database.DmlOptions
dmlOptions)
```

Parameters

recordToUpdate
Type: sObject

dmlOptions
Type: Database.DMLOptions

The optional *dmlOptions* parameter specifies additional data for the transaction, such as assignment rule information or rollback behavior when errors occur during record insertions.

Return Value

Type: Database.SaveResult

Usage

`update` is analogous to the UPDATE statement in SQL.

Apex classes and triggers saved (compiled) using API version 15.0 and higher produce a runtime error if you assign a String value that is too long for the field.

Each executed `update` method counts against the governor limit for DML statements.

update(recordsToUpdate, dmlOptions)

Modifies one or more existing sObject records, such as individual accounts or contacts, in your organization's data.

Signature

```
public static Database.SaveResult[] update(sObject[] recordsToUpdate, Database.DMLOptions dmlOptions)
```

Parameters

recordsToUpdate
Type: sObject []

dmlOptions
Type: Database.DMLOptions

The optional *dmlOptions* parameter specifies additional data for the transaction, such as assignment rule information or rollback behavior when errors occur during record insertions.

Return Value

Type: Database.SaveResult[]

Usage

`update` is analogous to the UPDATE statement in SQL.

Apex classes and triggers saved (compiled) using API version 15.0 and higher produce a runtime error if you assign a String value that is too long for the field.

Each executed `update` method counts against the governor limit for DML statements.

update(recordToUpdate, allOrNone, accessLevel)

Modifies an existing sObject record, such as an individual account or contact, in your organization's data.

Signature

```
public static Database.SaveResult update(SObject recordToUpdate, Boolean allOrNone, System.AccessLevel accessLevel)
```

Parameters

recordToUpdate
Type: SObject

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.SaveResult

Usage

If you use the `accessLevel` parameter to specify that the method runs in user mode, we report all encountered inaccessible fields. The way to retrieve the names of these inaccessible fields depends on the value of this method's `allOrNone` parameter, or the equivalent `DmlOptions.optAllOrNone` property. If you specify that:

- `allOrNone=true` or `DmlOptions.optAllOrNone=true`: Catch the `DMLException` and use the `DMLException.getDMLFieldNames()` method to retrieve the list of inaccessible fields. See Exception Class and Built-In Exceptions for more information.
- `allOrNone=false` or `DmlOptions.optAllOrNone=false`: For each failing record, we update the `Database.Error` object that results from the DML operation. Use the `Error.getFields()` method to retrieve the list of inaccessible fields. See the Error Class methods for more information.

update(recordsToUpdate, allOrNone, accessLevel)

Modifies one or more existing sObject records, such as individual accounts or contacts, in your organization's data.

Signature

```
public static List<Database.SaveResult> update(List<SObject> recordsToUpdate, Boolean
allOrNone, System.AccessLevel accessLevel)
```

Parameters

recordsToUpdate
Type: List<sObject>

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: List<Database.SaveResult>

Usage

If you use the `accessLevel` parameter to specify that the method runs in user mode, we report all encountered inaccessible fields. The way to retrieve the names of these inaccessible fields depends on the value of this method's `allOrNone` parameter, or the equivalent `DmlOptions.optAllOrNone` property. If you specify that:

- `allOrNone=true` or `DmlOptions.optAllOrNone=true`: Catch the `DMLException` and use the `DMLException.getDMLFieldNames()` method to retrieve the list of inaccessible fields. See Exception Class and Built-In Exceptions for more information.
- `allOrNone=false` or `DmlOptions.optAllOrNone=false`: For each failing record, we update the `Database.Error` object that results from the DML operation. Use the `Error.getFields()` method to retrieve the list of inaccessible fields. See the Error Class methods for more information.

update(recordToUpdate, dmlOptions, accessLevel)

Modifies an existing sObject record, such as an individual account or contact, in your organization's data.

Signature

```
public static Database.SaveResult update(SObject recordToUpdate, Database.DMLOptions
dmlOptions, System.AccessLevel accessLevel)
```

Parameters

recordToUpdate
Type: SObject

dmlOptions
Type: Database.DMLOptions

The optional *dmlOptions* parameter specifies additional data for the transaction, such as assignment rule information or rollback behavior when errors occur during record insertions.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.SaveResult

update(recordsToUpdate, dmlOptions, accessLevel)

Modifies one or more existing sObject records, such as individual accounts or contacts, in your organization's data.

Signature

```
public static List<Database.SaveResult> update(List<SObject> recordsToUpdate,
Database.DMLOptions dmlOptions, System.AccessLevel accessLevel)
```

Parameters

recordsToUpdate
Type: List<sObject>

dmlOptions
Type: Database.DMLOptions

The optional *dmlOptions* parameter specifies additional data for the transaction, such as assignment rule information or rollback behavior when errors occur during record insertions.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: List<Database.SaveResult>

upsert(recordToUpsert, externalIdField, allOrNone)

Creates a new sObject record or updates an existing sObject record within a single statement, using a specified field to determine the presence of existing objects, or the ID field if no field is specified.

Signature

```
public static Database.UpsertResult upsert(sObject recordToUpsert, Schema.SObjectField
externalIDField, Boolean allOrNone)
```

Parameters

recordToUpsert
Type: sObject

externalIdField
Type: Schema.SObjectField

(Optional) The *externalIdField* is of type `Schema.SObjectField`, that is, a field token. Find the token for the field by using the `fields` special method. For example, `Schema.SObjectField f = Account.Fields.MyExternalId`. The *externalIdField* parameter is the field that `upsert()` uses to match sObjects with existing records. This field can be a custom field marked as external ID, or a standard field with the `idLookup` attribute.

Note: If *externalIdField* isn't specified, then the ID field is used to determine a match with existing records.

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

Return Value

Type: Database.UpsertResult

Usage

Apex classes and triggers saved (compiled) using API version 15.0 and higher produce a runtime error if you assign a String value that is too long for the field.

Each executed `upsert` method counts against the governor limit for DML statements.

For more information on how the upsert operation works, see the upsert() statement.

upsert(recordsToUpsert, externalIdField, allOrNone)

Creates new sObject records or updates existing sObject records within a single statement, using a specified field to determine the presence of existing objects, or the ID field if no field is specified.

Signature

```
public static Database.UpsertResult[] upsert(sObject[] recordsToUpsert,
Schema.SObjectField externalIdField, Boolean allOrNone)
```

Parameters

recordsToUpsert
Type: sObject []

externalIdField
Type: Schema.SObjectField

(Optional) The *externalIdField* is of type `Schema.SObjectField`, that is, a field token. Find the token for the field by using the `fields` special method. For example, `Schema.SObjectField f = Account.Fields.MyExternalId`. The *externalIdField* parameter is the field that `upsert()` uses to match sObjects with existing records. This field can be a custom field marked as external ID, or a standard field with the `idLookup` attribute.

Note: If *externalIdField* isn't specified, then the ID field is used to determine a match with existing records.

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

If *allOrNone* is set to false and a before-trigger assigns an invalid value to a field, the partial set of valid records isn't inserted.

Return Value

Type: Database.UpsertResult[]

Usage

Apex classes and triggers saved (compiled) using API version 15.0 and higher produce a runtime error if you assign a String value that is too long for the field.

Each executed upsert method counts against the governor limit for DML statements.

For more information on how the upsert operation works, see the upsert() statement.

upsert(recordToUpsert, externalIdField, allOrNone, accessLevel)

Creates a new sObject record or updates an existing sObject record within a single statement, using a specified field to determine the presence of existing objects, or the ID field if no field is specified.

Signature

```
public static Database.UpsertResult upsert(SObject recordToUpsert, Schema.SObjectField
externalIdField, Boolean allOrNone, System.AccessLevel accessLevel)
```

Parameters

recordToUpsert
Type: SObject

externalIdField
Type: Schema.SObjectField

(Optional) The *externalIdField* is of type Schema.SObjectField, that is, a field token. Find the token for the field by using the fields special method. For example, Schema.SObjectField f = Account.Fields.MyExternalId. The *externalIdField* parameter is the field that upsert() uses to match sObjects with existing records. This field can be a custom field marked as external ID, or a standard field with the idLookup attribute.

Note: If *externalIdField* isn't specified, then the ID field is used to determine a match with existing records.

allOrNone
Type: Boolean

(Optional) The *allOrNone* parameter specifies whether the operation allows partial success. If *allOrNone* is set to false and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify which records succeeded or failed. If *allOrNone* is set to true and the method isn't successful, an exception is thrown. The default for the parameter is true.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (AccessLevel.SYSTEM_MODE) or user mode (AccessLevel.USER_MODE). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.UpsertResult

Usage

If you use the `accessLevel` parameter to specify that the method runs in user mode, we report all encountered inaccessible fields. The way to retrieve the names of these inaccessible fields depends on the value of this method's `allOrNone` parameter, or the equivalent `DmlOptions.optAllOrNone` property. If you specify that:

- `allOrNone=true` or `DmlOptions.optAllOrNone=true`: Catch the `DMLException` and use the `DMLException.getDMLFieldNames()` method to retrieve the list of inaccessible fields. See Exception Class and Built-In Exceptions for more information.
- `allOrNone=false` or `DmlOptions.optAllOrNone=false`: For each failing record, we update the `Database.Error` object that results from the DML operation. Use the `Error.getFields()` method to retrieve the list of inaccessible fields. See the Error Class methods for more information.

Apex classes and triggers saved (compiled) using API version 15.0 and higher produce a runtime error if you assign a String value that is too long for the field.

Each executed `upsert` method counts against the governor limit for DML statements.

For more information on how the upsert operation works, see the upsert() statement.

`upsert(recordsToUpsert, externalIdField, allOrNone, accessLevel)`

Creates new sObject records or updates existing sObject records within a single statement, using a specified field to determine the presence of existing objects, or the ID field if no field is specified.

Signature

```
public static List<Database.UpsertResult> upsert(List<SObject> recordsToUpsert,
Schema.SObjectField externalIdField, Boolean allOrNone, System.AccessLevel accessLevel)
```

Parameters

`recordsToUpsert`
Type: List<sObject >

`externalIdField`
Type: Schema.SObjectField

(Optional) The *`externalIdField`* is of type `Schema.SObjectField`, that is, a field token. Find the token for the field by using the `fields` special method. For example, `Schema.SObjectField f = Account.Fields.MyExternalId`. The *`externalIdField`* parameter is the field that `upsert()` uses to match sObjects with existing records. This field can be a custom field marked as external ID, or a standard field with the `idLookup` attribute.

Note: If *`externalIdField`* isn't specified, then the ID field is used to determine a match with existing records.

`allOrNone`
Type: Boolean

(Optional) The *`allOrNone`* parameter specifies whether the operation allows partial success. If *`allOrNone`* is set to `false` and a record fails, the remainder of the DML operation can still succeed. You must iterate through the returned results to identify

which records succeeded or failed. If *allOrNone* is set to `true` and the method isn't successful, an exception is thrown. The default for the parameter is `true`.

If *allOrNone* is set to `false` and a before-trigger assigns an invalid value to a field, the partial set of valid records isn't inserted.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: List<Database.UpsertResult>

Usage

If you use the `accessLevel` parameter to specify that the method runs in user mode, we report all encountered inaccessible fields. The way to retrieve the names of these inaccessible fields depends on the value of this method's `allOrNone` parameter, or the equivalent `DmlOptions.optAllOrNone` property. If you specify that:

- `allOrNone=true` or `DmlOptions.optAllOrNone=true`: Catch the `DMLException` and use the `DMLException.getDMLFieldNames()` method to retrieve the list of inaccessible fields. See Exception Class and Built-In Exceptions for more information.
- `allOrNone=false` or `DmlOptions.optAllOrNone=false`: For each failing record, we update the `Database.Error` object that results from the DML operation. Use the `Error.getFields()` method to retrieve the list of inaccessible fields. See the Error Class methods for more information.

Apex classes and triggers saved (compiled) using API version 15.0 and higher produce a runtime error if you assign a String value that is too long for the field.

Each executed `upsert` method counts against the governor limit for DML statements.

For more information on how the upsert operation works, see the upsert() statement.

updateAsync(sobjects, callback)

Initiates requests to update external object data on the relevant external systems. The requests are executed asynchronously, as background operations, and are sent to the external systems that are defined by the external objects' associated external data sources. Allows referencing a callback class whose `processSave` method is called for each record after the remote operations are completed.

Signature

```
public static List<Database.SaveResult> updateAsync(List<SObject> sobjects,
DataSource.AsyncSaveCallback callback)
```

Parameters

sobjects
Type: List<SObject>

List of external object records to modify.

callback
Type: DataSource.AsyncSaveCallback

The callback object that contains the state in the originating context and an action (the `processSave` method) that executes after the insert operation is completed. Use the action callback to update org data according to the operation's results. The callback object must extend `DataSource.AsyncSaveCallback`.

Return Value

Type: List<Database.SaveResult>

Status results for the update operation. Each result corresponds to a record processed by this asynchronous operation and is associated with a unique identifier (`asyncLocator`). The `asyncLocator` value is included in the errors array of the result. You can retrieve this identifier with `Database.getAsyncLocator()`. Retrieve the final result with `Database.getAsyncSaveResult()`.

updateAsync(sobject, callback)

Initiates a request to update external object data on the relevant external system. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source. Allows referencing a callback class whose `processSave` method is called after the remote operation is completed.

Signature

```
public static Database.SaveResult updateAsync(SObject sobject,
DataSource.AsyncSaveCallback callback)
```

Parameters

sobject
Type: SObject

External object record to modify.

callback
Type: DataSource.AsyncSaveCallback

The callback object that contains the state in the originating context and an action (the `processSave` method) that executes after the insert operation is completed. Use the action callback to update org data according to the operation's results. The callback object must extend `DataSource.AsyncSaveCallback`.

Return Value

Type: Database.SaveResult

Status result for the insert operation. The result corresponds to a record processed by this asynchronous operation and is associated with a unique identifier (`asyncLocator`). The `asyncLocator` value is included in the errors array of the result. You can retrieve this identifier with `Database.getAsyncLocator()`. Retrieve the final result with `Database.getAsyncSaveResult()`.

updateAsync(sobjects)

Initiates requests to update external object data on the relevant external systems. The requests are executed asynchronously, as background operations, and are sent to the external systems that are defined by the external objects' associated external data sources.

Signature

```
public static List<Database.SaveResult> updateAsync(List<SObject> sobjects)
```

Parameters

sobjects
Type: List<SObject>

List of external object records to modify.

Return Value

Type: List<Database.SaveResult>

Status results for the update operation. Each result corresponds to a record processed by this asynchronous operation and is associated with a unique identifier (`asyncLocator`). The `asyncLocator` value is included in the errors array of the result. You can retrieve this identifier with `Database.getAsyncLocator()`. Retrieve the final result with `Database.getAsyncSaveResult()`.

`updateAsync(sobject)`

Initiates a request to update external object data on the relevant external system. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source.

Signature

```
public static Database.SaveResult updateAsync(SObject sobject)
```

Parameters

sobject
Type: SObject

External object record to modify.

Return Value

Type: Database.SaveResult

Status result for the insert operation. The result corresponds to a record processed by this asynchronous operation and is associated with a unique identifier (`asyncLocator`). The `asyncLocator` value is included in the errors array of the result. You can retrieve this identifier with `Database.getAsyncLocator()`. Retrieve the final result with `Database.getAsyncSaveResult()`.

updateAsync(sobjects, callback, accessLevel)

Initiates requests to update external object data on the relevant external systems. The requests are executed asynchronously, as background operations, and are sent to the external systems that are defined by the external objects' associated external data sources. Allows referencing a callback class whose `processSave` method is called for each record after the remote operations are completed.

Signature

```
public static List<Database.SaveResult> updateAsync(List<SObject> sobjects,
DataSource.AsyncSaveCallback callback, System.AccessLevel accessLevel)
```

Parameters

sobjects
Type: List<SObject>

List of external object records to modify.

callback
Type: DataSource.AsyncSaveCallback

The callback object that contains the state in the originating context and an action (the `processSave` method) that executes after the insert operation is completed. The execution is in system mode regardless of the `accessLevel` parameter. Use the action callback to update org data according to the operation's results. The callback object must extend `DataSource.AsyncSaveCallback`.

accessLevel
Type: System.AccessLevel

(Optional) The *`accessLevel`* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: List<Database.SaveResult>

Status results for the update operation. Each result corresponds to a record processed by this asynchronous operation and is associated with a unique identifier (`asyncLocator`). The `asyncLocator` value is included in the errors array of the result. You can retrieve this identifier with `Database.getAsyncLocator()`. Retrieve the final result with `Database.getAsyncSaveResult()`.

updateAsync(sobject, callback, accessLevel)

Initiates a request to update external object data on the relevant external system. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source. Allows referencing a callback class whose `processSave` method is called after the remote operation is completed.

Signature

```
public static Database.SaveResult updateAsync(SObject sobject,
DataSource.AsyncSaveCallback callback, System.AccessLevel accessLevel)
```

Parameters

sobject
Type: SObject

External object record to modify.

callback
Type: DataSource.AsyncSaveCallback

The callback object that contains the state in the originating context and an action (the `processSave` method) that executes after the insert operation is completed. The execution is in system mode regardless of the `accessLevel` parameter. Use the action callback to update org data according to the operation's results. The callback object must extend `DataSource.AsyncSaveCallback`.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.SaveResult

Status result for the insert operation. The result corresponds to a record processed by this asynchronous operation and is associated with a unique identifier (`asyncLocator`). The `asyncLocator` value is included in the errors array of the result. You can retrieve this identifier with `Database.getAsyncLocator()`. Retrieve the final result with `Database.getAsyncSaveResult()`.

updateAsync(sobjects, accessLevel)

Initiates requests to update external object data on the relevant external systems. The requests are executed asynchronously, as background operations, and are sent to the external systems that are defined by the external objects' associated external data sources.

Signature

```
public static List<Database.SaveResult> updateAsync(List<SObject> sobjects,
System.AccessLevel accessLevel)
```

Parameters

sobjects
Type: List<SObject>

List of external object records to modify.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: List<Database.SaveResult>

Status results for the update operation. Each result corresponds to a record processed by this asynchronous operation and is associated with a unique identifier (`asyncLocator`). The `asyncLocator` value is included in the errors array of the result. You can retrieve this identifier with `Database.getAsyncLocator()`. Retrieve the final result with `Database.getAsyncSaveResult()`.

updateAsync(sobject, accessLevel)

Initiates a request to update external object data on the relevant external system. The request is executed asynchronously, as a background operation, and is sent to the external system that's defined by the external object's associated external data source.

Signature

```
public static Database.SaveResult updateAsync(SObject sobject, System.AccessLevel
accessLevel)
```

Parameters

sobject
Type: SObject

External object record to modify.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.SaveResult

Status result for the insert operation. The result corresponds to a record processed by this asynchronous operation and is associated with a unique identifier (`asyncLocator`). The `asyncLocator` value is included in the errors array of the result. You can retrieve this identifier with `Database.getAsyncLocator()`. Retrieve the final result with `Database.getAsyncSaveResult()`.

`updateImmediate(sobjects)`

Initiates requests to update external object data on the relevant external systems. The requests are executed synchronously and are sent to the external systems that are defined by the external objects' associated external data sources. If the Apex transaction contains pending changes, the synchronous operations can't be completed and throw exceptions.

Signature

```
public static List<Database.SaveResult> updateImmediate(List<SObject> sobjects)
```

Parameters

sobjects
Type: List<SObject>

List of external object records to modify.

Return Value

Type: List<Database.SaveResult>

Status results for the update operation.

Usage

The operation allows partial success. If one or more record updates fail, the method doesn't throw an exception and the remainder of the DML operation can still succeed. The returned `SaveResult` objects indicate whether the operation was successful. If it wasn't successful, the objects also return the error code and description.

updateImmediate(sobject)

Initiates a request to update external object data on the relevant external system. The request is executed synchronously and is sent to the external system that's defined by the external object's associated external data source. If the Apex transaction contains pending changes, the synchronous operation can't be completed and throws an exception.

Signature

```
public static Database.SaveResult updateImmediate(SObject sobject)
```

Parameters

sobject
Type: SObject

External object record to modify.

Return Value

Type: Database.SaveResult

Status result for the update operation.

Usage

If a record update fails, the method doesn't throw an exception. The returned `SaveResult` object indicates whether the operation was successful. If it wasn't successful, the object returns the error code and description.

updateImmediate(sobjects, accessLevel)

Initiates requests to update external object data on the relevant external systems. The requests are executed synchronously and are sent to the external systems that are defined by the external objects' associated external data sources. If the Apex transaction contains pending changes, the synchronous operations can't be completed and throw exceptions.

Signature

```
public static List<Database.SaveResult> updateImmediate(List<SObject> sobjects,
System.AccessLevel accessLevel)
```

Parameters

sobjects
Type: List<SObject>

List of external object records to modify.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: List<Database.SaveResult>

Status results for the update operation.

Usage

The operation allows partial success. If one or more record updates fail, the method doesn't throw an exception and the remainder of the DML operation can still succeed. The returned `SaveResult` objects indicate whether the operation was successful. If it wasn't successful, the objects also return the error code and description.

`updateImmediate(sobject, accessLevel)`

Initiates a request to update external object data on the relevant external system. The request is executed synchronously and is sent to the external system that's defined by the external object's associated external data source. If the Apex transaction contains pending changes, the synchronous operation can't be completed and throws an exception.

Signature

```
public static Database.SaveResult updateImmediate(SObject sobject, System.AccessLevel accessLevel)
```

Parameters

sobject
Type: SObject

External object record to modify.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Database.SaveResult

Status result for the update operation.

Usage

If a record update fails, the method doesn't throw an exception. The returned `SaveResult` object indicates whether the operation was successful. If it failed, the object returns the error code and description.

Date Class

Contains methods for the Date primitive data type.

Namespace

System

Usage

For more information on Dates, see Date Data Type.

Date Methods

The following are methods for `Date`.

IN THIS SECTION:

addDays(additionalDays)
Adds the specified number of additional days to a Date.

addMonths(additionalMonths)
Adds the specified number of additional months to a Date

addYears(additionalYears)
Adds the specified number of additional years to a Date

day()
Returns the day-of-month component of a Date.

dayOfYear()
Returns the day-of-year component of a Date.

daysBetween(secondDate)
Returns the number of days between the Date that called the method and the specified date.

daysInMonth(year, month)
Returns the number of days in the month for the specified *`year`* and *`month`* (1=Jan).

format()
Returns the Date as a string using the locale of the context user

isLeapYear(year)
Returns `true` if the specified year is a leap year.

isSameDay(dateToCompare)
Returns `true` if the Date that called the method is the same as the specified date.

month()
Returns the month component of a Date (1=Jan).

monthsBetween(secondDate)
Returns the number of months between the Date that called the method and the specified date, ignoring the difference in days.

newInstance(year, month, day)
Constructs a Date from Integer representations of the *`year`*, *`month`* (1=Jan), and *`day`*.

parse(stringDate)
Constructs a Date from a String. The format of the String depends on the local date format.

today()
Returns the current date in the current user's time zone.

toStartOfMonth()
Returns the first of the month for the Date that called the method.

toStartOfWeek()
Returns the start of the week for the Date that called the method, depending on the context user's locale.

valueOf(stringDate)
Returns a Date that contains the value of the specified String.

valueOf(fieldValue)
Converts the specified object to a `Date`. Use this method to convert a history tracking field value or an object that represents a `Date` value.

year()
Returns the year component of a Date

addDays(additionalDays)

Adds the specified number of additional days to a Date.

Signature

```
public Date addDays(Integer additionalDays)
```

Parameters

additionalDays
Type: Integer

Return Value

Type: Date

Example

```
Date myDate = Date.newInstance(1960, 2, 17);
Date newDate = mydate.addDays(2);
```

addMonths(additionalMonths)

Adds the specified number of additional months to a Date

Signature

```
public Date addMonths(Integer additionalMonths)
```

Parameters

additionalMonths
Type: Integer

Return Value

Type: Date

Example

```
date myDate = date.newInstance(1990, 11, 21);
date newDate = myDate.addMonths(3);
date expectedDate = date.newInstance(1991, 2, 21);
system.assertEquals(expectedDate, newDate);
```

addYears(additionalYears)

Adds the specified number of additional years to a Date

Signature

```
public Date addYears(Integer additionalYears)
```

Parameters

additionalYears
Type: Integer

Return Value

Type: Date

Example

```
date myDate = date.newInstance(1983, 7, 15);
date newDate = myDate.addYears(2);
date expectedDate = date.newInstance(1985, 7, 15);
system.assertEquals(expectedDate, newDate);
```

day()

Returns the day-of-month component of a Date.

Signature

```
public Integer day()
```

Return Value

Type: Integer

Example

```
date myDate = date.newInstance(1989, 4, 21);
Integer day = myDate.day();
system.assertEquals(21, day);
```

dayOfYear()

Returns the day-of-year component of a Date.

Signature

```
public Integer dayOfYear()
```

Return Value

Type: Integer

Example

```
date myDate = date.newInstance(1998, 10, 21);
Integer day = myDate.dayOfYear();
system.assertEquals(294, day);
```

daysBetween(secondDate)

Returns the number of days between the Date that called the method and the specified date.

Signature

```
public Integer daysBetween(Date secondDate)
```

Parameters

secondDate
Type: Date

Return Value

Type: Integer

Usage

If the Date that calls the method occurs after the *secondDate*, the return value is negative.

Example

```
Date startDate = Date.newInstance(2008, 1, 1);
Date dueDate = Date.newInstance(2008, 1, 30);
Integer numberDaysDue = startDate.daysBetween(dueDate);
```

daysInMonth(year, month)

Returns the number of days in the month for the specified *year* and *month* (1=Jan).

Signature

```
public static Integer daysInMonth(Integer year, Integer month)
```

Parameters

year
Type: Integer

month
Type: Integer

Return Value

Type: Integer

Example

The following example finds the number of days in the month of February in the year 1960.

```
Integer numberDays = date.daysInMonth(1960, 2);
```

format()

Returns the Date as a string using the locale of the context user

Signature

```
public String format()
```

Return Value

Type: String

Example

```
// In American-English locale
date myDate = date.newInstance(2001, 3, 21);
String dayString = myDate.format();
system.assertEquals('3/21/2001', dayString);
```

isLeapYear(year)

Returns `true` if the specified year is a leap year.

Signature

```
public static Boolean isLeapYear(Integer year)
```

Parameters

year
Type: Integer

Return Value

Type: Boolean

Example

```
system.assert(Date.isLeapYear(2004));
```

isSameDay(dateToCompare)

Returns `true` if the Date that called the method is the same as the specified date.

Signature

```
public Boolean isSameDay(Date dateToCompare)
```

Parameters

dateToCompare
Type: Date

Return Value

Type: Boolean

Example

```
date myDate = date.today();
date dueDate = date.newInstance(2008, 1, 30);
boolean dueNow = myDate.isSameDay(dueDate);
```

month()

Returns the month component of a Date (1=Jan).

Signature

```
public Integer month()
```

Return Value

Type: Integer

Example

```
date myDate = date.newInstance(2004, 11, 21);
Integer month = myDate.month();
system.assertEquals(11, month);
```

monthsBetween(secondDate)

Returns the number of months between the Date that called the method and the specified date, ignoring the difference in days.

Signature

```
public Integer monthsBetween(Date secondDate)
```

Parameters

secondDate
Type: Date

Return Value

Type: Integer

Example

```
Date firstDate = Date.newInstance(2006, 12, 2);
Date secondDate = Date.newInstance(2012, 12, 8);
Integer monthsBetween = firstDate.monthsBetween(secondDate);
System.assertEquals(72, monthsBetween);
```

newInstance(year, month, day)

Constructs a Date from Integer representations of the *year*, *month* (1=Jan), and *day*.

Signature

```
public static Date newInstance(Integer year, Integer month, Integer day)
```

Parameters

year
Type: Integer

month
Type: Integer

day
Type: Integer

Return Value

Type: Date

Example

The following example creates the date February 17th, 1960:

```
Date myDate = date.newinstance(1960, 2, 17);
```

parse(stringDate)

Constructs a Date from a String. The format of the String depends on the local date format.

Signature

```
public static Date parse(String stringDate)
```

Parameters

stringDate
Type: String

Return Value

Type: Date

Example

The following example works in some locales.

```
date mydate = date.parse('12/27/2009');
```

today()

Returns the current date in the current user's time zone.

Signature

```
public static Date today()
```

Return Value

Type: Date

toStartOfMonth()

Returns the first of the month for the Date that called the method.

Signature

```
public Date toStartOfMonth()
```

Return Value

Type: Date

Example

```
date myDate = date.newInstance(1987, 12, 17);
date firstDate = myDate.toStartOfMonth();
date expectedDate = date.newInstance(1987, 12, 1);
system.assertEquals(expectedDate, firstDate);
```

toStartOfWeek()

Returns the start of the week for the Date that called the method, depending on the context user's locale.

Signature

```
public Date toStartOfWeek()
```

Return Value

Type: Date

Example

For example, the start of a week is Sunday in the United States locale, and Monday in European locales. For example:

```
Date myDate = Date.today();
Date weekStart = myDate.toStartofWeek();
```

valueOf(stringDate)

Returns a Date that contains the value of the specified String.

Signature

```
public static Date valueOf(String stringDate)
```

Parameters

stringDate
Type: String

Return Value

Type: Date

Usage

The specified string should use the standard date format "yyyy-MM-dd HH:mm:ss" in the local time zone.

Example

```
string year = '2008';
string month = '10';
```

```
string day = '5';
string hour = '12';
string minute = '20';
string second = '20';
string stringDate = year + '-' + month
 + '-' + day + ' ' + hour + ':' +
minute + ':' + second;

Date myDate = date.valueOf(stringDate);
```

valueOf(fieldValue)

Converts the specified object to a `Date`. Use this method to convert a history tracking field value or an object that represents a `Date` value.

Signature

```
public static Date valueOf(Object fieldValue)
```

Parameters

`fieldValue`
Type: Object

Return Value

Type: Date

Usage

Use this method with the `OldValue` or `NewValue` fields of history sObjects, such as `AccountHistory`, when the field is a Date field.

Example

This example converts history tracking fields to `Date` values.

```
List<AccountHistory> ahlist = [SELECT Field,OldValue,NewValue FROM AccountHistory];
for(AccountHistory ah : ahlist) {
  System.debug('Field: ' + ah.Field);
  if (ah.field == 'MyDate__c') {
    Date oldValue = Date.valueOf(ah.OldValue);
    Date newValue = Date.valueOf(ah.NewValue);
  }
}
```

Versioned Behavior Changes

`Date.valueOf` has been versioned in these releases.

API version 33.0 or earlier

If you call `Date.valueOf` with a `Datetime` object, the method returns a `Date` value that contains the hours, minutes, seconds, and milliseconds set.

API version 34.0 to API version 53.0

If you call `Date.valueOf` with a `Datetime` object, the method converts `Datetime` to a valid `Date` without the time information, but the result depends on the manner in which the `Datetime` object was initialized. For example, if the `Datetime` object was initialized using `Datetime.valueOf(stringDate)`, the returned `Date` value contains time (hours) information. If the `Datetime` object is initialized using `Datetime.newInstance(year, month, day, hour, minute, second)` the returned `Date` value doesn't contain time information.

API version 54.0 and later

If you call `Date.valueOf` with a `Datetime` object, the method converts the object to a valid `Date` without the time information.

`year()`

Returns the year component of a Date

Signature

```
public Integer year()
```

Return Value

Type: Integer

Example

```
date myDate = date.newInstance(1988, 12, 17);
system.assertEquals(1988, myDate.year());
```

Datetime Class

Contains methods for the Datetime primitive data type.

Namespace

System

Usage

Apex supports both implicit and explicit casting of Date values to Datetime, with the time component being zeroed out in the resulting Datetime value. For more information about the Datetime, see Datetime Data Type.

Datetime Methods

The following are methods for `Datetime`.

IN THIS SECTION:

addDays(additionalDays)
Adds the specified number of days to a Datetime.

addHours(additionalHours)
Adds the specified number of hours to a Datetime.

addMinutes(additionalMinutes)
Adds the specified number of minutes to a Datetime.

addMonths(additionalMonths)
Adds the specified number of months to a Datetime.

addSeconds(additionalSeconds)
Adds the specified number of seconds to a Datetime.

addYears(additionalYears)
Adds the specified number of years to a Datetime.

date()
Returns the Date component of a Datetime in the local time zone of the context user.

dateGMT()
Return the Date component of a Datetime in the GMT time zone.

day()
Returns the day-of-month component of a Datetime in the local time zone of the context user.

dayGmt()
Returns the day-of-month component of a Datetime in the GMT time zone.

dayOfYear()
Returns the day-of-year component of a Datetime in the local time zone of the context user.

dayOfYearGmt()
Returns the day-of-year component of a Datetime in the GMT time zone.

format()
Converts the date to the local time zone and returns the converted date as a formatted string using the locale of the context user. If the time zone cannot be determined, GMT is used.

format(dateFormatString)
Converts the date to the local time zone and returns the converted date as a string using the supplied Java simple date format. If the time zone cannot be determined, GMT is used.

format(dateFormatString, timezone)
Converts the date to the specified time zone and returns the converted date as a string using the supplied Java simple date format. If the supplied time zone is not in the correct format, GMT is used.

formatGmt(dateFormatString)
Returns a Datetime as a string using the supplied Java simple date format and the GMT time zone.

formatLong()
Converts the date to the local time zone and returns the converted date in long date format.

getTime()
Returns the number of milliseconds since January 1, 1970, 00:00:00 GMT represented by this DateTime object.

hour()

Returns the hour component of a Datetime in the local time zone of the context user.

hourGmt()

Returns the hour component of a Datetime in the GMT time zone.

isSameDay(dateToCompare)

Returns true if the Datetime that called the method is the same as the specified Datetime in the local time zone of the context user.

millisecond()

Return the millisecond component of a Datetime in the local time zone of the context user.

millisecondGmt()

Return the millisecond component of a Datetime in the GMT time zone.

minute()

Returns the minute component of a Datetime in the local time zone of the context user.

minuteGmt()

Returns the minute component of a Datetime in the GMT time zone.

month()

Returns the month component of a Datetime in the local time zone of the context user (1=Jan).

monthGmt()

Returns the month component of a Datetime in the GMT time zone (1=Jan).

newInstance(milliseconds)

Constructs a Datetime and initializes it to represent the specified number of milliseconds since January 1, 1970, 00:00:00 GMT.

newInstance(date, time)

Constructs a DateTime from the specified date and time in the local time zone.

newInstance(year, month, day)

Constructs a Datetime from Integer representations of the specified year, month (1=Jan), and day at midnight in the local time zone.

newInstance(year, month, day, hour, minute, second)

Constructs a Datetime from Integer representations of the specified year, month (1=Jan), day, hour, minute, and second in the local time zone.

newInstanceGmt(date, time)

Constructs a DateTime from the specified date and time in the GMT time zone.

newInstanceGmt(year, month, date)

Constructs a Datetime from Integer representations of the specified year, month (1=Jan), and day at midnight in the GMT time zone

newInstanceGmt(year, month, date, hour, minute, second)

Constructs a Datetime from Integer representations of the specified year, month (1=Jan), day, hour, minute, and second in the GMT time zone

now()

Returns the current Datetime based on a GMT calendar.

parse(datetimeString)

Constructs a Datetime from the given String in the local time zone and in the format of the user locale.

second()

Returns the second component of a Datetime in the local time zone of the context user.

secondGmt()
Returns the second component of a Datetime in the GMT time zone.

time()
Returns the time component of a Datetime in the local time zone of the context user.

timeGmt()
Returns the time component of a Datetime in the GMT time zone.

valueOf(dateTimeString)
Returns a Datetime that contains the value of the specified string.

valueOf(fieldValue)
Converts the specified object to a Datetime. Use this method to convert a history tracking field value or an object that represents a Datetime value.

valueOfGmt(dateTimeString)
Returns a Datetime that contains the value of the specified String.

year()
Returns the year component of a Datetime in the local time zone of the context user.

yearGmt()
Returns the year component of a Datetime in the GMT time zone.

addDays(additionalDays)

Adds the specified number of days to a Datetime.

Signature

```
public Datetime addDays(Integer additionalDays)
```

Parameters

additionalDays
Type: Integer

Return Value

Type: Datetime

Example

```
Datetime myDateTime = Datetime.newInstance(1960, 2, 17);
Datetime newDateTime = myDateTime.addDays(2);
Datetime expected = Datetime.newInstance(1960, 2, 19);
System.assertEquals(expected, newDateTime);
```

addHours(additionalHours)

Adds the specified number of hours to a Datetime.

Signature

```
public Datetime addHours(Integer additionalHours)
```

Parameters

additionalHours
Type: Integer

Return Value

Type: Datetime

Example

```
DateTime myDateTime = DateTime.newInstance(1997, 1, 31, 7, 8, 16);
DateTime newDateTime = myDateTime.addHours(3);
DateTime expected = DateTime.newInstance(1997, 1, 31, 10, 8, 16);
System.assertEquals(expected, newDateTime);
```

addMinutes(additionalMinutes)

Adds the specified number of minutes to a Datetime.

Signature

```
public Datetime addMinutes(Integer additionalMinutes)
```

Parameters

additionalMinutes
Type: Integer

Return Value

Type: Datetime

Example

```
DateTime myDateTime = DateTime.newInstance(1999, 2, 11, 8, 6, 16);
DateTime newDateTime = myDateTime.addMinutes(7);
DateTime expected = DateTime.newInstance(1999, 2, 11, 8, 13, 16);
System.assertEquals(expected, newDateTime);
```

addMonths(additionalMonths)

Adds the specified number of months to a Datetime.

Signature

```
public Datetime addMonths(Integer additionalMonths)
```

Parameters

additionalMonths
Type: Integer

Return Value

Type: Datetime

Example

```
DateTime myDateTime = DateTime.newInstance(2000, 7, 7, 7, 8, 12);
DateTime newDateTime = myDateTime.addMonths(1);
DateTime expected = DateTime.newInstance(2000, 8, 7, 7, 8, 12);
System.assertEquals(expected, newDateTime);
```

addSeconds(additionalSeconds)

Adds the specified number of seconds to a Datetime.

Signature

```
public Datetime addSeconds(Integer additionalSeconds)
```

Parameters

additionalSeconds
Type: Integer

Return Value

Type: Datetime

Example

```
DateTime myDateTime = DateTime.newInstance(2001, 7, 19, 10, 7, 12);
DateTime newDateTime = myDateTime.addSeconds(4);
DateTime expected = DateTime.newInstance(2001, 7, 19, 10, 7, 16);
System.assertEquals(expected, newDateTime);
```

addYears(additionalYears)

Adds the specified number of years to a Datetime.

Signature

```
public Datetime addYears(Integer additionalYears)
```

Parameters

additionalYears
Type: Integer

Return Value

Type: Datetime

Example

```
DateTime myDateTime = DateTime.newInstance(2009, 12, 17, 13, 6, 6);
DateTime newDateTime = myDateTime.addYears(1);
DateTime expected = DateTime.newInstance(2010, 12, 17, 13, 6, 6);
System.assertEquals(expected, newDateTime);
```

date()

Returns the Date component of a Datetime in the local time zone of the context user.

Signature

```
public Date date()
```

Return Value

Type: Date

Example

```
DateTime myDateTime = DateTime.newInstance(2006, 3, 16, 12, 6, 13);
Date myDate = myDateTime.date();
Date expected = Date.newInstance(2006, 3, 16);
System.assertEquals(expected, myDate);
```

dateGMT()

Return the Date component of a Datetime in the GMT time zone.

Signature

```
public Date dateGMT()
```

Return Value

Type: Date

Example

```
// California local time, PST
DateTime myDateTime = DateTime.newInstance(2006, 3, 16, 23, 0, 0);
```

```
Date myDate = myDateTime.dateGMT();
Date expected = Date.newInstance(2006, 3, 17);
System.assertEquals(expected, myDate);
```

day()

Returns the day-of-month component of a Datetime in the local time zone of the context user.

Signature

```
public Integer day()
```

Return Value

Type: Integer

Example

```
DateTime myDateTime = DateTime.newInstance(1986, 2, 21, 23, 0, 0);
System.assertEquals(21, myDateTime.day());
```

dayGmt()

Returns the day-of-month component of a Datetime in the GMT time zone.

Signature

```
public Integer dayGmt()
```

Return Value

Type: Integer

Example

```
// California local time, PST
DateTime myDateTime = DateTime.newInstance(1987, 1, 14, 23, 0, 3);
System.assertEquals(15, myDateTime.dayGMT());
```

dayOfYear()

Returns the day-of-year component of a Datetime in the local time zone of the context user.

Signature

```
public Integer dayOfYear()
```

Return Value

Type: Integer

Example

For example, February 5, 2008 08:30:12 would be day 36.

```
Datetime myDate = Datetime.newInstance(2008, 2, 5, 8, 30, 12);
system.assertEquals(myDate.dayOfYear(), 36);
```

dayOfYearGmt()

Returns the day-of-year component of a Datetime in the GMT time zone.

Signature

```
public Integer dayOfYearGmt()
```

Return Value

Type: Integer

Example

```
// This sample assumes we are in the PST timezone
DateTime myDateTime = DateTime.newInstance(1999, 2, 5, 23, 0, 3);
// January has 31 days + 5 days in February = 36 days
// dayOfYearGmt() adjusts the time zone from the current time zone to GMT
// by adding 8 hours to the PST time zone, so it's 37 days and not 36 days
System.assertEquals(37, myDateTime.dayOfYearGmt());
```

format()

Converts the date to the local time zone and returns the converted date as a formatted string using the locale of the context user. If the time zone cannot be determined, GMT is used.

Signature

```
public String format()
```

Return Value

Type: String

Example

Note: The sample is executed in an org where the "Enable ICU Locale Formats" crucial update is enabled. See https://releasenotes.docs.salesforce.com/en-us/spring20/release-notes/rn_forcecom_globalization_enable_icu_cruc.htm.

```
DateTime.myDateTime = DateTime.newInstance(1993, 6, 6, 3, 3, 3);
system.assertEquals('6/6/1993, 3:03 AM', mydatetime.format());
```

format(dateFormatString)

Converts the date to the local time zone and returns the converted date as a string using the supplied Java simple date format. If the time zone cannot be determined, GMT is used.

Signature

```
public String format(String dateFormatString)
```

Parameters

dateFormatString
Type: String

Return Value

Type: String

Usage

For more information on the Java simple date format, see Java SimpleDateFormat.

Example

```
Datetime myDT = DateTime.newInstance(2022, 5, 4, 19, 37, 55);
String myDate = myDT.format('yyyy-MM-dd h:mm a');
String expected = '2022-05-04 7:37 PM';
System.assertEquals(expected, myDate);
```

format(dateFormatString, timezone)

Converts the date to the specified time zone and returns the converted date as a string using the supplied Java simple date format. If the supplied time zone is not in the correct format, GMT is used.

Signature

```
public String format(String dateFormatString, String timezone)
```

Parameters

dateFormatString
Type: String

timezone
Type: String

Valid time zone values for the *timezone* argument are the time zones of the Java TimeZone class that correspond to the time zones returned by the TimeZone.getAvailableIDs method in Java. We recommend you use full time zone names, not the three-letter abbreviations.

Return Value

Type: String

Usage

For more information on the Java simple date format, see Java SimpleDateFormat.

Example

This example uses `format` to convert a GMT date to the America/New_York time zone and formats the date using the specified date format.

```
Datetime GMTDate =
  Datetime.newInstanceGmt(2011,6,1,12,1,5);
String strConvertedDate =
  GMTDate.format('MM/dd/yyyy HH:mm:ss',
                 'America/New_York');
// Date is converted to
// the new time zone and is adjusted
// for daylight saving time.
System.assertEquals(
  '06/01/2011 08:01:05', strConvertedDate);
```

formatGmt(dateFormatString)

Returns a Datetime as a string using the supplied Java simple date format and the GMT time zone.

Signature

```
public String formatGmt(String dateFormatString)
```

Parameters

dateFormatString
Type: String

Return Value

Type: String

Usage

For more information on the Java simple date format, see Java SimpleDateFormat.

Example

```
DateTime myDateTime = DateTime.newInstance(1993, 6, 6, 3, 3, 3);
String formatted = myDateTime.formatGMT('EEE, MMM d yyyy HH:mm:ss');
String expected = 'Sun, Jun 6 1993 10:03:03';
System.assertEquals(expected, formatted);
```

formatLong()

Converts the date to the local time zone and returns the converted date in long date format.

Signature

```
public String formatLong()
```

Return Value

Type: String

Example

```
// Passing local date based on the PST time zone
Datetime dt = DateTime.newInstance(2012,12,28,10,0,0);
// Writes 12/28/2012 10:00:00 AM PST
System.debug('dt.formatLong()=' + dt.formatLong());
```

getTime()

Returns the number of milliseconds since January 1, 1970, 00:00:00 GMT represented by this DateTime object.

Signature

```
public Long getTime()
```

Return Value

Type: Long

Example

```
DateTime dt = DateTime.newInstanceGMT(2007, 6, 23, 3, 3, 3);
Long gettime = dt.getTime();
Long expected = 1182567783000L;
System.assertEquals(expected, gettime);
```

hour()

Returns the hour component of a Datetime in the local time zone of the context user.

Signature

```
public Integer hour()
```

Return Value

Type: Integer

Example

```
DateTime myDateTime = DateTime.newInstance(1998, 11, 21, 3, 3, 3);
System.assertEquals(3 , myDateTime.hour());
```

`hourGmt()`

Returns the hour component of a Datetime in the GMT time zone.

Signature

```
public Integer hourGmt()
```

Return Value

Type: Integer

Example

```
// California local time
DateTime myDateTime = DateTime.newInstance(2000, 4, 27, 3, 3, 3);
System.assertEquals(10 , myDateTime.hourGMT());
```

`isSameDay(dateToCompare)`

Returns true if the Datetime that called the method is the same as the specified Datetime in the local time zone of the context user.

Signature

```
public Boolean isSameDay(Datetime dateToCompare)
```

Parameters

dateToCompare
Type: Datetime

Return Value

Type: Boolean

Example

```
datetime myDate = datetime.now();
datetime dueDate =
     datetime.newInstance(2008, 1, 30);
boolean dueNow = myDate.isSameDay(dueDate);
```

`millisecond()`

Return the millisecond component of a Datetime in the local time zone of the context user.

Signature

```
public Integer millisecond()
```

Return Value

Type: Integer

Example

```
DateTime myDateTime = DateTime.now();
system.debug(myDateTime.millisecond());
```

millisecondGmt()

Return the millisecond component of a Datetime in the GMT time zone.

Signature

```
public Integer millisecondGmt()
```

Return Value

Type: Integer

Example

```
DateTime myDateTime = DateTime.now();
system.debug(myDateTime.millisecondGMT());
```

minute()

Returns the minute component of a Datetime in the local time zone of the context user.

Signature

```
public Integer minute()
```

Return Value

Type: Integer

Example

```
DateTime myDateTime = DateTime.newInstance(2001, 2, 27, 3, 3, 3);
system.assertEquals(3, myDateTime.minute());
```

minuteGmt()

Returns the minute component of a Datetime in the GMT time zone.

Signature

```
public Integer minuteGmt()
```

Return Value

Type: Integer

Example

```
DateTime myDateTime = DateTime.newInstance(2002, 12, 3, 3, 3, 3);
system.assertEquals(3, myDateTime.minuteGMT());
```

`month()`

Returns the month component of a Datetime in the local time zone of the context user (1=Jan).

Signature

```
public Integer month()
```

Return Value

Type: Integer

Example

```
DateTime myDateTime = DateTime.newInstance(2004, 11, 4, 3, 3, 3);
system.assertEquals(11, myDateTime.month());
```

`monthGmt()`

Returns the month component of a Datetime in the GMT time zone (1=Jan).

Signature

```
public Integer monthGmt()
```

Return Value

Type: Integer

Example

```
DateTime myDateTime = DateTime.newInstance(2006, 11, 19, 3, 3, 3);
system.assertEquals(11, myDateTime.monthGMT());
```

`newInstance(milliseconds)`

Constructs a Datetime and initializes it to represent the specified number of milliseconds since January 1, 1970, 00:00:00 GMT.

Signature

```
public static Datetime newInstance(Long milliseconds)
```

Parameters

milliseconds
Type: Long

Return Value

Type: Datetime

The returned date is in the GMT time zone.

Example

```
Long longtime = 1341828183000L;
DateTime dt = DateTime.newInstance(longtime);
DateTime expected = DateTime.newInstance(2012, 7, 09, 3, 3, 3);
System.assertEquals(expected, dt);
```

newInstance(date, time)

Constructs a DateTime from the specified date and time in the local time zone.

Signature

```
public static Datetime newInstance(Date date, Time time)
```

Parameters

date
Type: Date

time
Type: Time

Return Value

Type: Datetime

The returned date is in the GMT time zone.

Example

```
Date myDate = Date.newInstance(2011, 11, 18);
Time myTime = Time.newInstance(3, 3, 3, 0);
DateTime dt = DateTime.newInstance(myDate, myTime);
DateTime expected = DateTime.newInstance(2011, 11, 18, 3, 3, 3);
System.assertEquals(expected, dt);
```

newInstance(year, month, day)

Constructs a Datetime from Integer representations of the specified year, month (1=Jan), and day at midnight in the local time zone.

Signature

```
public static Datetime newInstance(Integer year, Integer month, Integer day)
```

Parameters

year
Type: Integer

month
Type: Integer

day
Type: Integer

Return Value

Type: Datetime

The returned date is in the GMT time zone.

Example

```
datetime myDate = datetime.newInstance(2008, 12, 1);
```

newInstance(year, month, day, hour, minute, second)

Constructs a Datetime from Integer representations of the specified year, month (1=Jan), day, hour, minute, and second in the local time zone.

Signature

```
public static Datetime newInstance(Integer year, Integer month, Integer day, Integer
hour, Integer minute, Integer second)
```

Parameters

year
Type: Integer

month
Type: Integer

day
Type: Integer

hour
Type: Integer

minute
Type: Integer

second
Type: Integer

Return Value

Type: Datetime

The returned date is in the GMT time zone.

Example

```
Datetime myDate = Datetime.newInstance(2008, 12, 1, 12, 30, 2);
```

newInstanceGmt(date, time)

Constructs a DateTime from the specified date and time in the GMT time zone.

Signature

```
public static Datetime newInstanceGmt(Date date, Time time)
```

Parameters

date
Type: Date

time
Type: Time

Return Value

Type: Datetime

Example

```
Date myDate = Date.newInstance(2013, 11, 12);
Time myTime = Time.newInstance(3, 3, 3, 0);
DateTime dt = DateTime.newInstanceGMT(myDate, myTime);
DateTime expected = DateTime.newInstanceGMT(2013, 11, 12, 3, 3, 3);
System.assertEquals(expected, dt);
```

newInstanceGmt(year, month, date)

Constructs a Datetime from Integer representations of the specified year, month (1=Jan), and day at midnight in the GMT time zone

Signature

```
public static Datetime newInstanceGmt(Integer year, Integer month, Integer date)
```

Parameters

year
Type: Integer

month
Type: Integer

date
Type: Integer

Return Value

Type: Datetime

Example

```
DateTime dt = DateTime.newInstanceGMT(1996, 3, 22);
```

newInstanceGmt(year, month, date, hour, minute, second)

Constructs a Datetime from Integer representations of the specified year, month (1=Jan), day, hour, minute, and second in the GMT time zone

Signature

```
public static Datetime newInstanceGmt(Integer year, Integer month, Integer date, Integer hour, Integer minute, Integer second)
```

Parameters

year
Type: Integer

month
Type: Integer

date
Type: Integer

hour
Type: Integer

minute
Type: Integer

second
Type: Integer

Return Value

Type: Datetime

Example

```
//California local time
DateTime dt = DateTime.newInstanceGMT(1998, 1, 29, 2, 2, 3);
DateTime expected = DateTime.newInstance(1998, 1, 28, 18, 2, 3);
System.assertEquals(expected, dt);
```

now()

Returns the current Datetime based on a GMT calendar.

Signature

```
public static Datetime now()
```

Return Value

Type: Datetime

The format of the returned datetime is: `'MM/DD/YYYY HH:MM PERIOD'`

Example

```
datetime myDateTime = datetime.now();
```

parse(datetimeString)

Constructs a Datetime from the given String in the local time zone and in the format of the user locale.

Signature

```
public static Datetime parse(String datetimeString)
```

Parameters

datetimeString
Type: String

Return Value

Type: Datetime

The returned date is in the GMT time zone.

Example

This example uses `parse` to create a Datetime from a date passed in as a string and that is formatted for the English (United States) locale. You may need to change the format of the date string if you have a different locale.

Note: This sample is executed in an org where the "Enable ICU Locale Formats" crucial update is enabled. See https://releasenotes.docs.salesforce.com/en-us/spring20/release-notes/rn_forcecom_globalization_enable_icu_cruc.htm.

```
Datetime dt = DateTime.parse('10/14/2011, 11:46 AM');
String myDtString = dt.format();
system.assertEquals(myDtString, '10/14/2011, 11:46 AM');
```

second()

Returns the second component of a Datetime in the local time zone of the context user.

Signature

```
public Integer second()
```

Return Value

Type: Integer

Example

```
DateTime dt = DateTime.newInstanceGMT(1999, 9, 22, 3, 1, 2);
System.assertEquals(2, dt.second());
```

secondGmt()

Returns the second component of a Datetime in the GMT time zone.

Signature

```
public Integer secondGmt()
```

Return Value

Type: Integer

Example

```
DateTime dt = DateTime.newInstance(2000, 2, 3, 3, 1, 5);
System.assertEquals(5, dt.secondGMT());
```

time()

Returns the time component of a Datetime in the local time zone of the context user.

Signature

```
public Time time()
```

Return Value

Type: Time

Example

```
DateTime dt = DateTime.newInstance(2002, 11, 21, 0, 2, 2);
Time expected = Time.newInstance(0, 2, 2, 0);
System.assertEquals(expected, dt.time());
```

`timeGmt()`

Returns the time component of a Datetime in the GMT time zone.

Signature

```
public Time timeGmt()
```

Return Value

Type: Time

Example

```
// This sample is based on the PST time zone
DateTime dt = DateTime.newInstance(2004, 1, 27, 4, 1, 2);
Time expected = Time.newInstance(12, 1, 2, 0);
// 8 hours are added to the time to convert it from
// PST to GMT
System.assertEquals(expected, dt.timeGMT());
```

`valueOf(dateTimeString)`

Returns a Datetime that contains the value of the specified string.

Signature

```
public static Datetime valueOf(String dateTimeString)
```

Parameters

`dateTimeString`
Type: String

Return Value

Type: Datetime

The returned date is in the GMT time zone.

Usage

The specified string should use the standard date format "yyyy-MM-dd HH:mm:ss" in the local time zone.

Example

```
string year = '2008';
string month = '10';
string day = '5';
string hour = '12';
string minute = '20';
string second = '20';
string stringDate = year + '-' + month + '-' + day + ' ' + hour + ':'
    + minute +  ':' + second;

Datetime myDate = Datetime.valueOf(stringDate);
```

valueOf(fieldValue)

Converts the specified object to a Datetime. Use this method to convert a history tracking field value or an object that represents a Datetime value.

Signature

```
public static Datetime valueOf(Object fieldValue)
```

Parameters

fieldValue
Type: Object

Return Value

Type: Datetime

Usage

Use this method with the `OldValue` or `NewValue` fields of history sObjects, such as `AccountHistory`, when the field is a Date/Time field.

Example

```
List<AccountHistory> ahlist = [SELECT Field,OldValue,NewValue FROM AccountHistory];
for(AccountHistory ah : ahlist) {
  System.debug('Field: ' + ah.Field);
  if (ah.field == 'MyDatetime__c') {
    Datetime oldValue = Datetime.valueOf(ah.OldValue);
    Datetime newValue = Datetime.valueOf(ah.NewValue);
  }
}
```

valueOfGmt(dateTimeString)

Returns a Datetime that contains the value of the specified String.

Signature

```
public static Datetime valueOfGmt(String dateTimeString)
```

Parameters

dateTimeString
Type: String

Return Value

Type: Datetime

Usage

The specified string should use the standard date format "yyyy-MM-dd HH:mm:ss" in the GMT time zone.

Example

```
// California locale time
string year = '2009';
string month = '3';
string day = '5';
string hour = '5';
string minute = '2';
string second = '2';
string stringDate = year + '-' + month + '-' + day + ' ' + hour + ':'
    + minute +  ':' + second;

Datetime myDate = Datetime.valueOfGMT(stringDate);

DateTime expected = DateTime.newInstance(2009, 3, 4, 21, 2, 2);
System.assertEquals(expected, myDate);
```

year()

Returns the year component of a Datetime in the local time zone of the context user.

Signature

```
public Integer year()
```

Return Value

Type: Integer

Example

```
DateTime dt = DateTime.newInstance(2012, 1, 26, 5, 2, 4);
System.assertEquals(2012, dt.year());
```

`yearGmt()`

Returns the year component of a Datetime in the GMT time zone.

Signature

```
public Integer yearGmt()
```

Return Value

Type: Integer

Example

```
DateTime dt = DateTime.newInstance(2012, 10, 4, 6, 4, 6);
System.assertEquals(2012, dt.yearGMT());
```

Decimal Class

Contains methods for the Decimal primitive data type.

Namespace

System

Usage

Note: Two Decimal objects that are numerically equivalent but differ in scale (such as 1.1 and 1.10) generally do not have the same hashcode. Use caution when such Decimal objects are used in Sets or as Map keys.

For more information on Decimal, see Decimal Data Type.

IN THIS SECTION:

Rounding Mode
Rounding mode specifies the rounding behavior for numerical operations capable of discarding precision.

Decimal Methods

Rounding Mode

Rounding mode specifies the rounding behavior for numerical operations capable of discarding precision.

Each rounding mode indicates how the least significant returned digit of a rounded result is to be calculated. The following are the valid values for *roundingMode*.

Name	Description

CEILING	Rounds towards positive infinity. That is, if the result is positive, this mode behaves the same as the UP rounding mode; if the result is negative, it behaves the same as the DOWN rounding mode. Note that this rounding mode never decreases the calculated value. For example:
	• Input number 5.5: CEILING round mode result: 6
	• Input number 1.1: CEILING round mode result: 2
	• Input number -1.1: CEILING round mode result: -1
	• Input number -2.7: CEILING round mode result: -2
	`Decimal[] example = new Decimal[]{5.5, 1.1, -1.1, -2.7};`
	`Long[] expected = new Long[]{6, 2, -1, -2};`
	`for(integer x = 0; x < example.size(); x++){`
	`  System.assertEquals(expected[x],`
	`  example[x].round(System.RoundingMode.CEILING));`
	`}`

DOWN	Rounds towards zero. This rounding mode always discards any fractions (decimal points) prior to executing. Note that this rounding mode never increases the magnitude of the calculated value. For example:
	• Input number 5.5: DOWN round mode result: 5
	• Input number 1.1: DOWN round mode result: 1
	• Input number -1.1: DOWN round mode result: -1
	• Input number -2.7: DOWN round mode result: -2
	`Decimal[] example = new Decimal[]{5.5, 1.1, -1.1, -2.7};`
	`Long[] expected = new Long[]{5, 1, -1, -2};`
	`for(integer x = 0; x < example.size(); x++){`
	`  System.assertEquals(expected[x],`
	`  example[x].round(System.RoundingMode.DOWN));`
	`}`

FLOOR	Rounds towards negative infinity. That is, if the result is positive, this mode behaves the same as theDOWN rounding mode; if negative, this mode behaves the same as the UP rounding mode. Note that this rounding mode never increases the calculated value. For example:
	• Input number 5.5: FLOOR round mode result: 5
	• Input number 1.1: FLOOR round mode result: 1
	• Input number -1.1: FLOOR round mode result: -2
	• Input number -2.7: FLOOR round mode result: -3
	`Decimal[] example = new Decimal[]{5.5, 1.1, -1.1, -2.7};`
	`Long[] expected = new Long[]{5, 1, -2, -3};`
	`for(integer x = 0; x < example.size(); x++){`
	`  System.assertEquals(expected[x],`
	`  example[x].round(System.RoundingMode.FLOOR));`
	`}`

Name	Description

HALF_DOWN	Rounds towards the "nearest neighbor" unless both neighbors are equidistant, in which case this mode rounds down. This rounding mode behaves the same as the UP rounding mode if the discarded fraction (decimal point) is > 0.5; otherwise, it behaves the same as DOWN rounding mode. For example:
	• Input number 5.5: HALF_DOWN round mode result: 5
	• Input number 1.1: HALF_DOWN round mode result: 1
	• Input number -1.1: HALF_DOWN round mode result: -1
	• Input number -2.7: HALF_DOWN round mode result: -3
	`Decimal[] example = new Decimal[]{5.5, 1.1, -1.1, -2.7};`
	`Long[] expected = new Long[]{5, 1, -1, -3};`
	`for(integer x = 0; x < example.size(); x++){`
	`System.assertEquals(expected[x],`
	`example[x].round(System.RoundingMode.HALF_DOWN));`
	`}`

HALF_EVEN	Rounds towards the "nearest neighbor" unless both neighbors are equidistant, in which case, this mode rounds towards the even neighbor. This rounding mode behaves the same as the HALF_UP rounding mode if the digit to the left of the discarded fraction (decimal point) is odd. It behaves the same as the HALF_DOWN rounding method if it is even. For example:
	• Input number 5.5: HALF_EVEN round mode result: 6
	• Input number 1.1: HALF_EVEN round mode result: 1
	• Input number -1.1: HALF_EVEN round mode result: -1
	• Input number -2.7: HALF_EVEN round mode result: -3
	`Decimal[] example = new Decimal[]{5.5, 1.1, -1.1, -2.7};`
	`Long[] expected = new Long[]{6, 1, -1, -3};`
	`for(integer x = 0; x < example.size(); x++){`
	`System.assertEquals(expected[x],`
	`example[x].round(System.RoundingMode.HALF_EVEN));`
	`}`
	Note that this rounding mode statistically minimizes cumulative error when applied repeatedly over a sequence of calculations.

HALF_UP	Rounds towards the "nearest neighbor" unless both neighbors are equidistant, in which case, this mode rounds up. This rounding method behaves the same as the UP rounding method if the discarded fraction (decimal point) is >= 0.5; otherwise, this rounding method behaves the same as the DOWN rounding method. For example:
	• Input number 5.5: HALF_UP round mode result: 6
	• Input number 1.1: HALF_UP round mode result: 1
	• Input number -1.1: HALF_UP round mode result: -1
	• Input number -2.7: HALF_UP round mode result: -3
	`Decimal[] example = new Decimal[]{5.5, 1.1, -1.1, -2.7};`
	`Long[] expected = new Long[]{6, 1, -1, -3};`
	`for(integer x = 0; x < example.size(); x++){`

Name	Description

	``
	    System.assertEquals(expected[x],
	        example[x].round(System.RoundingMode.HALF_UP));
	}
	``

UNNECESSARY	Asserts that the requested operation has an exact result, which means that no rounding is necessary. If this rounding mode is specified on an operation that yields an inexact result, a MathException is thrown. For example:
	• Input number 5.5: UNNECESSARY round mode result: MathException
	• Input number 1.1: UNNECESSARY round mode result: MathException
	• Input number 1.0: UNNECESSARY round mode result: 1
	• Input number -1.0: UNNECESSARY round mode result: -1
	• Input number -2.2: UNNECESSARY round mode result: MathException
	``
	Decimal example1 = 5.5;
	Decimal example2 = 1.0;
	system.assertEquals(1,
	    example2.round(System.RoundingMode.UNNECESSARY));
	try{
	    example1.round(System.RoundingMode.UNNECESSARY);
	} catch(Exception E) {
	   system.assertEquals('System.MathException', E.getTypeName());
	}
	``

UP	Rounds away from zero. This rounding mode always truncates any fractions (decimal points) prior to executing. Note that this rounding mode never decreases the magnitude of the calculated value. For example:
	• Input number 5.5: UP round mode result: 6
	• Input number 1.1: UP round mode result: 2
	• Input number -1.1: UP round mode result: -2
	• Input number -2.7: UP round mode result: -3
	``
	Decimal[] example = new Decimal[]{5.5, 1.1, -1.1, -2.7};
	Long[] expected = new Long[]{6, 2, -2, -3};
	for(integer x = 0; x < example.size(); x++){
	    System.assertEquals(expected[x],
	        example[x].round(System.RoundingMode.UP));
	}
	``

Decimal Methods

The following are methods for `Decimal`.

IN THIS SECTION:

abs()

Returns the absolute value of the Decimal.

divide(divisor, scale)

Divides this Decimal by the specified divisor, and sets the scale, that is, the number of decimal places, of the result using the specified scale.

divide(divisor, scale, roundingMode)

Divides this Decimal by the specified divisor, sets the scale, that is, the number of decimal places, of the result using the specified scale, and if necessary, rounds the value using the rounding mode.

doubleValue()

Returns the Double value of this Decimal.

format()

Returns the String value of this Decimal using the locale of the context user.

intValue()

Returns the Integer value of this Decimal.

longValue()

Returns the Long value of this Decimal.

pow(exponent)

Returns the value of this decimal raised to the power of the specified exponent.

precision()

Returns the total number of digits for the Decimal.

round()

Returns the rounded approximation of this Decimal. The number is rounded to zero decimal places using half-even rounding mode, that is, it rounds towards the "nearest neighbor" unless both neighbors are equidistant, in which case, this mode rounds towards the even neighbor.

round(roundingMode)

Returns the rounded approximation of this Decimal. The number is rounded to zero decimal places using the rounding mode specified by the rounding mode.

scale()

Returns the scale of the Decimal, that is, the number of decimal places.

setScale(scale)

Returns the Decimal scaled to the specified number of decimal places, using half-even rounding, if necessary. Half-even rounding mode rounds toward the "nearest neighbor." If both neighbors are equidistant, the number is rounded toward the even neighbor.

setScale(scale, roundingMode)

Returns the Decimal scaled to the specified number of decimal places, using the specified rounding mode, if necessary.

stripTrailingZeros()

Returns the Decimal with any trailing zeros removed.

toPlainString()

Returns the String value of this Decimal, without using scientific notation.

valueOf(doubleToDecimal)

Returns a Decimal that contains the value of the specified Double.

valueOf(longToDecimal)

Returns a Decimal that contains the value of the specified Long.

valueOf(stringToDecimal)

Returns a Decimal that contains the value of the specified String. As in Java, the string is interpreted as representing a signed Decimal.

abs()

Returns the absolute value of the Decimal.

Signature

```
public Decimal abs()
```

Return Value

Type: Decimal

Example

```
Decimal myDecimal = -6.02214129;
System.assertEquals(6.02214129, myDecimal.abs());
```

divide(divisor, scale)

Divides this Decimal by the specified divisor, and sets the scale, that is, the number of decimal places, of the result using the specified scale.

Signature

```
public Decimal divide(Decimal divisor, Integer scale)
```

Parameters

divisor

Type: Decimal

scale

Type: Integer

Return Value

Type: Decimal

Example

```
Decimal decimalNumber = 19;
Decimal result = decimalNumber.divide(100, 3);
System.assertEquals(0.190, result);
```

divide(divisor, scale, roundingMode)

Divides this Decimal by the specified divisor, sets the scale, that is, the number of decimal places, of the result using the specified scale, and if necessary, rounds the value using the rounding mode.

Signature

```
public Decimal divide(Decimal divisor, Integer scale, System.RoundingMode roundingMode)
```

Parameters

divisor
Type: Decimal

scale
Type: Integer

roundingMode
Type: System.RoundingMode

Return Value

Type: Decimal

Example

```
Decimal myDecimal = 12.4567;
Decimal divDec = myDecimal.divide(7, 2, System.RoundingMode.UP);
System.assertEquals(divDec, 1.78);
```

doubleValue()

Returns the Double value of this Decimal.

Signature

```
public Double doubleValue()
```

Return Value

Type: Double

Example

```
Decimal myDecimal = 6.62606957;
Double value = myDecimal.doubleValue();
System.assertEquals(6.62606957, value);
```

format()

Returns the String value of this Decimal using the locale of the context user.

Signature

```
public String format()
```

Return Value

Type: String

Usage

Scientific notation will be used if an exponent is needed.

Example

```
// U.S. locale
Decimal myDecimal = 12345.6789;
system.assertEquals('12,345.679', myDecimal.format());
```

intValue()

Returns the Integer value of this Decimal.

Signature

```
public Integer intValue()
```

Return Value

Type: Integer

Example

```
Decimal myDecimal = 1.602176565;
system.assertEquals(1, myDecimal.intValue());
```

longValue()

Returns the Long value of this Decimal.

Signature

```
public Long longValue()
```

Return Value

Type: Long

Example

```
Decimal myDecimal = 376.730313461;
system.assertEquals(376, myDecimal.longValue());
```

pow(exponent)

Returns the value of this decimal raised to the power of the specified exponent.

Signature

```
public Decimal pow(Integer exponent)
```

Parameters

exponent
Type: Integer

The value of *exponent* must be between 0 and 32,767.

Return Value

Type: Decimal

Usage

If you use `MyDecimal.pow(0)`, 1 is returned.

The `Math.pow` method does accept negative values.

Example

```
Decimal myDecimal = 4.12;
Decimal powDec = myDecimal.pow(2);
System.assertEquals(powDec, 16.9744);
```

precision()

Returns the total number of digits for the Decimal.

Signature

```
public Integer precision()
```

Return Value

Type: Integer

Example

For example, if the Decimal value was 123.45, `precision` returns 5. If the Decimal value is 123.123, `precision` returns 6.

```
Decimal D1 = 123.45;
Integer precision1 = D1.precision();
system.assertEquals(precision1, 5);
Decimal D2 = 123.123;
Integer precision2 = D2.precision();
system.assertEquals(precision2, 6);
```

round()

Returns the rounded approximation of this Decimal. The number is rounded to zero decimal places using half-even rounding mode, that is, it rounds towards the "nearest neighbor" unless both neighbors are equidistant, in which case, this mode rounds towards the even neighbor.

Signature

```
public Long round()
```

Return Value

Type: Long

Usage

Note that this rounding mode statistically minimizes cumulative error when applied repeatedly over a sequence of calculations.

Example

```
Decimal D = 4.5;
Long L = D.round();
System.assertEquals(4, L);

Decimal D1 = 5.5;
Long L1 = D1.round();
System.assertEquals(6, L1);

Decimal D2 = 5.2;
Long L2 = D2.round();
System.assertEquals(5, L2);

Decimal D3 = -5.7;
Long L3 = D3.round();
System.assertEquals(-6, L3);
```

round(roundingMode)

Returns the rounded approximation of this Decimal. The number is rounded to zero decimal places using the rounding mode specified by the rounding mode.

Signature

```
public Long round(System.RoundingMode roundingMode)
```

Parameters

roundingMode
Type: System.RoundingMode

Return Value

Type: Long

scale()

Returns the scale of the Decimal, that is, the number of decimal places.

Signature

```
public Integer scale()
```

Return Value

Type: Integer

Example

```
Decimal myDecimal = 9.27400968;
system.assertEquals(8, myDecimal.scale());
```

setScale(scale)

Returns the Decimal scaled to the specified number of decimal places, using half-even rounding, if necessary. Half-even rounding mode rounds toward the "nearest neighbor." If both neighbors are equidistant, the number is rounded toward the even neighbor.

Signature

```
public Decimal setScale(Integer scale)
```

Parameters

scale

Type: Integer

The value of *scale* must be between –33 and 33. If the value of *scale* is negative, your unscaled value is multiplied by 10 to the power of the negation of *scale*. For example, after this operation, the value of *d* is `4*10^3`.

```
Decimal d = 4000;
d = d.setScale(-3);
```

Return Value

Type: Decimal

Usage

If you do not explicitly set the scale for a Decimal, the item from which the Decimal is created determines the scale.

- If the Decimal is created as part of a query, the scale is based on the scale of the field returned from the query.
- If the Decimal is created from a String, the scale is the number of characters after the decimal point of the String.

- If the Decimal is created from a non-decimal number, the number is first converted to a String. The scale is then set using the number of characters after the decimal point.

Example

```
Decimal myDecimal = 8.987551787;
Decimal setScaled = myDecimal.setscale(3);
System.assertEquals(8.988, setScaled);
```

setScale(scale, roundingMode)

Returns the Decimal scaled to the specified number of decimal places, using the specified rounding mode, if necessary.

Signature

```
public Decimal setScale(Integer scale, System.RoundingMode roundingMode)
```

Parameters

scale
Type: Integer

The value of *scale* must be between –33 and 33. If the value of *scale* is negative, your unscaled value is multiplied by 10 to the power of the negation of *scale*. For example, after this operation, the value of *d* is `4*10^3`.

```
Decimal d = 4000;
d = d.setScale(-3);
```

roundingMode
Type: System.RoundingMode

Return Value

Type: Decimal

Usage

If you do not explicitly set the scale for a Decimal, the item from which the Decimal is created determines the scale.

- If the Decimal is created as part of a query, the scale is based on the scale of the field returned from the query.
- If the Decimal is created from a String, the scale is the number of characters after the decimal point of the String.
- If the Decimal is created from a non-decimal number, the number is first converted to a String. The scale is then set using the number of characters after the decimal point.

stripTrailingZeros()

Returns the Decimal with any trailing zeros removed.

Signature

```
public Decimal stripTrailingZeros()
```

Return Value

Type: Decimal

Example

```
Decimal myDecimal = 1.10000;
Decimal stripped = myDecimal.stripTrailingZeros();
System.assertEquals(stripped, 1.1);
```

toPlainString()

Returns the String value of this Decimal, without using scientific notation.

Signature

```
public String toPlainString()
```

Return Value

Type: String

Example

```
Decimal myDecimal = 12345.6789;
System.assertEquals('12345.6789', myDecimal.toPlainString());
```

valueOf(doubleToDecimal)

Returns a Decimal that contains the value of the specified Double.

Signature

```
public static Decimal valueOf(Double doubleToDecimal)
```

Parameters

doubleToDecimal
Type: Double

Return Value

Type: Decimal

Example

```
Double myDouble = 2.718281828459045;
Decimal myDecimal = Decimal.valueOf(myDouble);
System.assertEquals(2.718281828459045, myDecimal);
```

valueOf(longToDecimal)

Returns a Decimal that contains the value of the specified Long.

Signature

```
public static Decimal valueOf(Long longToDecimal)
```

Parameters

longToDecimal
Type: Long

Return Value

Type: Decimal

Example

```
Long myLong = 299792458;
Decimal myDecimal = Decimal.valueOf(myLong);
System.assertEquals(299792458, myDecimal);
```

valueOf(stringToDecimal)

Returns a Decimal that contains the value of the specified String. As in Java, the string is interpreted as representing a signed Decimal.

Signature

```
public static Decimal valueOf(String stringToDecimal)
```

Parameters

stringToDecimal
Type: String

Return Value

Type: Decimal

Example

```
String temp = '12.4567';
Decimal myDecimal = Decimal.valueOf(temp);
```

Domain Class

Represents an existing domain hosted by Salesforce that serves the org or its content. Contains methods to obtain information about these domains, such as the domain type, My Domain name, and sandbox name.

Namespace

System

Usage

Use the Domain class to obtain information about the domains that Salesforce hosts for your org. This class only applies to domains hosted by Salesforce, and can't be used to generate a new domain.

Example

This code uses the System.DomainParser class to parse a hostname. It then gets the associated domain type.

```
System.Domain d = DomainParser.parse('mycompany.lightning.force.com');
String myDomainName = d.getMyDomainName();
System.DomainType domainType = d.getDomainType();
```

IN THIS SECTION:

Domain Methods

Domain Methods

The following are methods for `Domain`.

IN THIS SECTION:

getDomainType()
Returns the domain's type, such as `CONTENT_DOMAIN`, `EXPERIENCE_CLOUD_SITES_DOMAIN`, or `LIGHTNING_DOMAIN`.

getMyDomainName()
Returns the domain's My Domain name.

getPackageName()
For a domain that includes the package name, such as a Lightning Component domain or Visualforce page domain, returns the package name. For a domain that doesn't contain a package name, this method returns `null`.

getSandboxName()
For a sandbox org domain, returns the sandbox name. For a production org domain, returns `null`.

getSitesSubdomainName()
For a system-managed Experience Cloud site domain or Salesforce Site domain, returns the sites subdomain name. If enhanced domains are enabled, this method always returns `null`. When enhanced domains are enabled, the org's My Domain name is the subdomain for the system-managed domains for Experience Cloud sites and Salesforce Sites domains.

`getDomainType()`

Returns the domain's type, such as `CONTENT_DOMAIN`, `EXPERIENCE_CLOUD_SITES_DOMAIN`, or `LIGHTNING_DOMAIN`.

Signature

```
public System.DomainType getDomainType()
```

Return Value

Type: System.DomainType

`getMyDomainName()`

Returns the domain's My Domain name.

Signature

```
public String getMyDomainName()
```

Return Value

Type: String

`getPackageName()`

For a domain that includes the package name, such as a Lightning Component domain or Visualforce page domain, returns the package name. For a domain that doesn't contain a package name, this method returns `null`.

Signature

```
public String getPackageName()
```

Return Value

Type: String

`getSandboxName()`

For a sandbox org domain, returns the sandbox name. For a production org domain, returns `null`.

Signature

```
public String getSandboxName()
```

Return Value

Type: String

`getSitesSubdomainName()`

For a system-managed Experience Cloud site domain or Salesforce Site domain, returns the sites subdomain name. If enhanced domains are enabled, this method always returns `null`. When enhanced domains are enabled, the org's My Domain name is the subdomain for the system-managed domains for Experience Cloud sites and Salesforce Sites domains.

Signature

```
public String getSitesSubdomainName()
```

Return Value

Type: String

DomainCreator Class

Use the DomainCreator class to return a hostname specific to the org. For example, get the org's Visualforce hostname. Values are returned as a hostname, such as ***MyDomainName***`.lightning.force.com`.

Namespace

System

Examples

This example code fetches the org's My Domain login hostname and the Visualforce hostname for the `uat` package.

```
//Get the My Domain login hostname
String myDomainHostname = DomainCreator.getOrgMyDomainHostname();

//Get the Visualforce hostname
String vfHostname = DomainCreator.getVisualforceHostname('uat');
```

In this case, in a production org with a My Domain name of `mycompany`, `myDomainHostname` returns `mycompany.my.salesforce.com`. And in the same production org, `vfHostname` returns `mycompany--uat.vf.force.com`.

This example code creates a link to a Salesforce Account record. It gets the Lightning hostname associated with this org. It then gets the Account record ID and uses concatenation to build the link URL.

```
//Get the org's Lightning hostname
String myLightningHostname = DomainCreator.getLightningHostname();

//Get the ID of a record Account with the name 'Acme'
Account acct = [SELECT Id FROM Account WHERE Name = 'Acme' LIMIT 1];

//Build the URL to view the account record
String fullRecordURL = 'https://' + myLightningHostname + '/lightning/r/Account/' + acct.Id
 + '/view';
```

IN THIS SECTION:

DomainCreator Methods

DomainCreator Methods

The following are methods for `DomainCreator`.

IN THIS SECTION:

getContentHostname()
Returns the hostname for content stored in the org, such as files.

getExperienceCloudSitesBuilderHostname()
Returns the hostname to access Experience Builder for the org's Experience Cloud sites.

getExperienceCloudSitesHostname()
Returns the system-managed hostname for the org's Experience Cloud sites, such as ***ExperienceCloudSitesSubdomainName***`.force.com`. If Digital Experiences aren't enabled, this method throws an `InvalidParameterValueException`.

getExperienceCloudSitesLivePreviewHostname()
Returns the hostname to access Experience Builder Live Preview for the org's Experience Cloud sites.

getExperienceCloudSitesPreviewHostname()
Returns the hostname to access Experience Builder Preview for the org's Experience Cloud sites.

getLightningContainerComponentHostname(packageName)
Returns the hostname for the org's Lightning Container Components.

getLightningHostname()
Returns the hostname for the org's Lightning pages.

getOrgMyDomainHostname()
Returns the hostname for the org's My Domain login domain.

getSalesforceSitesHostname()
Returns the hostname for the org's Salesforce Sites. If Salesforce Sites aren't enabled, this method throws an `InvalidParameterValueException`.

getSetupHostname()
Returns the hostname for the org's setup domain, which hosts Setup pages in Salesforce.

getVisualforceHostname(packageName)
Returns the hostname for the org's Visualforce pages.

`getContentHostname()`

Returns the hostname for content stored in the org, such as files.

Signature

```
public static String getContentHostname()
```

Return Value

Type: String

`getExperienceCloudSitesBuilderHostname()`

Returns the hostname to access Experience Builder for the org's Experience Cloud sites.

Signature

```
public static String getExperienceCloudSitesBuilderHostname()
```

Return Value

Type: String

`getExperienceCloudSitesHostname()`

Returns the system-managed hostname for the org's Experience Cloud sites, such as ***`ExperienceCloudSitesSubdomainName`***`.force.com`. If Digital Experiences aren't enabled, this method throws an `InvalidParameterValueException`.

Signature

```
public static String getExperienceCloudSitesHostname()
```

Return Value

Type: String

`getExperienceCloudSitesLivePreviewHostname()`

Returns the hostname to access Experience Builder Live Preview for the org's Experience Cloud sites.

Signature

```
public static String getExperienceCloudSitesLivePreviewHostname()
```

Return Value

Type: String

`getExperienceCloudSitesPreviewHostname()`

Returns the hostname to access Experience Builder Preview for the org's Experience Cloud sites.

Signature

```
public static String getExperienceCloudSitesPreviewHostname()
```

Return Value

Type: String

`getLightningContainerComponentHostname(packageName)`

Returns the hostname for the org's Lightning Container Components.

Signature

```
public static String getLightningContainerComponentHostname(String packageName)
```

Parameters

packageName
Type: String

The package name for this component.

If packageName is `null`, this method uses the org's namespace prefix as the package name. Otherwise, it uses the default namespace.

Return Value

Type: String

getLightningHostname()

Returns the hostname for the org's Lightning pages.

Signature

```
public static String getLightningHostname()
```

Return Value

Type: String

getOrgMyDomainHostname()

Returns the hostname for the org's My Domain login domain.

Signature

```
public static String getOrgMyDomainHostname()
```

Return Value

Type: String

getSalesforceSitesHostname()

Returns the hostname for the org's Salesforce Sites. If Salesforce Sites aren't enabled, this method throws an `InvalidParameterValueException`.

Signature

```
public static String getSalesforceSitesHostname()
```

Return Value

Type: String

getSetupHostname()

Returns the hostname for the org's setup domain, which hosts Setup pages in Salesforce.

Signature

```
public static String getSetupHostname()
```

Return Value

Type: String

`getVisualforceHostname(packageName)`

Returns the hostname for the org's Visualforce pages.

Signature

```
public static String getVisualforceHostname(String packageName)
```

Parameters

`packageName`
Type: String

The package name for this component.

If packageName is `null`, this method uses the org's namespace prefix as the package name. Otherwise, it uses the default namespace.

Return Value

Type: String

DomainParser Class

Use the DomainParser class to parse a domain that Salesforce hosts for the org and extract information about the domain.

Namespace

System

Examples

This example code parses the org's Lightning domain and gets the My Domain name and domain type from the `System.Domain` object.

```
System.Domain d = DomainParser.parse('mycompany.lightning.force.com');
String myDomainName = d.getMyDomainName();
System.DomainType domainType = d.getDomainType();
```

This example code parses a known Visualforce URL to get the domain type, the org's My Domain name, and the package name.

```
//Parse a known URL
System.Domain domain = DomainParser.parse('https://mycompany--abcpackage.vf.force.com');

//Get the domain type
System.DomainType domainType = domain.getDomainType(); // Returns VISUALFORCE_DOMAIN
```

```
//Get the org's My Domain name
String myDomainName = domain.getMyDomainName(); // Returns mycompany

//Get the package name
String packageName = domain.getPackageName(); // Returns abcpackage
```

IN THIS SECTION:

DomainParser Methods

DomainParser Methods

The following are methods for `DomainParser`.

IN THIS SECTION:

parse(hostname)
Parses a passed hostname of a domain that Salesforce hosts for the org, and returns the System.Domain.

parse(url)
Parses a passed uniform resource locator (URL) of a domain that Salesforce hosts for the org, and returns the System.Domain.

`parse(hostname)`

Parses a passed hostname of a domain that Salesforce hosts for the org, and returns the System.Domain.

Signature

```
public static System.Domain parse(String hostname)
```

Parameters

`hostname`
Type: String

The label that identifies a Salesforce host, including all subdomains but without the protocol, path, or any parameters. For example, `mycompany.my.site.com` or `mycompany--sandbox1.sandbox.my.salesforceforce.com`.

If the hostname format is invalid, it isn't a Salesforce hosted domain, or it isn't owned by this org, this method throws an `InvalidParameterValueException`.

Return Value

Type: System.Domain

`parse(url)`

Parses a passed uniform resource locator (URL) of a domain that Salesforce hosts for the org, and returns the System.Domain.

Signature

```
public static System.Domain parse(System.Url url)
```

Parameters

url
Type: System.Url

A uniform resource locator (URL) for a Salesforce org, including all subdomains and the protocol. For example, `https://mycompany--sandbox1.sandbox.my.salesforceforce.com`.

The URL can also include paths and parameters. For example, `https://mycompany.my.site.com/en/us/help` or `https://mycompany.file.force.com/servlet/servlet.FileDownload?file=015300000000xvU`.

If the URL format is invalid, it isn't a Salesforce hosted domain, or it isn't owned by this org, this method throws an `InvalidParameterValueException`.

Return Value

Type: System.Domain

DomainType Enum

Specifies the domain type for a System.Domain.

Usage

Use the `DomainType` enum to obtain the type of a domain parsed through the System.DomainParser class.

Enum Values

The following are the values of the `System.DomainType` enum. These values only apply to Salesforce-hosted domains.

Value	Description
`CMS_DOMAIN`	Content Management System (CMS) public channel domains.
`CONTENT_DOMAIN`	Domains that serve content (files) stored in Salesforce.
`CUSTOMER_360_ADMIN_DOMAIN`	Customer 360 Data Manager domains.
`CUSTOMER_360_DOMAIN`	Customer 360 Data Manager Admin domains.
`EXPERIENCE_CLOUD_SITES_BUILDER_DOMAIN`	Experience Builder for Experience Cloud sites domains.
`EXPERIENCE_CLOUD_SITES_DOMAIN`	Salesforce-hosted domains that serve Experience Cloud sites.
`EXPERIENCE_CLOUD_SITES_LIVE_PREVIEW_DOMAIN`	Experience Builder Live Preview domains.
`EXPERIENCE_CLOUD_SITES_PREVIEW_DOMAIN`	Experience Builder Preview domains.
`LIGHTNING_CONTAINER_COMPONENT_DOMAIN`	Lightning Container Component domains.
`LIGHTNING_DOMAIN`	Domains that serve Lighting pages.

Value	Description
`ORG_MY_DOMAIN`	My Domain login domains.
`SALESFORCE_SITES_DOMAIN`	Salesforce-hosted domains that serve Salesforce Sites.
`SETUP_DOMAIN`	The Salesforce-hosted domain that serves Setup pages.
`VISUALFORCE_DOMAIN`	Domains that serve Visualforce pages.

Double Class

Contains methods for the Double primitive data type.

Namespace

System

Usage

For more information on Double, see Double Data Type.

Double Methods

The following are methods for `Double`.

IN THIS SECTION:

format()
Returns the String value for this Double using the locale of the context user

intValue()
Returns the Integer value of this Double by casting it to an Integer.

longValue()
Returns the Long value of this Double.

round()
Returns the closest Long to this Double value.

valueOf(stringToDouble)
Returns a Double that contains the value of the specified String. As in Java, the String is interpreted as representing a signed decimal.

valueOf(fieldValue)
Converts the specified object to a Double value. Use this method to convert a history tracking field value or an object that represents a Double value.

`format()`

Returns the String value for this Double using the locale of the context user

Signature

```
public String format()
```

Return Value

Type: String

Example

```
Double myDouble = 1261992;
system.assertEquals('1,261,992', myDouble.format());
```

intValue()

Returns the Integer value of this Double by casting it to an Integer.

Signature

```
public Integer intValue()
```

Return Value

Type: Integer

Example

```
Double DD1 = double.valueOf('3.14159');
Integer value = DD1.intValue();
system.assertEquals(value, 3);
```

longValue()

Returns the Long value of this Double.

Signature

```
public Long longValue()
```

Return Value

Type: Long

Example

```
Double myDouble = 421994;
Long value = myDouble.longValue();
System.assertEquals(421994, value);
```

round()

Returns the closest Long to this Double value.

Signature

```
public Long round()
```

Return Value

Type: Long

Example

```
Double D1 = 4.5;
Long L1 = D1.round();
System.assertEquals(5, L1);

Double D2= 4.2;
Long L2= D2.round();
System.assertEquals(4, L2);

Double D3= -4.7;
Long L3= D3.round();
System.assertEquals(-5, L3);
```

valueOf(stringToDouble)

Returns a Double that contains the value of the specified String. As in Java, the String is interpreted as representing a signed decimal.

Signature

```
public static Double valueOf(String stringToDouble)
```

Parameters

stringToDouble
Type: String

Return Value

Type: Double

Example

```
Double DD1 = double.valueOf('3.14159');
```

valueOf(fieldValue)

Converts the specified object to a Double value. Use this method to convert a history tracking field value or an object that represents a Double value.

Signature

```
public static Double valueOf(Object fieldValue)
```

Parameters

fieldValue
Type: Object

Return Value

Type: Double

Usage

Use this method with the `OldValue` or `NewValue` fields of history sObjects, such as `AccountHistory`, when the field type corresponds to a Double type, like a number field.

Example

```
List<AccountHistory> ahlist =
  [SELECT Field,OldValue,NewValue
   FROM AccountHistory];
for(AccountHistory ah : ahlist) {
  System.debug('Field: ' + ah.Field);
  if (ah.field == 'NumberOfEmployees') {
    Double oldValue =
      Double.valueOf(ah.OldValue);
    Double newValue =
      Double.valueOf(ah.NewValue);
}
```

EmailMessages Class

Use the methods in the `EmailMessages` class to interact with emails and email threading.

Namespace

System

EmailMessages Methods

The following are static methods for `EmailMessages`.

IN THIS SECTION:

getFormattedThreadingToken(recordId)
Returns an email threading token that's formatted with the correct prefix and suffix. This token can be embedded in an outbound email body, email subject, or both the body and subject. When users reply to the email, threading tokens can be used to attach responses to a record, such as a Case record in Email-to-Case.

getRecordIdFromEmail(subject, textBody, htmlBody)

Returns the record ID corresponding to the specified email threading token, or returns null if none is found.

getFormattedThreadingToken(recordId)

Returns an email threading token that's formatted with the correct prefix and suffix. This token can be embedded in an outbound email body, email subject, or both the body and subject. When users reply to the email, threading tokens can be used to attach responses to a record, such as a Case record in Email-to-Case.

Signature

```
public static Id getFormattedThreadingToken(Id recordId)
```

Parameters

recordId

Type:Id

The record ID associated with the threading token.

Return Value

Type: String

The returned value is a formatted string that includes a prefix and suffix, for example:

```
thread::pp5XPGfmNf2hRZdRCWnrohc::
```

Usage

When sending emails in Apex, use the returned string to match emails to a record, such as a Case record, that's associated with the email thread. Embed the formatted token in the body or subject of outgoing emails. To find the corresponding record ID in incoming emails, use EmailMessages.getRecordIdFromEmail(subject, textBody, htmlBody) on page 3488.

If there is no existing token, `getFormattedThreadingToken` may perform a Data Manipulation Language (DML) operation to generate one.

Example

In this sample, we send an email with a threading token so that the email and any responses are associated with the related case.

```
// Get your Record ID. Here, we're using a dummy Case ID.
ID caseId = Id.valueOf('500xx000000bpkTAAQ');

// Get the formatted threading token.
String formattedToken = EmailMessages.getFormattedThreadingToken(caseId);

// Create a SingleEmailMessage object.
Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();

// Set recipients and other fields.
email.setToAddresses(new String[] {'test@example.com'});

// Append the threading token to the email body (text or html), subject,
```

```
// or both body and subject.
email.setPlainTextBody('Test Email Notification text body' + '\n\n' + formattedToken);
email.setHtmlBody('Test Email Notification html body' + '<br><br>' + formattedToken);
email.setSubject('Test Notification ' + '[ ' + formattedToken + ' ]');
// ........... more fields ...........

// Send out the email.
Messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
```

getRecordIdFromEmail(subject, textBody, htmlBody)

Returns the record ID corresponding to the specified email threading token, or returns null if none is found.

Signature

```
public static Id getRecordIdFromEmail(String subject, String textBody, String htmlBody)
```

Parameters

subject
Type: String

The subject of the email.

textBody
Type: String

The body of the email in text format.

htmlBody
Type: String

The body of the email in HTML format.

Return Value

Type: Id

The record ID that corresponds to the embedded threading token.

Usage

When you send emails with threading tokens embedded in the email subject, the email body, or in both the subject and body, most email clients quote the email body and maintain the email subject in a response. This method finds a corresponding record that matches the embedded threading token in a response.

Typically this method is used in Email Services so that you can provide your own handling of inbound emails using Apex code.

Example

If you implement header-based threading in your Email Services currently, we recommend that you use Lightning threading, which combines token-based threading and header-based threading. For header-based threading to continue to work, store emails as EmailMessage records with the MessageIdentifier field set properly. With Lightning threading, you can use threading tokens as the primary threading method and rely on header-based threading as a fallback, or vice versa.

In this example, we rely on threading tokens and use header-based threading as a fallback.

```
global class AttachEmailMessageToCaseExample implements Messaging.InboundEmailHandler {
    global Messaging.InboundEmailResult handleInboundEmail(Messaging.inboundEmail email,

                Messaging.InboundEnvelope env) {

        // Create an InboundEmailResult object for returning the result of the
        // Apex Email Service.
        Messaging.InboundEmailResult result = new Messaging.InboundEmailResult();

        // Try to find the Case ID using threading tokens in email attributes.
       Id caseId = EmailMessages.getRecordIdFromEmail(email.subject, email.plainTextBody,
 email.htmlBody);

        // If we haven't found the Case ID, try finding it using headers.
        if (caseId == null) {
            caseId = Cases.getCaseIdFromEmailHeaders(email.headers);
        }

        // If a Case isn’t found, create a new Case record.
        if (caseId == null) {
            Case c = new Case(Subject = email.subject);
            insert c;
            System.debug('New Case Object: ' + c);
            caseId = c.Id;
        }

        // Process recipients
        String toAddresses;
        if (email.toAddresses != null) {
            toAddresses = String.join(email.toAddresses, '; ');
        }

        // To store an EmailMessage for threading, you need at minimum
        // the Status, the MessageIdentifier, and the ParentId fields.
        EmailMessage em = new EmailMessage(
            Status = '0',
            MessageIdentifier = email.messageId,
            ParentId = caseId,
            // Other important fields.
            FromAddress = email.fromAddress,
            FromName = email.fromName,
            ToAddress = toAddresses,
            TextBody = email.plainTextBody,
            HtmlBody = email.htmlBody,
            Subject = email.subject,
            // Parse thread-index header to remain consistent with Email-to-Case.
            ClientThreadIdentifier = getClientThreadIdentifier(email.headers)
            // Other fields you wish to add.
        );

        // Insert the new EmailMessage.
        insert em;
        System.debug('New EmailMessage Object: ' + em );
```

```
    // Set the result to true. No need to send an email back to the user
    // with an error message.
    result.success = true;

    // Return the result for the Apex Email Service.
    return result;
  }

  private String getClientThreadIdentifier(List<Messaging.InboundEmail.Header> headers) {

    if (headers == null || headers.size() == 0) return null;
    try {
        for (Messaging.InboundEmail.Header header : headers) {
            if (header.name.equalsIgnoreCase('thread-index')) {
                Blob threadIndex = EncodingUtil.base64Decode(header.value.trim());
              return EncodingUtil.convertToHex(threadIndex).substring(0, 44).toUpperCase();

            }
        }
    } catch (Exception e){
        return null;
    }
    return null;
  }
}
```

EncodingUtil Class

Use the methods in the `EncodingUtil` class to encode and decode URL strings, and convert strings to hexadecimal format.

Namespace

System

Usage

Note: You cannot use the EncodingUtil methods to move documents with non-ASCII characters to Salesforce. You can, however, download a document from Salesforce. To do so, query the ID of the document using the API `query` call, then request it by ID.

EncodingUtil Methods

The following are methods for `EncodingUtil`. All methods are static.

IN THIS SECTION:

base64Decode(inputString)
Converts a Base64-encoded String to a Blob representing its normal form.

base64Encode(inputBlob)
Converts a Blob to an unencoded String representing its normal form.

convertFromHex(inputString)

Converts the specified hexadecimal (base 16) string to a Blob value and returns this Blob value.

convertToHex(inputBlob)

Returns a hexadecimal (base 16) representation of the *inputBlob*. This method can be used to compute the client response (for example, HA1 or HA2) for HTTP Digest Authentication (RFC2617).

urlDecode(inputString, encodingScheme)

Decodes a string in `application/x-www-form-urlencoded` format using a specific encoding scheme, for example "UTF-8."

urlEncode(inputString, encodingScheme)

Encodes a string into the `application/x-www-form-urlencoded` format using a specific encoding scheme, for example "UTF-8."

base64Decode(inputString)

Converts a Base64-encoded String to a Blob representing its normal form.

Signature

```
public static Blob base64Decode(String inputString)
```

Parameters

inputString
Type: String

Return Value

Type: Blob

base64Encode(inputBlob)

Converts a Blob to an unencoded String representing its normal form.

Signature

```
public static String base64Encode(Blob inputBlob)
```

Parameters

inputBlob
Type: Blob

Return Value

Type: String

convertFromHex(inputString)

Converts the specified hexadecimal (base 16) string to a Blob value and returns this Blob value.

Signature

```
public static Blob convertFromHex(String inputString)
```

Parameters

inputString
Type: String

The hexadecimal string to convert. The string can contain only valid hexadecimal characters (0-9, a-f, A-F) and must have an even number of characters.

Return Value

Type: Blob

Usage

Each byte in the Blob is constructed from two hexadecimal characters in the input string.

The `convertFromHex` method throws the following exceptions.

- `NullPointerException` — the *inputString* is `null`.
- `InvalidParameterValueException` — the *inputString* contains invalid hexadecimal characters or doesn't contain an even number of characters.

Example

```
Blob blobValue = EncodingUtil.convertFromHex('4A4B4C');
System.assertEquals('JKL', blobValue.toString());
```

convertToHex(inputBlob)

Returns a hexadecimal (base 16) representation of the *inputBlob*. This method can be used to compute the client response (for example, HA1 or HA2) for HTTP Digest Authentication (RFC2617).

Signature

```
public static String convertToHex(Blob inputBlob)
```

Parameters

inputBlob
Type: Blob

Return Value

Type: String

urlDecode(inputString, encodingScheme)

Decodes a string in `application/x-www-form-urlencoded` format using a specific encoding scheme, for example "UTF-8."

Signature

```
public static String urlDecode(String inputString, String encodingScheme)
```

Parameters

inputString
Type: String

encodingScheme
Type: String

Return Value

Type: String

Usage

This method uses the supplied encoding scheme to determine which characters are represented by any consecutive sequence of the form `\"%xy\"`. For more information about the format, see The form-urlencoded Media Type in *Hypertext Markup Language - 2.0*.

urlEncode(inputString, encodingScheme)

Encodes a string into the `application/x-www-form-urlencoded` format using a specific encoding scheme, for example "UTF-8."

Signature

```
public static String urlEncode(String inputString, String encodingScheme)
```

Parameters

inputString
Type: String

encodingScheme
Type: String

Return Value

Type: String

Usage

The rules that apply when the method encodes a string:

- These characters remain the same:
  - Alphanumeric characters A - Z, a - z, and 0 -9.
  - Special characters dot (.), hyphen (-), asterisk (*), and under score (_).
- The space character is converted to a plus sign (+).
- All other characters are unsafe. This method uses the supplied encoding scheme to obtain the bytes for unsafe characters. For more information about the format, see The form-urlencoded Media Type in *Hypertext Markup Language - 2.0*.

Example

```
String encoded = EncodingUtil.urlEncode(url, 'UTF-8');
```

Enum Methods

An enum is an abstract data type with values that each take on exactly one of a finite set of identifiers that you specify. Apex provides built-in enums, such as `LoggingLevel`, and you can define your own enum.

All Apex enums, whether user-defined enums or built-in enums, have these common methods:

`values`
: This method returns the values of the Enum as a list of the same Enum type.

`valueOf(string enumStr)`
: This method converts a specified string to an enum constant value. An exception is thrown if the input string doesn't match an enum value.

Each Enum value has the following methods that take no arguments.

`name`
: Returns the name of the Enum item as a String.

`ordinal`
: Returns the position of the item, as an Integer, in the list of Enum values starting with zero.

Enum values cannot have user-defined methods added to them.

For more information about Enum, see Enums.

Example

```
Integer i = StatusCode.DELETE_FAILED.ordinal();

String s = StatusCode.DELETE_FAILED.name();

List<StatusCode> values = StatusCode.values();

StatusCode statusCodeValue = StatusCode.valueOf('delete_failed');
```

EventBus Class

Contains methods for publishing platform events.

Namespace

System

IN THIS SECTION:

EventBus Methods

SEE ALSO:

Platform Events Developer Guide: Publishing Platform Events

EventBus Methods

The following are methods for `EventBus`. All methods are static.

IN THIS SECTION:

getOperationId(result)
Returns the event UUID, which identifies a published event message.

publish(event)
Publishes the given platform event.

publish(events)
Publishes the given list of platform events.

publish(event, callback)
Publishes the given platform event using the specified callback. To track asynchronous publish failures, you can implement an Apex publish callback.

publish(events, callback)
Publishes the given list of platform events using the specified callback. To track asynchronous publish failures, you can implement an Apex publish callback.

`getOperationId(result)`

Returns the event UUID, which identifies a published event message.

Signature

```
public static String getOperationId(Object result)
```

Parameters

`result`
Type: Object

The SaveResult that is returned by the `EventBus.publish` call.

Return Value

Type: String

Usage

- If the event publish request fails to be enqueued in Salesforce, and `EventBus.publish` returns a synchronous error, `getOperationId` returns null. Also in this case, `getOperationId` returns null even when the event was created using the newSObject(recordTypeId, loadDefaults) method and contains a prepopulated UUID.

`publish(event)`

Publishes the given platform event.

Signature

```
public static Database.SaveResult publish(SObject event)
```

Parameters

`event`
Type: SObject

An instance of a platform event. For example, an instance of *`MyEvent__e`*. You must first define your platform event object in your org.

Return Value

Type: Database.SaveResult

The result of publishing the given event. `Database.SaveResult` contains information about whether the operation was successful and the errors encountered. If the `isSuccess()` method returns `true`, the publish request is queued in Salesforce and the event message is published asynchronously. For more information, see High-Volume Platform Event Persistence. If `isSuccess()` returns `false`, the event publish operation resulted in errors, which are returned in the `Database.Error` object. This method doesn't throw an exception due to an unsuccessful publish operation.

`Database.SaveResult` also contains the `Id` system field. The `Id` field value isn't included in the event message delivered to subscribers. It isn't used to identify an event message, and isn't always unique.

This method returns a `System.UnexpectedException` if you attempt to publish an `SObject` that represents an object that isn't a platform event.

Usage

- The platform event message is published either immediately or after a transaction is committed, depending on the publish behavior you set in the platform event definition. For more information, see Platform Event Fields in the *Platform Events Developer Guide*.
- Apex governor limits apply. For events configured with the **Publish After Commit** behavior, each method execution is counted as one DML statement against the Apex DML statement limit. You can check limit usage using the `Limits.getDMLStatements()` on page 3595 method. For events configured with the **Publish Immediately** behavior, each method execution is counted against a separate event publishing limit of 150 `EventBus.publish()` calls. You can check limit usage using the `Limits.getPublishImmediateDML()` on page 3598 method.

`publish(events)`

Publishes the given list of platform events.

Signature

```
public static List<Database.SaveResult> publish(List<SObject> events)
```

Parameters

events
Type: List<sObject>

A list of platform event instances. For example, a list of *MyEvent__e* objects. You must first define your platform event object in your Salesforce org.

Return Value

Type: List<Database.SaveResult>

A list of results, each corresponding to the result of publishing one event. For each event, `Database.SaveResult` contains information about whether the operation was successful and the errors encountered. If the `isSuccess()` method returns `true`, the publish request is queued in Salesforce and the event message is published asynchronously. For more information, see High-Volume Platform Event Persistence. If `isSuccess()` returns `false`, the event publish operation resulted in errors, which are returned in the `Database.Error` object. `EventBus.publish()` can publish some passed-in events, even when other events can't be published due to errors. The `EventBus.publish()` method doesn't throw exceptions caused by an unsuccessful publish operation. It's similar in behavior to the Apex `Database.insert` method when called with the partial success option.

`Database.SaveResult` also contains the `Id` system field. The `Id` field value isn't included in the event message delivered to subscribers. It isn't used to identify an event message, and isn't always unique.

If an empty list is passed in for the *events* parameter, no event is published, and an empty `List<Database.SaveResult>` is returned.

This method returns a `System.UnexpectedException` if you attempt to publish a list of type `List<SObject>` that contains objects that aren't platform events.

Usage

- The platform event message is published either immediately or after a transaction is committed, depending on the publish behavior you set in the platform event definition. For more information, see Platform Event Fields in the *Platform Events Developer Guide*.
- Apex governor limits apply. For events configured with the **Publish After Commit** behavior, each method execution is counted as one DML statement against the Apex DML statement limit. You can check limit usage using the `Limits.getDMLStatements()` on page 3595 method. For events configured with the **Publish Immediately** behavior, each method execution is counted against a separate event publishing limit of 150 `EventBus.publish()` calls. You can check limit usage using the `Limits.getPublishImmediateDML()` on page 3598 method.

publish(event, callback)

Publishes the given platform event using the specified callback. To track asynchronous publish failures, you can implement an Apex publish callback.

Signature

```
public static Database.SaveResult publish(SObject event, Object callback)
```

Parameters

event
Type: SObject

An instance of a platform event. For example, an instance of *MyEvent__e*. You must first define your platform event object in your Salesforce org.

callback
Type: Object

An Apex class that implements the EventPublishFailureCallback Interface or EventPublishSuccessCallback Interface.

Return Value

Type: Database.SaveResult

The result of publishing the given event. `Database.SaveResult` contains information about whether the operation was successful and the errors encountered. If the `isSuccess()` method returns `true`, the publish request is queued in Salesforce and the event message is published asynchronously. For more information, see High-Volume Platform Event Persistence. If `isSuccess()` returns `false`, the event publish operation resulted in errors, which are returned in the `Database.Error` object. This method doesn't throw an exception due to an unsuccessful publish operation.

This method returns a `System.UnexpectedException` if you attempt to publish an `SObject` that represents an object that isn't a platform event.

Usage

- Use this method with Apex publish callbacks. For more information, see Get the Result of Asynchronous Platform Event Publishing with Apex Publish Callbacks in the *Platform Events Developer Guide*.
- The platform event message is published either immediately or after a transaction is committed, depending on the publish behavior you set in the platform event definition. For more information, see Platform Event Fields in the *Platform Events Developer Guide*.
- Apex governor limits apply. For events configured with the **Publish After Commit** behavior, each method execution is counted as one DML statement against the Apex DML statement limit. You can check limit usage using the `Limits.getDMLStatements()` on page 3595 method. For events configured with the **Publish Immediately** behavior, each method execution is counted against a separate event publishing limit of 150 `EventBus.publish()` calls. You can check limit usage using the `Limits.getPublishImmediateDML()` on page 3598 method.

`publish(events, callback)`

Publishes the given list of platform events using the specified callback. To track asynchronous publish failures, you can implement an Apex publish callback.

Signature

```
public static List<Database.SaveResult> publish(List<SObject> sobjects, Object callback)
```

Parameters

sobjects
Type: List<SObject>

A list of platform event instances. For example, a list of *MyEvent__e* objects. You must first define your platform event object in your Salesforce org.

callback
Type: Object

An Apex class that implements the EventPublishFailureCallback Interface or EventPublishSuccessCallback Interface.

Return Value

Type: List<Database.SaveResult>

A list of results, each corresponding to the result of publishing one event. For each event, `Database.SaveResult` contains information about whether the operation was successful and the errors encountered. If the `isSuccess()` method returns `true`, the publish request is queued in Salesforce and the event message is published asynchronously. For more information, see High-Volume Platform Event Persistence. If `isSuccess()` returns `false`, the event publish operation resulted in errors, which are returned in the `Database.Error` object. `EventBus.publish()` can publish some passed-in events, even when other events can't be published due to errors. The `EventBus.publish()` method doesn't throw exceptions caused by an unsuccessful publish operation. It's similar in behavior to the Apex `Database.insert` method when called with the partial success option.

If an empty list is passed in for the *`events`* parameter, no event is published, and an empty `List<Database.SaveResult>` is returned.

This method returns a `System.UnexpectedException` if you attempt to publish a list of type `List<SObject>` that contains objects that aren't platform events.

Usage

- Use this method with Apex publish callbacks. For more information, see Get the Result of Asynchronous Platform Event Publishing with Apex Publish Callbacks in the *Platform Events Developer Guide*.
- The platform event message is published either immediately or after a transaction is committed, depending on the publish behavior you set in the platform event definition. For more information, see Platform Event Fields in the *Platform Events Developer Guide*.
- Apex governor limits apply. For events configured with the **Publish After Commit** behavior, each method execution is counted as one DML statement against the Apex DML statement limit. You can check limit usage using the `Limits.getDMLStatements()` on page 3595 method. For events configured with the **Publish Immediately** behavior, each method execution is counted against a separate event publishing limit of 150 `EventBus.publish()` calls. You can check limit usage using the `Limits.getPublishImmediateDML()` on page 3598 method.

Exception Class and Built-In Exceptions

An exception denotes an error that disrupts the normal flow of code execution. You can use Apex built-in exceptions or create custom exceptions. All exceptions have common methods.

All exceptions support built-in methods for returning the error message and exception type. In addition to the standard `exception` class, there are several different types of exceptions:

The following are exceptions in the `System` namespace.

Exception	Description
`AssertException`	A System.assert failure that halts code execution. Optionally contains the custom message specified in the last (`msg`) argument to the `assert()` method.
`AuraException`	Legacy Aura-related exception. Use System.AuraHandledException instead.

Exception	Description
`AuraHandledException`	Returns a custom error message to a JavaScript controller. See Returning Errors from an Apex Server-Side Controller.
`AsyncException`	Any problem with an asynchronous operation, such as failing to enqueue an asynchronous call.
`BigObjectException`	Any problem with big object records, such as connection timeouts during attempts to access or insert big object records.
`CalloutException`	Any problem with a Web service operation, such as failing to make a callout to an external system.
`DataWeaveScriptException`	Any run-time script errors that occur within DataWeave in Apex.
`DmlException`	Any problem with a DML statement, such as an `insert` statement missing a required field on a record.
`DuplicateMessageException`	Attempt to enqueue job with duplicate queueable signature
`EmailException`	Any problem with email, such as failure to deliver. For more information, see Outbound Email.
`ExternalObjectException`	Any problem with external object records, such as connection timeouts during attempts to access the data that's stored on external systems.
`FatalCursorException`	Any problem with Apex cursors in a transaction.
`FinalException`	Any attempt to mutate a read-only collection or record such as an sObject in an after-update trigger, or a final variable. This exception causes execution to halt.
`FlowException`	Any problem with starting flow interviews from Apex. For example, if an active version of the flow can't be found or it can't be started from Apex.
`HandledException`	A generic handled exception.
`IllegalArgumentException`	An illegal argument was provided to a method call. For example, a method that requires a non-null argument throws this exception if a null value is passed into the method.
`InvalidHeaderException`	An illegal header argument was provided to an Apex REST call. For example, a call to the `RestResponse.addHeader(name, value)` method throws this exception if the header name is `cookie`.
`InvalidParameterValueException`	This exception is used with both Visualforce pages and Salesforce Functions. **Visualforce** The exception is thrown when an invalid parameter is supplied for a method, or any problem is encountered with a URL used with Visualforce pages. For more information on Visualforce, see the *Visualforce Developer's Guide*. **Salesforce Functions** The exception is thrown when the `functionName` parameter to `Function.get()` doesn't have the correct `project name.function name` format. For more information on Salesforce functions, see `Function.get()`.

Exception	Description
`LimitException`	A governor limit has been exceeded. This exception can't be caught.
`JSONException`	Any problem with JSON serialization and deserialization operations. For more information, see the methods of `System.JSON`, `System.JSONParser`, and `System.JSONGenerator`.
`ListException`	Any problem with a list, such as attempting to access an index that is out of bounds.
`MathException`	Any problem with a mathematical operation, such as dividing by zero.
`NoAccessException`	Any problem with unauthorized access, such as trying to access an sObject that the current user doesn't have access to. This exception is used with Visualforce pages. For more information on Visualforce, see the *Visualforce Developer's Guide*.
`NoDataFoundException`	This exception is used with both Visualforce pages and Salesforce Functions. **Visualforce** The exception is thrown with data that doesn't exist, such as trying to access an sObject that has been deleted. For more information on Visualforce, see the *Visualforce Developer's Guide*. **Salesforce Functions** The exception is thrown when the project or function name provided in the `functionName` parameter to the `Function.get()` method can't be found. For more information on Salesforce functions, see `Function.get()`.
`NoSuchElementException`	This exception is thrown if you try to access items that are outside the bounds of a list. This exception is used by the Iterator `next` method. For example, if `iterator.hasNext() == false` and you call `iterator.next()`, this exception is thrown. This exception is also used by the Apex Flex Queue methods and is thrown if you attempt to access a job at an invalid position in the flex queue.

`NullPointerException`	Any problem with dereferencing null, such as in the following code:
	`String s;`
	`s.toLowerCase(); // Since s is null, this call causes`
	`// a NullPointerException`

`QueryException`	Any problem with SOQL queries, such as assigning a query that returns no records or more than one record to a singleton sObject variable.
`RequiredFeatureMissing`	A Chatter feature is required for code that has been deployed to an organization that doesn't have Chatter enabled.
`SearchException`	Any problem with SOSL queries executed with SOAP API `search()` call, for example, when the `searchString` parameter contains fewer than two characters. For more information, see the *SOAP API Developer Guide*.
`SecurityException`	Any problem with static methods in the Crypto utility class. For more information, see Crypto Class.
`SerializationException`	Any problem with the serialization of data. This exception is used with Visualforce pages. For more information on Visualforce, see the *Visualforce Developer's Guide*.

Exception	Description
`SObjectException`	Any problem with sObject records, such as attempting to change a field in an `update` statement that can only be changed during `insert`.
`StringException`	Any problem with Strings, such as a String that is exceeding your heap size.
`TransientCursorException`	A transient problem with an Apex cursor transaction. The failed transaction can be retried.
`TypeException`	Any problem with type conversions, such as attempting to convert the String 'a' to an Integer using the `valueOf` method.
`UnexpectedException`	A non-recoverable internal error within Salesforce has occurred. This exception causes execution to halt. If necessary, contact Salesforce Customer Support for more information.
`VisualforceException`	Any problem with a Visualforce page. For more information on Visualforce, see the *Visualforce Developer's Guide*.
`XmlException`	Any problem with the XmlStream classes, such as failing to read or write XML.

The following is an example using the DmlException exception:

```
Account[] accts = new Account[]{new Account(billingcity = 'San Jose')};
try {
    insert accts;
} catch (System.DmlException e) {
    for (Integer i = 0; i < e.getNumDml(); i++) {
        // Process exception here
        System.debug(e.getDmlMessage(i));
    }
}
```

For exceptions in other namespaces, see:

- Cache Exceptions
- Canvas Exceptions
- Compression Exceptions
- `ConnectApi` Exceptions
- DataSource Exceptions
- Reports Exceptions
- Site Exceptions

Common Exception Methods

Exception methods are all called by and operate on an instance of an exception. This table describes all instance exception methods. All types of exceptions have these methods in common.

Name	Arguments	Return Type	Description
`getCause`		Exception	Returns the cause of the exception as an exception object.

Name	Arguments	Return Type	Description
`getLineNumber`		Integer	Returns the line number from where the exception was thrown.
`getMessage`		String	Returns the error message that displays for the user.
`getStackTraceString`		String	Returns the stack trace of a thrown exception as a string.
`getTypeName`		String	Returns the type of exception, such as DmlException, ListException, MathException, and so on.
`initCause`	Exception *cause*	Void	Sets the cause for this exception, if one hasn't already been set.
`setMessage`	String *s*	Void	Sets the error message that displays for the user.

DMLException and EmailException Methods

In addition to the common exception methods, DMLException and EmailException have these methods:

Name	Arguments	Return Type	Description
`getDmlFieldNames`	Integer *i*	String []	Returns the names of the field or fields that caused the error described by the ith failed row.
`getDmlFields`	Integer *i*	Schema.sObjectField []	Returns the field token or tokens for the field or fields that caused the error described by the ith failed row. For more information on field tokens, see Dynamic Apex.
`getDmlId`	Integer *i*	String	Returns the ID of the failed record that caused the error described by the ith failed row.
`getDmlIndex`	Integer *i*	Integer	Returns the original row position of the ith failed row.
`getDmlMessage`	Integer *i*	String	Returns the user message for the ith failed row.
`getDmlStatusCode`	Integer *i*	String	Deprecated. Use getDmlType instead. Returns the Apex failure code for the ith failed row.
`getDmlType`	Integer *i*	System.StatusCode	Returns the value of the System.StatusCode enum. For example: `try { insert new Account(); } catch (System.DmlException ex) { System.assertEquals( StatusCode.REQUIRED_FIELD_MISSING, ex.getDmlType(0)); }` For more information about System.StatusCode, see Enums.
`getNumDml`		Integer	Returns the number of failed rows for DML exceptions.

QueryException Method

In addition to the common exception methods, QueryException has this method.

Name	Arguments	Return Type	Description
`getInaccessibleFields`		Map on page 3624<String,Set< on page 3767String>>	Returns a map in which each key is an `sObjectType` and its corresponding value is the set of inaccessible field names in fully qualified format (Namespace__FieldName__c). Use this method to determine the cause of the `QueryException`. The returned map contains data only if the method that threw the `QueryException` is running in user mode (as opposed to the default system mode). In this code sample, it's assumed that the user doesn't have field level security access to the `Contact.Email` and `Account.Website` fields. `try { List<Account> accounts = [SELECT Website, (SELECT Email FROM Contacts) FROM Account WITH USER_MODE]; } catch (QueryException qe) { // Handle inaccessible fields Map<String, Set<String>> inaccessible = qe.getInaccessibleFields(); Set<String> accountFields = inaccessible.get('Account'); Set<String> contactFields = inaccessible.get('Contact'); }`

ExternalServiceTest Class

Provides methods to test an external service's asynchronous callouts, enables sending a mock request, asserts the expected request payload, then triggers the mocked external service's asynchronous callback response.

Namespace

System

Usage

See Create Unit Testing for Asynchronous Callouts in the "Extend Salesforce with Clicks, Not Code" Help Guide.

IN THIS SECTION:

ExternalServiceTest Methods
An instance of the ExternalServiceTest method is used when the test class triggers a mocked external service's callback response. You can access ExternalServiceTest through `Test.getExternalService()`

ExternalServiceTest Methods

An instance of the ExternalServiceTest method is used when the test class triggers a mocked external service's callback response. You can access ExternalServiceTest through `Test.getExternalService()`

The following are methods for `ExternalServiceTest`.

IN THIS SECTION:

sendCallback(request)
Sends the HTTP request back as an external service asynchronous response.

sendCallback(request)

Sends the HTTP request back as an external service asynchronous response.

Signature

```
public System.HttpResponse sendCallback(System.HttpRequest request)
```

```
System.ExternalServiceTest, sendCallback, [System.HttpRequest], System.HttpResponse
```

Parameters

request
Type: System.HttpRequest on page 3521

Return Value

Type: System.HttpResponse on page 3530

FlexQueue Class

Contains methods that reorder batch jobs in the Apex flex queue.

Namespace

System

Usage

You can place up to 100 batch jobs in a holding status for future execution. When system resources become available, the jobs are taken from the top of the Apex flex queue and moved to the batch job queue. Up to five queued or active jobs can be processed simultaneously

for each org. When a job is moved out of the flex queue for processing, its status changes from Holding to Queued. Queued jobs are executed when the system is ready to process new jobs.

Use this class's methods to reorder your `Holding` jobs in the flex queue. As best practice and for safe usage, a `FlexQueue` reorder method must be the final statement in a transaction.

Example

This example moves a job to the front of the flex queue so that it's executed immediately. The job is moved by calling the `System.FlexQueue.moveJobToFront()` method with the high priority job ID as the parameter.

```
ID highPriorityJobId = Database.executeBatch(new HighPriorityBatchClass(), 200);
boolean jobMovedToFrontOfQueue = FlexQueue.moveJobToFront(highPriorityJobId);
```

IN THIS SECTION:

FlexQueue Methods

SEE ALSO:

Monitoring the Apex Flex Queue

Apex Developer Guide: Using Batch Apex

FlexQueue Methods

The following are methods for `FlexQueue`.

IN THIS SECTION:

moveAfterJob(jobToMoveId, jobInQueueId)

Moves the job with the ID *`jobToMoveId`* immediately after the job with the ID *`jobInQueueId`* in the flex queue. You can move *`jobToMoveId`* forward or backward in the queue. If either job isn't in the queue, it throws an element-not-found exception. Returns `true` if the job is moved, or `false` if *`jobToMoveId`* is already immediately after *`jobInQueueId`*, so no change is made.

moveBeforeJob(jobToMoveId, jobInQueueId)

Moves the job with the ID *`jobToMoveId`* immediately before the job with the ID *`jobInQueueId`* in the flex queue. You can move *`jobToMoveId`* forward or backward in the queue. If either job isn't in the queue, it throws an element-not-found exception. Returns `true` if the job is moved, or `false` if *`jobToMoveId`* is already immediately before *`jobInQueueId`*, so no change is made.

moveJobToEnd(jobId)

Moves the specified job the end of the flex queue, to index position `(size - 1)`. All jobs after the job's starting position move one spot forward. If the job isn't in the queue, it throws an element-not-found exception. Returns `true` if the job is moved, or `false` if the job is already at the end of the queue, so no change is made.

moveJobToFront(jobId)

Moves the specified job to the front of the flex queue, to index position `0`. All other jobs move back one spot. If the job isn't in the queue, it throws an element-not-found exception. Returns `true` if the job is moved, or `false` if the job is already at the front of the queue, so no change is made.

moveAfterJob(jobToMoveId, jobInQueueId)

Moves the job with the ID *jobToMoveId* immediately after the job with the ID *jobInQueueId* in the flex queue. You can move *jobToMoveId* forward or backward in the queue. If either job isn't in the queue, it throws an element-not-found exception. Returns `true` if the job is moved, or `false` if *jobToMoveId* is already immediately after *jobInQueueId*, so no change is made.

Signature

```
public static Boolean moveAfterJob(Id jobToMoveId, Id jobInQueueId)
```

Parameters

jobToMoveId
Type: Id

The ID of the job to move.

jobInQueueId
Type: Id

The ID of the job to move after.

Return Value

Type: Boolean

moveBeforeJob(jobToMoveId, jobInQueueId)

Moves the job with the ID *jobToMoveId* immediately before the job with the ID *jobInQueueId* in the flex queue. You can move *jobToMoveId* forward or backward in the queue. If either job isn't in the queue, it throws an element-not-found exception. Returns `true` if the job is moved, or `false` if *jobToMoveId* is already immediately before *jobInQueueId*, so no change is made.

Signature

```
public static Boolean moveBeforeJob(Id jobToMoveId, Id jobInQueueId)
```

Parameters

jobToMoveId
Type: Id

The ID of the job to move.

jobInQueueId
Type: Id

The ID of the job to use as a reference point.

Return Value

Type: Boolean

moveJobToEnd(jobId)

Moves the specified job the end of the flex queue, to index position `(size - 1)`. All jobs after the job's starting position move one spot forward. If the job isn't in the queue, it throws an element-not-found exception. Returns `true` if the job is moved, or `false` if the job is already at the end of the queue, so no change is made.

Signature

```
public static Boolean moveJobToEnd(Id jobId)
```

Parameters

jobId
Type: Id

The ID of the job to move.

Return Value

Type: Boolean

moveJobToFront(jobId)

Moves the specified job to the front of the flex queue, to index position `0`. All other jobs move back one spot. If the job isn't in the queue, it throws an element-not-found exception. Returns `true` if the job is moved, or `false` if the job is already at the front of the queue, so no change is made.

Signature

```
public static Boolean moveJobToFront(Id jobId)
```

Parameters

jobId
Type: Id

The ID of the job to move.

Return Value

Type: Boolean

FeatureManagement Class

Use the methods in the `System.FeatureManagement` class to check and modify the values of feature parameters, and to show or hide custom objects and custom permissions in your subscribers' orgs.

Namespace

System

Usage

For information about feature parameters, see Manage Features in Second Generation Managed Packages in the *Second-Generation Managed Packaging Developer Guide*, or Manage Features in First-Generation Managed Packages in the *First-Generation Managed Packaging Developer Guide*.

The set methods (setPackageBooleanValue, setPackageDateValue, setPackageIntegerValue) use DML operations on setup sObjects. To learn more about mixing operations in a test, see Mixed DML Operations in Test Methods.

IN THIS SECTION:

FeatureManagement Methods

FeatureManagement Methods

The following are methods for `FeatureManagement`.

IN THIS SECTION:

changeProtection(apiName, typeApiName, protection)

Hides or reveals custom permissions, or reveals custom objects, in your subscriber's org.

checkPackageBooleanValue(apiName)

Checks the `value__c` value of the `FeatureParameterBoolean__c` record for a feature parameter in your subscriber's org. You set the record's value using `setPackageBooleanValue(apiName, value)`.

checkPackageDateValue(apiName)

Checks the `value__c` value of the `FeatureParameterDate__c` record for a feature parameter in your subscriber's org. You can set the record's value using `setPackageDateValue(apiName, value)`.

checkPackageIntegerValue(apiName)

Checks the `value__c` value of the `FeatureParameterInteger__c` record for a feature parameter in your subscriber's org. You can set the record's value using `setPackageIntegerValue(apiName, value)`.

checkPermission(apiName)

Checks whether a custom permission is enabled.

setPackageBooleanValue(apiName, value)

Sets the `value__c` value of the `FeatureParameterBoolean__c` record for a subscriber-to-LMO feature parameter in your subscriber's org. You can check the record's value using `checkPackageBooleanValue(apiName)`.

setPackageDateValue(apiName, value)

Sets the `value__c` value of the `FeatureParameterDate__c` record for a subscriber-to-LMO feature parameter in your subscriber's org. You can check the record's value using `checkPackageDateValue(apiName)`.

setPackageIntegerValue(apiName, value)

Sets the `value__c` value of the `FeatureParameterInteger__c` record for a subscriber-to-LMO feature parameter in your subscriber's org. You can check the record's value using `checkPackageIntegerValue(apiName)`.

`changeProtection(apiName, typeApiName, protection)`

Hides or reveals custom permissions, or reveals custom objects, in your subscriber's org.

Signature

```
public static void changeProtection(String apiName, String typeApiName, String
protection)
```

Parameters

apiName
Type: String

The API name of the custom object or custom permission to show or hide—for example, `'MyCustomObject__c'` or `'MyCustomPermission'`.

typeApiName
Type: String

The API name of the type that you want to show or hide: `'CustomObject'` or `'CustomPermission'`.

protection
Type: String

To show a custom object or custom permission, `'Unprotected'`.

To hide a custom permission, `'Protected'`.

Return Value

Type: void

Usage

Warning: For custom permissions, you can toggle the protected value indefinitely. However, after you've released unprotected objects to subscribers, you can't set visibility to `Protected`. Be sure to protect any custom objects that you want to hide before you release the first package version that contains them.

To hide custom permissions in released packages:

```
FeatureManagement.changeProtection('YourCustomPermissionName', 'CustomPermission',
    'Protected');
```

To unhide custom permissions and custom objects in released packages:

```
FeatureManagement.changeProtection('YourCustomPermissionName', 'CustomPermission',
    'Unprotected');
```

```
FeatureManagement.changeProtection('YourCustomObjectName__c', 'CustomObject',
    'Unprotected');
```

checkPackageBooleanValue(apiName)

Checks the `value__c` value of the `FeatureParameterBoolean__c` record for a feature parameter in your subscriber's org. You set the record's value using `setPackageBooleanValue(apiName, value)`.

Signature

```
public static Boolean checkPackageBooleanValue(String apiName)
```

Parameters

apiName
Type: String

The `fullName__c` value of the feature parameter whose value you want to check—for example, `'SpecialAccessAvailable'`.

Return Value

Type: Boolean

The value that's currently assigned to the `value__c` field on the `FeatureParameterBoolean__c` record that associates the feature parameter with its related license.

checkPackageDateValue(apiName)

Checks the `value__c` value of the `FeatureParameterDate__c` record for a feature parameter in your subscriber's org. You can set the record's value using `setPackageDateValue(apiName, value)`.

Signature

```
public static Date checkPackageDateValue(String apiName)
```

Parameters

apiName
Type: String

The `fullName__c` value of the feature parameter whose value you want to check—for example, `'TrialExpirationDate'`.

Return Value

Type: Date

The value that's currently assigned to the `value__c` field on the `FeatureParameterDate__c` record that associates the feature parameter with its related license.

checkPackageIntegerValue(apiName)

Checks the `value__c` value of the `FeatureParameterInteger__c` record for a feature parameter in your subscriber's org. You can set the record's value using `setPackageIntegerValue(apiName, value)`.

Signature

```
public static Integer checkPackageIntegerValue(String apiName)
```

Parameters

apiName
Type: String

The `fullName__c` value of the feature parameter whose value you want to check—for example, `'NumberOfLicenses'`.

Return Value

Type: Integer

The value that's currently assigned to the `value__c` field on the `FeatureParameterInteger__c` record that associates the feature parameter with its related license.

checkPermission(apiName)

Checks whether a custom permission is enabled.

Signature

```
public static Boolean checkPermission(String apiName)
```

Parameters

apiName
Type: String

The API name of the custom permission to check the value of—for example, `'MyCustomPermission'`.

Return Value

Type: Boolean

Shows whether the permission is enabled (`true`) or disabled (`false`).

setPackageBooleanValue(apiName, value)

Sets the `value__c` value of the `FeatureParameterBoolean__c` record for a subscriber-to-LMO feature parameter in your subscriber's org. You can check the record's value using `checkPackageBooleanValue(apiName)`.

Signature

```
public static void setPackageBooleanValue(String apiName, Boolean value)
```

Parameters

apiName
Type: String

The `fullName__c` value of the feature parameter whose value you want to set—for example, `'SpecialAccessAvailable'`.

value
Type: Boolean

The value to assign to the `value__c` field on the `FeatureParameterBoolean__c` record that associates the feature parameter with its related license.

Return Value

Type: void

`setPackageDateValue(apiName, value)`

Sets the `value__c` value of the `FeatureParameterDate__c` record for a subscriber-to-LMO feature parameter in your subscriber's org. You can check the record's value using `checkPackageDateValue(apiName)`.

Signature

```
public static void setPackageDateValue(String apiName, Date value)
```

Parameters

apiName
Type: String

The `fullName__c` value of the feature parameter whose value you want to set—for example, `'TrialExpirationDate'`.

value
Type: Date

The value to assign to the `value__c` field on the `FeatureParameterDate__c` record that associates the feature parameter with its related license.

Return Value

Type: void

`setPackageIntegerValue(apiName, value)`

Sets the `value__c` value of the `FeatureParameterInteger__c` record for a subscriber-to-LMO feature parameter in your subscriber's org. You can check the record's value using `checkPackageIntegerValue(apiName)`.

Signature

```
public static void setPackageIntegerValue(String apiName, Integer value)
```

Parameters

apiName
Type: String

The `fullName__c` value of the feature parameter whose value you want to set—for example, `'NumberOfLicenses'`.

value
Type: Integer

The value to assign to the `value__c` field on the `FeatureParameterInteger__c` record that associates the feature parameter with its related license.

Return Value

Type: void

Formula Class

Contains methods to get a builder for creating a formula instance and to update all formula fields on the input SObjects..

Namespace

System

IN THIS SECTION:

Formula Methods

Formula Methods

The following are methods for `Formula`.

IN THIS SECTION:

builder()
Creates an instance of `FormulaBuilder` for configuring the formula with formula expression, context type, and output data type as inputs.

recalculateFormulas(sobjects)
Updates (recalculates) all formula fields on the input SObjects.

`builder()`

Creates an instance of `FormulaBuilder` for configuring the formula with formula expression, context type, and output data type as inputs.

Signature

```
public static formulaeval.FormulaBuilder builder()
```

Return Value

Type: FormulaEval.FormulaBuilder

`recalculateFormulas(sobjects)`

Updates (recalculates) all formula fields on the input SObjects.

Signature

```
public static List<System.FormulaRecalcResult> recalculateFormulas(List<SObject>
sobjects)
```

Parameters

`sobjects`
Type: List<SObject>

List of sObjects whose formula fields are to be recalculated.

Return Value

Type: List<FormulaRecalcResult Class>

Usage

Recalculate formula fields on new or queried SObjects. If all data is present on the SObjects, SOQL limits are not affected. If the data required to evaluate a formula field is missing, that data is retrieved and limits are changed accordingly.

The new formula values are stored in the SObjects themselves and overwrite previous values of formula fields.

Example

```
Account a = new Account();
a.Name = 'Salesforce';
a.BillingCity = 'San Francisco';
List<Account> accounts = new List<Account>{a};

List<FormulaRecalcResult> results = Formula.recalculateFormulas(accounts);
System.assert(results[0].isSuccess());
// Option 1
System.debug('New value: ' + accounts[0].get('My_Formula_Field__c'));
// Option 2
System.debug('New value: ' + results[0].getSObject().get('My_Formula_Field__c'));
```

FormulaRecalcFieldError Class

The return type of the `FormulaRecalcResult.getErrors` method.

Namespace

System

IN THIS SECTION:

FormulaRecalcFieldError Methods

FormulaRecalcFieldError Methods

The following are methods for `FormulaRecalcFieldError`.

IN THIS SECTION:

getFieldError()

Returns a message describing the errors encountered during formula recalculation.

getFieldName()

Returns the name of the formula recalculation error field.

`getFieldError()`

Returns a message describing the errors encountered during formula recalculation.

Signature

```
public String getFieldError()
```

Return Value

Type: String

`getFieldName()`

Returns the name of the formula recalculation error field.

Signature

```
public String getFieldName()
```

Return Value

Type: String

FormulaRecalcResult Class

The return type of the `Formula.recalculateFormulas` method.

Namespace

System

Usage

Indicates the result and status of recalculating formulas on a single sObject. Holds a reference to the sObject and a list of all the fields that were recalculated.

Example

This example assumes that you have a formula field called `divide__c` with formula `"1 / LEN(Name)`.

```
List<Account> accounts = [SELECT Name FROM Account WHERE Name='Acme'];
accounts[0].Name = '';
List<FormulaRecalcResult> results = Formula.recalculateFormulas(accounts);
FormulaRecalcResult result0 = results[0];
FormulaRecalcFieldError fieldError = result0.getErrors()[0];
System.debug(fieldError.getFieldName()); // 'divide'
System.debug(fieldError.getFieldError()); // 'Division by zero'
```

IN THIS SECTION:

FormulaRecalcResult Methods

FormulaRecalcResult Methods

The following are methods for `FormulaRecalcResult`.

IN THIS SECTION:

getErrors()
If an error occurs during formula recalculation, an array of one or more database error objects, along with error codes and descriptions, is returned.

getSObject()
Returns the sObject with formulas recalculated.

isSuccess()
Returns a Boolean value that is set to `true` if the formula recalculation process completed successfully; otherwise, it is set to `false`.

`getErrors()`

If an error occurs during formula recalculation, an array of one or more database error objects, along with error codes and descriptions, is returned.

Signature

```
public List<System.FormulaRecalcFieldError> getErrors()
```

Return Value

Type: List<FormulaRecalcFieldError Class>

`getSObject()`

Returns the sObject with formulas recalculated.

Signature

```
public SObject getSObject()
```

Return Value

Type: SObject

`isSuccess()`

Returns a Boolean value that is set to `true` if the formula recalculation process completed successfully; otherwise, it is set to `false`.

Signature

```
public Boolean isSuccess()
```

Return Value

Type: Boolean

Http Class

Use the `Http` class to initiate an HTTP request and response.

Namespace

System

Http Methods

The following are methods for `Http`. All are instance methods.

IN THIS SECTION:

send(request)
Sends an HttpRequest and returns the response.

toString()
Returns a string that displays and identifies the object's properties.

`send(request)`

Sends an HttpRequest and returns the response.

Signature

```
public HttpResponse send(HttpRequest request)
```

Parameters

`request`
Type: System.HttpRequest

Return Value

Type: System.HttpResponse

`toString()`

Returns a string that displays and identifies the object's properties.

Signature

```
public String toString()
```

Return Value

Type: String

HttpCalloutMock Interface

Enables sending fake responses when testing HTTP callouts.

Namespace

System

Usage

For an implementation example, see Testing HTTP Callouts by Implementing the `HttpCalloutMock` Interface.

HttpCalloutMock Methods

The following are methods for `HttpCalloutMock`.

IN THIS SECTION:

respond(request)
Returns an HTTP response for the given request. The implementation of this method is called by the Apex runtime to send a fake response when an HTTP callout is made after `Test.setMock` has been called.

`respond(request)`

Returns an HTTP response for the given request. The implementation of this method is called by the Apex runtime to send a fake response when an HTTP callout is made after `Test.setMock` has been called.

Signature

```
public HttpResponse respond(HttpRequest request)
```

Parameters

request
Type: System.HttpRequest

Return Value

Type: System.HttpResponse

HttpRequest Class

Use the `HttpRequest` class to programmatically create HTTP requests like GET, POST, PATCH, PUT, and DELETE.

Namespace

System

Usage

Use the XML classes or JSON classes to parse XML or JSON content in the body of a request created by `HttpRequest`.

Example

The following example illustrates how you can use an authorization header with a request and handle the response.

```
public class AuthCallout {

   public void basicAuthCallout(){
     HttpRequest req = new HttpRequest();
     req.setEndpoint('http://www.yahoo.com');
     req.setMethod('GET');

     // Specify the required user name and password to access the endpoint
     // As well as the header and header information

     String username = 'myname';
     String password = 'mypwd';

     Blob headerValue = Blob.valueOf(username + ':' + password);
     String authorizationHeader = 'Basic ' +
     EncodingUtil.base64Encode(headerValue);
     req.setHeader('Authorization', authorizationHeader);

     // Create a new http object to send the request object
     // A response object is generated as a result of the request

     Http http = new Http();
     HTTPResponse res = http.send(req);
     System.debug(res.getBody());
   }
}
```

Note: You can set the endpoint as a named credential URL. A named credential URL contains the scheme `callout:`, the name of the named credential, and an optional path. For example: `callout:My_Named_Credential/some_path`. A named credential specifies the URL of a callout endpoint and its required authentication parameters in one definition. Salesforce manages all authentication for Apex callouts that specify a named credential as the callout endpoint so that your code doesn't have to. See Named Credentials as Callout Endpoints.

Compression

To compress the data you send, use `setCompressed`.

```
HttpRequest req = new HttpRequest();
req.setEndPoint('my_endpoint');
req.setCompressed(true);
```

```
req.setBody('some post body');
```

If a response comes back in compressed format, `getBody` recognizes the format, uncompresses it, and returns the uncompressed value.

IN THIS SECTION:

HttpRequest Constructors

HttpRequest Methods

SEE ALSO:

Apex Developer Guide: JSON Support

Apex Developer Guide: XML Support

HttpRequest Constructors

The following are constructors for `HttpRequest`.

IN THIS SECTION:

HttpRequest()
Creates a new instance of the `HttpRequest` class.

`HttpRequest()`

Creates a new instance of the `HttpRequest` class.

Signature

```
public HttpRequest()
```

HttpRequest Methods

The following are methods for `HttpRequest`. All are instance methods.

IN THIS SECTION:

getBody()
Retrieves the body of this request.

getBodyAsBlob()
Retrieves the body of this request as a Blob.

getBodyDocument()
Retrieves the body of this request as a DOM document.

getCompressed()
If `true`, the request body is compressed, `false` otherwise.

getEndpoint()

Retrieves the URL for the endpoint of the external server for this request.

getHeader(key)

Retrieves the contents of the request header.

getMethod()

Returns the type of method used by `HttpRequest`.

setBody(body)

Sets the contents of the body for this request.

setBodyAsBlob(body)

Sets the contents of the body for this request using a Blob.

setBodyDocument(document)

Sets the contents of the body for this request. The contents represent a DOM document.

setClientCertificate(clientCert, password)

This method is deprecated. Use `setClientCertificateName` instead.

setClientCertificateName(certDevName)

If the external service requires a client certificate for authentication, set the certificate name.

setCompressed(flag)

If `true`, the data in the body is delivered to the endpoint in the gzip compressed format. If `false`, no compression format is used.

setEndpoint(endpoint)

Specifies the endpoint for this request.

setHeader(key, value)

Sets the contents of the request header.

setMethod(method)

Sets the type of method to be used for the HTTP request.

setTimeout(timeout)

Sets a timeout for the request between 1 and 120,000 milliseconds. The timeout is the maximum time to wait for establishing the HTTP connection. The same timeout is used for waiting for the request to start. When the request is executing, such as retrieving or posting data, the connection is kept alive until the request finishes.

toString()

Returns a string containing the URL for the endpoint of the external server for this request and the method used, for example, `Endpoint=http://YourServer, Method=POST`

`getBody()`

Retrieves the body of this request.

Signature

```
public String getBody()
```

Return Value

Type: String

getBodyAsBlob()

Retrieves the body of this request as a Blob.

Signature

```
public Blob getBodyAsBlob()
```

Return Value

Type: Blob

getBodyDocument()

Retrieves the body of this request as a DOM document.

Signature

```
public Dom.Document getBodyDocument()
```

Return Value

Type: Dom.Document

Example

Use this method as a shortcut for:

```
String xml = httpRequest.getBody();
Dom.Document domDoc = new Dom.Document(xml);
```

getCompressed()

If `true`, the request body is compressed, `false` otherwise.

Signature

```
public Boolean getCompressed()
```

Return Value

Type: Boolean

getEndpoint()

Retrieves the URL for the endpoint of the external server for this request.

Signature

```
public String getEndpoint()
```

Return Value

Type: String

`getHeader(key)`

Retrieves the contents of the request header.

Signature

```
public String getHeader(String key)
```

Parameters

key
Type: String

Return Value

Type: String

`getMethod()`

Returns the type of method used by `HttpRequest`.

Signature

```
public String getMethod()
```

Return Value

Type: String

Usage

Examples of return values:

- DELETE
- GET
- HEAD
- PATCH
- POST
- PUT
- TRACE

`setBody(body)`

Sets the contents of the body for this request.

Signature

```
public Void setBody(String body)
```

Parameters

body
Type: String

Return Value

Type: Void

Usage

Limit: 6 MB for synchronous Apex or 12 MB for asynchronous Apex.

The HTTP request and response sizes count towards the total heap size.

setBodyAsBlob(body)

Sets the contents of the body for this request using a Blob.

Signature

```
public Void setBodyAsBlob(Blob body)
```

Parameters

body
Type: Blob

Return Value

Type: Void

Usage

Limit: 6 MB for synchronous Apex or 12 MB for asynchronous Apex.

The HTTP request and response sizes count towards the total heap size.

setBodyDocument(document)

Sets the contents of the body for this request. The contents represent a DOM document.

Signature

```
public Void setBodyDocument(Dom.Document document)
```

Parameters

document
Type: Dom.Document

Return Value

Type: Void

Usage

Limit: 6 MB for synchronous Apex or 12 MB for asynchronous Apex.

setClientCertificate(clientCert, password)

This method is deprecated. Use `setClientCertificateName` instead.

Signature

```
public Void setClientCertificate(String clientCert, String password)
```

Parameters

clientCert
Type: String

password
Type: String

Return Value

Type: Void

Usage

If the server requires a client certificate for authentication, set the client certificate PKCS12 key store and password.

setClientCertificateName(certDevName)

If the external service requires a client certificate for authentication, set the certificate name.

Signature

```
public Void setClientCertificateName(String certDevName)
```

Parameters

certDevName
Type: String

Return Value

Type: Void

Usage

See Using Certificates with HTTP Requests.

setCompressed(flag)

If `true`, the data in the body is delivered to the endpoint in the gzip compressed format. If `false`, no compression format is used.

Signature

```
public Void setCompressed(Boolean flag)
```

Parameters

flag
Type: Boolean

Return Value

Type: Void

setEndpoint(endpoint)

Specifies the endpoint for this request.

Signature

```
public Void setEndpoint(String endpoint)
```

Parameters

endpoint
Type: String

Possible values for the endpoint:

- Endpoint URL

  ```
  https://my_endpoint.example.com/some_path
  ```

- Named credential URL, which contains the scheme `callout`, the name of the named credential, and, optionally, an appended path

  ```
  callout:My_Named_Credential/some_path
  ```

Return Value

Type: Void

SEE ALSO:

Apex Developer Guide: Named Credentials as Callout Endpoints

`setHeader(key, value)`

Sets the contents of the request header.

Signature

```
public Void setHeader(String key, String value)
```

Parameters

key
Type: String

value
Type: String

Return Value

Type: Void

Usage

Limit 100 KB.

`setMethod(method)`

Sets the type of method to be used for the HTTP request.

Signature

```
public Void setMethod(String method)
```

Parameters

method
Type: String

Possible values for the method type include:

- DELETE
- GET
- HEAD
- PATCH
- POST

- PUT
- TRACE

Return Value

Type: Void

Usage

You can also use this method to set any required options.

`setTimeout(timeout)`

Sets a timeout for the request between 1 and 120,000 milliseconds. The timeout is the maximum time to wait for establishing the HTTP connection. The same timeout is used for waiting for the request to start. When the request is executing, such as retrieving or posting data, the connection is kept alive until the request finishes.

Signature

```
public Void setTimeout(Integer timeout)
```

Parameters

timeout
Type: Integer

Return Value

Type: Void

`toString()`

Returns a string containing the URL for the endpoint of the external server for this request and the method used, for example, `Endpoint=http://YourServer, Method=POST`

Signature

```
public String toString()
```

Return Value

Type: String

HttpResponse Class

Use the `HttpResponse` class to handle the HTTP response returned by the `Http` class.

Namespace

System

Usage

Use the XML classes or JSON Classes to parse XML or JSON content in the body of a response accessed by `HttpResponse`.

Example

In the following `getXmlStreamReader` example, content is retrieved via an HTTP callout, then the XML is parsed using the `XmlStreamReader` class.

```
public class ReaderFromCalloutSample {
    public void getAndParse() {

        // Get the XML document from the endpoint
        Http http = new Http();
        HttpRequest req = new HttpRequest();
        req.setEndpoint(URL.getOrgDomainUrl().toExternalForm() + '/services/data');
        req.setMethod('GET');
        req.setHeader('Accept', 'application/xml');
        HttpResponse res = http.send(req);

        // Log the XML content
        System.debug(res.getBody());

        // Generate the HTTP response as an XML stream
        XmlStreamReader reader = res.getXmlStreamReader();

        // Read through the XML
        while(reader.hasNext()) {
            System.debug('Event Type:' + reader.getEventType());
            if (reader.getEventType() == XmlTag.START_ELEMENT) {
            System.debug(reader.getLocalName());
            }
        reader.next();
        }

  }
}
```

SEE ALSO:

Apex Developer Guide: JSON Support

Apex Developer Guide: XML Support

HttpResponse Methods

The following are methods for `HttpResponse`. All are instance methods.

IN THIS SECTION:

getBody()

Retrieves the body returned in the response.

getBodyAsBlob()
Retrieves the body returned in the response as a Blob.

getBodyDocument()
Retrieves the body returned in the response as a DOM document.

getHeader(key)
Retrieves the contents of the response header.

getHeaderKeys()
Retrieves an array of header keys returned in the response.

getStatus()
Retrieves the status message returned for the response.

getStatusCode()
Retrieves the value of the status code returned in the response.

getXmlStreamReader()
Returns an `XmlStreamReader` that parses the body of the callout response.

setBody(body)
Specifies the body returned in the response.

setBodyAsBlob(body)
Specifies the body returned in the response using a Blob.

setHeader(key, value)
Specifies the contents of the response header.

setStatus(status)
Specifies the status message returned in the response.

setStatusCode(statusCode)
Specifies the value of the status code returned in the response.

toString()
Returns the status message and status code returned in the response, for example:

`getBody()`

Retrieves the body returned in the response.

Signature

```
public String getBody()
```

Return Value

Type: String

Usage

Limit 6 MB for synchronous Apex or 12 MB for asynchronous Apex. The HTTP request and response sizes count towards the total heap size.

getBodyAsBlob()

Retrieves the body returned in the response as a Blob.

Signature

```
public Blob getBodyAsBlob()
```

Return Value

Type: Blob

Usage

Limit 6 MB for synchronous Apex or 12 MB for asynchronous Apex. The HTTP request and response sizes count towards the total heap size.

getBodyDocument()

Retrieves the body returned in the response as a DOM document.

Signature

```
public Dom.Document getBodyDocument()
```

Return Value

Type: `Dom.Document`

Example

Use it as a shortcut for:

```
String xml = httpResponse.getBody();
Dom.Document domDoc = new Dom.Document(xml);
```

getHeader(key)

Retrieves the contents of the response header.

Signature

```
public String getHeader(String key)
```

Parameters

key
Type: String

Return Value

Type: String

getHeaderKeys()

Retrieves an array of header keys returned in the response.

Signature

```
public String[] getHeaderKeys()
```

Return Value

Type: String[]

getStatus()

Retrieves the status message returned for the response.

Signature

```
public String getStatus()
```

Return Value

Type: String

getStatusCode()

Retrieves the value of the status code returned in the response.

Signature

```
public Integer getStatusCode()
```

Return Value

Type: Integer

getXmlStreamReader()

Returns an `XmlStreamReader` that parses the body of the callout response.

Signature

```
public XmlStreamReader getXmlStreamReader()
```

Return Value

Type: System.XmlStreamReader

Usage

Use it as a shortcut for:

```
String xml = httpResponse.getBody();
XmlStreamReader xsr = new XmlStreamReader(xml);
```

setBody(body)

Specifies the body returned in the response.

Signature

```
public Void setBody(String body)
```

Parameters

body
Type: String

Return Value

Type: Void

setBodyAsBlob(body)

Specifies the body returned in the response using a Blob.

Signature

```
public Void setBodyAsBlob(Blob body)
```

Parameters

body
Type: Blob

Return Value

Type: Void

setHeader(key, value)

Specifies the contents of the response header.

Signature

```
public Void setHeader(String key, String value)
```

Parameters

key
Type: String

value
Type: String

Return Value

Type: Void

setStatus(status)

Specifies the status message returned in the response.

Signature

```
public Void setStatus(String status)
```

Parameters

status
Type: String

Return Value

Type: Void

setStatusCode(statusCode)

Specifies the value of the status code returned in the response.

Signature

```
public Void setStatusCode(Integer statusCode)
```

Parameters

statusCode
Type: Integer

Return Value

Type: Void

toString()

Returns the status message and status code returned in the response, for example:

Signature

```
public String toString()
```

Return Value

Type: String

Example

```
Status=OK, StatusCode=200
```

Id Class

Contains methods for the ID primitive data type.

Namespace

System

Example: Getting an sObject Token From an ID

This sample shows how to use the `getSObjectType` method to obtain an sObject token from an ID. The `updateOwner` method in this sample accepts a list of IDs of the sObjects to update the ownerId field of. This list contains IDs of sObjects of the same type. The second parameter is the new owner ID. Note that since it is a future method, it doesn't accept sObject types as parameters; this is why it accepts IDs of sObjects. This method gets the sObject token from the first ID in the list, then does a describe to obtain the object name and constructs a query dynamicallly. It then queries for all sObjects and updates their owner ID fields to the new owner ID.

```
public class MyDynamicSolution {
    @future
    public static void updateOwner(List<ID> objIds, ID newOwnerId) {
        // Validate input
        System.assert(objIds != null);
        System.assert(objIds.size() > 0);
        System.assert(newOwnerId != null);

        // Get the sObject token from the first ID
        // (the List contains IDs of sObjects of the same type).
        Schema.SObjectType token = objIds[0].getSObjectType();

        // Using the token, do a describe
        // and construct a query dynamically.
        Schema.DescribeSObjectResult dr = token.getDescribe();
        String queryString = 'SELECT ownerId FROM ' + dr.getName() +
            ' WHERE ';
        for(ID objId : objIds) {
            queryString += 'Id=\'' + objId + '\' OR ';
        }
        // Remove the last ' OR'
        queryString = queryString.subString(0, queryString.length() - 4);

        sObject[] objDBList = Database.query(queryString);
```

```
        System.assert(objDBList.size() > 0);

        // Update the owner ID on the sObjects
        for(Integer i=0;i<objDBList.size();i++) {
            objDBList[i].put('ownerId', newOwnerId);
        }
        Database.SaveResult[] srList = Database.update(objDBList, false);
        for(Database.SaveResult sr : srList) {
            if (sr.isSuccess()) {
                System.debug('Updated owner ID successfully for ' +
                    dr.getName() + ' ID ' + sr.getId());
            }
            else {
              System.debug('Updating ' + dr.getName() + ' returned the following errors.');

                for(Database.Error e : sr.getErrors()) {
                    System.debug(e.getMessage());
                }
            }
        }
    }
}
```

Id Methods

The following are methods for `Id`.

IN THIS SECTION:

addError(errorMsg)

Marks a trigger record with a custom error message and prevents any DML operation from occurring.

addError(errorMsg, escape)

Marks a trigger record with a custom error message, specifies if the error message should be escaped, and prevents any DML operation from occurring.

addError(exceptionError)

Marks a trigger record with a custom error message and prevents any DML operation from occurring.

addError(exceptionError, escape)

Marks a trigger record with a custom error message and prevents any DML operation from occurring.

getSObjectType()

Returns the token for the sObject corresponding to this ID. This method is primarily used with describe information.

to15()

Converts an 18-character Id value to a 15-character case-sensitive string.

valueOf(toID)

Converts the specified String into an ID and returns the ID.

valueOf(str, restoreCasing)

Converts the specified string into an ID and returns the ID. If `restoreCasing` is `true`, and the string represents an 18-character ID that has incorrect casing, the method returns an 18-character ID that is correctly aligned with its encoded casing.

addError(errorMsg)

Marks a trigger record with a custom error message and prevents any DML operation from occurring.

Signature

```
public Void addError(String errorMsg)
```

Parameters

errorMsg
Type: String

The error message to mark the record with.

Return Value

Type: Void

Usage

This method is similar to the `addError(errorMsg)` sObject method.

Note: This method escapes any HTML markup in the specified error message. The escaped characters are: `\n`, `<`, `>`, `&`, `"`, `\`, `\u2028`, `\u2029`, and `\u00a9`. As a result, HTML markup is not rendered; instead, it is displayed as text in the Salesforce user interface.

Example

```
Trigger.new[0].Id.addError('bad');
```

addError(errorMsg, escape)

Marks a trigger record with a custom error message, specifies if the error message should be escaped, and prevents any DML operation from occurring.

Signature

```
public Void addError(String errorMsg, Boolean escape)
```

Parameters

errorMsg
Type: String

The error message to mark the record with.

escape
Type: Boolean

Indicates whether any HTML markup in the custom error message should be escaped (`true`) or not (`false`). This parameter is ignored in both Lightning Experience and the Salesforce mobile app, and the HTML is always escaped. The escape parameter only applies in Salesforce Classic.

Return Value

Type: Void

Usage

The escaped characters are: `\n`, `<`, `>`, `&`, `"`, `\`, `\u2028`, `\u2029`, and `\u00a9`. As a result, HTML markup is not rendered; instead, it is displayed as text in the Salesforce user interface.

Warning: Be cautious if you specify `false` for the *`escape`* argument. Unescaped strings displayed in the Salesforce user interface can represent a vulnerability in the system because these strings might contain harmful code. If you want to include HTML markup in the error message, call this method with a `false` *`escape`* argument. Make sure that you escape any dynamic content, such as input field values. Otherwise, specify `true` for the *`escape`* argument or call `addError(String` ***`errorMsg`***`)` instead.

Example

```
Trigger.new[0].Id.addError('Fix & resubmit', false);
```

addError(exceptionError)

Marks a trigger record with a custom error message and prevents any DML operation from occurring.

Signature

```
public Void addError(Exception exceptionError)
```

Parameters

`exceptionError`
Type: System.Exception

An Exception object or a custom exception object that contains the error message to mark the record with.

Return Value

Type: Void

Usage

This method is similar to the `addError(exceptionError)` sObject method.

This method escapes any HTML markup in the specified error message. The escaped characters are: `\n`, `<`, `>`, `&`, `"`, `\`, `\u2028`, `\u2029`, and `\u00a9`. As a result, HTML markup is not rendered; instead, it is displayed as text in the Salesforce user interface.

Example

```
public class MyException extends Exception{}

Trigger.new[0].Id.addError(new myException('Invalid Id'));
```

addError(exceptionError, escape)

Marks a trigger record with a custom error message and prevents any DML operation from occurring.

Signature

```
public Void addError(Exception exceptionError, Boolean escape)
```

Parameters

exceptionError
Type: System.Exception

An Exception object or a custom exception object that contains the error message to mark the record with.

escape
Type: Boolean

Indicates whether any HTML markup in the custom error message should be escaped (`true`) or not (`false`). This parameter is ignored in both Lightning Experience and the Salesforce mobile app, and the HTML is always escaped. The escape parameter only applies in Salesforce Classic.

Return Value

Type: Void

Usage

The escaped characters are: `\n`, `<`, `>`, `&`, `"`, `\`, `\u2028`, `\u2029`, and `\u00a9`. As a result, HTML markup is not rendered; instead, it is displayed as text in the Salesforce user interface.

Warning: Be cautious if you specify `false` for the *escape* argument. Unescaped strings displayed in the Salesforce user interface can represent a vulnerability in the system because these strings might contain harmful code. If you want to include HTML markup in the error message, call this method with a `false` *escape* argument. Make sure that you escape any dynamic content, such as input field values. Otherwise, specify `true` for the *escape* argument or call `addError(Exception e)` instead.

Example

```
public class MyException extends Exception{}

account a = new account();
a.addError(new MyException('Invalid Id & other issues'), false);
```

getSObjectType()

Returns the token for the sObject corresponding to this ID. This method is primarily used with describe information.

Signature

```
public Schema.SObjectType getSObjectType()
```

Return Value

Type: Schema.SObjectType

Usage

For more information about describes, see Understanding Apex Describe Information.

Example

```
account a = new account(name = 'account');
insert a;
Id myId = a.id;
system.assertEquals(Schema.Account.SObjectType, myId.getSobjectType());
```

to15()

Converts an 18-character Id value to a 15-character case-sensitive string.

Signature

```
public static string to15()
```

Return Value

Type: String

Example

```
String Id_15_char = '0D5B000001DVM9t';
String Id_18_char = '0D5B000001DVM9tkAh';
ID testId = Id_18_char;
System.assertEquals(testId.to15(),Id_15_char);
```

valueOf(toID)

Converts the specified String into an ID and returns the ID.

Signature

```
public static ID valueOf(String toID)
```

Parameters

toID
Type: String

Return Value

Type: ID

Example

```
Id myId = Id.valueOf('001xa000003DIlo');
```

Versioned Behavior Changes

In API version 54.0 and later, assignment of an invalid 15 or 18 character ID to a variable results in a `System.StringException` exception.

`valueOf(str, restoreCasing)`

Converts the specified string into an ID and returns the ID. If `restoreCasing` is `true`, and the string represents an 18-character ID that has incorrect casing, the method returns an 18-character ID that is correctly aligned with its encoded casing.

Signature

```
public static Id valueOf(String str, Boolean restoreCasing)
```

Parameters

str
Type: String

String to be converted to an ID

restoreCasing
Type: Boolean

If set to `true`, and *str* represents an 18-character ID, the method returns an 18-character ID that is correctly aligned with its casing.

Return Value

Type: Id

The return value depends on both the *str* and the *restoreCasing* parameter values.

Note: If the *str* is invalid, the method throws a `System.StringException` exception.

Parameters	restoreCasing=true	restoreCasing=false
Valid 15-character *str* value	Because the 15-character input value can't be encoded for casing, the method throws a `System.StringException`.	The method returns a 15-character ID.
Valid 18-character *str* value	The method returns an 18-character ID that is correctly aligned with its encoded casing.	The method doesn't attempt to correctly align the casing of the 18-character ID and returns an 18-character ID.

Ideas Class

Represents zone ideas.

Namespace

System

Usage

Ideas is a zone of users who post, vote for, and comment on ideas. An Ideas zone provides an online, transparent way for you to attract, manage, and showcase innovation.

A set of *recent replies* (returned by methods, see below) includes ideas that a user posted or commented on that already have comments posted by another user. The returned ideas are listed based on the time of the last comment made by another user, with the most recent ideas appearing first.

The `userID` argument is a required argument that filters the results so only the ideas that the specified user has posted or commented on are returned.

The `communityID` argument filters the results so only the ideas within the specified zone are returned. If this argument is the empty string, then all recent replies for the specified user are returned regardless of the zone.

For more information on ideas, see "Using Ideas" in the Salesforce online help.

Example

The following example finds ideas in a specific zone that have similar titles as a new idea:

```
public class FindSimilarIdeasController {

   public static void test() {
      // Instantiate a new idea
      Idea idea = new Idea ();

      // Specify a title for the new idea
      idea.Title = 'Increase Vacation Time for Employees';

      // Specify the communityID (INTERNAL_IDEAS) in which to find similar ideas.
      Community community = [ SELECT Id FROM Community WHERE Name = 'INTERNAL_IDEAS' ];

      idea.CommunityId = community.Id;

      ID[] results = Ideas.findSimilar(idea);
   }
}
```

The following example uses a Visualforce page in conjunction with a *custom controller*, that is, a special Apex class. For more information on Visualforce, see the Visualforce Developer's Guide.

This example creates an Apex method in the controller that returns unread recent replies. You can leverage this same example for the `getAllRecentReplies` and `getReadRecentReplies` methods. For this example to work, there must be ideas posted to the zone. In addition, at least one zone member must have posted a comment to another zone member's idea or comment.

```
// Create an Apex method to retrieve the recent replies marked as unread in all communities
public class IdeasController {

   public Idea[] getUnreadRecentReplies() {
       Idea[] recentReplies;
       if (recentReplies == null) {
```

```
            Id[] recentRepliesIds = Ideas.getUnreadRecentReplies(UserInfo.getUserId(), '');

             recentReplies = [SELECT Id, Title FROM Idea WHERE Id IN :recentRepliesIds];
        }
        return recentReplies;

    }

}
```

The following is the markup for a Visualforce page that uses the above custom controller to list unread recent replies.

```
<apex:page controller="IdeasController" showHeader="false">
    <apex:dataList value="{!unreadRecentReplies}" var="recentReplyIdea">
           <a href="/apex/viewIdea?id={!recentReplyIdea.Id}">
                 <apex:outputText value="{!recentReplyIdea.Title}" escape="true"/></a>
     </apex:dataList>
</apex:page>
```

The following example uses a Visualforce page in conjunction with a custom controller to list ideas. Then, a second Visualforce page and custom controller is used to display a specific idea and mark it as read. For this example to work, there must be ideas posted to the zone.

```
// Create a controller to use on a VisualForce page to list ideas
public class IdeaListController {

    public final Idea[] ideas {get; private set;}

    public IdeaListController() {
        Integer i = 0;
        ideas = new Idea[10];
        for (Idea tmp : Database.query
('SELECT Id, Title FROM Idea WHERE Id != null AND parentIdeaId = null LIMIT 10')) {
            i++;
            ideas.add(tmp);
        }
    }
}
```

The following is the markup for a Visualforce page that uses the above custom controller to list ideas:

```
<apex:page controller="IdeaListController" tabStyle="Idea" showHeader="false">

        <apex:dataList value="{!ideas}" var="idea" id="ideaList">
            <a href="/apex/viewIdea?id={!idea.id}">
<apex:outputText value="{!idea.title}" escape="true"/></a>
        </apex:dataList>

</apex:page>
```

The following example also uses a Visualforce page and custom controller, this time, to display the idea that is selected on the above idea list page. In this example, the `markRead` method marks the selected idea and associated comments as read by the user that is currently logged in. Note that the `markRead` method is in the constructor so that the idea is marked read immediately when the user

goes to a page that uses this controller. For this example to work, there must be ideas posted to the zone. In addition, at least one zone member must have posted a comment to another zone member's idea or comment.

```
// Create an Apex method in the controller that marks all comments as read for the
// selected idea
public class ViewIdeaController {

   private final String id = System.currentPage().getParameters().get('id');

   public ViewIdeaController(ApexPages.StandardController controller) {
               Ideas.markRead(id);
   }

}
```

The following is the markup for a Visualforce page that uses the above custom controller to display the idea as read.

```
<apex:page standardController="Idea" extensions="ViewIdeaController" showHeader="false">

     <h2><apex:outputText value="{!idea.title}" /></h2>
     <apex:outputText value="{!idea.body}" />

</apex:page>
```

Ideas Methods

The following are methods for `Ideas`. All methods are static.

IN THIS SECTION:

findSimilar(idea)
Returns a list of similar ideas based on the title of the specified idea.

getAllRecentReplies(userID, communityID)
Returns ideas that have recent replies for the specified user or zone. This includes all read and unread replies.

getReadRecentReplies(userID, communityID)
Returns ideas that have recent replies marked as read.

getUnreadRecentReplies(userID, communityID)
Returns ideas that have recent replies marked as unread.

markRead(ideaID)
Marks all comments as read for the user that is currently logged in.

`findSimilar(idea)`

Returns a list of similar ideas based on the title of the specified idea.

Signature

```
public static ID[] findSimilar(Idea idea)
```

Parameters

idea
Type: Idea

Return Value

Type: ID[]

Usage

Each `findSimilar` call counts against the SOSL query limits. See Execution Governors and Limits.

getAllRecentReplies(userID, communityID)

Returns ideas that have recent replies for the specified user or zone. This includes all read and unread replies.

Signature

```
public static ID[] getAllRecentReplies(String userID, String communityID)
```

Parameters

userID
Type: String

communityID
Type: String

Return Value

Type: ID[]

Usage

Each `getAllRecentReplies` call counts against the SOQL query limits. See Execution Governors and Limits.

getReadRecentReplies(userID, communityID)

Returns ideas that have recent replies marked as read.

Signature

```
public static ID[] getReadRecentReplies(String userID, String communityID)
```

Parameters

userID
Type: String

communityID
Type: String

Return Value

Type: ID[]

Usage

Each `getReadRecentReplies` call counts against the SOQL query limits. See Execution Governors and Limits.

`getUnreadRecentReplies(userID, communityID)`

Returns ideas that have recent replies marked as unread.

Signature

```
public static ID[] getUnreadRecentReplies(String userID, String communityID)
```

Parameters

`userID`
Type: String

`communityID`
Type: String

Return Value

Type: ID[]

Usage

Each `getUnreadRecentReplies` call counts against the SOQL query limits. See Execution Governors and Limits.

`markRead(ideaID)`

Marks all comments as read for the user that is currently logged in.

Signature

```
public static Void markRead(String ideaID)
```

Parameters

`ideaID`
Type: String

Return Value

Type: Void

InstallHandler Interface

Enables custom code to run after a managed package installation or upgrade.

Namespace

System

Usage

App developers can implement this interface to specify Apex code that runs automatically after a subscriber installs or upgrades a managed package. The package install or upgrade can be customized based on details of the subscriber's organization. For instance, you can use the script to populate custom settings, create sample data, send an email to the installer, notify an external system, or kick off a batch operation to populate a new field across a large set of data.

The post install script is invoked after tests have been run, and is subject to default governor limits. It runs as a special system user that represents your package, so all operations performed by the script appear to be done by your package. You can access this user by using UserInfo. You only see this user at runtime, not while running tests.

If the script fails, the install or upgrade is aborted. Any errors in the script are emailed to the user specified in the **Notify on Apex Error** field of the package. If no user is specified, the install or upgrade details are unavailable.

The post install script has the following additional properties.

- It can initiate batch, scheduled, and future jobs.
- It can't access Session IDs.
- It can only perform callouts using an async operation. The callout occurs after the script is run and the install is complete and committed.
- It can't call another Apex class in the package if that Apex class uses the `with sharing` or `inherit sharing` keyword. These keywords can prevent the package from successfully installing. To learn more, see the Apex Developer Guide.

The `InstallHandler` interface has a single method called `onInstall`, which specifies the actions to be performed on install or upgrade.

```
public interface InstallHandler {
  void onInstall(InstallContext context)
};
```

The `onInstall` method takes a context object as its argument, which provides the following information.

- The org ID of the organization in which the installation takes place.
- The user ID of the user who initiated the installation.
- The version number of the previously installed package (specified using the `Version` class). The version is always a three-part number, such as 1.2.0.
- Whether the installation is an upgrade.
- Whether the installation is a push.

The context argument is an object whose type is the `InstallContext` interface. This interface is automatically implemented by the system. The following definition of the `InstallContext` interface shows the methods you can call on the context argument.

```
public interface InstallContext {
  ID organizationId();
  ID installerId();
  Boolean isUpgrade();
  Boolean isPush();
  Version previousVersion();
}
```

IN THIS SECTION:

InstallHandler Methods

InstallHandler Example Implementation

InstallHandler Methods

The following are methods for `InstallHandler`.

IN THIS SECTION:

onInstall(context)
Specifies the actions to be performed on install/upgrade.

onInstall(context)

Specifies the actions to be performed on install/upgrade.

Signature

```
public Void onInstall(InstallContext context)
```

Parameters

context
Type: System.InstallContext

Return Value

Type: Void

InstallHandler Example Implementation

The following sample post install script performs these actions on package install/upgrade.

- If the previous version is null, that is, the package is being installed for the first time, the script:
  - Creates a new Account called Newco and verifies that it was created.
  - Creates a new instance of the custom object Survey, called Client Satisfaction Survey.
  - Sends an email message to the subscriber confirming installation of the package.
- If the previous version is 1.0, the script creates a new instance of Survey called "Upgrading from Version 1.0".
- If the package is an upgrade, the script creates a new instance of Survey called "Sample Survey during Upgrade".
- If the upgrade is being pushed, the script creates a new instance of Survey called "Sample Survey during Push".

```
public class PostInstallClass implements InstallHandler {
  global void onInstall(InstallContext context) {
    if(context.previousVersion() == null) {
      Account a = new Account(name='Newco');
      insert(a);
```

```
      Survey__c obj = new Survey__c(name='Client Satisfaction Survey');
      insert obj;

      User u = [Select Id, Email from User where Id =:context.installerID()];
      String toAddress= u.Email;
      String[] toAddresses = new String[]{toAddress};
      Messaging.SingleEmailMessage mail =
        new Messaging.SingleEmailMessage();
      mail.setToAddresses(toAddresses);
      mail.setReplyTo('support@package.dev');
      mail.setSenderDisplayName('My Package Support');
      mail.setSubject('Package install successful');
      mail.setPlainTextBody('Thanks for installing the package.');
      Messaging.sendEmail(new Messaging.Email[] { mail });
      }
    else
      if(context.previousVersion().compareTo(new Version(1,0)) == 0) {
      Survey__c obj = new Survey__c(name='Upgrading from Version 1.0');
      insert(obj);
      }
    if(context.isUpgrade()) {
      Survey__c obj = new Survey__c(name='Sample Survey during Upgrade');
      insert obj;
      }
    if(context.isPush()) {
      Survey__c obj = new Survey__c(name='Sample Survey during Push');
      insert obj;
      }
    }
  }
```

You can test a post install script using the new `testInstall` method of the `Test` class. This method takes the following arguments.

- A class that implements the `InstallHandler` interface.
- A `Version` object that specifies the version number of the existing package.
- An optional Boolean value that is `true` if the installation is a push. The default is `false`.

This sample shows how to test a post install script implemented in the `PostInstallClass` Apex class.

```
@isTest
static void testInstallScript() {
  PostInstallClass postinstall = new PostInstallClass();
    Test.testInstall(postinstall, null);
    Test.testInstall(postinstall, new Version(1,0), true);
    List<Account> a = [Select id, name from Account where name ='Newco'];
    System.assertEquals(1, a.size(), 'Account not found');
  }
```

Integer Class

Contains methods for the Integer primitive data type.

Namespace

System

Usage

For more information on integers, see Integer Data Type.

Integer Methods

The following are methods for `Integer`.

IN THIS SECTION:

format()
Returns the integer as a string using the locale of the context user.

valueOf(stringToInteger)
Returns an Integer that contains the value of the specified String. As in Java, the String is interpreted as representing a signed decimal integer.

valueOf(fieldValue)
Converts the specified object to an Integer. Use this method to convert a history tracking field value or an object that represents an Integer value.

`format()`

Returns the integer as a string using the locale of the context user.

Signature

```
public String format()
```

Return Value

Type: String

Example

```
integer myInt = 22;
system.assertEquals('22', myInt.format());
```

`valueOf(stringToInteger)`

Returns an Integer that contains the value of the specified String. As in Java, the String is interpreted as representing a signed decimal integer.

Signature

```
public static Integer valueOf(String stringToInteger)
```

Parameters

stringToInteger
Type: String

Return Value

Type: Integer

Example

```
Integer myInt = Integer.valueOf('123');
```

valueOf(fieldValue)

Converts the specified object to an Integer. Use this method to convert a history tracking field value or an object that represents an Integer value.

Signature

```
public static Integer valueOf(Object fieldValue)
```

Parameters

fieldValue
Type: Object

Return Value

Type: Integer

Usage

Use this method with the `OldValue` or `NewValue` fields of history sObjects, such as `AccountHistory`, when the field type corresponds to an Integer type, like a number field.

Example:

Example

```
List<AccountHistory> ahlist =
  [SELECT Field,OldValue,NewValue
   FROM AccountHistory];
for(AccountHistory ah : ahlist) {
  System.debug('Field: ' + ah.Field);
  if (ah.field == 'NumberOfEmployees') {
    Integer oldValue =
      Integer.valueOf(ah.OldValue);
    Integer newValue =
      Integer.valueOf(ah.NewValue);
}
```

JSON Class

Contains methods for serializing Apex objects into JSON format and deserializing JSON content that was serialized using the `serialize` method in this class.

Namespace

System

Usage

Use the methods in the `System.JSON` class to perform round-trip JSON serialization and deserialization of Apex objects.

SEE ALSO:

Apex Developer Guide: Roundtrip Serialization and Deserialization

JSON Methods

The following are methods for `JSON`. All methods are static.

IN THIS SECTION:

createGenerator(prettyPrint)
Returns a new JSON generator.

createParser(jsonString)
Returns a new JSON parser.

deserialize(jsonString, apexType)
Deserializes the specified JSON string into an Apex object of the specified type.

deserializeStrict(jsonString, apexType)
Deserializes the specified JSON string into an Apex object of the specified type.

deserializeUntyped(jsonString)
Deserializes the specified JSON string into collections of primitive data types.

serialize(objectToSerialize)
Serializes Apex objects into JSON content.

serialize(objectToSerialize, suppressApexObjectNulls)
Suppresses `null` values when serializing Apex objects into JSON content.

serializePretty(objectToSerialize)
Serializes Apex objects into JSON content and generates indented content using the pretty-print format.

serializePretty(objectToSerialize, suppressApexObjectNulls)
Suppresses `null` values when serializing Apex objects into JSON content and generates indented content using the pretty-print format.

`createGenerator(prettyPrint)`

Returns a new JSON generator.

Signature

```
public static System.JSONGenerator createGenerator(Boolean prettyPrint)
```

Parameters

prettyPrint
Type: Boolean

Determines whether the JSON generator creates JSON content in pretty-print format with the content indented. Set to `true` to create indented content.

Return Value

Type: System.JSONGenerator

createParser(jsonString)

Returns a new JSON parser.

Signature

```
public static System.JSONParser createParser(String jsonString)
```

Parameters

jsonString
Type: String

The JSON content to parse.

Return Value

Type: System.JSONParser

deserialize(jsonString, apexType)

Deserializes the specified JSON string into an Apex object of the specified type.

Signature

```
public static Object deserialize(String jsonString, System.Type apexType)
```

Parameters

jsonString
Type: String

The JSON content to deserialize.

apexType
Type: System.Type

The Apex type of the object that this method creates after deserializing the JSON content.

Return Value

Type: Object

Usage

If the JSON content contains attributes not present in the `System.Type` argument, such as a missing field or object, deserialization fails in some circumstances. When deserializing JSON content into a custom object or an sObject using Salesforce API version 34.0 or earlier, this method throws a runtime exception when passed extraneous attributes. When deserializing JSON content into an Apex class in any API version, or into an object in API version 35.0 or later, no exception is thrown. When no exception is thrown, this method ignores extraneous attributes and parses the rest of the JSON content.

Example

The following example deserializes a `Decimal` value.

```
Decimal n = (Decimal)JSON.deserialize(
                '100.1', Decimal.class);
System.assertEquals(n, 100.1);
```

deserializeStrict(jsonString, apexType)

Deserializes the specified JSON string into an Apex object of the specified type.

Signature

```
public static Object deserializeStrict(String jsonString, System.Type apexType)
```

Parameters

jsonString
Type: String

The JSON content to deserialize.

apexType
Type: System.Type

The Apex type of the object that this method creates after deserializing the JSON content.

Return Value

Type: Object

Usage

All attributes in the JSON string must be present in the specified type. If the JSON content contains attributes not present in the `System.Type` argument, such as a missing field or object, deserialization fails in some circumstances. When deserializing JSON content with extraneous attributes into an Apex class, this method throws an exception in all API versions. However, no exception is thrown when you use this method to deserialize JSON content into a custom object or an sObject.

Example

The following example deserializes a JSON string into an object of a user-defined type represented by the `Car` class, which this example also defines.

```
public class Car {
    public String make;
    public String year;
}

public void parse() {
    Car c = (Car)JSON.deserializeStrict(
        '{"make":"SFDC","year":"2020"}',
        Car.class);
    System.assertEquals(c.make, 'SFDC');
    System.assertEquals(c.year, '2020');
}
```

deserializeUntyped(jsonString)

Deserializes the specified JSON string into collections of primitive data types.

Signature

```
public static Object deserializeUntyped(String jsonString)
```

Parameters

jsonString
Type: String

The JSON content to deserialize.

Return Value

Type: Object

Example

The following example deserializes a JSON representation of an appliance object into a map that contains primitive data types and further collections of primitive types. It then verifies the deserialized values.

```
String jsonInput = '{\n' +
    ' "description" :"An appliance",\n' +
    ' "accessories" : [ "powerCord", ' +
      '{ "right":"door handle1", ' +
        '"left":"door handle2" } ],\n' +
    ' "dimensions" : ' +
      '{ "height" : 5.5 , ' +
        '"width" : 3.0 , ' +
        '"depth" : 2.2 },\n' +
    ' "type" : null,\n' +
    ' "inventory" : 2000,\n' +
    ' "price" : 1023.45,\n' +
```

```
    ' "isShipped" : true,\n' +
    ' "modelNumber" : "123"\n' +
    '}';

Map<String, Object> m =
   (Map<String, Object>)
      JSON.deserializeUntyped(jsonInput);

System.assertEquals(
   'An appliance', m.get('description'));

List<Object> a =
   (List<Object>)m.get('accessories');
System.assertEquals('powerCord', a[0]);
Map<String, Object> a2 =
   (Map<String, Object>)a[1];
System.assertEquals(
   'door handle1', a2.get('right'));
System.assertEquals(
   'door handle2', a2.get('left'));

Map<String, Object> dim =
   (Map<String, Object>)m.get('dimensions');
System.assertEquals(
   5.5, dim.get('height'));
System.assertEquals(
   3.0, dim.get('width'));
System.assertEquals(
   2.2, dim.get('depth'));

System.assertEquals(null, m.get('type'));
System.assertEquals(
   2000, m.get('inventory'));
System.assertEquals(
   1023.45, m.get('price'));
System.assertEquals(
   true, m.get('isShipped'));
System.assertEquals(
   '123', m.get('modelNumber'));
```

serialize(objectToSerialize)

Serializes Apex objects into JSON content.

Signature

```
public static String serialize(Object objectToSerialize)
```

Parameters

objectToSerialize
Type: Object

The Apex object to serialize.

Return Value

Type: String

Example

The following example serializes a new `Datetime` value.

```
Datetime dt = Datetime.newInstance(
                Date.newInstance(
                   2011, 3, 22),
                Time.newInstance(
                   1, 15, 18, 0));
    String str = JSON.serialize(dt);
    System.assertEquals(
       '"2011-03-22T08:15:18.000Z"',
       str);
```

serialize(objectToSerialize, suppressApexObjectNulls)

Suppresses `null` values when serializing Apex objects into JSON content.

Signature

```
public static String serialize(Object objectToSerialize, Boolean suppressApexObjectNulls)
```

Parameters

objectToSerialize
Type: Object

The Apex object to serialize.

suppressApexObjectNulls
Type: Boolean

If true, remove `null` values before serializing the JSON object.

Note: This parameter doesn't apply to sObjects retrieved via SOQL.

Return Value

Type: String

Usage

This method allows you to specify whether to suppress `null` values when serializing Apex objects into JSON content.

serializePretty(objectToSerialize)

Serializes Apex objects into JSON content and generates indented content using the pretty-print format.

Signature

```
public static String serializePretty(Object objectToSerialize)
```

Parameters

objectToSerialize
Type: Object

The Apex object to serialize.

Return Value

Type: String

serializePretty(objectToSerialize, suppressApexObjectNulls)

Suppresses `null` values when serializing Apex objects into JSON content and generates indented content using the pretty-print format.

Signature

```
public static String serializePretty(Object objectToSerialize, Boolean
suppressApexObjectNulls)
```

Parameters

objectToSerialize
Type: Object

The Apex object to serialize.

suppressApexObjectNulls
Type: Boolean

If true, remove `null` values before serializing the JSON object.

Note: This parameter doesn't apply to sObjects retrieved via SOQL.

Return Value

Type: String

JSONGenerator Class

Contains methods used to serialize objects into JSON content using the standard JSON encoding.

Namespace

System

Usage

The `System.JSONGenerator` class is provided to enable the generation of standard JSON-encoded content and gives you more control on the structure of the JSON output.

SEE ALSO:

Apex Developer Guide: JSON Generator

JSONGenerator Methods

The following are methods for `JSONGenerator`. All are instance methods.

IN THIS SECTION:

close()
Closes the JSON generator.

getAsString()
Returns the generated JSON content.

isClosed()
Returns `true` if the JSON generator is closed; otherwise, returns `false`.

writeBlob(blobValue)
Writes the specified `Blob` value as a base64-encoded string.

writeBlobField(fieldName, blobValue)
Writes a field name and value pair using the specified field name and BLOB value.

writeBoolean(blobValue)
Writes the specified Boolean value.

writeBooleanField(fieldName, booleanValue)
Writes a field name and value pair using the specified field name and Boolean value.

writeDate(dateValue)
Writes the specified date value in the ISO-8601 format.

writeDateField(fieldName, dateValue)
Writes a field name and value pair using the specified field name and date value. The date value is written in the ISO-8601 format.

writeDateTime(datetimeValue)
Writes the specified date and time value in the ISO-8601 format.

writeDateTimeField(fieldName, datetimeValue)
Writes a field name and value pair using the specified field name and date and time value. The date and time value is written in the ISO-8601 format.

writeEndArray()
Writes the ending marker of a JSON array (']').

writeEndObject()
Writes the ending marker of a JSON object ('}').

writeFieldName(fieldName)
Writes a field name.

writeId(identifier)

Writes the specified ID value.

writeIdField(fieldName, identifier)

Writes a field name and value pair using the specified field name and identifier value.

writeNull()

Writes the JSON null literal value.

writeNullField(fieldName)

Writes a field name and value pair using the specified field name and the JSON null literal value.

writeNumber(number)

Writes the specified decimal value.

writeNumber(number)

Writes the specified double value.

writeNumber(number)

Writes the specified integer value.

writeNumber(number)

Writes the specified long value.

writeNumberField(fieldName, number)

Writes a field name and value pair using the specified field name and decimal value.

writeNumberField(fieldName, number)

Writes a field name and value pair using the specified field name and double value.

writeNumberField(fieldName, number)

Writes a field name and value pair using the specified field name and integer value.

writeNumberField(fieldName, number)

Writes a field name and value pair using the specified field name and long value.

writeObject(anyObject)

Writes the specified Apex object in JSON format.

writeObjectField(fieldName, value)

Writes a field name and value pair using the specified field name and Apex object.

writeStartArray()

Writes the starting marker of a JSON array ('[').

writeStartObject()

Writes the starting marker of a JSON object ('{').

writeString(stringValue)

Writes the specified string value.

writeStringField(fieldName, stringValue)

Writes a field name and value pair using the specified field name and string value.

writeTime(timeValue)

Writes the specified time value in the ISO-8601 format.

writeTimeField(fieldName, timeValue)

Writes a field name and value pair using the specified field name and time value in the ISO-8601 format.

close()

Closes the JSON generator.

Signature

```
public Void close()
```

Return Value

Type: Void

Usage

No more content can be written after the JSON generator is closed.

getAsString()

Returns the generated JSON content.

Signature

```
public String getAsString()
```

Return Value

Type: String

Usage

This method closes the JSON generator if it isn't closed already.

isClosed()

Returns `true` if the JSON generator is closed; otherwise, returns `false`.

Signature

```
public Boolean isClosed()
```

Return Value

Type: Boolean

writeBlob(blobValue)

Writes the specified `Blob` value as a base64-encoded string.

Signature

```
public Void writeBlob(Blob blobValue)
```

Parameters

blobValue
Type: Blob

Return Value

Type: Void

writeBlobField(fieldName, blobValue)

Writes a field name and value pair using the specified field name and BLOB value.

Signature

```
public Void writeBlobField(String fieldName, Blob blobValue)
```

Parameters

fieldName
Type: String

blobValue
Type: Blob

Return Value

Type: Void

writeBoolean(blobValue)

Writes the specified Boolean value.

Signature

```
public Void writeBoolean(Boolean blobValue)
```

Parameters

blobValue
Type: Boolean

Return Value

Type: Void

writeBooleanField(fieldName, booleanValue)

Writes a field name and value pair using the specified field name and Boolean value.

Signature

```
public Void writeBooleanField(String fieldName, Boolean booleanValue)
```

Parameters

fieldName
Type: String

booleanValue
Type: Boolean

Return Value

Type: Void

writeDate(dateValue)

Writes the specified date value in the ISO-8601 format.

Signature

```
public Void writeDate(Date dateValue)
```

Parameters

dateValue
Type: Date

Return Value

Type: Void

writeDateField(fieldName, dateValue)

Writes a field name and value pair using the specified field name and date value. The date value is written in the ISO-8601 format.

Signature

```
public Void writeDateField(String fieldName, Date dateValue)
```

Parameters

fieldName
Type: String

dateValue
Type: Date

Return Value

Type: Void

writeDateTime(datetimeValue)

Writes the specified date and time value in the ISO-8601 format.

Signature

```
public Void writeDateTime(Datetime datetimeValue)
```

Parameters

datetimeValue
Type: Datetime

Return Value

Type: Void

writeDateTimeField(fieldName, datetimeValue)

Writes a field name and value pair using the specified field name and date and time value. The date and time value is written in the ISO-8601 format.

Signature

```
public Void writeDateTimeField(String fieldName, Datetime datetimeValue)
```

Parameters

fieldName
Type: String

datetimeValue
Type: Datetime

Return Value

Type: Void

writeEndArray()

Writes the ending marker of a JSON array (']').

Signature

```
public Void writeEndArray()
```

Return Value

Type: Void

writeEndObject()

Writes the ending marker of a JSON object ('}').

Signature

```
public Void writeEndObject()
```

Return Value

Type: Void

writeFieldName(fieldName)

Writes a field name.

Signature

```
public Void writeFieldName(String fieldName)
```

Parameters

fieldName
Type: String

Return Value

Type: Void

writeId(identifier)

Writes the specified ID value.

Signature

```
public Void writeId(ID identifier)
```

Parameters

identifier
Type: ID

Return Value

Type: Void

writeIdField(fieldName, identifier)

Writes a field name and value pair using the specified field name and identifier value.

Signature

```
public Void writeIdField(String fieldName, Id identifier)
```

Parameters

fieldName
Type: String

identifier
Type: ID

Return Value

Type: Void

writeNull()

Writes the JSON null literal value.

Signature

```
public Void writeNull()
```

Return Value

Type: Void

writeNullField(fieldName)

Writes a field name and value pair using the specified field name and the JSON null literal value.

Signature

```
public Void writeNullField(String fieldName)
```

Parameters

fieldName
Type: String

Return Value

Type: Void

writeNumber(number)

Writes the specified decimal value.

Signature

```
public Void writeNumber(Decimal number)
```

Parameters

number
Type: Decimal

Return Value

Type: Void

writeNumber(number)

Writes the specified double value.

Signature

```
public Void writeNumber(Double number)
```

Parameters

number
Type: Double

Return Value

Type: Void

writeNumber(number)

Writes the specified integer value.

Signature

```
public Void writeNumber(Integer number)
```

Parameters

number
Type: Integer

Return Value

Type: Void

writeNumber(number)

Writes the specified long value.

Signature

```
public Void writeNumber(Long number)
```

Parameters

number
Type: Long

Return Value

Type: Void

writeNumberField(fieldName, number)

Writes a field name and value pair using the specified field name and decimal value.

Signature

```
public Void writeNumberField(String fieldName, Decimal number)
```

Parameters

fieldName
Type: String

number
Type: Decimal

Return Value

Type: Void

writeNumberField(fieldName, number)

Writes a field name and value pair using the specified field name and double value.

Signature

```
public Void writeNumberField(String fieldName, Double number)
```

Parameters

fieldName
Type: String

number
Type: Double

Return Value

Type: Void

writeNumberField(fieldName, number)

Writes a field name and value pair using the specified field name and integer value.

Signature

```
public Void writeNumberField(String fieldName, Integer number)
```

Parameters

fieldName
Type: String

number
Type: Integer

Return Value

Type: Void

writeNumberField(fieldName, number)

Writes a field name and value pair using the specified field name and long value.

Signature

```
public Void writeNumberField(String fieldName, Long number)
```

Parameters

fieldName
Type: String

number
Type: Long

Return Value

Type: Void

writeObject(anyObject)

Writes the specified Apex object in JSON format.

Signature

```
public Void writeObject(Object anyObject)
```

Parameters

anyObject
Type: Object

Return Value

Type: Void

writeObjectField(fieldName, value)

Writes a field name and value pair using the specified field name and Apex object.

Signature

```
public Void writeObjectField(String fieldName, Object value)
```

Parameters

fieldName
Type: String

value
Type: Object

Return Value

Type: Void

writeStartArray()

Writes the starting marker of a JSON array ('[').

Signature

```
public Void writeStartArray()
```

Return Value

Type: Void

writeStartObject()

Writes the starting marker of a JSON object ('{').

Signature

```
public Void writeStartObject()
```

Return Value

Type: Void

writeString(stringValue)

Writes the specified string value.

Signature

```
public Void writeString(String stringValue)
```

Parameters

stringValue
Type: String

Return Value

Type: Void

writeStringField(fieldName, stringValue)

Writes a field name and value pair using the specified field name and string value.

Signature

```
public Void writeStringField(String fieldName, String stringValue)
```

Parameters

fieldName
Type: String

stringValue
Type: String

Return Value

Type: Void

writeTime(timeValue)

Writes the specified time value in the ISO-8601 format.

Signature

```
public Void writeTime(Time timeValue)
```

Parameters

timeValue
Type: Time

Return Value

Type: Void

writeTimeField(fieldName, timeValue)

Writes a field name and value pair using the specified field name and time value in the ISO-8601 format.

Signature

```
public Void writeTimeField(String fieldName, Time timeValue)
```

Parameters

fieldName
Type: String

timeValue
Type: Time

Return Value

Type: Void

JSONParser Class

Represents a parser for JSON-encoded content.

Namespace

System

Usage

Use the `System.JSONParser` methods to parse a response that's returned from a call to an external service that is in JSON format, such as a JSON-encoded response of a Web service callout.

SEE ALSO:

Apex Developer Guide: JSON Parsing

JSONParser Methods

The following are methods for `JSONParser`. All are instance methods.

IN THIS SECTION:

clearCurrentToken()
Removes the current token.

getBlobValue()
Returns the current token as a BLOB value.

getBooleanValue()
Returns the current token as a Boolean value.

getCurrentName()
Returns the name associated with the current token.

getCurrentToken()
Returns the token that the parser currently points to or `null` if there's no current token.

getDatetimeValue()
Returns the current token as a date and time value.

getDateValue()
Returns the current token as a date value.

getDecimalValue()
Returns the current token as a decimal value.

getDoubleValue()
Returns the current token as a double value.

getIdValue()
Returns the current token as an ID value.

getIntegerValue()
Returns the current token as an integer value.

getLastClearedToken()
Returns the last token that was cleared by the `clearCurrentToken` method.

getLongValue()
Returns the current token as a long value.

getText()
Returns the textual representation of the current token or `null` if there's no current token.

getTimeValue()
Returns the current token as a time value.

hasCurrentToken()
Returns `true` if the parser currently points to a token; otherwise, returns `false`.

nextToken()
Returns the next token or `null` if the parser has reached the end of the input stream.

nextValue()
Returns the next token that is a value type or `null` if the parser has reached the end of the input stream.

readValueAs(apexType)
Deserializes JSON content into an object of the specified Apex type and returns the deserialized object.

readValueAsStrict(apexType)
Deserializes JSON content into an object of the specified Apex type and returns the deserialized object. All attributes in the JSON content must be present in the specified type.

skipChildren()
Skips all child tokens of type `JSONToken.START_ARRAY` and `JSONToken.START_OBJECT` that the parser currently points to.

`clearCurrentToken()`

Removes the current token.

Signature

```
public Void clearCurrentToken()
```

Return Value

Type: Void

Usage

After this method is called, a call to `hasCurrentToken` returns `false` and a call to `getCurrentToken` returns `null`. You can retrieve the cleared token by calling `getLastClearedToken`.

getBlobValue()

Returns the current token as a BLOB value.

Signature

```
public Blob getBlobValue()
```

Return Value

Type: Blob

Usage

The current token must be of type `JSONToken.VALUE_STRING` and must be Base64-encoded.

getBooleanValue()

Returns the current token as a Boolean value.

Signature

```
public Boolean getBooleanValue()
```

Return Value

Type: Boolean

Usage

The current token must be of type `JSONToken.VALUE_TRUE` or `JSONToken.VALUE_FALSE`.

The following example parses a sample JSON string and retrieves a Boolean value.

```
String JSONContent =
    '{"isActive":true}';
JSONParser parser =
   JSON.createParser(JSONContent);
// Advance to the start object marker.
parser.nextToken();
// Advance to the next value.
parser.nextValue();
// Get the Boolean value.
Boolean isActive = parser.getBooleanValue();
```

getCurrentName()

Returns the name associated with the current token.

Signature

```
public String getCurrentName()
```

Return Value

Type: String

Usage

If the current token is of type `JSONToken.FIELD_NAME`, this method returns the same value as `getText`. If the current token is a value, this method returns the field name that precedes this token. For other values such as array values or root-level values, this method returns `null`.

The following example parses a sample JSON string. It advances to the field value and retrieves its corresponding field name.

Example

```
String JSONContent = '{"firstName":"John"}';
JSONParser parser =
   JSON.createParser(JSONContent);
// Advance to the start object marker.
parser.nextToken();
// Advance to the next value.
parser.nextValue();
// Get the field name for the current value.
String fieldName = parser.getCurrentName();
// Get the textual representation
// of the value.
String fieldValue = parser.getText();
```

getCurrentToken()

Returns the token that the parser currently points to or `null` if there's no current token.

Signature

```
public System.JSONToken getCurrentToken()
```

Return Value

Type: System.JSONToken

Usage

The following example iterates through all the tokens in a sample JSON string.

```
String JSONContent = '{"firstName":"John"}';
JSONParser parser =
```

```
   JSON.createParser(JSONContent);
// Advance to the next token.
while (parser.nextToken() != null) {
    System.debug('Current token: ' +
        parser.getCurrentToken());
}
```

getDatetimeValue()

Returns the current token as a date and time value.

Signature

```
public Datetime getDatetimeValue()
```

Return Value

Type: Datetime

Usage

The current token must be of type `JSONToken.VALUE_STRING` and must represent a `Datetime` value in the ISO-8601 format.

The following example parses a sample JSON string and retrieves a Datetime value.

```
String JSONContent =
'{"transactionDate":"2011-03-22T13:01:23"}';
JSONParser parser =
   JSON.createParser(JSONContent);
// Advance to the start object marker.
parser.nextToken();
// Advance to the next value.
parser.nextValue();
// Get the transaction date.
Datetime transactionDate =
   parser.getDatetimeValue();
```

getDateValue()

Returns the current token as a date value.

Signature

```
public Date getDateValue()
```

Return Value

Type: Date

Usage

The current token must be of type `JSONToken.VALUE_STRING` and must represent a `Date` value in the ISO-8601 format.

The following example parses a sample JSON string and retrieves a Date value.

```
String JSONContent =
   '{"dateOfBirth":"2011-03-22"}';
JSONParser parser =
   JSON.createParser(JSONContent);
// Advance to the start object marker.
parser.nextToken();
// Advance to the next value.
parser.nextValue();
// Get the date of birth.
Date dob = parser.getDateValue();
```

getDecimalValue()

Returns the current token as a decimal value.

Signature

```
public Decimal getDecimalValue()
```

Return Value

Type: Decimal

Usage

The current token must be of type `JSONToken.VALUE_NUMBER_FLOAT` or `JSONToken.VALUE_NUMBER_INT` and is a numerical value that can be converted to a value of type `Decimal`.

The following example parses a sample JSON string and retrieves a Decimal value.

```
String JSONContent =
   '{"GPA":3.8}';
JSONParser parser =
   JSON.createParser(JSONContent);
// Advance to the start object marker.
parser.nextToken();
// Advance to the next value.
parser.nextValue();
// Get the GPA score.
Decimal gpa = parser.getDecimalValue();
```

getDoubleValue()

Returns the current token as a double value.

Signature

```
public Double getDoubleValue()
```

Return Value

Type: Double

Usage

The current token must be of type `JSONToken.VALUE_NUMBER_FLOAT` and is a numerical value that can be converted to a value of type `Double`.

The following example parses a sample JSON string and retrieves a Double value.

```
String JSONContent =
   '{"GPA":3.8}';
JSONParser parser =
   JSON.createParser(JSONContent);
// Advance to the start object marker.
parser.nextToken();
// Advance to the next value.
parser.nextValue();
// Get the GPA score.
Double gpa = parser.getDoubleValue();
```

getIdValue()

Returns the current token as an ID value.

Signature

```
public ID getIdValue()
```

Return Value

Type: ID

Usage

The current token must be of type `JSONToken.VALUE_STRING` and must be a valid `ID`.

The following example parses a sample JSON string and retrieves an ID value.

```
String JSONContent =
   '{"recordId":"001R0000002nO6H"}';
JSONParser parser =
   JSON.createParser(JSONContent);
// Advance to the start object marker.
parser.nextToken();
// Advance to the next value.
parser.nextValue();
// Get the record ID.
ID recordID = parser.getIdValue();
```

getIntegerValue()

Returns the current token as an integer value.

Signature

```
public Integer getIntegerValue()
```

Return Value

Type: Integer

Usage

The current token must be of type `JSONToken.VALUE_NUMBER_INT` and must represent an `Integer`.

The following example parses a sample JSON string and retrieves an Integer value.

```
String JSONContent =
   '{"recordCount":10}';
JSONParser parser =
   JSON.createParser(JSONContent);
// Advance to the start object marker.
parser.nextToken();
// Advance to the next value.
parser.nextValue();
// Get the record count.
Integer count = parser.getIntegerValue();
```

`getLastClearedToken()`

Returns the last token that was cleared by the `clearCurrentToken` method.

Signature

```
public System.JSONToken getLastClearedToken()
```

Return Value

Type: System.JSONToken

`getLongValue()`

Returns the current token as a long value.

Signature

```
public Long getLongValue()
```

Return Value

Type: Long

Usage

The current token must be of type `JSONToken.VALUE_NUMBER_INT` and is a numerical value that can be converted to a value of type `Long` .

The following example parses a sample JSON string and retrieves a Long value.

```
String JSONContent =
   '{"recordCount":2097531021}';
JSONParser parser =
   JSON.createParser(JSONContent);
// Advance to the start object marker.
parser.nextToken();
// Advance to the next value.
parser.nextValue();
// Get the record count.
Long count = parser.getLongValue();
```

`getText()`

Returns the textual representation of the current token or `null` if there's no current token.

Signature

```
public String getText()
```

Return Value

Type: String

Usage

No current token exists, and therefore this method returns `null`, if `nextToken` has not been called yet for the first time or if the parser has reached the end of the input stream.

`getTimeValue()`

Returns the current token as a time value.

Signature

```
public Time getTimeValue()
```

Return Value

Type: Time

Usage

The current token must be of type `JSONToken.VALUE_STRING` and must represent a `Time` value in the ISO-8601 format.

The following example parses a sample JSON string and retrieves a Datetime value.

```
String JSONContent =
   '{"arrivalTime":"18:05"}';
JSONParser parser =
   JSON.createParser(JSONContent);
// Advance to the start object marker.
```

```
parser.nextToken();
// Advance to the next value.
parser.nextValue();
// Get the arrival time.
Time arrivalTime = parser.getTimeValue();
```

hasCurrentToken()

Returns `true` if the parser currently points to a token; otherwise, returns `false`.

Signature

```
public Boolean hasCurrentToken()
```

Return Value

Type: Boolean

nextToken()

Returns the next token or `null` if the parser has reached the end of the input stream.

Signature

```
public System.JSONToken nextToken()
```

Return Value

Type: System.JSONToken

Usage

Advances the stream enough to determine the type of the next token, if any.

nextValue()

Returns the next token that is a value type or `null` if the parser has reached the end of the input stream.

Signature

```
public System.JSONToken nextValue()
```

Return Value

Type: System.JSONToken

Usage

Advances the stream enough to determine the type of the next token that is of a value type, if any, including a JSON array and object start and end markers.

readValueAs(apexType)

Deserializes JSON content into an object of the specified Apex type and returns the deserialized object.

Signature

```
public Object readValueAs(System.Type apexType)
```

Parameters

apexType
Type: System.Type

The *apexType* argument specifies the type of the object that this method returns after deserializing the current value.

Return Value

Type: Object

Usage

If the JSON content contains attributes not present in the `System.Type` argument, such as a missing field or object, deserialization fails in some circumstances. When deserializing JSON content into a custom object or an sObject using Salesforce API version 34.0 or earlier, this method throws a runtime exception when passed extraneous attributes. When deserializing JSON content into an Apex class in any API version, or into an object in API version 35.0 or later, no exception is thrown. When no exception is thrown, this method ignores extraneous attributes and parses the rest of the JSON content.

Example

The following example parses a sample JSON string and retrieves a Datetime value. Before being able to run this sample, you must create a new Apex class as follows:

```
public class Person {
    public String name;
    public String phone;
}
```

Next, insert the following sample in a class method:

```
// JSON string that contains a Person object.
String JSONContent =
    '{"person":{' +
        '"name":"John Smith",' +
        '"phone":"555-1212"}}';
JSONParser parser =
   JSON.createParser(JSONContent);
// Make calls to nextToken()
// to point to the second
// start object marker.
parser.nextToken();
parser.nextToken();
parser.nextToken();
// Retrieve the Person object
// from the JSON string.
```

```
Person obj =
   (Person)parser.readValueAs(
      Person.class);
System.assertEquals(
   obj.name, 'John Smith');
System.assertEquals(
   obj.phone, '555-1212');
```

readValueAsStrict(apexType)

Deserializes JSON content into an object of the specified Apex type and returns the deserialized object. All attributes in the JSON content must be present in the specified type.

Signature

```
public Object readValueAsStrict(System.Type apexType)
```

Parameters

apexType
Type: System.Type

The *apexType* argument specifies the type of the object that this method returns after deserializing the current value.

Return Value

Type: Object

Usage

If the JSON content contains attributes not present in the `System.Type` argument, such as a missing field or object, deserialization fails in some circumstances. When deserializing JSON content with extraneous attributes into an Apex class, this method throws an exception in all API versions. However, no exception is thrown when you use this method to deserialize JSON content into a custom object or an sObject.

The following example parses a sample JSON string and retrieves a Datetime value. Before being able to run this sample, you must create a new Apex class as follows:

```
public class Person {
    public String name;
    public String phone;
}
```

Next, insert the following sample in a class method:

```
// JSON string that contains a Person object.
String JSONContent =
    '{"person":{' +
        '"name":"John Smith",' +
        '"phone":"555-1212"}}';
JSONParser parser =
   JSON.createParser(JSONContent);
// Make calls to nextToken()
// to point to the second
```

```
// start object marker.
parser.nextToken();
parser.nextToken();
parser.nextToken();
// Retrieve the Person object
// from the JSON string.
Person obj =
   (Person)parser.readValueAsStrict(
      Person.class);
System.assertEquals(
   obj.name, 'John Smith');
System.assertEquals(
   obj.phone, '555-1212');
```

skipChildren()

Skips all child tokens of type `JSONToken.START_ARRAY` and `JSONToken.START_OBJECT` that the parser currently points to.

Signature

```
public Void skipChildren()
```

Return Value

Type: Void

JSONToken Enum

Contains all token values used for parsing JSON content.

Namespace

System

Enum Value	Description
END_ARRAY	The ending of an array value. This token is returned when ']' is encountered.
END_OBJECT	The ending of an object value. This token is returned when '}' is encountered.
FIELD_NAME	A string token that is a field name.
NOT_AVAILABLE	The requested token isn't available.
START_ARRAY	The start of an array value. This token is returned when '[' is encountered.
START_OBJECT	The start of an object value. This token is returned when '{' is encountered.

Enum Value	Description
VALUE_EMBEDDED_OBJECT	An embedded object that isn't accessible as a typical object structure that includes the start and end object tokens START_OBJECT and END_OBJECT but is represented as a raw object.
VALUE_FALSE	The literal "false" value.
VALUE_NULL	The literal "null" value.
VALUE_NUMBER_FLOAT	A float value.
VALUE_NUMBER_INT	An integer value.
VALUE_STRING	A string value.
VALUE_TRUE	A value that corresponds to the "true" string literal.

Label Class

Provides methods to retrieve a custom label or to check if translation exists for a label in a specific language and namespace. Label names are dynamically resolved at run time, overriding the user's current language if a translation exists for the requested language. You can't access labels that are protected in a different namespace.

Namespace

System

Usage

Custom labels enable developers to create multilingual applications by automatically presenting information (for example, help text or error messages) in a user's native language. Custom labels have a limit of 1000 characters and can be accessed from Apex classes, Visualforce pages, Lightning pages, or Lightning components. For more information on custom labels, see Custom Labels in *Salesforce Help*.The label values can be translated into any language Salesforce supports. For supported languages, see Supported Languages in *Salesforce Help*.

- To define custom labels, from Setup, in the Quick Find box, enter `Custom Labels`, and then select **Custom Labels**.
- To assign translated values, turn on Translation Workbench and add translation mappings.
- To retrieve the label for a default language setting or for a language and namespace, use `System.Label.get(namespace, label, language)`.
- To check if translation exists for a label and language in a namespace, use `Label.translationExists(namespace, label, language)`.

In Apex code, you can refer to or instantiate a Label like this:

```
System.Label.myLabelName
```

For information on passing in labels into Aura components, see Getting Labels in Apex in the *Lightning Aura Components Developer Guide*.

Examples

This example returns `True` if an English label called *`MyLabel`* exists in the *`MyNamespace`* namespace.

```
boolean exists = Label.translationExists('MyNamespace', 'MyLabel', 'en')
```

This example retrieves the custom label translation text for *`MyLabel`* in French.

```
String value = Label.get('MyNamespace', 'MyLabel', 'fr')
```

IN THIS SECTION:

Label Methods

Label Methods

The following are methods for `Label`.

IN THIS SECTION:

get(namespace, label)
Retrieve a custom label for the specified namespace and a default language setting.

get(namespace, label, language)
Retrieve a custom label for the specified namespace and language.

translationExists(namespace, label, language)
Check if translation exists for a label and language in a namespace.

`get(namespace, label)`

Retrieve a custom label for the specified namespace and a default language setting.

Signature

```
public static String get(String namespace, String label)
```

Parameters

`namespace`
Type: String

If the namespace name is null, it defaults to the package namespace. If the namespace name is an empty string, it implies the org namespace.

`label`
Type: String

The label name cannot be null or an empty string.

Return Value

Type: String

get(namespace, label, language)

Retrieve a custom label for the specified namespace and language.

Signature

```
public static String get(String namespace, String label, String language)
```

Parameters

namespace
Type: String

If the namespace name is null, it defaults to the package namespace. If the namespace name is an empty string, it implies the org namespace.

label
Type: String

The label name cannot be null or an empty string.

language
Type: String

This parameter must be a valid language ISO code. See the Platform-Only Languages section in Supported Languages in Salesforce Help.

Return Value

Type: String

translationExists(namespace, label, language)

Check if translation exists for a label and language in a namespace.

Signature

```
public static Boolean translationExists(string namespace, string label, string language)
```

Parameters

namespace
Type: String

If the namespace name is null, it defaults to the package namespace. If the namespace name is an empty string, it implies the org namespace.

label
Type: String

The label name cannot be null or an empty string.

language
Type: String

This parameter must be a valid language ISO code. See the Platform-Only Languages section in Supported Languages in Salesforce Help.

Return Value

Type: Boolean

Limits Class

Contains methods that return limit information for specific resources.

Namespace

System

Usage

The Limits methods return the specific limit for the particular governor, such as the number of calls of a method or the amount of heap size remaining.

Because Apex runs in a multitenant environment, the Apex runtime engine strictly enforces a number of limits to ensure that runaway Apex doesn't monopolize shared resources.

None of the Limits methods require an argument. The format of the limits methods is as follows:

```
myDMLLimit = Limits.getDMLStatements();
```

There are two versions of every method: the first returns the amount of the resource that has been used while the second version contains the word limit and returns the total amount of the resource that is available.

See Execution Governors and Limits.

Limits Methods

The following are methods for `Limits`. All methods are static.

IN THIS SECTION:

getAggregateQueries()
Returns the number of aggregate queries that have been processed with any SOQL query statement.

getLimitAggregateQueries()
Returns the total number of aggregate queries that can be processed with SOQL query statements.

getApexCursorRows()
Gets the number of rows returned by an Apex cursor.

getLimitApexCursorRows()
Gets the maximum number of rows that can be returned by an Apex cursor.

getFetchCallsOnApexCursor()
Gets the number of fetch calls on an Apex cursor.

getLimitFetchCallsOnApexCursor()
Gets the maximum number of fetch calls that can be made on an Apex cursor.

getAsyncCalls()
Reserved for future use.

getLimitAsyncCalls()

Reserved for future use.

getCallouts()

Returns the number of Web service statements that have been processed.

getChildRelationshipsDescribes()

Deprecated. Returns the number of child relationship objects that have been returned.

getLimitCallouts()

Returns the total number of Web service statements that can be processed.

getCpuTime()

Returns the CPU time (in milliseconds) that has been used in the current transaction.

getLimitCpuTime()

Returns the maximum CPU time (in milliseconds) that can be used in a transaction.

getDMLRows()

Returns the number of records that have been processed with any statement that counts against DML limits, such as DML statements, the `Database.emptyRecycleBin` method, and other methods.

getLimitDMLRows()

Returns the total number of records that can be processed with any statement that counts against DML limits, such as DML statements, the `database.EmptyRecycleBin` method, and other methods.

getDMLStatements()

Returns the number of DML statements (such as `insert`, `update` or the `database.EmptyRecycleBin` method) that have been called.

getLimitDMLStatements()

Returns the total number of DML statements or the `database.EmptyRecycleBin` methods that can be called.

getEmailInvocations()

Returns the number of email invocations (such as `sendEmail`) that have been called.

getLimitEmailInvocations()

Returns the total number of email invocation (such as `sendEmail`) that can be called.

getFindSimilarCalls()

Deprecated. Returns the same value as `getSoslQueries`. The number of `findSimilar` methods is no longer a separate limit, but is tracked as the number of SOSL queries issued.

getLimitFindSimilarCalls()

Deprecated. Returns the same value as `getLimitSoslQueries`. The number of `findSimilar` methods is no longer a separate limit, but is tracked as the number of SOSL queries issued.

getFutureCalls()

Returns the number of methods with the `future` annotation that have been executed (not necessarily completed).

getLimitFutureCalls()

Returns the total number of methods with the `future` annotation that can be executed (not necessarily completed).

getHeapSize()

Returns the approximate amount of memory (in bytes) that has been used for the heap.

getLimitHeapSize()

Returns the total amount of memory (in bytes) that can be used for the heap.

getMobilePushApexCalls()

Returns the number of Apex calls that have been used by mobile push notifications during the current metering interval.

getLimitMobilePushApexCalls()

Returns the total number of Apex calls that are allowed per transaction for mobile push notifications.

getPublishImmediateDML()

Returns the number of `EventBus.publish` calls that have been made for platform events configured to publish immediately.

getLimitPublishImmediateDML()

Returns the total number of `EventBus.publish` statements that can be called for platform events configured to publish immediately.

getQueries()

Returns the number of SOQL queries that have been issued.

getLimitQueries()

Returns the total number of SOQL queries that can be issued.

getQueryLocatorRows()

Returns the number of records that have been returned by the `Database.getQueryLocator` method.

getLimitQueryLocatorRows()

Returns the total number of records that can be returned by the `Database.getQueryLocator` method.

getQueryRows()

Returns the number of records that have been returned by issuing SOQL queries.

getLimitQueryRows()

Returns the total number of records that can be returned by issuing SOQL queries.

getQueueableJobs()

Returns the number of queueable jobs that have been added to the queue per transaction. A queueable job corresponds to a class that implements the `Queueable` interface.

getLimitQueueableJobs()

Returns the maximum number of queueable jobs that can be added to the queue per transaction. A queueable job corresponds to a class that implements the `Queueable` interface.

getRunAs()

Deprecated. Returns the same value as `getDMLStatements`.

getLimitRunAs()

Deprecated. Returns the same value as `getLimitDMLStatements`.

getSavepointRollbacks()

Deprecated. Returns the same value as `getDMLStatements`.

getLimitSavepointRollbacks()

Deprecated. Returns the same value as `getLimitDMLStatements`.

getSavepoints()

Deprecated. Returns the same value as `getDMLStatements`.

getLimitSavepoints()

Deprecated. Returns the same value as `getLimitDMLStatements`.

getSoslQueries()

Returns the number of SOSL queries that have been issued.

getLimitSoslQueries()
Returns the total number of SOSL queries that can be issued.

getAggregateQueries()

Returns the number of aggregate queries that have been processed with any SOQL query statement.

Signature

```
public static Integer getAggregateQueries()
```

Return Value

Type: Integer

getLimitAggregateQueries()

Returns the total number of aggregate queries that can be processed with SOQL query statements.

Signature

```
public static Integer getLimitAggregateQueries()
```

Return Value

Type: Integer

getApexCursorRows()

Gets the number of rows returned by an Apex cursor.

Signature

```
public static Integer getApexCursorRows()
```

Return Value

Type: Integer

getLimitApexCursorRows()

Gets the maximum number of rows that can be returned by an Apex cursor.

Signature

```
public static Integer getLimitApexCursorRows()
```

Return Value

Type: Integer

getFetchCallsOnApexCursor()

Gets the number of fetch calls on an Apex cursor.

Signature

```
public static Integer getFetchCallsOnApexCursor()
```

Return Value

Type: Integer

getLimitFetchCallsOnApexCursor()

Gets the maximum number of fetch calls that can be made on an Apex cursor.

Signature

```
public static Integer getLimitFetchCallsOnApexCursor()
```

Return Value

Type: Integer

getAsyncCalls()

Reserved for future use.

Signature

```
public static Integer getAsyncCalls()
```

Return Value

Type: Integer

getLimitAsyncCalls()

Reserved for future use.

Signature

```
public static Integer getLimitAsyncCalls()
```

Return Value

Type: Integer

getCallouts()

Returns the number of Web service statements that have been processed.

Signature

```
public static Integer getCallouts()
```

Return Value

Type: Integer

getChildRelationshipsDescribes()

Deprecated. Returns the number of child relationship objects that have been returned.

Signature

```
public static Integer getChildRelationshipsDescribes()
```

Return Value

Type: Integer

Usage

Note: Because describe limits are no longer enforced in any API version, this method is no longer available. In API version 30.0 and earlier, this method is available but is deprecated.

getLimitCallouts()

Returns the total number of Web service statements that can be processed.

Signature

```
public static Integer getLimitCallouts()
```

Return Value

Type: Integer

getCpuTime()

Returns the CPU time (in milliseconds) that has been used in the current transaction.

Signature

```
public static Integer getCpuTime()
```

Return Value

Type: Integer

getLimitCpuTime()

Returns the maximum CPU time (in milliseconds) that can be used in a transaction.

Signature

```
public static Integer getLimitCpuTime()
```

Return Value

Type: Integer

getDMLRows()

Returns the number of records that have been processed with any statement that counts against DML limits, such as DML statements, the `Database.emptyRecycleBin` method, and other methods.

Signature

```
public static Integer getDMLRows()
```

Return Value

Type: Integer

getLimitDMLRows()

Returns the total number of records that can be processed with any statement that counts against DML limits, such as DML statements, the `database.EmptyRecycleBin` method, and other methods.

Signature

```
public static Integer getLimitDMLRows()
```

Return Value

Type: Integer

getDMLStatements()

Returns the number of DML statements (such as `insert`, `update` or the `database.EmptyRecycleBin` method) that have been called.

Signature

```
public static Integer getDMLStatements()
```

Return Value

Type: Integer

`getLimitDMLStatements()`

Returns the total number of DML statements or the `database.EmptyRecycleBin` methods that can be called.

Signature

```
public static Integer getLimitDMLStatements()
```

Return Value

Type: Integer

`getEmailInvocations()`

Returns the number of email invocations (such as `sendEmail`) that have been called.

Signature

```
public static Integer getEmailInvocations()
```

Return Value

Type: Integer

`getLimitEmailInvocations()`

Returns the total number of email invocation (such as `sendEmail`) that can be called.

Signature

```
public static Integer getLimitEmailInvocations()
```

Return Value

Type: Integer

`getFindSimilarCalls()`

Deprecated. Returns the same value as `getSoslQueries`. The number of `findSimilar` methods is no longer a separate limit, but is tracked as the number of SOSL queries issued.

Signature

```
public static Integer getFindSimilarCalls()
```

Return Value

Type: Integer

`getLimitFindSimilarCalls()`

Deprecated. Returns the same value as `getLimitSoslQueries`. The number of `findSimilar` methods is no longer a separate limit, but is tracked as the number of SOSL queries issued.

Signature

```
public static Integer getLimitFindSimilarCalls()
```

Return Value

Type: Integer

`getFutureCalls()`

Returns the number of methods with the `future` annotation that have been executed (not necessarily completed).

Signature

```
public static Integer getFutureCalls()
```

Return Value

Type: Integer

`getLimitFutureCalls()`

Returns the total number of methods with the `future` annotation that can be executed (not necessarily completed).

Signature

```
public static Integer getLimitFutureCalls()
```

Return Value

Type: Integer

`getHeapSize()`

Returns the approximate amount of memory (in bytes) that has been used for the heap.

Signature

```
public static Integer getHeapSize()
```

Return Value

Type: Integer

getLimitHeapSize()

Returns the total amount of memory (in bytes) that can be used for the heap.

Signature

```
public static Integer getLimitHeapSize()
```

Return Value

Type: Integer

getMobilePushApexCalls()

Returns the number of Apex calls that have been used by mobile push notifications during the current metering interval.

Signature

```
public static Integer getMobilePushApexCalls()
```

Return Value

Type:Integer

getLimitMobilePushApexCalls()

Returns the total number of Apex calls that are allowed per transaction for mobile push notifications.

Signature

```
public static Integer getLimitMobilePushApexCalls()
```

Return Value

Type:Integer

getPublishImmediateDML()

Returns the number of `EventBus.publish` calls that have been made for platform events configured to publish immediately.

Signature

```
public static Integer getPublishImmediateDML()
```

Return Value

Type: Integer

getLimitPublishImmediateDML()

Returns the total number of `EventBus.publish` statements that can be called for platform events configured to publish immediately.

Signature

```
public static Integer getLimitPublishImmediateDML()
```

Return Value

Type: Integer

getQueries()

Returns the number of SOQL queries that have been issued.

Signature

```
public static Integer getQueries()
```

Return Value

Type: Integer

getLimitQueries()

Returns the total number of SOQL queries that can be issued.

Signature

```
public static Integer getLimitQueries()
```

Return Value

Type: Integer

getQueryLocatorRows()

Returns the number of records that have been returned by the `Database.getQueryLocator` method.

Signature

```
public static Integer getQueryLocatorRows()
```

Return Value

Type: Integer

getLimitQueryLocatorRows()

Returns the total number of records that can be returned by the `Database.getQueryLocator` method.

Signature

```
public static Integer getLimitQueryLocatorRows()
```

Return Value

Type: Integer

getQueryRows()

Returns the number of records that have been returned by issuing SOQL queries.

Signature

```
public static Integer getQueryRows()
```

Return Value

Type: Integer

getLimitQueryRows()

Returns the total number of records that can be returned by issuing SOQL queries.

Signature

```
public static Integer getLimitQueryRows()
```

Return Value

Type: Integer

getQueueableJobs()

Returns the number of queueable jobs that have been added to the queue per transaction. A queueable job corresponds to a class that implements the `Queueable` interface.

Signature

```
public static Integer getQueueableJobs()
```

Return Value

Type: Integer

getLimitQueueableJobs()

Returns the maximum number of queueable jobs that can be added to the queue per transaction. A queueable job corresponds to a class that implements the `Queueable` interface.

Signature

```
public static Integer getLimitQueueableJobs()
```

Return Value

Type: Integer

getRunAs()

Deprecated. Returns the same value as `getDMLStatements`.

Signature

```
public static Integer getRunAs()
```

Return Value

Type: Integer

Usage

The number of `RunAs` methods is no longer a separate limit, but is tracked as the number of DML statements issued.

getLimitRunAs()

Deprecated. Returns the same value as `getLimitDMLStatements`.

Signature

```
public static Integer getLimitRunAs()
```

Return Value

Type: Integer

Usage

The number of `RunAs` methods is no longer a separate limit, but is tracked as the number of DML statements issued.

getSavepointRollbacks()

Deprecated. Returns the same value as `getDMLStatements`.

Signature

```
public static Integer getSavepointRollbacks()
```

Return Value

Type: Integer

Usage

The number of `Rollback` methods is no longer a separate limit, but is tracked as the number of DML statements issued.

getLimitSavepointRollbacks()

Deprecated. Returns the same value as `getLimitDMLStatements`.

Signature

```
public static Integer getLimitSavepointRollbacks()
```

Return Value

Type: Integer

Usage

The number of `Rollback` methods is no longer a separate limit, but is tracked as the number of DML statements issued.

getSavepoints()

Deprecated. Returns the same value as `getDMLStatements`.

Signature

```
public static Integer getSavepoints()
```

Return Value

Type: Integer

Usage

The number of `setSavepoint` methods is no longer a separate limit, but is tracked as the number of DML statements issued.

getLimitSavepoints()

Deprecated. Returns the same value as `getLimitDMLStatements`.

Signature

```
public static Integer getLimitSavepoints()
```

Return Value

Type: Integer

Usage

The number of `setSavepoint` methods is no longer a separate limit, but is tracked as the number of DML statements issued.

getSoslQueries()

Returns the number of SOSL queries that have been issued.

Signature

```
public static Integer getSoslQueries()
```

Return Value

Type: Integer

`getLimitSoslQueries()`

Returns the total number of SOSL queries that can be issued.

Signature

```
public static Integer getLimitSoslQueries()
```

Return Value

Type: Integer

List Class

Contains methods for the List collection type.

Namespace

System

Usage

The list methods are all instance methods, that is, they operate on a particular instance of a list. For example, the following removes all elements from `myList`:

```
myList.clear();
```

Even though the `clear` method does not include any parameters, the list that calls it is its implicit parameter.

Note:

- When using a custom type for the list elements, provide an `equals` method in your class. Apex uses this method to determine equality and uniqueness for your objects. For more information on providing an `equals` method, see Using Custom Types in Map Keys and Sets.
- If the list contains String elements, the elements are case-sensitive. Two list elements that differ only by case are considered distinct.

For more information on lists, see Lists.

IN THIS SECTION:

List Constructors

List Methods

List Constructors

The following are constructors for `List`.

IN THIS SECTION:

List<T>()
Creates a new instance of the `List` class. A list can hold elements of any data type T.

List<T>(listToCopy)
Creates a new instance of the `List` class by copying the elements from the specified list. T is the data type of the elements in both lists and can be any data type.

List<T>(setToCopy)
Creates a new instance of the `List` class by copying the elements from the specified set. T is the data type of the elements in the set and list and can be any data type.

List<T>()

Creates a new instance of the `List` class. A list can hold elements of any data type T.

Signature

```
public List<T>()
```

Example

```
// Create a list
List<Integer> ls1 = new List<Integer>();
// Add two integers to the list
ls1.add(1);
ls1.add(2);
```

List<T>(listToCopy)

Creates a new instance of the `List` class by copying the elements from the specified list. T is the data type of the elements in both lists and can be any data type.

Signature

```
public List<T>(List<T> listToCopy)
```

Parameters

listToCopy
Type: List<T>

The list containing the elements to initialize this list from. T is the data type of the list elements.

Example

```
List<Integer> ls1 = new List<Integer>();
ls1.add(1);
ls1.add(2);
// Create a list based on an existing one
List<Integer> ls2 = new List<Integer>(ls1);
// ls2 elements are copied from ls1
System.debug(ls2);// DEBUG|(1, 2)
```

`List<T>(setToCopy)`

Creates a new instance of the `List` class by copying the elements from the specified set. T is the data type of the elements in the set and list and can be any data type.

Signature

```
public List<T>(Set<T> setToCopy)
```

Parameters

setToCopy

Type: Set<T>

The set containing the elements to initialize this list with. T is the data type of the set elements.

Example

```
Set<Integer> s1 = new Set<Integer>();
s1.add(1);
s1.add(2);
// Create a list based on a set
List<Integer> ls = new List<Integer>(s1);
// ls elements are copied from s1
Assert.isTrue(ls.contains(2));
Assert.isTrue(ls.contains(1));
```

List Methods

The following are methods for `List`. All are instance methods.

IN THIS SECTION:

add(listElement)

Adds an element to the end of the list.

add(index, listElement)

Inserts an element into the list at the specified index position and shifts all subsequent elements one index position to the right.

addAll(fromList)

Adds all of the elements in the specified list to the list that calls the method. Both lists must be of the same type.

addAll(fromSet)
Add all of the elements in specified set to the list that calls the method. The set and the list must be of the same type.

clear()
Removes all elements from a list, consequently setting the list's length to zero.

clone()
Makes a duplicate copy of a list.

contains(listElement)
Returns `true` if the list contains the specified element.

deepClone(preserveId, preserveReadonlyTimestamps, preserveAutonumber)
Makes a duplicate copy of a list of sObject records, including the sObject records themselves.

equals(list2)
Compares this list with the specified list and returns `true` if both lists are equal; otherwise, returns `false`.

get(index)
Returns the list element stored at the specified index.

getSObjectType()
Returns the token of the sObject type that makes up a list of sObjects.

hashCode()
Returns the hashcode corresponding to this list and its contents.

indexOf(listElement)
Returns the index of the first occurrence of the specified element in this list. If this list does not contain the element, returns -1.

isEmpty()
Returns true if the list has zero elements.

iterator()
Returns an instance of an iterator for this list.

remove(index)
Removes the list element stored at the specified index, returning the element that was removed.

set(index, listElement)
Sets the specified value for the element at the given index.

size()
Returns the number of elements in the list.

sort()
Sorts the items in the list in ascending order.

toString()
Returns the string representation of the list.

`add(listElement)`

Adds an element to the end of the list.

Signature

```
public Void add(Object listElement)
```

Parameters

listElement
Type: Object

Return Value

Type: Void

Example

```
List<Integer> myList = new List<Integer>();
myList.add(47);
Integer myNumber = myList.get(0);
system.assertEquals(47, myNumber);
```

add(index, listElement)

Inserts an element into the list at the specified index position and shifts all subsequent elements one index position to the right.

Signature

```
public Void add(Integer index, Object listElement)
```

Parameters

index
Type: Integer

listElement
Type: Object

Return Value

Type: Void

Example

In this example, a list with six elements is created. Integers are added to the first and second index positions, and the subsequent elements are shifted to the right. After the two methods are called, the list has eight total elements.

```
List<Integer> myList = new Integer[6];
myList.add(0, 47);
myList.add(1, 52);
system.debug(myList); // DEBUG|(47, 52, null, null, null, null, null, null)
system.assertEquals(52, myList.get(1));
```

addAll(fromList)

Adds all of the elements in the specified list to the list that calls the method. Both lists must be of the same type.

Signature

```
public Void addAll(List fromList)
```

Parameters

fromList
Type: List

Return Value

Type: Void

addAll(fromSet)

Add all of the elements in specified set to the list that calls the method. The set and the list must be of the same type.

Signature

```
public Void addAll(Set fromSet)
```

Parameters

fromSet
Type: Set

Return Value

Type: Void

clear()

Removes all elements from a list, consequently setting the list's length to zero.

Signature

```
public Void clear()
```

Return Value

Type: Void

clone()

Makes a duplicate copy of a list.

Signature

```
public List<Object> clone()
```

Return Value

Type: List<Object>

Usage

The cloned list is of the same type as the current list.

Note that if this is a list of sObject records, the duplicate list will only be a shallow copy of the list. That is, the duplicate will have references to each object, but the sObject records themselves will not be duplicated. For example:

To also copy the sObject records, you must use the `deepClone` method.

Example

```
Account a = new Account(Name='Acme', BillingCity='New York');

Account b = new Account();
Account[] q1 = new Account[]{a,b};

Account[] q2 = q1.clone();
q1[0].BillingCity = 'San Francisco';

System.assertEquals(
      'San Francisco',
      q1[0].BillingCity);
System.assertEquals(
      'San Francisco',
      q2[0].BillingCity);
```

`contains(listElement)`

Returns `true` if the list contains the specified element.

Signature

```
public Boolean contains(Object listElement)
```

Parameters

`listElement`
Type: Object

Return Value

Type: Boolean

Example

```
List<String> myStrings = new List<String>{'a', 'b'};
Boolean result = myStrings.contains('z');
System.assertEquals(false, result);
```

`deepClone(preserveId, preserveReadonlyTimestamps, preserveAutonumber)`

Makes a duplicate copy of a list of sObject records, including the sObject records themselves.

Signature

```
public List<Object> deepClone(Boolean preserveId, Boolean preserveReadonlyTimestamps,
Boolean preserveAutonumber)
```

Parameters

preserveId
Type: Boolean

The optional *preserveId* argument determines whether the IDs of the original objects are preserved or cleared in the duplicates. If set to `true`, the IDs are copied to the cloned objects. The default is `false`, that is, the IDs are cleared.

preserveReadonlyTimestamps
Type: Boolean

The optional *preserveReadonlyTimestamps* argument determines whether the read-only timestamp and user ID fields are preserved or cleared in the duplicates. If set to `true`, the read-only fields `CreatedById`, `CreatedDate`, `LastModifiedById`, and `LastModifiedDate` are copied to the cloned objects. The default is `false`, that is, the values are cleared.

preserveAutonumber
Type: Boolean

The optional *preserveAutonumber* argument determines whether the autonumber fields of the original objects are preserved or cleared in the duplicates. If set to `true`, auto number fields are copied to the cloned objects. The default is `false`, that is, auto number fields are cleared.

Return Value

Type: List<Object>

Usage

The returned list is of the same type as the current list.

Note:

- `deepClone` only works with lists of sObjects, not with lists of primitives.
- For Apex saved using Salesforce API version 22.0 or earlier, the default value for the *preserve_id* argument is `true`, that is, the IDs are preserved.

To make a shallow copy of a list without duplicating the sObject records it contains, use the `clone` method.

Example

This example performs a deep clone for a list with two accounts.

```
Account a = new Account(Name='Acme', BillingCity='New York');

Account b = new Account(Name='Salesforce');
```

```
Account[] q1 = new Account[]{a,b};

Account[] q2 = q1.deepClone();
q1[0].BillingCity = 'San Francisco';

System.assertEquals(
     'San Francisco',
     q1[0].BillingCity);

System.assertEquals(
     'New York',
     q2[0].BillingCity);
```

This example is based on the previous example and shows how to clone a list with preserved read-only timestamp and user ID fields.

```
insert q1;

List<Account> accts = [SELECT CreatedById, CreatedDate, LastModifiedById,
                       LastModifiedDate, BillingCity
                       FROM Account
                       WHERE Name='Acme' OR Name='Salesforce'];

// Clone list while preserving timestamp and user ID fields.
Account[] q3 = accts.deepClone(false,true,false);

// Verify timestamp fields are preserved for the first list element.
System.assertEquals(
    accts[0].CreatedById,
    q3[0].CreatedById);
System.assertEquals(
    accts[0].CreatedDate,
    q3[0].CreatedDate);
System.assertEquals(
    accts[0].LastModifiedById,
    q3[0].LastModifiedById);
System.assertEquals(
    accts[0].LastModifiedDate,
    q3[0].LastModifiedDate);
```

`equals(list2)`

Compares this list with the specified list and returns `true` if both lists are equal; otherwise, returns `false`.

Signature

```
public Boolean equals(List list2)
```

Parameters

`list2`

Type: List

The list to compare this list with.

Return Value

Type: Boolean

Usage

Two lists are equal if their elements are equal and are in the same order. The `==` operator is used to compare the elements of the lists.

The `==` operator is equivalent to calling the `equals` method, so you can call `list1.equals(list2);` instead of `list1 == list2;`.

get(index)

Returns the list element stored at the specified index.

Signature

```
public Object get(Integer index)
```

Parameters

index
Type: Integer

Return Value

Type: Object

Usage

To reference an element of a one-dimensional list of primitives or sObjects, you can also follow the name of the list with the element's index position in square brackets as shown in the example.

Example

```
List<Integer> myList = new List<Integer>();
myList.add(47);
Integer myNumber = myList.get(0);
system.assertEquals(47, myNumber);
```

```
List<String> colors = new String[3];
colors[0] = 'Red';
colors[1] = 'Blue';
colors[2] = 'Green';
```

getSObjectType()

Returns the token of the sObject type that makes up a list of sObjects.

Signature

```
public Schema.SObjectType getSObjectType()
```

Return Value

Type: Schema.SObjectType

Usage

Use this method with describe information to determine if a list contains sObjects of a particular type.

Note that this method can only be used with lists that are composed of sObjects.

For more information, see Understanding Apex Describe Information.

Example

```
// Create a generic sObject variable.
SObject sObj = Database.query('SELECT Id FROM Account LIMIT 1');

// Verify if that sObject  variable is an Account token.
System.assertEquals(
   Account.sObjectType,
   sObj.getSObjectType());

// Create a list of generic sObjects.
List<sObject> q = new Account[]{};

// Verify if the list of sObjects
// contains Account tokens.
System.assertEquals(
   Account.sObjectType,
   q.getSObjectType());
```

hashCode()

Returns the hashcode corresponding to this list and its contents.

Signature

```
public Integer hashCode()
```

Return Value

Type: Integer

indexOf(listElement)

Returns the index of the first occurrence of the specified element in this list. If this list does not contain the element, returns -1.

Signature

```
public Integer indexOf(Object listElement)
```

Parameters

listElement
Type: Object

Return Value

Type: Integer

Example

```
List<String> myStrings = new List<String>{'a', 'b', 'a'};
Integer result = myStrings.indexOf('a');
System.assertEquals(0, result);
```

isEmpty()

Returns true if the list has zero elements.

Signature

```
public Boolean isEmpty()
```

Return Value

Type: Boolean

iterator()

Returns an instance of an iterator for this list.

Signature

```
public Iterator iterator()
```

Return Value

Type: Iterator

Usage

From the returned iterator, you can use the iterable methods `hasNext` and `next` to iterate through the list.

Note: You don't have to implement the `iterable` interface to use the `iterable` methods with a list.

See Custom Iterators.

Example

```
public class CustomIterator
   implements Iterator<Account>{
```

```
    private List<Account> accounts;
    private Integer currentIndex;

    public CustomIterator(List<Account> accounts){
        this.accounts = accounts;
        this.currentIndex = 0;
    }

    public Boolean hasNext(){
        return currentIndex < accounts.size();
    }

    public Account next(){
        if(hasNext()) {
            return accounts[currentIndex++];
        } else {
            throw new NoSuchElementException('Iterator has no more elements.');
        }
    }
}
```

remove(index)

Removes the list element stored at the specified index, returning the element that was removed.

Signature

```
public Object remove(Integer index)
```

Parameters

index
Type: Integer

Return Value

Type: Object

Example

```
List<String> colors = new String[3];
colors[0] = 'Red';
colors[1] = 'Blue';
colors[2] = 'Green';
String s1 = colors.remove(2);
system.assertEquals('Green', s1);
```

set(index, listElement)

Sets the specified value for the element at the given index.

Signature

```
public Void set(Integer index, Object listElement)
```

Parameters

index
Type: Integer

The index of the list element to set.

listElement
Type: Object

The value of the list element to set.

Return Value

Type: Void

Usage

To set an element of a one-dimensional list of primitives or sObjects, you can also follow the name of the list with the element's index position in square brackets.

Example

```
List<Integer> myList = new Integer[6];
myList.set(0, 47);
myList.set(1, 52);
system.assertEquals(52, myList.get(1));
```

```
List<String> colors = new String[3];
colors[0] = 'Red';
colors[1] = 'Blue';
colors[2] = 'Green';
```

size()

Returns the number of elements in the list.

Signature

```
public Integer size()
```

Return Value

Type: Integer

Example

```
List<Integer> myList = new List<Integer>();
Integer size = myList.size();
```

```
system.assertEquals(0, size);

List<Integer> myList2 = new Integer[6];
Integer size2 = myList2.size();
system.assertEquals(6, size2);
```

sort()

Sorts the items in the list in ascending order.

Signature

```
public Void sort()
```

Return Value

Type: Void

Usage

Using this method, you can sort primitive types, SelectOption elements, and sObjects (standard objects and custom objects). For more information on the sort order used for sObjects, see Sorting Lists of sObjects. You can sort custom types (your Apex classes) if they implement the Comparable interface. Alternatively, a class implementing the Comparator interface can be passed as a parameter to the `List.sort` method.

When you use `sort()` methods on `List<Id>s` that contain both 15-character and 18-character IDs, IDs for the same record sort together in API version 35.0 and later.

Example

In the following example, the list has three elements. When the list is sorted, the first element is null because it has no value assigned. The second element and third element have values of 5 and 10.

```
List<Integer> q1 = new Integer[3];



// Assign values to the first two elements.

q1[0] = 10;

q1[1] = 5;



q1.sort();

// Verify sorted list. Elements are sorted in nulls-first order: null, 5, and 10

system.assertEquals(null, q1.get(0));

system.assertEquals(5, q1.get(1));
```

```
system.assertEquals(10, q1.get(2));
```

toString()

Returns the string representation of the list.

Signature

```
public String toString()
```

Return Value

Type: String

Usage

When used in cyclic references, the output is truncated to prevent infinite recursion. When used with large collections, the output is truncated to avoid exceeding total heap size and maximum CPU time.

- Up to 10 items per collection are included in the output, followed by an ellipsis (…).
- If the same object is included multiple times in a collection, it's shown in the output only once; subsequent references are shown as `(already output)`.

Location Class

Contains methods for accessing the component fields of geolocation compound fields.

Namespace

system

Usage

Each of these methods is also equivalent to a read-only property. For each getter method you can access the property using dot notation. For example, `myLocation.getLatitude()` is equivalent to `myLocation.latitude`.

You can't use dot notation to access compound fields' subfields directly on the parent field. Instead, assign the parent field to a variable of type `Location`, and then access its components.

```
Location loc = myAccount.MyLocation__c;
Double lat = loc.latitude;
```

Important: "Location" in Salesforce can also refer to the Location standard object. When referencing the Location object in your Apex code, always use `Schema.Location` instead of `Location` to prevent confusion with the standard Location compound field. If referencing both the location object and the Location field in the same snippet, you can differentiate between the two by using `System.Location` for the field and `Schema.Location` for the object.

Example

```
// Select and access the Location field. MyLocation__c is the name of a geolocation field
 on Account.
Account[] records = [SELECT id, MyLocation__c FROM Account LIMIT 10];
for(Account acct : records) {
   Location loc = acct.MyLocation__c;
   Double lat = loc.latitude;
   Double lon = loc.longitude;
}

// Instantiate new Location objects and compute the distance between them in different
ways.
Location loc1 = Location.newInstance(28.635308,77.22496);
Location loc2 = Location.newInstance(37.7749295,-122.4194155);
Double dist = Location.getDistance(loc1, loc2, 'mi');
Double dist2 = loc1.getDistance(loc2, 'mi');
```

IN THIS SECTION:

Location Methods

Location Methods

The following are methods for `Location`.

IN THIS SECTION:

getDistance(toLocation, unit)

Calculates the distance between this location and the specified location, using an approximation of the haversine formula and the specified unit.

getDistance(firstLocation, secondLocation, unit)

Calculates the distance between the two specified locations, using an approximation of the haversine formula and the specified unit.

getLatitude()

Returns the latitude field of this location.

getLongitude()

Returns the longitude field of this location.

newInstance(latitude, longitude)

Creates an instance of the `Location` class, with the specified latitude and longitude.

`getDistance(toLocation, unit)`

Calculates the distance between this location and the specified location, using an approximation of the haversine formula and the specified unit.

Signature

```
public Double getDistance(Location toLocation, String unit)
```

Parameters

toLocation
Type: Location

The `Location` to which you want to calculate the distance from the current `Location`.

unit
Type: String

The distance unit you want to use: `mi` or `km`.

Return Value

Type: Double

getDistance(firstLocation, secondLocation, unit)

Calculates the distance between the two specified locations, using an approximation of the haversine formula and the specified unit.

Signature

```
public static Double getDistance(Location firstLocation, Location secondLocation, String unit)
```

Parameters

firstLocation
Type: Location

The first of two locations used to calculate distance.

secondLocation
Type: Location

The second of two locations used to calculate distance.

unit
Type: String

The distance unit you want to use: `mi` or `km`.

Return Value

Type: Double

getLatitude()

Returns the latitude field of this location.

Signature

```
public Double getLatitude()
```

Return Value

Type: Double

`getLongitude()`

Returns the longitude field of this location.

Signature

```
public Double getLongitude()
```

Return Value

Type: Double

`newInstance(latitude, longitude)`

Creates an instance of the `Location` class, with the specified latitude and longitude.

Signature

```
public static Location newInstance(Decimal latitude, Decimal longitude)
```

Parameters

`latitude`
Type: Decimal

`longitude`
Type: Decimal

Return Value

Type: Location

LoggingLevel Enum

Specifies the logging level for the `System.debug` method.

Enum Values

The following are the values of the `System.LoggingLevel` enum, listed from the lowest to the highest levels. The level is cumulative, that is, if you select FINE, the log also includes all events logged at the DEBUG, INFO, WARN, and ERROR levels.

Value	Description
`NONE`	No logging.
`ERROR`	Error and exception logging.
`WARN`	Warning logging.

Value	Description
INFO	Informational logging.
DEBUG	User-specified debug logging.
FINE	High level of logging.
FINER	Higher level of logging than FINE.
FINEST	Highest level of logging.

Usage

Log levels are cumulative. For example, if the lowest level, ERROR, is specified for Apex code, only System.debug methods with the log level of ERROR are logged. If the next log level, WARN, is specified, System.debug methods specified with either ERROR or WARN levels are logged.

In this example, if the log level is set to ERROR, the string MsgTxt isn't written to the debug log because the debug method has a level of INFO.

```
System.debug(logginglevel.INFO, 'MsgTxt');
```

For more information on log levels, see Debug Log Levels in Salesforce Help.

Long Class

Contains methods for the Long primitive data type.

Namespace

System

Usage

For more information on Long, see Long Data Type.

Long Methods

The following are methods for Long.

IN THIS SECTION:

format()
Returns the String format for this Long using the locale of the context user.

intValue()
Returns the Integer value for this Long.

valueOf(stringToLong)
Returns a Long that contains the value of the specified String. As in Java, the string is interpreted as representing a signed decimal Long.

format()

Returns the String format for this Long using the locale of the context user.

Signature

```
public String format()
```

Return Value

Type: String

Example

```
Long myLong = 4271990;
system.assertEquals('4,271,990', myLong.format());
```

intValue()

Returns the Integer value for this Long.

Signature

```
public Integer intValue()
```

Return Value

Type: Integer

Example

```
Long myLong = 7191991;
Integer value = myLong.intValue();
system.assertEquals(7191991, myLong.intValue());
```

valueOf(stringToLong)

Returns a Long that contains the value of the specified String. As in Java, the string is interpreted as representing a signed decimal Long.

Signature

```
public static Long valueOf(String stringToLong)
```

Parameters

stringToLong
Type: String

Return Value

Type: Long

Example

```
Long L1 = long.valueOf('123456789');
```

Map Class

Contains methods for the Map collection type.

Namespace

System

Usage

The Map methods are all instance methods, that is, they operate on a particular instance of a map. The following are the instance methods for maps.

Note:

- Map keys and values can be of any data type—primitive types, collections, sObjects, user-defined types, and built-in Apex types.
- Uniqueness of map keys of user-defined types is determined by the `equals` and `hashCode` methods, which you provide in your classes. Uniqueness of keys of all other non-primitive types, such as sObject keys, is determined by comparing the objects' field values. Use caution when you use an sObject as a map key because when the sObject is changed, it no longer maps to the same value. For information and examples, see https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_map_sobject_considerations.htm
- Map keys of type String are case-sensitive. Two keys that differ only by the case are considered unique and have corresponding distinct Map entries. Subsequently, the Map methods, including `put`, `get`, `containsKey`, and `remove` treat these keys as distinct.
- With the `keySet()` method, the returned keySet is backed by the map and reflects any changes made to the map, and vice versa.

For more information on maps, see Maps.

IN THIS SECTION:

Map Constructors

Map Methods

Map Constructors

The following are constructors for `Map`.

IN THIS SECTION:

Map<T1,T2>()

Creates a new instance of the `Map` class. T1 is the data type of the keys and T2 is the data type of the values.

Map<T1,T2>(mapToCopy)

Creates a new instance of the `Map` class and initializes it by copying the entries from the specified map. T1 is the data type of the keys and T2 is the data type of the values.

Map<ID,sObject>(recordList)

Creates a new instance of the `Map` class and populates it with the passed-in list of sObject records. The keys are populated with the sObject IDs and the values are the sObjects.

Map<T1,T2>()

Creates a new instance of the `Map` class. T1 is the data type of the keys and T2 is the data type of the values.

Signature

```
public Map<T1,T2>()
```

Example

```
Map<Integer, String> m1 = new Map<Integer, String>();
m1.put(1, 'First item');
m1.put(2, 'Second item');
```

Map<T1,T2>(mapToCopy)

Creates a new instance of the `Map` class and initializes it by copying the entries from the specified map. T1 is the data type of the keys and T2 is the data type of the values.

Signature

```
public Map<T1,T2>(Map<T1,T2> mapToCopy)
```

Parameters

mapToCopy

Type: Map<T1, T2>

The map to initialize this map with. T1 is the data type of the keys and T2 is the data type of the values. All map keys and values are copied to this map.

Example

```
Map<Integer, String> m1 = new Map<Integer, String>();
m1.put(1, 'First item');
m1.put(2, 'Second item');
Map<Integer, String> m2 = new Map<Integer, String>(m1);
// The map elements of m2 are copied from m1
System.debug(m2);
```

Map<ID,sObject>(recordList)

Creates a new instance of the `Map` class and populates it with the passed-in list of sObject records. The keys are populated with the sObject IDs and the values are the sObjects.

Signature

```
public Map<ID,sObject>(List<sObject> recordList)
```

Parameters

recordList
Type: List<sObject>

The list of sObjects to populate the map with.

Example

```
List<Account> ls = [select Id,Name from Account];
Map<Id, Account> m = new Map<Id, Account>(ls);
```

Map Methods

The following are methods for `Map`. All are instance methods.

IN THIS SECTION:

clear()
Removes all of the key-value mappings from the map.

clone()
Makes a duplicate copy of the map.

containsKey(key)
Returns `true` if the map contains a mapping for the specified key.

deepClone()
Makes a duplicate copy of a map, including sObject records if this is a map with sObject record values.

equals(map2)
Compares this map with the specified map and returns `true` if both maps are equal; otherwise, returns `false`.

get(key)
Returns the value to which the specified key is mapped, or `null` if the map contains no value for this key.

getSObjectType()
Returns the token of the sObject type that makes up the map values.

hashCode()
Returns the hashcode corresponding to this map.

isEmpty()
Returns true if the map has zero key-value pairs.

keySet()
Returns a set that contains all of the keys in the map.

put(key, value)
Associates the specified value with the specified key in the map.

putAll(fromMap)
Copies all of the mappings from the specified map to the original map.

putAll(sobjectArray)
Adds the list of sObject records to a map declared as Map<ID, sObject> or Map<String, sObject>.

remove(key)
Removes the mapping for the specified key from the map, if present, and returns the corresponding value.

size()
Returns the number of key-value pairs in the map.

toString()
Returns the string representation of the map.

values()
Returns a list that contains all the values in the map.

`clear()`

Removes all of the key-value mappings from the map.

Signature

```
public Void clear()
```

Return Value

Type: Void

`clone()`

Makes a duplicate copy of the map.

Signature

```
public Map<Object, Object> clone()
```

Return Value

Type: Map (of same type)

Usage

If this is a map with sObject record values, the duplicate map will only be a shallow copy of the map. That is, the duplicate will have references to each sObject record, but the records themselves are not duplicated. For example:

To also copy the sObject records, you must use the `deepClone` method.

Example

```
Account a = new Account(
   Name='Acme',
   BillingCity='New York');

Map<Integer, Account> map1 = new Map<Integer, Account> {};
map1.put(1, a);

Map<Integer, Account> map2 = map1.clone();
map1.get(1).BillingCity =
'San Francisco';

System.assertEquals(
   'San Francisco',
   map1.get(1).BillingCity);

System.assertEquals(
   'San Francisco',
   map2.get(1).BillingCity);
```

containsKey(key)

Returns `true` if the map contains a mapping for the specified key.

Signature

```
public Boolean containsKey(Object key)
```

Parameters

key
Type: Object

Return Value

Type: Boolean

Usage

If the key is a string, the key value is case-sensitive.

Example

```
Map<String, String> colorCodes = new Map<String, String>();

colorCodes.put('Red', 'FF0000');
colorCodes.put('Blue', '0000A0');

Boolean contains = colorCodes.containsKey('Blue');
System.assertEquals(true, contains);
```

deepClone()

Makes a duplicate copy of a map, including sObject records if this is a map with sObject record values.

Signature

```
public Map<Object, Object> deepClone()
```

Return Value

Type: Map (of the same type)

Usage

To make a shallow copy of a map without duplicating the sObject records it contains, use the `clone()` method.

Example

```
Account a = new Account(
   Name='Acme',
   BillingCity='New York');

Map<Integer, Account> map1 = new Map<Integer, Account> {};

map1.put(1, a);

Map<Integer, Account> map2 = map1.deepClone();

// Update the first entry of map1
map1.get(1).BillingCity = 'San Francisco';
// Verify that the BillingCity is updated in map1 but not in map2
System.assertEquals('San Francisco', map1.get(1).BillingCity);
System.assertEquals('New York', map2.get(1).BillingCity);
```

equals(map2)

Compares this map with the specified map and returns `true` if both maps are equal; otherwise, returns `false`.

Signature

```
public Boolean equals(Map map2)
```

Parameters

map2
Type: Map

The *map2* argument is the map to compare this map with.

Return Value

Type: Boolean

Usage

Two maps are equal if their key/value pairs are identical, regardless of the order of those pairs. The `==` operator is used to compare the map keys and values.

The `==` operator is equivalent to calling the `equals` method, so you can call `map1.equals(map2);` instead of `map1 == map2;`.

get(key)

Returns the value to which the specified key is mapped, or `null` if the map contains no value for this key.

Signature

```
public Object get(Object key)
```

Parameters

key
Type: Object

Return Value

Type: Object

Usage

If the key is a string, the key value is case-sensitive.

Example

```
Map<String, String> colorCodes = new Map<String, String>();

colorCodes.put('Red', 'FF0000');
colorCodes.put('Blue', '0000A0');

String code = colorCodes.get('Blue');

System.assertEquals('0000A0', code);

// The following is not a color in the map
String code2 = colorCodes.get('Magenta');
System.assertEquals(null, code2);
```

getSObjectType()

Returns the token of the sObject type that makes up the map values.

Signature

```
public Schema.SObjectType getSObjectType()
```

Return Value

Type: Schema.SObjectType

Usage

Use this method with describe information, to determine if a map contains sObjects of a particular type.

Note that this method can only be used with maps that have sObject values.

For more information, see Understanding Apex Describe Information.

Example

```
// Create a generic sObject variable.
SObject sObj = Database.query('SELECT Id FROM Account LIMIT 1');

// Verify if that sObject variable is an Account token.
System.assertEquals(
   Account.sObjectType,
   sObj.getSObjectType());

// Create a map of generic sObjects
Map<Integer, Account> m = new Map<Integer, Account>();

// Verify if the map contains Account tokens.
System.assertEquals(
   Account.sObjectType,
   m.getSObjectType());
```

hashCode()

Returns the hashcode corresponding to this map.

Signature

```
public Integer hashCode()
```

Return Value

Type: Integer

isEmpty()

Returns true if the map has zero key-value pairs.

Signature

```
public Boolean isEmpty()
```

Return Value

Type: Boolean

Example

```
Map<String, String> colorCodes = new Map<String, String>();
Boolean empty = colorCodes.isEmpty();
System.assertEquals(true, empty);
```

keySet()

Returns a set that contains all of the keys in the map.

Signature

```
public Set<Object> keySet()
```

Return Value

Type: Set (of key type)

The returned keySet is backed by the map, so the keySet reflects any changes made to the map, and vice-versa.

Example

```
Map<String, String> colorCodes = new Map<String, String>();

colorCodes.put('Red', 'FF0000');
colorCodes.put('Blue', '0000A0');

Set <String> colorSet = new Set<String>();
colorSet = colorCodes.keySet();
```

put(key, value)

Associates the specified value with the specified key in the map.

Signature

```
public Object put(Object key, Object value)
```

Parameters

key
Type: Object

value
Type: Object

Return Value

Type: Object

Usage

If the map previously contained a mapping for this key, the old value is returned by the method and then replaced.

If the key is a string, the key value is case-sensitive.

Example

```
Map<String, String> colorCodes = new Map<String, String>();

colorCodes.put('Red', 'ff0000');
colorCodes.put('Red', '#FF0000');
// Red is now #FF0000
```

putAll(fromMap)

Copies all of the mappings from the specified map to the original map.

Signature

```
public Void putAll(Map fromMap)
```

Parameters

fromMap
Type: Map

Return Value

Type: Void

Usage

The new mappings from *fromMap* are merged with any mappings that existed in the original map. If any of the keys match, the original map values are replaced by corresponding values in the new mapping.

Example

```
Map<String, String> map1 = new Map<String, String>();
map1.put('Red','FF0000');
Map<String, String> map2 = new Map<String, String>();
map2.put('Blue','0000FF');
// Add map1 entries to map2
map2.putAll(map1);
System.assertEquals(2, map2.size());
```

putAll(sobjectArray)

Adds the list of sObject records to a map declared as Map<ID, sObject> or Map<String, sObject>.

Signature

```
public Void putAll(sObject[] sobjectArray)
```

Parameters

sobjectArray
Type: sObject[]

Return Value

Type: Void

Usage

This method is similar to calling the Map constructor with the same input.

Example

```
List<Account> accts = new List<Account>();
accts.add(new Account(Name='Account1'));
accts.add(new Account(Name='Account2'));
// Insert accounts so their IDs are populated.
insert accts;
Map<Id, Account> m = new Map<Id, Account>();
// Add all the records to the map.
m.putAll(accts);
System.assertEquals(2, m.size());
```

remove(key)

Removes the mapping for the specified key from the map, if present, and returns the corresponding value.

Signature

```
public Object remove(Key key)
```

Parameters

key
Type: Key

Return Value

Type: Object

Usage

If the key is a string, the key value is case-sensitive.

Example

```
Map<String, String> colorCodes = new Map<String, String>();

colorCodes.put('Red', 'FF0000');
colorCodes.put('Blue', '0000A0');

String myColor = colorCodes.remove('Blue');
String code2 = colorCodes.get('Blue');

System.assertEquals(null, code2);
```

size()

Returns the number of key-value pairs in the map.

Signature

```
public Integer size()
```

Return Value

Type: Integer

Example

```
Map<String, String> colorCodes = new Map<String, String>();

colorCodes.put('Red', 'FF0000');
colorCodes.put('Blue', '0000A0');

Integer mSize = colorCodes.size();
system.assertEquals(2, mSize);
```

toString()

Returns the string representation of the map.

Signature

```
public String toString()
```

Return Value

Type: String

Usage

When used in cyclic references, the output is truncated to prevent infinite recursion. When used with large collections, the output is truncated to avoid exceeding total heap size and maximum CPU time.

- Up to 10 items per collection are included in the output, followed by an ellipsis (…).

- If the same object is included multiple times in a collection, it's shown in the output only once; subsequent references are shown as `(already output)`.

`values()`

Returns a list that contains all the values in the map.

Signature

```
public List<Object> values()
```

Return Value

Type: List<Object>

Usage

The order of map elements is deterministic. You can rely on the order being the same in each subsequent execution of the same code. For example, suppose the `values()` method returns a list containing `value1` and index 0 and `value2` and index 1. Subsequent runs of the same code result in those values being returned in the same order.

Example

```
Map<String, String> colorCodes = new Map<String, String>();

colorCodes.put('Red', 'FF0000');
colorCodes.put('Blue', '0000A0');

List<String> colors = new List<String>();
colors = colorCodes.values();
```

Matcher Class

Matchers use Patterns to perform match operations on a character string.

Namespace

System

Matcher Methods

The following are methods for `Matcher`.

IN THIS SECTION:

end()
Returns the position after the last matched character.

end(groupIndex)

Returns the position after the last character of the subsequence captured by the group index during the previous match operation. If the match was successful but the group itself did not match anything, this method returns -1.

find()

Attempts to find the next subsequence of the input sequence that matches the pattern. This method returns true if a subsequence of the input sequence matches this Matcher object's pattern.

find(group)

Resets the Matcher object and then tries to find the next subsequence of the input sequence that matches the pattern. This method returns `true` if a subsequence of the input sequence matches this Matcher object's pattern.

group()

Returns the input subsequence returned by the previous match.

group(groupIndex)

Returns the input subsequence captured by the specified group index during the previous match operation. If the match was successful but the specified group failed to match any part of the input sequence, `null` is returned.

groupCount()

Returns the number of capturing groups in this Matching object's pattern. Group zero denotes the entire pattern and is not included in this count.

hasAnchoringBounds()

Returns true if the Matcher object has anchoring bounds, false otherwise. By default, a Matcher object uses anchoring bounds regions.

hasTransparentBounds()

Returns true if the Matcher object has transparent bounds, false if it uses opaque bounds. By default, a Matcher object uses opaque region boundaries.

hitEnd()

Returns true if the end of input was found by the search engine in the last match operation performed by this Matcher object. When this method returns true, it is possible that more input would have changed the result of the last search.

lookingAt()

Attempts to match the input sequence, starting at the beginning of the region, against the pattern.

matches()

Attempts to match the entire region against the pattern.

pattern()

Returns the Pattern object from which this Matcher object was created.

quoteReplacement(inputString)

Returns a literal replacement string for the specified string *`inputString`*. The characters in the returned string match the sequence of characters in *`inputString`*.

region(start, end)

Sets the limits of this Matcher object's region. The region is the part of the input sequence that is searched to find a match.

regionEnd()

Returns the end index (exclusive) of this Matcher object's region.

regionStart()

Returns the start index (inclusive) of this Matcher object's region.

replaceAll(replacementString)

Replaces every subsequence of the input sequence that matches the pattern with the replacement string.

replaceFirst(replacementString)

Replaces the first subsequence of the input sequence that matches the pattern with the replacement string.

requireEnd()

Returns true if more input could change a positive match into a negative one.

reset()

Resets this Matcher object. Resetting a Matcher object discards all of its explicit state information.

reset(inputSequence)

Resets this Matcher object with the new input sequence. Resetting a Matcher object discards all of its explicit state information.

start()

Returns the start index of the first character of the previous match.

start(groupIndex)

Returns the start index of the subsequence captured by the group specified by the group index during the previous match operation. Captured groups are indexed from left to right, starting at one. Group zero denotes the entire pattern, so the expression `m.start(0)` is equivalent to `m.start()`.

useAnchoringBounds(anchoringBounds)

Sets the anchoring bounds of the region for the Matcher object. By default, a Matcher object uses anchoring bounds regions.

usePattern(pattern)

Changes the Pattern object that the Matcher object uses to find matches. This method causes the Matcher object to lose information about the groups of the last match that occurred. The Matcher object's position in the input is maintained.

useTransparentBounds(transparentBounds)

Sets the transparency bounds for this Matcher object. By default, a Matcher object uses anchoring bounds regions.

`end()`

Returns the position after the last matched character.

Signature

```
public Integer end()
```

Return Value

Type: Integer

`end(groupIndex)`

Returns the position after the last character of the subsequence captured by the group index during the previous match operation. If the match was successful but the group itself did not match anything, this method returns -1.

Signature

```
public Integer end(Integer groupIndex)
```

Parameters

groupIndex
Type: Integer

Return Value

Type: Integer

Usage

Captured groups are indexed from left to right, starting at one. Group zero denotes the entire pattern, so the expression`m.end(0)` is equivalent to `m.end()`.

See Understanding Capturing Groups.

find()

Attempts to find the next subsequence of the input sequence that matches the pattern. This method returns true if a subsequence of the input sequence matches this Matcher object's pattern.

Signature

```
public Boolean find()
```

Return Value

Type: Boolean

Usage

This method starts at the beginning of this Matcher object's region, or, if a previous invocation of the method was successful and the Matcher object has not since been reset, at the first character not matched by the previous match.

If the match succeeds, more information can be obtained using the `start`, `end`, and `group` methods.

For more information, see Using Regions.

find(group)

Resets the Matcher object and then tries to find the next subsequence of the input sequence that matches the pattern. This method returns `true` if a subsequence of the input sequence matches this Matcher object's pattern.

Signature

```
public Boolean find(Integer group)
```

Parameters

group
Type: Integer

Return Value

Type: Boolean

Usage

If the match succeeds, more information can be obtained using the `start`, `end`, and `group` methods.

group()

Returns the input subsequence returned by the previous match.

Signature

```
public String group()
```

Return Value

Type: String

Usage

Note that some groups, such as `(a*)`, match the empty string. This method returns the empty string when such a group successfully matches the empty string in the input.

group(groupIndex)

Returns the input subsequence captured by the specified group index during the previous match operation. If the match was successful but the specified group failed to match any part of the input sequence, `null` is returned.

Signature

```
public String group(Integer groupIndex)
```

Parameters

groupIndex
Type: Integer

Return Value

Type: String

Usage

Captured groups are indexed from left to right, starting at one. Group zero denotes the entire pattern, so the expression `m.group(0)` is equivalent to `m.group()`.

Note that some groups, such as `(a*)`, match the empty string. This method returns the empty string when such a group successfully matches the empty string in the input.

See Understanding Capturing Groups.

groupCount()

Returns the number of capturing groups in this Matching object's pattern. Group zero denotes the entire pattern and is not included in this count.

Signature

```
public Integer groupCount()
```

Return Value

Type: Integer

Usage

See Understanding Capturing Groups.

hasAnchoringBounds()

Returns true if the Matcher object has anchoring bounds, false otherwise. By default, a Matcher object uses anchoring bounds regions.

Signature

```
public Boolean hasAnchoringBounds()
```

Return Value

Type: Boolean

Usage

If a Matcher object uses anchoring bounds, the boundaries of this Matcher object's region match start and end of line anchors such as ^ and $.

For more information, see Using Bounds.

hasTransparentBounds()

Returns true if the Matcher object has transparent bounds, false if it uses opaque bounds. By default, a Matcher object uses opaque region boundaries.

Signature

```
public Boolean hasTransparentBounds()
```

Return Value

Type: Boolean

Usage

For more information, see Using Bounds.

hitEnd()

Returns true if the end of input was found by the search engine in the last match operation performed by this Matcher object. When this method returns true, it is possible that more input would have changed the result of the last search.

Signature

```
public Boolean hitEnd()
```

Return Value

Type: Boolean

lookingAt()

Attempts to match the input sequence, starting at the beginning of the region, against the pattern.

Signature

```
public Boolean lookingAt()
```

Return Value

Type: Boolean

Usage

Like the `matches` method, this method always starts at the beginning of the region; unlike that method, it does not require the entire region be matched.

If the match succeeds, more information can be obtained using the `start`, `end`, and `group` methods.

See Using Regions.

matches()

Attempts to match the entire region against the pattern.

Signature

```
public Boolean matches()
```

Return Value

Type: Boolean

Usage

If the match succeeds, more information can be obtained using the `start`, `end`, and `group` methods.

See Using Regions.

pattern()

Returns the Pattern object from which this Matcher object was created.

Signature

```
public Pattern object pattern()
```

Return Value

Type: System.Pattern

quoteReplacement(inputString)

Returns a literal replacement string for the specified string *inputString*. The characters in the returned string match the sequence of characters in *inputString*.

Signature

```
public static String quoteReplacement(String inputString)
```

Parameters

inputString
Type: String

Return Value

Type: String

Usage

Metacharacters (such as `$` or `^`) and escape sequences in the input string are treated as literal characters with no special meaning.

region(start, end)

Sets the limits of this Matcher object's region. The region is the part of the input sequence that is searched to find a match.

Signature

```
public Matcher object region(Integer start, Integer end)
```

Parameters

start
Type: Integer

end
Type: Integer

Return Value

Type: Matcher

Usage

This method first resets the Matcher object, then sets the region to start at the index specified by `start` and end at the index specified by `end`.

Depending on the transparency boundaries being used, certain constructs such as anchors may behave differently at or around the boundaries of the region.

See Using Regions and Using Bounds.

`regionEnd()`

Returns the end index (exclusive) of this Matcher object's region.

Signature

```
public Integer regionEnd()
```

Return Value

Type: Integer

Usage

See Using Regions.

`regionStart()`

Returns the start index (inclusive) of this Matcher object's region.

Signature

```
public Integer regionStart()
```

Return Value

Type: Integer

Usage

See Using Regions.

`replaceAll(replacementString)`

Replaces every subsequence of the input sequence that matches the pattern with the replacement string.

Signature

```
public String replaceAll(String replacementString)
```

Parameters

replacementString
Type: String

Return Value

Type: String

Usage

This method first resets the Matcher object, then scans the input sequence looking for matches of the pattern. Characters that are not part of any match are appended directly to the result string; each match is replaced in the result by the replacement string. The replacement string may contain references to captured subsequences.

Note that backslashes (\) and dollar signs ($) in the replacement string may cause the results to be different than if the string was treated as a literal replacement string. Dollar signs may be treated as references to captured subsequences, and backslashes are used to escape literal characters in the replacement string.

Invoking this method changes this Matcher object's state. If the Matcher object is to be used in further matching operations it should first be reset.

Given the regular expression `a*b`, the input `"aabxyzaabxyzabxyzb"`, and the replacement string `"-"`, an invocation of this method on a Matcher object for that expression would yield the string `"-xyz-xyz-xyz-"`.

replaceFirst(replacementString)

Replaces the first subsequence of the input sequence that matches the pattern with the replacement string.

Signature

```
public String replaceFirst(String replacementString)
```

Parameters

replacementString
Type: String

Return Value

Type: String

Usage

Note that backslashes (\) and dollar signs ($) in the replacement string may cause the results to be different than if the string was treated as a literal replacement string. Dollar signs may be treated as references to captured subsequences, and backslashes are used to escape literal characters in the replacement string.

Invoking this method changes this Matcher object's state. If the Matcher object is to be used in further matching operations it should first be reset.

Given the regular expression `dog`, the input `"zzzdogzzzdogzzz"`, and the replacement string `"cat"`, an invocation of this method on a Matcher object for that expression would return the string `"zzzcatzzzdogzzz"`.

`requireEnd()`

Returns true if more input could change a positive match into a negative one.

Signature

```
public Boolean requireEnd()
```

Return Value

Type: Boolean

Usage

If this method returns true, and a match was found, then more input could cause the match to be lost.

If this method returns false and a match was found, then more input might change the match but the match won't be lost.

If a match was not found, then `requireEnd` has no meaning.

`reset()`

Resets this Matcher object. Resetting a Matcher object discards all of its explicit state information.

Signature

```
public Matcher object reset()
```

Return Value

Type: Matcher

Usage

This method does not change whether the Matcher object uses anchoring bounds. You must explicitly use the `useAnchoringBounds` method to change the anchoring bounds.

For more information, see Using Bounds.

`reset(inputSequence)`

Resets this Matcher object with the new input sequence. Resetting a Matcher object discards all of its explicit state information.

Signature

```
public Matcher reset(String inputSequence)
```

Parameters

`inputSequence`
Type: String

Return Value

Type: Matcher

`start()`

Returns the start index of the first character of the previous match.

Signature

```
public Integer start()
```

Return Value

Type: Integer

`start(groupIndex)`

Returns the start index of the subsequence captured by the group specified by the group index during the previous match operation. Captured groups are indexed from left to right, starting at one. Group zero denotes the entire pattern, so the expression `m.start(0)` is equivalent to `m.start()`.

Signature

```
public Integer start(Integer groupIndex)
```

Parameters

groupIndex
Type: Integer

Return Value

Type: Integer

Usage

See Understanding Capturing Groups.

`useAnchoringBounds(anchoringBounds)`

Sets the anchoring bounds of the region for the Matcher object. By default, a Matcher object uses anchoring bounds regions.

Signature

```
public Matcher object useAnchoringBounds(Boolean anchoringBounds)
```

Parameters

anchoringBounds
Type: Boolean

If you specify `true`, the Matcher object uses anchoring bounds. If you specify `false`, non-anchoring bounds are used.

Return Value

Type: Matcher

Usage

If a Matcher object uses anchoring bounds, the boundaries of this Matcher object's region match start and end of line anchors such as ^ and $.

For more information, see Using Bounds.

usePattern(pattern)

Changes the Pattern object that the Matcher object uses to find matches. This method causes the Matcher object to lose information about the groups of the last match that occurred. The Matcher object's position in the input is maintained.

Signature

```
public Matcher object usePattern(Pattern pattern)
```

Parameters

pattern
Type: System.Pattern

Return Value

Type: Matcher

useTransparentBounds(transparentBounds)

Sets the transparency bounds for this Matcher object. By default, a Matcher object uses anchoring bounds regions.

Signature

```
public Matcher object useTransparentBounds(Boolean transparentBounds)
```

Parameters

transparentBounds
Type: Boolean

If you specify `true`, the Matcher object uses transparent bounds. If you specify `false`, opaque bounds are used.

Return Value

Type: Matcher

Usage

For more information, see Using Bounds.

Math Class

Contains methods for mathematical operations.

Namespace

System

Math Fields

The following are fields for `Math`.

IN THIS SECTION:

E
Returns the mathematical constant *e*, which is the base of natural logarithms.

PI
Returns the mathematical constant *pi*, which is the ratio of the circumference of a circle to its diameter.

E

Returns the mathematical constant *e*, which is the base of natural logarithms.

Signature

```
public static final Double E
```

Property Value

Type: Double

PI

Returns the mathematical constant *pi*, which is the ratio of the circumference of a circle to its diameter.

Signature

```
public static final Double PI
```

Property Value

Type: Double

Math Methods

The following are methods for `Math`. All methods are static.

IN THIS SECTION:

abs(decimalValue)
Returns the absolute value of the specified Decimal.

abs(doubleValue)
Returns the absolute value of the specified Double.

abs(integerValue)
Returns the absolute value of the specified Integer.

abs(longValue)
Returns the absolute value of the specified Long.

acos(decimalAngle)
Returns the arc cosine of an angle, in the range of 0.0 through *pi*.

acos(doubleAngle)
Returns the arc cosine of an angle, in the range of 0.0 through *pi*.

asin(decimalAngle)
Returns the arc sine of an angle, in the range of -*pi*/2 through *pi*/2.

asin(doubleAngle)
Returns the arc sine of an angle, in the range of -*pi*/2 through *pi*/2.

atan(decimalAngle)
Returns the arc tangent of an angle, in the range of -*pi*/2 through *pi*/2.

atan(doubleAngle)
Returns the arc tangent of an angle, in the range of -*pi*/2 through *pi*/2.

atan2(xCoordinate, yCoordinate)
Converts rectangular coordinates (`xCoordinate` and `yCoordinate`) to polar (`r` and `theta`). This method computes the phase `theta` by computing an arc tangent of `xCoordinate`/`yCoordinate` in the range of -*pi* to *pi*.

atan2(xCoordinate, yCoordinate)
Converts rectangular coordinates (`xCoordinate` and `yCoordinate`) to polar (`r` and `theta`). This method computes the phase `theta` by computing an arc tangent of `xCoordinate`/`yCoordinate` in the range of -*pi* to *pi*.

cbrt(decimalValue)
Returns the cube root of the specified Decimal. The cube root of a negative value is the negative of the cube root of that value's magnitude.

cbrt(doubleValue)
Returns the cube root of the specified Double. The cube root of a negative value is the negative of the cube root of that value's magnitude.

ceil(decimalValue)
Returns the smallest (closest to negative infinity) Decimal that is not less than the argument and is equal to a mathematical integer.

ceil(doubleValue)
Returns the smallest (closest to negative infinity) Double that is not less than the argument and is equal to a mathematical integer.

cos(decimalAngle)
Returns the trigonometric cosine of the angle specified by `decimalAngle`.

cos(doubleAngle)
Returns the trigonometric cosine of the angle specified by `doubleAngle`.

cosh(decimalAngle)

Returns the hyperbolic cosine of `decimalAngle`. The hyperbolic cosine of `d` is defined to be $(e^x + e^{-x})/2$ where e is Euler's number.

cosh(doubleAngle)

Returns the hyperbolic cosine of `doubleAngle`. The hyperbolic cosine of `d` is defined to be $(e^x + e^{-x})/2$ where e is Euler's number.

exp(exponentDecimal)

Returns Euler's number *e* raised to the power of the specified Decimal.

exp(exponentDouble)

Returns Euler's number *e* raised to the power of the specified Double.

floor(decimalValue)

Returns the largest (closest to positive infinity) Decimal that is not greater than the argument and is equal to a mathematical integer.

floor(doubleValue)

Returns the largest (closest to positive infinity) Double that is not greater than the argument and is equal to a mathematical integer.

log(decimalValue)

Returns the natural logarithm (base *e*) of the specified Decimal.

log(doubleValue)

Returns the natural logarithm (base *e*) of the specified Double.

log10(decimalValue)

Returns the logarithm (base *10*) of the specified Decimal.

log10(doubleValue)

Returns the logarithm (base *10*) of the specified Double.

max(decimalValue1, decimalValue2)

Returns the larger of the two specified Decimals.

max(doubleValue1, doubleValue2)

Returns the larger of the two specified Doubles.

max(integerValue1, integerValue2)

Returns the larger of the two specified Integers.

max(longValue1, longValue2)

Returns the larger of the two specified Longs.

min(decimalValue1, decimalValue2)

Returns the smaller of the two specified Decimals.

min(doubleValue1, doubleValue2)

Returns the smaller of the two specified Doubles.

min(integerValue1, integerValue2)

Returns the smaller of the two specified Integers.

min(longValue1, longValue2)

Returns the smaller of the two specified Longs.

mod(integerValue1, integerValue2)

Returns the remainder of `integerValue1` divided by `integerValue2`.

mod(longValue1, longValue2)

Returns the remainder of `longValue1` divided by `longValue2`.

pow(doubleValue, exponent)

Returns the value of the first Double raised to the power of `exponent`.

random()

Returns a positive Double that is greater than or equal to 0.0 and less than 1.0.

rint(decimalValue)

Returns the value that is closest in value to `decimalValue` and is equal to a mathematical integer.

rint(doubleValue)

Returns the value that is closest in value to `doubleValue` and is equal to a mathematical integer.

round(doubleValue)

Do not use. This method is deprecated as of the Winter '08 release. Instead, use `Math.roundToLong`. Returns the closest Integer to the specified Double. If the result is less than -2,147,483,648 or greater than 2,147,483,647, Apex generates an error.

round(decimalValue)

Returns the rounded approximation of this Decimal. The number is rounded to zero decimal places using half-even rounding mode, that is, it rounds towards the "nearest neighbor" unless both neighbors are equidistant, in which case, this mode rounds towards the even neighbor. If the result is less than -2,147,483,648 or greater than 2,147,483,647, Apex generates an error.

roundToLong(decimalValue)

Returns the rounded approximation of this Decimal. The number is rounded to zero decimal places using half-even rounding mode, that is, it rounds towards the "nearest neighbor" unless both neighbors are equidistant, in which case, this mode rounds towards the even neighbor.

roundToLong(doubleValue)

Returns the closest Long to the specified Double.

signum(decimalValue)

Returns the signum function of the specified Decimal, which is 0 if `decimalValue` is 0, 1.0 if `decimalValue` is greater than 0, -1.0 if `decimalValue` is less than 0.

signum(doubleValue)

Returns the signum function of the specified Double, which is 0 if `doubleValue` is 0, 1.0 if `doubleValue` is greater than 0, -1.0 if `doubleValue` is less than 0.

sin(decimalAngle)

Returns the trigonometric sine of the angle specified by `decimalAngle`.

sin(doubleAngle)

Returns the trigonometric sine of the angle specified by `doubleAngle`.

sinh(decimalAngle)

Returns the hyperbolic sine of `decimalAngle`. The hyperbolic sine of `decimalAngle` is defined to be $(e^x - e^{-x})/2$ where e is Euler's number.

sinh(doubleAngle)

Returns the hyperbolic sine of `doubleAngle`. The hyperbolic sine of `doubleAngle` is defined to be $(e^x - e^{-x})/2$ where e is Euler's number.

sqrt(decimalValue)

Returns the correctly rounded positive square root of `decimalValue`.

sqrt(doubleValue)

Returns the correctly rounded positive square root of `doubleValue`.

tan(decimalAngle)

Returns the trigonometric tangent of the angle specified by *decimalAngle*.

tan(doubleAngle)

Returns the trigonometric tangent of the angle specified by *doubleAngle*.

tanh(decimalAngle)

Returns the hyperbolic tangent of *decimalAngle*. The hyperbolic tangent of *decimalAngle* is defined to be $(e^x - e^{-x})/(e^x + e^{-x})$ where *e* is Euler's number. In other words, it is equivalent to `sinh(x)/cosinh(x)`. The absolute value of the exact `tanh` is always less than 1.

tanh(doubleAngle)

Returns the hyperbolic tangent of *doubleAngle*. The hyperbolic tangent of *doubleAngle* is defined to be $(e^x - e^{-x})/(e^x + e^{-x})$ where *e* is Euler's number. In other words, it is equivalent to `sinh(x)/cosinh(x)`. The absolute value of the exact `tanh` is always less than 1.

abs(decimalValue)

Returns the absolute value of the specified Decimal.

Signature

```
public static Decimal abs(Decimal decimalValue)
```

Parameters

decimalValue
Type: Decimal

Return Value

Type: Decimal

abs(doubleValue)

Returns the absolute value of the specified Double.

Signature

```
public static Double abs(Double doubleValue)
```

Parameters

doubleValue
Type: Double

Return Value

Type: Double

abs(integerValue)

Returns the absolute value of the specified Integer.

Signature

```
public static Integer abs(Integer integerValue)
```

Parameters

integerValue
Type: Integer

Return Value

Type: Integer

Example

```
Integer i = -42;
Integer i2 = math.abs(i);
system.assertEquals(i2, 42);
```

abs(longValue)

Returns the absolute value of the specified Long.

Signature

```
public static Long abs(Long longValue)
```

Parameters

longValue
Type: Long

Return Value

Type: Long

acos(decimalAngle)

Returns the arc cosine of an angle, in the range of 0.0 through *pi*.

Signature

```
public static Decimal acos(Decimal decimalAngle)
```

Parameters

decimalAngle
Type: Decimal

Return Value

Type: Decimal

acos(doubleAngle)

Returns the arc cosine of an angle, in the range of 0.0 through *pi*.

Signature

```
public static Double acos(Double doubleAngle)
```

Parameters

doubleAngle
Type: Double

Return Value

Type: Double

asin(decimalAngle)

Returns the arc sine of an angle, in the range of *-pi*/2 through *pi*/2.

Signature

```
public static Decimal asin(Decimal decimalAngle)
```

Parameters

decimalAngle
Type: Decimal

Return Value

Type: Decimal

asin(doubleAngle)

Returns the arc sine of an angle, in the range of *-pi*/2 through *pi*/2.

Signature

```
public static Double asin(Double doubleAngle)
```

Parameters

doubleAngle
Type: Double

Return Value

Type: Double

atan(decimalAngle)

Returns the arc tangent of an angle, in the range of -*pi*/2 through *pi*/2.

Signature

```
public static Decimal atan(Decimal decimalAngle)
```

Parameters

decimalAngle
Type: Decimal

Return Value

Type: Decimal

atan(doubleAngle)

Returns the arc tangent of an angle, in the range of -*pi*/2 through *pi*/2.

Signature

```
public static Double atan(Double doubleAngle)
```

Parameters

doubleAngle
Type: Double

Return Value

Type: Double

atan2(xCoordinate, yCoordinate)

Converts rectangular coordinates (*xCoordinate* and *yCoordinate*) to polar (*r* and *theta*). This method computes the phase *theta* by computing an arc tangent of *xCoordinate*/*yCoordinate* in the range of -*pi* to *pi*.

Signature

```
public static Decimal atan2(Decimal xCoordinate, Decimal yCoordinate)
```

Parameters

xCoordinate
Type: Decimal

yCoordinate
Type: Decimal

Return Value

Type: Decimal

atan2(xCoordinate, yCoordinate)

Converts rectangular coordinates (*xCoordinate* and *yCoordinate*) to polar (*r* and *theta*). This method computes the phase *theta* by computing an arc tangent of *xCoordinate*/*yCoordinate* in the range of -*pi* to *pi*.

Signature

```
public static Double atan2(Double xCoordinate, Double yCoordinate)
```

Parameters

xCoordinate
Type: Double

yCoordinate
Type: Double

Return Value

Type: Double

cbrt(decimalValue)

Returns the cube root of the specified Decimal. The cube root of a negative value is the negative of the cube root of that value's magnitude.

Signature

```
public static Decimal cbrt(Decimal decimalValue)
```

Parameters

decimalValue
Type: Decimal

Return Value

Type: Decimal

cbrt(doubleValue)

Returns the cube root of the specified Double. The cube root of a negative value is the negative of the cube root of that value's magnitude.

Signature

```
public static Double cbrt(Double doubleValue)
```

Parameters

doubleValue
Type: Double

Return Value

Type: Double

ceil(decimalValue)

Returns the smallest (closest to negative infinity) Decimal that is not less than the argument and is equal to a mathematical integer.

Signature

```
public static Decimal ceil(Decimal decimalValue)
```

Parameters

decimalValue
Type: Decimal

Return Value

Type: Decimal

ceil(doubleValue)

Returns the smallest (closest to negative infinity) Double that is not less than the argument and is equal to a mathematical integer.

Signature

```
public static Double ceil(Double doubleValue)
```

Parameters

doubleValue
Type: Double

Return Value

Type: Double

cos(decimalAngle)

Returns the trigonometric cosine of the angle specified by *decimalAngle*.

Signature

```
public static Decimal cos(Decimal decimalAngle)
```

Parameters

decimalAngle
Type: Decimal

Return Value

Type: Decimal

cos(doubleAngle)

Returns the trigonometric cosine of the angle specified by *doubleAngle*.

Signature

```
public static Double cos(Double doubleAngle)
```

Parameters

doubleAngle
Type: Double

Return Value

Type: Double

cosh(decimalAngle)

Returns the hyperbolic cosine of *decimalAngle*. The hyperbolic cosine of *d* is defined to be $(e^X + e^{-X})/2$ where *e* is Euler's number.

Signature

```
public static Decimal cosh(Decimal decimalAngle)
```

Parameters

decimalAngle
Type: Decimal

Return Value

Type: Decimal

cosh(doubleAngle)

Returns the hyperbolic cosine of *doubleAngle*. The hyperbolic cosine of *d* is defined to be $(e^x + e^{-x})/2$ where *e* is Euler's number.

Signature

```
public static Double cosh(Double doubleAngle)
```

Parameters

doubleAngle
Type: Double

Return Value

Type: Double

exp(exponentDecimal)

Returns Euler's number *e* raised to the power of the specified Decimal.

Signature

```
public static Decimal exp(Decimal exponentDecimal)
```

Parameters

exponentDecimal
Type: Decimal

Return Value

Type: Decimal

exp(exponentDouble)

Returns Euler's number *e* raised to the power of the specified Double.

Signature

```
public static Double exp(Double exponentDouble)
```

Parameters

exponentDouble
Type: Double

Return Value

Type: Double

floor(decimalValue)

Returns the largest (closest to positive infinity) Decimal that is not greater than the argument and is equal to a mathematical integer.

Signature

```
public static Decimal floor(Decimal decimalValue)
```

Parameters

decimalValue
Type: Decimal

Return Value

Type: Decimal

floor(doubleValue)

Returns the largest (closest to positive infinity) Double that is not greater than the argument and is equal to a mathematical integer.

Signature

```
public static Double floor(Double doubleValue)
```

Parameters

doubleValue
Type: Double

Return Value

Type: Double

log(decimalValue)

Returns the natural logarithm (base *e*) of the specified Decimal.

Signature

```
public static Decimal log(Decimal decimalValue)
```

Parameters

decimalValue
Type: Decimal

Return Value

Type: Decimal

log(doubleValue)

Returns the natural logarithm (base *e*) of the specified Double.

Signature

```
public static Double log(Double doubleValue)
```

Parameters

doubleValue
Type: Double

Return Value

Type: Double

log10(decimalValue)

Returns the logarithm (base *10*) of the specified Decimal.

Signature

```
public static Decimal log10(Decimal decimalValue)
```

Parameters

decimalValue
Type: Decimal

Return Value

Type: Decimal

log10(doubleValue)

Returns the logarithm (base *10*) of the specified Double.

Signature

```
public static Double log10(Double doubleValue)
```

Parameters

doubleValue
Type: Double

Return Value

Type: Double

max(decimalValue1, decimalValue2)

Returns the larger of the two specified Decimals.

Signature

```
public static Decimal max(Decimal decimalValue1, Decimal decimalValue2)
```

Parameters

decimalValue1
Type: Decimal

decimalValue2
Type: Decimal

Return Value

Type: Decimal

Example

```
Decimal larger = math.max(12.3, 156.6);
system.assertEquals(larger, 156.6);
```

max(doubleValue1, doubleValue2)

Returns the larger of the two specified Doubles.

Signature

```
public static Double max(Double doubleValue1, Double doubleValue2)
```

Parameters

doubleValue1
Type: Double

doubleValue2
Type: Double

Return Value

Type: Double

max(integerValue1, integerValue2)

Returns the larger of the two specified Integers.

Signature

```
public static Integer max(Integer integerValue1, Integer integerValue2)
```

Parameters

integerValue1
Type: Integer

integerValue2
Type: Integer

Return Value

Type: Integer

max(longValue1, longValue2)

Returns the larger of the two specified Longs.

Signature

```
public static Long max(Long longValue1, Long longValue2)
```

Parameters

longValue1
Type: Long

longValue2
Type: Long

Return Value

Type: Long

min(decimalValue1, decimalValue2)

Returns the smaller of the two specified Decimals.

Signature

```
public static Decimal min(Decimal decimalValue1, Decimal decimalValue2)
```

Parameters

decimalValue1
Type: Decimal

decimalValue2
Type: Decimal

Return Value

Type: Decimal

Example

```
Decimal smaller = math.min(12.3, 156.6);
system.assertEquals(smaller, 12.3);
```

min(doubleValue1, doubleValue2)

Returns the smaller of the two specified Doubles.

Signature

```
public static Double min(Double doubleValue1, Double doubleValue2)
```

Parameters

doubleValue1
Type: Double

doubleValue2
Type: Double

Return Value

Type: Double

min(integerValue1, integerValue2)

Returns the smaller of the two specified Integers.

Signature

```
public static Integer min(Integer integerValue1, Integer integerValue2)
```

Parameters

integerValue1
Type: Integer

integerValue2
Type: Integer

Return Value

Type: Integer

min(longValue1, longValue2)

Returns the smaller of the two specified Longs.

Signature

```
public static Long min(Long longValue1, Long longValue2)
```

Parameters

longValue1
Type: Long

longValue2
Type: Long

Return Value

Type: Long

mod(integerValue1, integerValue2)

Returns the remainder of *integerValue1* divided by *integerValue2*.

Signature

```
public static Integer mod(Integer integerValue1, Integer integerValue2)
```

Parameters

integerValue1
Type: Integer

integerValue2
Type: Integer

Return Value

Type: Integer

Example

```
Integer remainder = math.mod(12, 2);
system.assertEquals(remainder, 0);

Integer remainder2 = math.mod(8, 3);
system.assertEquals(remainder2, 2);
```

mod(longValue1, longValue2)

Returns the remainder of *longValue1* divided by *longValue2*.

Signature

```
public static Long mod(Long longValue1, Long longValue2)
```

Parameters

longValue1
Type: Long

longValue2
Type: Long

Return Value

Type: Long

pow(doubleValue, exponent)

Returns the value of the first Double raised to the power of *exponent*.

Signature

```
public static Double pow(Double doubleValue, Double exponent)
```

Parameters

doubleValue
Type: Double

exponent
Type: Double

Return Value

Type: Double

random()

Returns a positive Double that is greater than or equal to 0.0 and less than 1.0.

Signature

```
public static Double random()
```

Return Value

Type: Double

rint(decimalValue)

Returns the value that is closest in value to *decimalValue* and is equal to a mathematical integer.

Signature

```
public static Decimal rint(Decimal decimalValue)
```

Parameters

decimalValue
Type: Decimal

Return Value

Type: Decimal

rint(doubleValue)

Returns the value that is closest in value to *doubleValue* and is equal to a mathematical integer.

Signature

```
public static Double rint(Double doubleValue)
```

Parameters

doubleValue
Type: Double

Return Value

Type: Double

round(doubleValue)

Do not use. This method is deprecated as of the Winter '08 release. Instead, use `Math.roundToLong`. Returns the closest Integer to the specified Double. If the result is less than -2,147,483,648 or greater than 2,147,483,647, Apex generates an error.

Signature

```
public static Integer round(Double doubleValue)
```

Parameters

doubleValue
Type: Double

Return Value

Type: Integer

round(decimalValue)

Returns the rounded approximation of this Decimal. The number is rounded to zero decimal places using half-even rounding mode, that is, it rounds towards the "nearest neighbor" unless both neighbors are equidistant, in which case, this mode rounds towards the even neighbor. If the result is less than -2,147,483,648 or greater than 2,147,483,647, Apex generates an error.

Signature

```
public static Integer round(Decimal decimalValue)
```

Parameters

decimalValue
Type: Decimal

Return Value

Type: Integer

Usage

Note that this rounding mode statistically minimizes cumulative error when applied repeatedly over a sequence of calculations.

Example

```
Decimal d1 = 4.5;
Integer i1 = Math.round(d1);
System.assertEquals(4, i1);

Decimal d2 = 5.5;
Integer i2 = Math.round(d2);
System.assertEquals(6, i2);
```

roundToLong(decimalValue)

Returns the rounded approximation of this Decimal. The number is rounded to zero decimal places using half-even rounding mode, that is, it rounds towards the "nearest neighbor" unless both neighbors are equidistant, in which case, this mode rounds towards the even neighbor.

Signature

```
public static Long roundToLong(Decimal decimalValue)
```

Parameters

decimalValue
Type: Decimal

Return Value

Type: Long

Usage

Note that this rounding mode statistically minimizes cumulative error when applied repeatedly over a sequence of calculations.

Example

```
Decimal d1 = 4.5;
Long i1 = Math.roundToLong(d1);
System.assertEquals(4, i1);
```

```
Decimal d2 = 5.5;
Long i2 = Math.roundToLong(d2);
System.assertEquals(6, i2);
```

roundToLong(doubleValue)

Returns the closest Long to the specified Double.

Signature

```
public static Long roundToLong(Double doubleValue)
```

Parameters

doubleValue
Type: Double

Return Value

Type: Long

signum(decimalValue)

Returns the signum function of the specified Decimal, which is 0 if *decimalValue* is 0, 1.0 if *decimalValue* is greater than 0, -1.0 if *decimalValue* is less than 0.

Signature

```
public static Decimal signum(Decimal decimalValue)
```

Parameters

decimalValue
Type: Decimal

Return Value

Type: Decimal

signum(doubleValue)

Returns the signum function of the specified Double, which is 0 if *doubleValue* is 0, 1.0 if *doubleValue* is greater than 0, -1.0 if *doubleValue* is less than 0.

Signature

```
public static Double signum(Double doubleValue)
```

Parameters

doubleValue
Type: Double

Return Value

Type: Double

sin(decimalAngle)

Returns the trigonometric sine of the angle specified by *decimalAngle*.

Signature

```
public static Decimal sin(Decimal decimalAngle)
```

Parameters

decimalAngle
Type: Decimal

Return Value

Type: Decimal

sin(doubleAngle)

Returns the trigonometric sine of the angle specified by *doubleAngle*.

Signature

```
public static Double sin(Double doubleAngle)
```

Parameters

doubleAngle
Type: Double

Return Value

Type: Double

sinh(decimalAngle)

Returns the hyperbolic sine of *decimalAngle*. The hyperbolic sine of *decimalAngle* is defined to be $(e^x - e^{-x})/2$ where e is Euler's number.

Signature

```
public static Decimal sinh(Decimal decimalAngle)
```

Parameters

decimalAngle
Type: Decimal

Return Value

Type: Decimal

sinh(doubleAngle)

Returns the hyperbolic sine of *doubleAngle*. The hyperbolic sine of *doubleAngle* is defined to be $(e^X - e^{-X})/2$ where e is Euler's number.

Signature

```
public static Double sinh(Double doubleAngle)
```

Parameters

doubleAngle
Type: Double

Return Value

Type: Double

sqrt(decimalValue)

Returns the correctly rounded positive square root of *decimalValue*.

Signature

```
public static Decimal sqrt(Decimal decimalValue)
```

Parameters

decimalValue
Type: Decimal

Return Value

Type: Decimal

sqrt(doubleValue)

Returns the correctly rounded positive square root of *doubleValue*.

Signature

```
public static Double sqrt(Double doubleValue)
```

Parameters

doubleValue
Type: Double

Return Value

Type: Double

tan(decimalAngle)

Returns the trigonometric tangent of the angle specified by *decimalAngle*.

Signature

```
public static Decimal tan(Decimal decimalAngle)
```

Parameters

decimalAngle
Type: Decimal

Return Value

Type: Decimal

tan(doubleAngle)

Returns the trigonometric tangent of the angle specified by *doubleAngle*.

Signature

```
public static Double tan(Double doubleAngle)
```

Parameters

doubleAngle
Type: Double

Return Value

Type: Double

tanh(decimalAngle)

Returns the hyperbolic tangent of *decimalAngle*. The hyperbolic tangent of *decimalAngle* is defined to be $(e^x - e^{-x})/(e^x + e^{-x})$ where *e* is Euler's number. In other words, it is equivalent to `sinh(x)/cosinh(x)`. The absolute value of the exact `tanh` is always less than 1.

Signature

```
public static Decimal tanh(Decimal decimalAngle)
```

Parameters

decimalAngle
Type: Decimal

Return Value

Type: Decimal

tanh(doubleAngle)

Returns the hyperbolic tangent of *doubleAngle*. The hyperbolic tangent of *doubleAngle* is defined to be $(e^x - e^{-x})/(e^x + e^{-x})$ where *e* is Euler's number. In other words, it is equivalent to `sinh(x)/cosinh(x)`. The absolute value of the exact `tanh` is always less than 1.

Signature

```
public static Double tanh(Double doubleAngle)
```

Parameters

doubleAngle
Type: Double

Return Value

Type: Double

Messaging Class

Contains messaging methods used when sending a single or mass email.

Namespace

System

Messaging Methods

The following are methods for `Messaging`. All are instance methods.

IN THIS SECTION:

extractInboundEmail(source, includeForwardedAttachments)
Use this method in your email service code to control how to parse and process forwarded or attached emails. Returns an instance of `Messaging.InboundEmail` from a stream of data that is in RFC822 format. The data stream can be a forwarded email in an attachment to an existing InboundEmail, or a stream from another source.

reserveMassEmailCapacity(amountReserved)

Reserves email capacity to send mass email to the specified number of email addresses, after the current transaction commits.

reserveSingleEmailCapacity(amountReserved)

Reserves email capacity to send single email to the specified number of email addresses, after the current transaction commits.

sendEmail(emails, allOrNothing)

Sends the list of emails instantiated with either `SingleEmailMessage` or `MassEmailMessage` and returns a list of SendEmailResult objects. When org preferences are set to save EmailMessage objects and a trigger is defined for EmailMessage objects, the trigger is fired for each `SingleEmailMessage` individually. The `sendEmail` method can be called 10 times per Apex transaction and each method invocation can include up to 100 "To", 25 "Cc", and 25 "Bcc" recipients.

sendEmailMessage(emailMessageIds, allOrNothing)

Sends draft email messages as defined by the specified email message IDs and returns a list of SendEmailResult objects.

renderEmailTemplate(whoId, whatId, bodies)

Replaces merge fields in text bodies of email templates with values from Salesforce records. Returns an array of `RenderEmailTemplateBodyResult` objects, each of which corresponds to an element in the supplied array of text bodies. Each `RenderEmailTemplateBodyResult` provides a success or failure indication, along with either an error code or the rendered text.

renderStoredEmailTemplate(templateId, whoId, whatId)

Renders a text, custom, HTML, or Visualforce email template that exists in the database into an instance of `Messaging.SingleEmailMessage`. Includes all attachment content in the returned email message.

renderStoredEmailTemplate(templateId, whoId, whatId, attachmentRetrievalOption)

Renders a text, custom, HTML, or Visualforce email template that exists in the database into an instance of `Messaging.SingleEmailMessage`. Provides options for including attachment metadata only, attachment metadata and content, or excluding attachments.

renderStoredEmailTemplate(templateId, whoId, whatId, attachmentRetrievalOption, updateEmailTemplateUsage)

Renders a text, custom, HTML, or Visualforce email template that exists in the database into an instance of `Messaging.SingleEmailMessage`. Provides options for including attachment metadata only, attachment metadata and content, or excluding attachments.

`extractInboundEmail(source, includeForwardedAttachments)`

Use this method in your email service code to control how to parse and process forwarded or attached emails. Returns an instance of `Messaging.InboundEmail` from a stream of data that is in RFC822 format. The data stream can be a forwarded email in an attachment to an existing InboundEmail, or a stream from another source.

Signature

```
public static Messaging.InboundEmail extractInboundEmail(Object source, Boolean
includeForwardedAttachments)
```

Parameters

source

Type: Object

An instance of `Messaging.InboundEmail.BinaryAttachment` whose MimeTypeSubtype is `message/rfc822` or a Blob. If *source* is a Blob, then supply a byte array in RFC822 format.

includeForwardedAttachments
Type: Boolean

This parameter controls how attachments to embedded or forwarded emails are handled. Set to `true` to provide all attachments, even attachments in embedded emails in the `binaryAttachments` and `textAttachments` properties of the returned value. Set to `false` to provide only the attachments that are at the top level of the source email.

Return Value

Type: Messaging.InboundEmail

reserveMassEmailCapacity(amountReserved)

Reserves email capacity to send mass email to the specified number of email addresses, after the current transaction commits.

Signature

```
public Void reserveMassEmailCapacity(Integer amountReserved)
```

Parameters

amountReserved
Type: Integer

Return Value

Type: Void

Usage

This method can be called when you know in advance how many addresses emails will be sent to as a result of the transaction. If the transaction would cause the organization to exceed its daily email limit, using this method results in the following error: `System.HandledException: The daily limit for the org would be exceeded by this request.` If the organization doesn't have permission to send API or mass email, using this method results in the following error: `System.NoAccessException: The organization is not permitted to send email.`

reserveSingleEmailCapacity(amountReserved)

Reserves email capacity to send single email to the specified number of email addresses, after the current transaction commits.

Signature

```
public Void reserveSingleEmailCapacity(Integer amountReserved)
```

Parameters

amountReserved
Type: Integer

Return Value

Type: Void

Usage

This method can be called when you know in advance how many addresses emails will be sent to as a result of the transaction. If the transaction would cause the organization to exceed its daily email limit, using this method results in the following error: `System.HandledException: The daily limit for the org would be exceeded by this request.` If the organization doesn't have permission to send API or mass email, using this method results in the following error: `System.NoAccessException: The organization is not permitted to send email.`

`sendEmail(emails, allOrNothing)`

Sends the list of emails instantiated with either `SingleEmailMessage` or `MassEmailMessage` and returns a list of SendEmailResult objects. When org preferences are set to save EmailMessage objects and a trigger is defined for EmailMessage objects, the trigger is fired for each `SingleEmailMessage` individually. The `sendEmail` method can be called 10 times per Apex transaction and each method invocation can include up to 100 "To", 25 "Cc", and 25 "Bcc" recipients.

Signature

```
public Messaging.SendEmailResult[] sendEmail(Messaging.Email[] emails, Boolean
allOrNothing)
```

Parameters

`emails`
Type: Messaging.Email[]

`allOrNothing`
Type: Boolean

The optional *`opt_allOrNone`* parameter specifies whether `sendEmail` prevents delivery of all other messages when any of the messages fail due to an error (`true`), or whether it allows delivery of the messages that don't have errors (`false`). The default is `true`.

Return Value

Type: Messaging.SendEmailResult[]

`sendEmailMessage(emailMessageIds, allOrNothing)`

Sends draft email messages as defined by the specified email message IDs and returns a list of SendEmailResult objects.

Signature

```
public Messaging.SendEmailResult[] sendEmailMessage(List<ID> emailMessageIds, Boolean
allOrNothing)
```

Parameters

emailMessageIds
Type: List<ID>

allOrNothing
Type: Boolean

Return Value

Type: Messaging.SendEmailResult[]

If the *emailMessageIds* parameter is null, the method throws a `System.IllegalArgumentException` exception.

Usage

The `sendEmailMessage` method assumes that the optional *allOrNothing* parameter is always `false` and ignores the value you set. Delivery of all messages is attempted even if some messages fail due to an error.

The email address of the user calling the `sendEmailMessage` method is inserted in the From Address field of the email header and the Email Message record.

Example

This example shows how to send a draft email message. It creates a case and a new email message associated with the case. Next, the example sends a draft email message and checks the results. Before running this example, make sure to replace the email address with a valid address.

```
Case c = new Case();
insert c;

EmailMessage e = new EmailMessage();
e.parentid = c.id;
// Set to draft status.
// This status is required
// for sendEmailMessage().
e.Status = '5';
e.TextBody =
  'Sample email message.';
e.Subject = 'Apex sample';
e.ToAddress = 'customer@email.com';
insert e;

List<Messaging.SendEmailResult>
  results =
  Messaging.sendEmailMessage(new ID[]
    { e.id });

System.assertEquals(1, results.size());
System.assertEquals(true,
                    results[0].success);
```

Versioned Behavior Changes

In API version 54.0 and later, a null *emailMessageIds* parameter results in a `System.IllegalArgumentException` exception. In API version 53.0 and earlier, a null *emailMessageIds* parameter results in an error.

renderEmailTemplate(whoId, whatId, bodies)

Replaces merge fields in text bodies of email templates with values from Salesforce records. Returns an array of `RenderEmailTemplateBodyResult` objects, each of which corresponds to an element in the supplied array of text bodies. Each `RenderEmailTemplateBodyResult` provides a success or failure indication, along with either an error code or the rendered text.

Signature

```
public static List<Messaging.RenderEmailTemplateBodyResult> renderEmailTemplate(String whoId, String whatId, List<String> bodies)
```

Parameters

whoId
Type: String

The identifier of an object in the database, typically a contact, lead, or user. The database record for that object is read and used in merge field processing.

whatId
Type: String

Identifies an object in the database like an account or opportunity. The record for that object is read and used in merge field processing.

bodies
Type: List<String>

An array of strings that are examined for merge field references. The corresponding data from the object referenced by the `whoId` or `whatId` replaces the merge field reference.

Return Value

Type: List<Messaging.RenderEmailTemplateBodyResult>

Usage

Use this method in situations in which you want to dynamically compose blocks of text that are enriched with data from the database. You can then use the the rendered blocks of text to compose and send an email or update a text value in another database record.

Executing the `renderEmailTemplate` method counts toward the SOQL governor limit. The number of SOQL queries that this method consumes is the number of elements in the list of strings passed in the *bodies* parameter.

SEE ALSO:

Execution Governors and Limits

renderStoredEmailTemplate(templateId, whoId, whatId)

Renders a text, custom, HTML, or Visualforce email template that exists in the database into an instance of `Messaging.SingleEmailMessage`. Includes all attachment content in the returned email message.

Signature

```
public static Messaging.SingleEmailMessage renderStoredEmailTemplate(String templateId,
String whoId, String whatId)
```

Parameters

templateId
Type: String

An email template that exists in the database, such as text, HTML, custom, and Visualforce templates.

whoId
Type: String

The identifier of an object in the database, typically a contact, lead, or user. The database record for that object is read and used in merge field processing.

whatId
Type: String

Identifies an object in the database, like an account or opportunity. The record for that object is read and used in merge field processing.

Return Value

Type: Messaging.SingleEmailMessage

Usage

Executing the `renderStoredEmailTemplate` method counts toward the SOQL governor limit as one query.

SEE ALSO:

Execution Governors and Limits

renderStoredEmailTemplate(templateId, whoId, whatId, attachmentRetrievalOption)

Renders a text, custom, HTML, or Visualforce email template that exists in the database into an instance of `Messaging.SingleEmailMessage`. Provides options for including attachment metadata only, attachment metadata and content, or excluding attachments.

Signature

```
public static Messaging.SingleEmailMessage renderStoredEmailTemplate(String templateId,
String whoId, String whatId, Messaging.AttachmentRetrievalOption
attachmentRetrievalOption)
```

Parameters

templateId
Type: String

An email template that exists in the database, such as text, HTML, custom, and Visualforce templates.

whoId
Type: String

The identifier of an object in the database, typically a contact, lead, or user. The database record for that object is read and used in merge field processing.

whatId
Type: String

Identifies an object in the database, like an account or opportunity. The record for that object is read and used in merge field processing.

attachmentRetrievalOption
Type: Messaging.AttachmentRetrievalOption

Specifies options for including attachments in the `fileAttachments` property of the returned `Messaging.SingleEmailMessage`. Set to one of the Messaging.AttachmentRetrievalOption values to include attachment metadata only, attachment metadata and content, or to exclude attachments.

> **Note:** When the *attachmentRetrievalOption* parameter is *not* set to `NONE`, the `entityAttachments` property of `Messaging.SingleEmailMessage` contains the ID of the Salesforce content objects to attach (ContentVersion or Document). The `fileAttachments` property contains the IDs of attachments, in addition to all the IDs in the `entityAttachments` property. As a result, the ID values in `entityAttachments` are duplicates of the IDs in the `fileAttachments` property. If you call `renderStoredEmailTemplate()` by passing the `METADATA_WITH_BODY` option, and send the rendered email message, the email will contain duplicate attachments. Before using the returned email message with sendEmail(emails, allOrNothing), you can remove attachments from `fileAttachments` that are duplicated in `entityAttachments`.

Return Value

Type: Messaging.SingleEmailMessage

Usage

Executing the `renderStoredEmailTemplate` method counts toward the SOQL governor limit as one query.

renderStoredEmailTemplate(templateId, whoId, whatId, attachmentRetrievalOption, updateEmailTemplateUsage)

Renders a text, custom, HTML, or Visualforce email template that exists in the database into an instance of `Messaging.SingleEmailMessage`. Provides options for including attachment metadata only, attachment metadata and content, or excluding attachments.

Signature

```
public static Messaging.SingleEmailMessage renderStoredEmailTemplate(String templateId,
String whoId, String whatId, Messaging.AttachmentRetrievalOption
attachmentRetrievalOption, Boolean updateEmailTemplateUsage)
```

Parameters

templateId
Type: String

An email template that exists in the database, such as text, HTML, custom, and Visualforce templates.

whoId
Type: String

The identifier of an object in the database, typically a contact, lead, or user. The database record for that object is read and used in merge field processing.

whatId
Type: String

Identifies an object in the database, like an account or opportunity. The record for that object is read and used in merge field processing.

attachmentRetrievalOption
Type: Messaging.AttachmentRetrievalOption

Specifies options for including attachments in the `fileAttachments` property of the returned `Messaging.SingleEmailMessage`. Set to one of the Messaging.AttachmentRetrievalOption values to include attachment metadata only, attachment metadata and content, or to exclude attachments.

> **Note:** When the *`attachmentRetrievalOption`* parameter is *not* set to `NONE`, the `entityAttachments` property of `Messaging.SingleEmailMessage` contains the ID of the Salesforce content objects to attach (ContentVersion or Document). The `fileAttachments` property contains the IDs of attachments, in addition to all the IDs in the `entityAttachments` property. As a result, the ID values in `entityAttachments` are duplicates of the IDs in the `fileAttachments` property. If you call `renderStoredEmailTemplate()` by passing the `METADATA_WITH_BODY` option, and send the rendered email message, the email will contain duplicate attachments. Before using the returned email message with sendEmail(emails, allOrNothing), you can remove attachments from `fileAttachments` that are duplicated in `entityAttachments`.

updateEmailTemplateUsage
Type: Boolean

Specifies whether the usage field in the EmailTemplate record is updated upon successful rendering.

Return Value

Type: Messaging.SingleEmailMessage

Usage

Executing the `renderStoredEmailTemplate` method counts toward the SOQL governor limit as one query.

MultiStaticResourceCalloutMock Class

Utility class used to specify a fake response using multiple resources for testing HTTP callouts.

Namespace

System

Usage

Use the methods in this class to set the response properties for testing HTTP callouts. You can specify a resource for each endpoint.

IN THIS SECTION:

MultiStaticResourceCalloutMock Constructors

MultiStaticResourceCalloutMock Methods

MultiStaticResourceCalloutMock Constructors

The following are constructors for `MultiStaticResourceCalloutMock`.

IN THIS SECTION:

MultiStaticResourceCalloutMock()
Creates a new instance of the `System.MultiStaticResourceCalloutMock` class.

`MultiStaticResourceCalloutMock()`

Creates a new instance of the `System.MultiStaticResourceCalloutMock` class.

Signature

```
public MultiStaticResourceCalloutMock()
```

MultiStaticResourceCalloutMock Methods

The following are methods for `MultiStaticResourceCalloutMock`. All are instance methods.

IN THIS SECTION:

setHeader(headerName, headerValue)
Sets the specified header name and value for the fake response.

setStaticResource(endpoint, resourceName)
Sets the specified static resource corresponding to the endpoint. The static resource contains the response body.

setStatus(httpStatus)
Sets the specified HTTP status for the response.

setStatusCode(httpStatusCode)
Sets the specified HTTP status code for the response.

`setHeader(headerName, headerValue)`

Sets the specified header name and value for the fake response.

Signature

```
public Void setHeader(String headerName, String headerValue)
```

Parameters

headerName
Type: String

headerValue
Type: String

Return Value

Type: Void

setStaticResource(endpoint, resourceName)

Sets the specified static resource corresponding to the endpoint. The static resource contains the response body.

Signature

```
public Void setStaticResource(String endpoint, String resourceName)
```

Parameters

endpoint
Type: String

resourceName
Type: String

Return Value

Type: Void

setStatus(httpStatus)

Sets the specified HTTP status for the response.

Signature

```
public Void setStatus(String httpStatus)
```

Parameters

httpStatus
Type: String

Return Value

Type: Void

setStatusCode(httpStatusCode)

Sets the specified HTTP status code for the response.

Signature

```
public Void setStatusCode(Integer httpStatusCode)
```

Parameters

`httpStatusCode`
Type: Integer

Return Value

Type: Void

Network Class

Manage Experience Cloud sites.

Namespace

System

IN THIS SECTION:

Network Constructors
Create an instance of the `System.Network` class.

Network Methods
Get the default landing page, login page, and self-registration page of a site. Asynchronously create site users and records. Get the login and logout URLs for a site. Get a user's current site. Map dashboards and Insights reports.

Network Constructors

Create an instance of the `System.Network` class.

The following are constructors for `Network`.

IN THIS SECTION:

Network()
Creates a new instance of the `System.Network` class.

`Network()`

Creates a new instance of the `System.Network` class.

Signature

```
public Network()
```

Network Methods

Get the default landing page, login page, and self-registration page of a site. Asynchronously create site users and records. Get the login and logout URLs for a site. Get a user's current site. Map dashboards and Insights reports.

The following are methods for `Network`. All methods are static.

IN THIS SECTION:

communitiesLanding()
Returns a Page Reference to the default landing page for the Experience Cloud site. This is the first tab of the site.

createExternalUserAsync(user, contact, account)
Asynchronously creates an Experience Cloud site user for the given account or contact and associates it with the site. This method processes requests in batches and then sends an email with login information to the user.

createRecordAsync(processType, mbObject)
Asynchronously creates case, lead, and custom object records. This method collects record creation requests and processes them in batches.

forwardToAuthPage(startURL)
Returns a Page Reference to the default login page. StartURL is included as a query paremeter for where to redirect after a successful login.

getLoginUrl(networkId)
Returns the absolute URL of the login page used by the Experience Cloud site.

getLogoutUrl(networkId)
Returns the absolute URL of the logout page used by the Experience Cloud site.

getNetworkId()
Returns the user's current Experience Cloud site.

getSelfRegUrl(networkId)
Returns the absolute URL of the self-registration page used by the Experience Cloud site.

loadAllPackageDefaultNetworkDashboardSettings()
Maps the dashboards from the Salesforce Communities Management package onto each Experience Cloud site's unconfigured dashboard settings. Returns the number of settings it configures.

loadAllPackageDefaultNetworkPulseSettings()
Maps the Insights reports from the Salesforce Communities Management package onto each Experience Cloud site's unconfigured Insights settings. Returns the number of settings it configures.

`communitiesLanding()`

Returns a Page Reference to the default landing page for the Experience Cloud site. This is the first tab of the site.

Signature

```
public static String communitiesLanding()
```

Return Value

Type: PageReference

Usage

If digital experiences isn't enabled for the user's org or the user is currently in the internal org, returns `null`.

createExternalUserAsync(user, contact, account)

Asynchronously creates an Experience Cloud site user for the given account or contact and associates it with the site. This method processes requests in batches and then sends an email with login information to the user.

Signature

```
public static String createExternalUserAsync(SObject user, SObject contact, SObject account)
```

Parameters

user
Type: SObject (optional)

Information required to create a user.

contact
Type: SObject (optional)

The contact you want to associate the user with.

account
Type: SObject

The account you want to associate the user with.

Return Value

Type: String

Returns the UUID for the site user.

createRecordAsync(processType, mbObject)

Asynchronously creates case, lead, and custom object records. This method collects record creation requests and processes them in batches.

Signature

```
public static String createRecordAsync(String processType, SObject mbObject)
```

Parameters

processType
Type: String

The process you use to create records.

mbObject
Type: SObject

The records created for objects. Objects must be supported by the high-volume record creation.

Return Value

Type: String

Returns the UUID for the record created.

forwardToAuthPage(startURL)

Returns a Page Reference to the default login page. StartURL is included as a query paremeter for where to redirect after a successful login.

Signature

```
public static PageReference forwardToAuthPage(String startURL)
```

Parameters

startURL
Type: String

Return Value

Type: PageReference

Usage

If digital experiences isn't enabled for the user's org or the user is currently in the internal org, returns `null`.

getLoginUrl(networkId)

Returns the absolute URL of the login page used by the Experience Cloud site.

Signature

```
public static String getLoginUrl(String networkId)
```

Parameters

networkId
Type: String

The ID of the Experience Cloud site you're retrieving this information for.

Return Value

Type: String

Usage

Returns the full URL for the Lightning Platform or Experience Builder page used as the login page in the Experience Cloud site.

getLogoutUrl(networkId)

Returns the absolute URL of the logout page used by the Experience Cloud site.

Signature

```
public static String getLogoutUrl(String networkId)
```

Parameters

networkId
Type: String

The ID of the Experience Cloud site you're retrieving this information for.

Return Value

Type: String

Usage

Returns the full URL for the Lightning Platform page, Experience Builder page, or Web page used as the logout page in the Experience Cloud site.

getNetworkId()

Returns the user's current Experience Cloud site.

Signature

```
public static String getNetworkId()
```

Return Value

Type: String

Usage

If digital experiences isn't enabled for the user's org or the user is currently in the internal org, returns `null`.

getSelfRegUrl(networkId)

Returns the absolute URL of the self-registration page used by the Experience Cloud site.

Signature

```
public static String getSelfRegUrl(String networkId)
```

Parameters

networkId
Type: String

The ID of the Experience Cloud site you're retrieving this information for.

Return Value

Type: String

Usage

Returns the full URL for the Lightning Platform or Experience Builder page used as the self-registration page in the Experience Cloud site.

`loadAllPackageDefaultNetworkDashboardSettings()`

Maps the dashboards from the Salesforce Communities Management package onto each Experience Cloud site's unconfigured dashboard settings. Returns the number of settings it configures.

Signature

```
public static Integer loadAllPackageDefaultNetworkDashboardSettings()
```

Return Value

Type: Integer

Usage

If digital experiences is enabled, and the Salesforce Communities Management package is installed, maps the dashboards provided in the package onto each Experience Cloud site's unconfigured dashboard settings. Returns the number of settings it configures. This method is invoked automatically during site creation and package installation, but isn't typically invoked manually.

If digital experiences isn't enabled for the user's org or the user is in the internal org, returns `0`.

`loadAllPackageDefaultNetworkPulseSettings()`

Maps the Insights reports from the Salesforce Communities Management package onto each Experience Cloud site's unconfigured Insights settings. Returns the number of settings it configures.

Signature

```
public static Integer loadAllPackageDefaultNetworkPulseSettings()
```

Return Value

Type: Integer

Usage

If digital experiences is enabled, and the Salesforce Communities Management package is installed, maps the Insights reports provided in the package onto each Experience Cloud site's unconfigured Insights settings. Returns the number of settings it configures. This method is invoked automatically during site creation and package installation, but isn't typically invoked manually.

If digital experiences isn't enabled for the user's org or the user is in the internal org, returns `0`.

Object Class

Contains methods that are implemented by all Apex types.

Namespace

System

Usage

All Apex classes have the Object class as the base class, and therefore implement all the Object class methods.

IN THIS SECTION:

Object Methods

Object Methods

The following are methods for `Object`.

IN THIS SECTION:

equals(obj)
Compares an object to the specified object and returns true if both are equal. Otherwise, returns false.

hashCode()
Returns a hash code for the object.

toString()
Returns a string that represents the object. The string includes the class name of which the object is an instance, the at (@) character, and the unsigned hexadecimal representation of the object's hash code value.

`equals(obj)`

Compares an object to the specified object and returns true if both are equal. Otherwise, returns false.

Signature

```
public Boolean equals(Object obj)
```

Parameters

obj
Type: Object

The object with which to compare.

Return Value

Type: Boolean

Usage

If x, y, and z are non-null instances of a class, the `equals` method must be:

- Reflexive: `x.equals(x)`
- Symmetric: `x.equals(y)` returns `true` if and only if `y.equals(x)` returns `true`
- Transitive: If `x.equals(y)` returns `true` and `y.equals(z)` returns `true`, then `x.equals(z)` returns `true`
- Consistent: Multiple invocations of `x.equals(y)` consistently return `true` or consistently return `false`, provided the objects used in comparison are not modified.
- For any non-null reference value x, `x.equals(null)` returns `false`

Use the `equals` method in your class to simplify comparision of objects. You can use the `==` operator to compare objects, or the `equals` method. For example:

```
// obj1 and obj2 are instances of MyClass
if (obj1 == obj2) {
    // Do something
}

if (obj1.equals(obj2)) {
    // Do something
}
```

hashCode()

Returns a hash code for the object.

Signature

```
public Integer hashCode()
```

Return Value

Type: Integer

Usage

- If the `hashCode` method is invoked on the same object more than once during execution of an Apex request, it must return the same value.
  - The hash code value is same provided no information used in equals comparisons on the object is modified.
  - The hash code value need not remain consistent from one Apex execution request to another execution of the same application.
- If two objects are equal, based on the `equals` method, `hashCode` must return the same value.
- If two objects are unequal, based on the result of the `equals` method, it is not required that `hashCode` return distinct values.

toString()

Returns a string that represents the object. The string includes the class name of which the object is an instance, the at (@) character, and the unsigned hexadecimal representation of the object's hash code value.

Signature

```
public String toString()
```

Return Value

Type: String

Versioned Behavior Changes

In API version 57.0 and later, the `toString()` method only includes member variables of Apex objects that are visible in the current namespace. Non-global properties are suppressed from output when you invoke `toString()` on managed Apex types. To keep the non-global state of the object visible in debug output, you can explicitly override the `toString()` method.

OrgLimit Class

Contains methods that provide the name, maximum value, and current value of an org limit.

Namespace

System

Usage

Use the `System.OrgLimits` `getAll` and `getMap` methods to obtain either a list or a map of all your org limits. To get details on each limit, use instance methods from `System.OrgLimit`.

For comparison, the Limits Class returns Apex governor limits and not Salesforce API limits.

Note: Limit values are updated asynchronously, in near-real-time.

IN THIS SECTION:

OrgLimit Methods

OrgLimit Methods

The following are methods for `OrgLimit`.

IN THIS SECTION:

getLimit()
Returns the maximum allowed limit value.

getName()
Returns the limit's name.

getValue()
Returns the limit usage value.

toString()
Returns the string representation of the org limit.

getLimit()

Returns the maximum allowed limit value.

Signature

```
public Integer getLimit()
```

Return Value

Type: Integer

Example

```
List<System.OrgLimit> limits = OrgLimits.getAll();
for (System.OrgLimit aLimit: limits) {
    System.debug('Limit: ' + aLimit.getName());
    System.debug('Max Limit is: ' + aLimit.getLimit());
}
```

getName()

Returns the limit's name.

Signature

```
public String getName()
```

Return Value

Type: String

Example

```
List<System.OrgLimit> limits = OrgLimits.getAll();
for (System.OrgLimit aLimit: limits) {
    System.debug('Limit: ' + aLimit.getName());
    System.debug('Max Limit is: ' + aLimit.getLimit());
}
```

getValue()

Returns the limit usage value.

Signature

```
public Integer getValue()
```

Return Value

Type: Integer

Example

```
List<System.OrgLimit> limits = OrgLimits.getAll();
for (System.OrgLimit aLimit: limits) {
    System.debug('Limit: ' + aLimit.getName());
    System.debug('Usage Value is: ' + aLimit.getValue());
}
```

`toString()`

Returns the string representation of the org limit.

Signature

```
public String toString()
```

Return Value

Type: String

String denoting the name, current consumption, and maximum value of the org limit. For example:

```
OrgLimit[DailyBulkApiBatches: consumed 25 of 15000]
```

OrgLimits Class

Contains methods that provide a list or map of all OrgLimit instances for Salesforce your org, such as SOAP API requests, Bulk API requests, and Streaming API limits.

Namespace

System

Usage

Use the `System.OrgLimits` `getAll` and `getMap` methods to obtain either a list or a map of all your org limits. To get details on each limit, use instance methods from `System.OrgLimit`.

For comparison, the Limits Class returns Apex governor limits and not Salesforce API limits.

Note: Limit values are updated asynchronously, in near-real-time.

IN THIS SECTION:

OrgLimits Methods

SEE ALSO:

REST API Developer Guide: Limits

OrgLimits Methods

The following are methods for `OrgLimits`.

IN THIS SECTION:

getAll()
Returns a list of OrgLimit instances.

getMap()
Returns a map of all OrgLimit instances with the limit name as key.

`getAll()`

Returns a list of OrgLimit instances.

Signature

```
public static List<System.OrgLimit> getAll()
```

Return Value

Type: List<System.OrgLimit>

`getMap()`

Returns a map of all OrgLimit instances with the limit name as key.

Signature

```
public static Map<String,System.OrgLimit> getMap()
```

Return Value

Type: Map<String,System.OrgLimit>

Example

```
Map<String,System.OrgLimit> limitsMap = OrgLimits.getMap();
System.OrgLimit apiRequestsLimit = limitsMap.get('DailyApiRequests');
System.debug('Limit Name: ' + apiRequestsLimit.getName());
System.debug('Usage Value: ' + apiRequestsLimit.getValue());
System.debug('Maximum Limit: ' + apiRequestsLimit.getLimit());
```

PageReference Class

A PageReference is a reference to an instantiation of a page. Among other attributes, PageReferences consist of a URL and a set of query parameter names and values.

Namespace

System

Use a PageReference object:

- To view or set query string parameters and values for a page
- To navigate the user to a different page as the result of an action method

Instantiation

In a custom controller or controller extension, you can refer to or instantiate a PageReference in one of the following ways:

- ```
  Page.existingPageName
  ```

  Refers to a PageReference for a Visualforce page that has already been saved in your organization. By referring to a page in this way, the platform recognizes that this controller or controller extension is dependent on the existence of the specified page and will prevent the page from being deleted while the controller or extension exists.

- ```
  PageReference pageRef = new PageReference('partialURL');
  ```

  Creates a PageReference to any page that is hosted on the Lightning platform. For example, setting `'partialURL'` to `'/apex/HelloWorld'` refers to the Visualforce page located at `http://mySalesforceInstance/apex/HelloWorld`. Likewise, setting `'partialURL'` to `'/' + 'recordID'` refers to the detail page for the specified record.

  This syntax is less preferable for referencing other Visualforce pages than `Page.existingPageName` because the PageReference is constructed at runtime, rather than referenced at compile time. Runtime references are not available to the referential integrity system. Consequently, the platform doesn't recognize that this controller or controller extension is dependent on the existence of the specified page and won't issue an error message to prevent user deletion of the page.

- ```
  PageReference pageRef = new PageReference('fullURL');
  ```

  Creates a PageReference for an external URL. For example:

  ```
  PageReference pageRef = new PageReference('http://www.google.com');
  ```

You can also instantiate a PageReference object for the current page with the `currentPage` ApexPages method. For example:

```
PageReference pageRef = ApexPages.currentPage();
```

Request Headers

The following table is a non-exhaustive list of headers that are set on requests.

Header	Description
Host	The host name requested in the request URL. This header is always set on Lightning Platform Site requests and My Domain requests. This header is optional on other requests when HTTP/1.0 is used instead of HTTP/1.1.
Referer	The URL that is either included or linked to the current request's URL. This header is optional.
```

Header	Description
User-Agent	The name, version, and extension support of the program that initiated this request, such as a web browser. This header is optional and can be overridden in most browsers to be a different value. Therefore, this header should not be relied upon.
CipherSuite	If this header exists and has a non-blank value, this means that the request is using HTTPS. Otherwise, the request is using HTTP. The contents of a non-blank value are not defined by this API, and can be changed without notice.
X-Salesforce-SIP	The source IP address of the request. This header is always set on HTTP and HTTPS requests that are initiated outside of Salesforce's data centers. Note: If a request passes through a content delivery network (CDN) or proxy server, the source IP address might be altered, and no longer the original client IP address.
X-Salesforce-Forwarded-To	The fully qualified domain name of the Salesforce instance that is handling this request. This header is always set on HTTP and HTTPS requests that are initiated outside of Salesforce's data centers.

Example: Retrieving Query String Parameters

The following example shows how to use a PageReference object to retrieve a query string parameter in the current page URL. In this example, the `getAccount` method references the `id` query string parameter:

```
public with sharing class MyController {
   public Account getAccount() {
        return [SELECT Id, Name FROM Account WITH SECURITY_ENFORCED
                WHERE Id = :ApexPages.currentPage().getParameters().get('Id')];
    }
}
```

The following page markup calls the `getAccount` method from the controller above:

```
<apex:page controller="MyController">
    <apex:pageBlock title="Retrieving Query String Parameters">
        You are viewing the {!account.name} account.
    </apex:pageBlock>
</apex:page>
```

Note: For this example to render properly, you must associate the Visualforce page with a valid account record in the URL. For example, if `001D000000IRt53` is the account ID, the resulting URL should be:

```
https://Visualforce_Url/apex/MyFirstPage?id=001D000000IRt53
```

Replace *`Visualforce_URL`* with the Visualforce URL for your org. For production, this URL is in the format ***`MyDomainName--PackageName`***`.vf.force.com`, and if your installed package is unmanaged, the package name is `c`. For more information on the format of the URLs that Salesforce serves for your org, see My Domain Login and Application URL Formats and Partitioned Domains in Salesforce Help.

The `getAccount` method uses an embedded SOQL query to return the account specified by the `id` parameter in the URL of the page. To access `id`, the `getAccount` method uses the `ApexPages` namespace:

- First the `currentPage` method returns the `PageReference` instance for the current page. `PageReference` returns a reference to a Visualforce page, including its query string parameters.

- Using the page reference, use the `getParameters` method to return a map of the specified query string parameter names and values.
- Then a call to the `get` method specifying `id` returns the value of the `id` parameter itself.

Example: Navigating to a New Page as the Result of an Action Method

Any action method in a custom controller or controller extension can return a PageReference object as the result of the method. If the `redirect` attribute on the PageReference is set to `true`, the user navigates to the URL specified by the PageReference.

The following example shows how this can be implemented with a `save` method. In this example, the PageReference returned by the `save` method redirects the user to the detail page for the account record that was just saved:

```
public class mySecondController {
    Account account;

    public Account getAccount() {
        if(account == null) account = new Account();
        return account;
    }

    public PageReference save() {
        // Add the account to the database.
        insert account;
        // Send the user to the detail page for the new account.
        PageReference acctPage = new ApexPages.StandardController(account).view();
        acctPage.setRedirect(true);
        return acctPage;
    }
}
```

The following page markup calls the `save` method from the controller above. When a user clicks **Save**, he or she is redirected to the detail page for the account just created:

```
<apex:page controller="mySecondController" tabStyle="Account">
    <apex:sectionHeader title="New Account Edit Page" />
    <apex:form>
        <apex:pageBlock title="Create a New Account">
            <apex:pageBlockButtons location="bottom">
                <apex:commandButton action="{!save}" value="Save"/>
            </apex:pageBlockButtons>
            <apex:pageBlockSection title="Account Information">
                <apex:inputField id="accountName" value="{!account.name}"/>
                <apex:inputField id="accountSite" value="{!account.site}"/>
            </apex:pageBlockSection>
       </apex:pageBlock>
    </apex:form>
</apex:page>
```

Example: Redirect Users to a Replacement Experience Cloud Site

The following example shows how to redirect a user attempting to access a retired feedback site to a self-service help site. If the `redirect` attribute is set to `true` on the PageReference for the feedback site, the user navigates to the URL specified by the

PageReference. The `redirectCode` attribute defines the redirection type for search engine optimization in public Experience Cloud sites.

```
public class RedirectController {
    // Redirect users to the self-service help site    public PageReference redirect() {
        final PageReference target = new
        PageReference(Site.getBaseSecureUrl() + '/SiteLogin');
        target.setRedirect(true);
        // This is a permanent redirection
        target.setRedirectCode(301);
        return target;
    }
}
```

The following example shows how to call the RedirectController class from the retired site page.

```
<apex:page controller="RedirectController" action="{!redirect}"/>
```

IN THIS SECTION:

PageReference Constructors

PageReference Methods

PageReference Constructors

The following are constructors for `PageReference`.

IN THIS SECTION:

PageReference(partialURL)

Creates a new instance of the `PageReference` class using the specified URL.

PageReference(record)

Generate a new instance of the `PageReference` class for the specified sObject record.

`PageReference(partialURL)`

Creates a new instance of the `PageReference` class using the specified URL.

Signature

```
public PageReference(String partialURL)
```

Parameters

`partialURL`

Type: String

The partial URL of a page hosted on the Lightning Platform or a full external URL. The following are some examples of the *`partialURL`* parameter values:

- `/apex/HelloWorld`: refers to the Visualforce page located at `http://`*`MyDomainName`*`-`*`PackageName`*`.vf.force.com/apex/HelloWorld`.

- */recordID*: refers to the detail page of a specified record.
- `http://www.google.com`: refers to an external URL.

`PageReference(record)`

Generate a new instance of the `PageReference` class for the specified sObject record.

Signature

```
public PageReference(SObject record)
```

Parameters

record
Type: SObject

The sObject record that references the `ApexPage`. The reference must be an `ApexPage`.

SEE ALSO:

Visualforce Developer Guide: apex:page

SOAP API Developer Guide: ApexPage

PageReference Methods

The following are methods for `PageReference`. All are instance methods.

IN THIS SECTION:

forResource(resourceName, path)
Create a PageReference for nested content inside a zip static resource, by name and path.

forResource(resourceName)
Create a PageReference for a static resource, by name.

getAnchor()
Returns the name of the anchor referenced in the page's URL. That is, the part of the URL after the hashtag (#).

getContent()
Returns the output of the page, as displayed to a user in a web browser.

getContentAsPDF()
Returns the page in PDF, regardless of the `<apex:page>` component's `renderAs` attribute.

getCookies()
Returns a map of cookie names and cookie objects, where the key is a String of the cookie name and the the value contains the cookie object with that name.

getHeaders()
Returns a map of the request headers, where the key string contains the name of the header, and the value string contains the value of the header.

getParameters()

Returns a map of the query string parameters for the PageReference; both POST and GET parameters are included. The key string contains the name of the parameter, while the value string contains the value of the parameter.

getRedirect()

Returns the current value of the PageReference object's `redirect` attribute.

getRedirectCode()

Returns the HTTP redirect code used when getRedirect() is set to `true` for the PageReference object.

getUrl()

Returns the relative URL associated with the PageReference when it was originally defined, including any query string parameters and anchors.

setAnchor(anchor)

Sets the URL's anchor reference to the specified string.

setCookies(cookies)

Creates a list of cookie objects. Used in conjunction with the `Cookie` class.

setRedirect(redirect)

Sets the value of the PageReference object's `redirect` attribute. If set to `true`, a redirect is performed through a client side redirect.

setRedirectCode(redirectCode)

Sets the HTTP redirect code to use for the PageReference object when setRedirect(redirect) is set to `true`.

forResource(resourceName, path)

Create a PageReference for nested content inside a zip static resource, by name and path.

Signature

```
public static System.PageReference forResource(String resourceName, String path)
```

Parameters

resourceName

Type: String

The resource name

path

Type: String

The resource path

Return Value

Type: System.PageReference

forResource(resourceName)

Create a PageReference for a static resource, by name.

Signature

```
public static System.PageReference forResource(String resourceName)
```

Parameters

resourceName
Type: String

The resource name

Return Value

Type: System.PageReference

getAnchor()

Returns the name of the anchor referenced in the page's URL. That is, the part of the URL after the hashtag (#).

Signature

```
public String getAnchor()
```

Return Value

Type: String

Note: Instances of `PageReference` returned by `ApexPages.currentPage()` have a null anchor attribute, because URL fragments are not sent to the Salesforce server during a request.

getContent()

Returns the output of the page, as displayed to a user in a web browser.

Signature

```
public Blob getContent()
```

Return Value

Type: Blob

Usage

The content of the returned Blob depends on how the page is rendered. If the page is rendered as a PDF file, it returns the PDF document. If the page isn't rendered as PDF, it returns HTML. To access the content of the returned HTML as a string, use the `toString` Blob method. If the Visualforce page has an error, an `ExecutionException` is thrown.

You can't use the `getContent` method in:

- Triggers
- Test methods. If you use `getContent` in a test method, the test method fails. `getContent` is treated as a callout in API version 34.0 and later.

- Apex email services

You also can't use the method to retrieve the output of a different Visualforce page with the same controller and controller extensions. Instead, pass the base URL of the destination page.

```
new PageReference(Site.getBaseUrl() + '/apex/VisualforcePageName').getContent();
```

getContentAsPDF()

Returns the page in PDF, regardless of the <apex:page> component's renderAs attribute.

Signature

```
public Blob getContentAsPDF()
```

Return Value

Type: Blob

Usage

This method can't be used in:

- Triggers
- Test methods. If you use getContentAsPDF in a test method, the test method fails. getContentAsPDF is treated as a callout in API version 34.0 and later.
- Apex email services

You also can't use the method to retrieve the output of a different Visualforce page with the same controller and controller extensions. Instead, pass the base URL of the destination page.

```
new PageReference(Site.getBaseUrl() + '/apex/VisualforcePageName').getContentAsPDF();
```

getCookies()

Returns a map of cookie names and cookie objects, where the key is a String of the cookie name and the the value contains the cookie object with that name.

Signature

```
public Map<String, System.Cookie> getCookies()
```

Return Value

Type: Map<String, System.Cookie>

Usage

Used in conjunction with the Cookie class. Only returns cookies with the "apex__" prefix set by the setCookies method.

getHeaders()

Returns a map of the request headers, where the key string contains the name of the header, and the value string contains the value of the header.

Signature

```
public Map<String, String> getHeaders()
```

Return Value

Type: Map<String, String>

Usage

This map can be modified and remains in scope for the PageReference object. For instance, you could do:

```
PageReference.getHeaders().put('Date', '9/9/99');
```

For a description of request headers, see Request Headers.

getParameters()

Returns a map of the query string parameters for the PageReference; both POST and GET parameters are included. The key string contains the name of the parameter, while the value string contains the value of the parameter.

Signature

```
public Map<String, String> getParameters()
```

Return Value

Type: Map<String, String>

Usage

This map can be modified and remains in scope for the PageReference object. For instance, you could do:

```
PageReference.getParameters().put('id', myID);
```

Parameter keys are case-insensitive. For example:

```
System.assert(
    ApexPages.currentPage().getParameters().get('myParamName') ==
    ApexPages.currentPage().getParameters().get('myparamname'));
```

getRedirect()

Returns the current value of the PageReference object's `redirect` attribute.

Signature

```
public Boolean getRedirect()
```

Return Value

Type: Boolean

Usage

Note that if the URL of the PageReference object is set to a website outside of the `salesforce.com` domain, the redirect always occurs, regardless of whether the `redirect` attribute is set to `true` or `false`.

getRedirectCode()

Returns the HTTP redirect code used when getRedirect() is set to `true` for the PageReference object.

Signature

```
public Integer getRedirectCode()
```

Return Value

Type: Integer

Possible Values:

- 0 — Redirect using the default redirect action for this PageReference. Typically a JavaScript-based redirection or HTTP 302.

  Note: Site URLRewriter Interface implementations pointing to a PageReference with a *`redirectCode`* of 0 are not redirected.

- 301 — Moved Permanently. Redirect users by sending an HTTP GET request to the target location. Includes instructions to update any references to the requested URL with the target location.
- 302 — Moved Temporarily. Redirect users by sending an HTTP GET request to the target location. Because the redirection is temporary, it doesn't include update instructions.
- 303 — See Other. Redirect users by sending an HTTP GET request to the target location. Not commonly used. Useful when the client sends a POST request and you want the client to call the new web page using a GET request instead of a POST request.
- 307 — Temporary Redirect. Send the same HTTP request, regardless of the HTTP method, to the target location. Because the redirection is temporary, it doesn't include update instructions.
- 308 — Permanent Redirect. Send the same HTTP request, regardless of the HTTP method, to the target location. Includes instructions to update any references to the requested URL with the target location.

getUrl()

Returns the relative URL associated with the PageReference when it was originally defined, including any query string parameters and anchors.

Signature

```
public String getUrl()
```

Return Value

Type: String

`setAnchor(anchor)`

Sets the URL's anchor reference to the specified string.

Signature

```
public System.PageReference setAnchor(String anchor)
```

Parameters

anchor
Type: String

Return Value

Type: System.PageReference

`setCookies(cookies)`

Creates a list of cookie objects. Used in conjunction with the `Cookie` class.

Signature

```
public Void setCookies(Cookie[] cookies)
```

Parameters

cookies
Type: System.Cookie[]

Return Value

Type: Void

Usage

Important:

- Cookie names and values set in Apex are URL encoded, that is, characters such as @ are replaced with a percent sign and their hexadecimal representation.
- The `setCookies` method adds the prefix "`apex__`" to the cookie names.
- Setting a cookie's value to `null` sends a cookie with an empty string value instead of setting an expired attribute.
- After you create a cookie, the properties of the cookie can't be changed.
- Be careful when storing sensitive information in cookies. Pages are cached regardless of a cookie value. If you use a cookie value to generate dynamic content, you should disable page caching. For more information, see Configure Site Caching in Salesforce Help.

`setRedirect(redirect)`

Sets the value of the PageReference object's `redirect` attribute. If set to `true`, a redirect is performed through a client side redirect.

Signature

```
public System.PageReference setRedirect(Boolean redirect)
```

Parameters

redirect
Type: Boolean

Return Value

Type: System.PageReference

Usage

This type of redirect performs an HTTP GET request, and flushes the view state, which uses POST. If set to `false`, the redirect is a server-side forward that preserves the view state if and only if the target page uses the same controller and contains the proper subset of extensions used by the source page.

Note that if the URL of the PageReference object is set to a website outside of the `salesforce.com` domain, or to a page with a different controller or controller extension, the redirect always occurs, regardless of whether the `redirect` attribute is set to `true` or `false`.

`setRedirectCode(redirectCode)`

Sets the HTTP redirect code to use for the PageReference object when setRedirect(redirect) is set to `true`.

Signature

```
public System.PageReference setRedirectCode(Integer redirectCode)
```

Parameters

redirectCode
Type: Integer

Valid values:

- 0 — Redirect using the default redirect action for this PageReference. Typically a JavaScript-based redirection or HTTP 302.

  Note: Site URLRewriter Interface implementations pointing to a PageReference with a *redirectCode* of 0 are not redirected.

- 301 — Moved Permanently. Redirect users by sending an HTTP GET request to the target location. Includes instructions to update any references to the requested URL with the target location.
- 302 — Moved Temporarily. Redirect users by sending an HTTP GET request to the target location. Because the redirection is temporary, it doesn't include update instructions.
- 303 — See Other. Redirect users by sending an HTTP GET request to the target location. Not commonly used. Useful when the client sends a POST request and you want the client to call the new web page using a GET request instead of a POST request.
- 307 — Temporary Redirect. Send the same HTTP request, regardless of the HTTP method, to the target location. Because the redirection is temporary, it doesn't include update instructions.
- 308 — Permanent Redirect. Send the same HTTP request, regardless of the HTTP method, to the target location. Includes instructions to update any references to the requested URL with the target location.

If the redirect code contains an invalid integer, an error message is displayed when `PageReference` is used by Salesforce for redirection.

Return Value

Type: System.PageReference

Packaging Class

Contains a method for obtaining information about managed and unlocked packages.

Namespace

System

Usage

In the context of a package, use the getCurrentPackageId method to retrieve the packageID.

IN THIS SECTION:

Packaging Methods

Packaging Methods

The following are methods for `Packaging`.

IN THIS SECTION:

getCurrentPackageId()
Returns the context `packageID` in managed and unlocked packages.

`getCurrentPackageId()`

Returns the context `packageID` in managed and unlocked packages.

Signature

```
public String getCurrentPackageId()
```

Return Value

Type: String

Usage

For managed packages, this method can be combined with isCurrentUserLicensedForPackage(packageId) to retrieve the `packageId` at runtime. Then, use `packageId` to confirm that the contextual user is licensed to use that managed package.

Pattern Class

Represents a compiled representation of a regular expression.

Namespace

System

Pattern Methods

The following are methods for `Pattern`.

IN THIS SECTION:

compile(regExp)
Compiles the regular expression into a Pattern object.

matcher(stringtoMatch)
Creates a Matcher object that matches the input string *`stringtoMatch`* against this Pattern object.

matches(regExp, stringtoMatch)
Compiles the regular expression *`regExp`* and tries to match it against the specified string. This method returns `true` if the specified string matches the regular expression, `false` otherwise.

pattern()
Returns the regular expression from which this Pattern object was compiled.

quote(yourString)
Returns a string that can be used to create a pattern that matches the string *`yourString`* as if it were a literal pattern.

split(regExp)
Returns a list that contains each substring of the String that matches this pattern.

split(regExp, limit)
Returns a list that contains each substring of the String that is terminated either by the regular expression *`regExp`* that matches this pattern, or by the end of the String.

`compile(regExp)`

Compiles the regular expression into a Pattern object.

Signature

```
public static Pattern compile(String regExp)
```

Parameters

`regExp`
Type: String

Return Value

Type: System.Pattern

matcher(stringtoMatch)

Creates a Matcher object that matches the input string *stringtoMatch* against this Pattern object.

Signature

```
public Matcher matcher(String stringtoMatch)
```

Parameters

stringtoMatch
Type: String

Return Value

Type: Matcher

matches(regExp, stringtoMatch)

Compiles the regular expression *regExp* and tries to match it against the specified string. This method returns `true` if the specified string matches the regular expression, `false` otherwise.

Signature

```
public static Boolean matches(String regExp, String stringtoMatch)
```

Parameters

regExp
Type: String

stringtoMatch
Type: String

Return Value

Type: Boolean

Usage

If a pattern is to be used multiple times, compiling it once and reusing it is more efficient than invoking this method each time.

Example

Note that the following code example:

```
Pattern.matches(regExp, input);
```

produces the same result as this code example:

```
Pattern.compile(regex).
matcher(input).matches();
```

pattern()

Returns the regular expression from which this Pattern object was compiled.

Signature

```
public String pattern()
```

Return Value

Type: String

quote(yourString)

Returns a string that can be used to create a pattern that matches the string *yourString* as if it were a literal pattern.

Signature

```
public static String quote(String yourString)
```

Parameters

yourString
Type: String

Return Value

Type: String

Usage

Metacharacters (such as `$` or `^`) and escape sequences in the input string are treated as literal characters with no special meaning.

split(regExp)

Returns a list that contains each substring of the String that matches this pattern.

Signature

```
public String[] split(String regExp)
```

Parameters

regExp
Type: String

Return Value

Type: String[]

Note: In API version 34.0 and earlier, a zero-width *regExp* value produces an empty list item at the beginning of the method's output.

Usage

The substrings are placed in the list in the order in which they occur in the String. If *regExp* does not match the pattern, the resulting list has just one element containing the original String.

split(regExp, limit)

Returns a list that contains each substring of the String that is terminated either by the regular expression *regExp* that matches this pattern, or by the end of the String.

Signature

```
public String[] split(String regExp, Integer limit)
```

Parameters

regExp
Type: String

limit
Type: Integer

(Optional) Controls the number of times the pattern is applied and therefore affects the length of the list.

- If *limit* is greater than zero:
  - The pattern is applied a maximum of (*limit* – 1) times.
  - The list's length is no greater than *limit*.
  - The list's last entry contains all input beyond the last matched delimiter.
- If *limit* is non-positive, the pattern is applied as many times as possible, and the list can have any length.
- If *limit* is zero, the pattern is applied as many times as possible, the list can have any length, and trailing empty strings are discarded.

Return Value

Type: String[]

Note: In API version 34.0 and earlier, a zero-width *regExp* value produces an empty list item at the beginning of the method's output.

Queueable Interface

Enables the asynchronous execution of Apex jobs that can be monitored.

Namespace

System

Usage

To execute Apex as an asynchronous job, implement the `Queueable` interface and add the processing logic in your implementation of the `execute` method.

To implement the `Queueable` interface, you must first declare a class with the `implements` keyword as follows:

```
public class MyQueueableClass implements Queueable {
```

Next, your class must provide an implementation for the following method:

```
public void execute(QueueableContext context) {
    // Your code here
}
```

Your class and method implementation must be declared as `public` or `global`.

To submit your class for asynchronous execution, call the `System.enqueueJob` by passing it an instance of your class implementation of the `Queueable` interface as follows:

```
ID jobID = System.enqueueJob(new MyQueueableClass());
```

IN THIS SECTION:

Queueable Methods

Queueable Example Implementation

SEE ALSO:

Apex Developer Guide: Queueable Apex

Queueable Methods

The following are methods for `Queueable`.

IN THIS SECTION:

execute(context)
Executes the queueable job.

`execute(context)`

Executes the queueable job.

Signature

```
public void execute(QueueableContext context)
```

Parameters

`context`
Type: QueueableContext

Contains the job ID.

Return Value

Type: Void

Queueable Example Implementation

This example is an implementation of the `Queueable` interface. The `execute` method in this example inserts a new account.

```
public class AsyncExecutionExample implements Queueable {
    public void execute(QueueableContext context) {
        Account a = new Account(Name='Acme',Phone='(415) 555-1212');
        insert a;
    }
}
```

To add this class as a job on the queue, call this method:

```
ID jobID = System.enqueueJob(new AsyncExecutionExample());
```

After you submit your queueable class for execution, the job is added to the queue and will be processed when system resources become available. You can monitor the status of your job programmatically by querying AsyncApexJob or through the user interface in Setup by entering *`Apex Jobs`* in the `Quick Find` box, then selecting **Apex Jobs**.

To query information about your submitted job, perform a SOQL query on AsyncApexJob by filtering on the job ID that the `System.enqueueJob` method returns. This example uses the jobID variable that was obtained in the previous example.

```
AsyncApexJob jobInfo = [SELECT Status,NumberOfErrors FROM AsyncApexJob WHERE Id=:jobID];
```

Similar to future jobs, queueable jobs don't process batches, and so the number of processed batches and the number of total batches are always zero.

Testing Queueable Jobs

This example shows how to test the execution of a queueable job in a test method. A queueable job is an asynchronous process. To ensure that this process runs within the test method, the job is submitted to the queue between the `Test.startTest` and `Test.stopTest` block. The system executes all asynchronous processes started in a test method synchronously after the `Test.stopTest` statement. Next, the test method verifies the results of the queueable job by querying the account that the job created.

```
@isTest
public class AsyncExecutionExampleTest {
    static testmethod void test1() {
        // startTest/stopTest block to force async processes
        //   to run in the test.
        Test.startTest();
        System.enqueueJob(new AsyncExecutionExample());
        Test.stopTest();

        // Validate that the job has run
        // by verifying that the record was created.
        // This query returns only the account created in test context by the
        // Queueable class method.
        Account acct = [SELECT Name,Phone FROM Account WHERE Name='Acme' LIMIT 1];
        System.assertNotEquals(null, acct);
        System.assertEquals('(415) 555-1212', acct.Phone);
```

```
    }
}
```

Note: The ID of a queueable Apex job isn't returned in test context—`System.enqueueJob` returns `null` in a running test.

QueueableContext Interface

Represents the parameter type of the `execute()` method in a class that implements the `Queueable` interface and contains the job ID. This interface is implemented internally by Apex.

Namespace

System

QueueableContext Methods

The following are methods for `QueueableContext`.

IN THIS SECTION:

getJobId()
Returns the ID of the submitted job that uses the `Queueable` interface.

`getJobId()`

Returns the ID of the submitted job that uses the `Queueable` interface.

Signature

```
public ID getJobId()
```

Return Value

Type: ID

The ID of the submitted job.

SEE ALSO:

Apex Developer Guide: Queueable Apex

QueueableDuplicateSignature Class

Used in the `AsyncOptions` class to store the queueable job signature in the `DuplicateSignature` property.

Namespace

System

IN THIS SECTION:

QueueableDuplicateSignature Methods

QueueableDuplicateSignature Methods

The following are methods for `QueueableDuplicateSignature`.

IN THIS SECTION:

toString()
Returns the duplicate signature as a string value.

`toString()`

Returns the duplicate signature as a string value.

Signature

```
public String toString()
```

Return Value

Type: String

QueueableDuplicateSignature.Builder Class

Build a unique signature for your queueable job using this inner builder class. The `build()` class method builds a `QueueableDuplicateSignature` object, with input from the `addId()`, `addInteger()`, and `addString()` methods. Use the `DuplicateSignature` property in the `AsyncOptions` class to store the queueable job signature. Enqueue your job by using the `System.enqueueJob()` with the `AsyncOptions` parameter.

Namespace

System

Examples

This example builds the async job signature with UserId and the string `MyQueueable`.

```
AsyncOptions options = new AsyncOptions();
options.DuplicateSignature = new System.QueueableDuplicateSignature.Builder()
                                .addId(UserInfo.getUserId())
                                .addString('MyQueueable')
                                .build();
try {
    System.enqueueJob(new MyQueueable(), options);
} catch (DuplicateMessageException ex) {
    //Exception is thrown if there is already an enqueued job with the same signature
    Assert.areEqual('Attempt to enqueue job with duplicate queueable signature',
```

```
        ex.getMessage());
}
```

This example builds the async job signature using ApexClass Id and the hash value of an sObject.

```
AsyncOptions options = new AsyncOptions();
options.DuplicateSignature = new QueueableDuplicateSignature.Builder()
                                .addInteger(System.hashCode(someAccount))
                                .addId([SELECT Id FROM ApexClass
                                     WHERE Name='MyQueueable'].Id)
                                .build();
System.enqueueJob(new MyQueueable(), options);
```

IN THIS SECTION:

QueueableDuplicateSignature.Builder Methods

QueueableDuplicateSignature.Builder Methods

The following are methods for `QueueableDuplicateSignature.Builder`.

IN THIS SECTION:

addId(inputId)

Adds an ID to build a unique signature for a queueable job. You can then enqueue the job by using the signature as the `AsyncOptions` parameter to `System.enqueueJob()`.

addInteger(inputInteger)

Adds an integer to build a unique signature for a queueable job. You can then enqueue the job by using the signature as the `AsyncOptions` parameter to `System.enqueueJob()`.

addString(inputString)

Adds a string to build a unique signature for a queueable job. You can then enqueue the job by using the signature as the `AsyncOptions` parameter to `System.enqueueJob()`.

build()

Builds a unique signature for a queueable job. You can then enqueue the job by using the signature as the `AsyncOptions` parameter to `System.enqueueJob()`.

getMaxSize()

Gets the maximum size of the queueable job signature in bytes.

getRemainingSize()

Gets the remaining size of the queueable job signature in bytes, after subtracting what is already used by the signature from the maximum allowed number.

getSize()

Gets the size of the queueable job signature in bytes.

addId(inputId)

Adds an ID to build a unique signature for a queueable job. You can then enqueue the job by using the signature as the `AsyncOptions` parameter to `System.enqueueJob()`.

Signature

```
public System.QueueableDuplicateSignature.Builder addId(Id id)
```

Parameters

inputId
Type: Id

Return Value

Type: QueueableDuplicateSignature.Builder

addInteger(inputInteger)

Adds an integer to build a unique signature for a queueable job. You can then enqueue the job by using the signature as the `AsyncOptions` parameter to `System.enqueueJob()`.

Signature

```
public System.QueueableDuplicateSignature.Builder addInteger(Integer i)
```

Parameters

inputInteger
Type: Integer

Return Value

Type: QueueableDuplicateSignature.Builder

addString(inputString)

Adds a string to build a unique signature for a queueable job. You can then enqueue the job by using the signature as the `AsyncOptions` parameter to `System.enqueueJob()`.

Signature

```
public System.QueueableDuplicateSignature.Builder addString(String s)
```

Parameters

inputString
Type: String

Return Value

Type: QueueableDuplicateSignature.Builder

build()

Builds a unique signature for a queueable job. You can then enqueue the job by using the signature as the `AsyncOptions` parameter to `System.enqueueJob()`.

Signature

```
public System.QueueableDuplicateSignature build()
```

Return Value

Type: QueueableDuplicateSignature Class

getMaxSize()

Gets the maximum size of the queueable job signature in bytes.

Signature

```
public Integer getMaxSize()
```

Return Value

Type: Integer

getRemainingSize()

Gets the remaining size of the queueable job signature in bytes, after subtracting what is already used by the signature from the maximum allowed number.

Signature

```
public Integer getRemainingSize()
```

Return Value

Type: Integer

getSize()

Gets the size of the queueable job signature in bytes.

Signature

```
public Integer getSize()
```

Return Value

Type: Integer

QuickAction Class

Use Apex to request and process actions on objects that allow custom fields, on objects that appear in a Chatter feed, or on objects that are available globally.

Namespace

System

Example

In this sample, the trigger determines if the new contacts to be inserted are created by a quick action. If so, it sets the `WhereFrom__c` custom field to a value that depends on whether the quick action is global or local to the contact. Otherwise, if the inserted contacts don't originate from a quick action, the `WhereFrom__c` field is set to `'NoAction'`.

```
trigger accTrig2 on Contact (before insert) {
    for (Contact c : Trigger.new) {
        if (c.getQuickActionName() == QuickAction.CreateContact) {
            c.WhereFrom__c = 'GlobaActionl';
        } else if (c.getQuickActionName() == Schema.Account.QuickAction.CreateContact) {
            c.WhereFrom__c = 'AccountAction';
        } else if (c.getQuickActionName() == null) {
            c.WhereFrom__c = 'NoAction';
        } else {
            System.assert(false);
        }
    }
}
```

This sample performs a global action—`QuickAction.CreateContact`–on the passed-in contact object.

```
public Id globalCreate(Contact c) {
    QuickAction.QuickActionRequest req = new QuickAction.QuickActionRequest();
    req.quickActionName = QuickAction.CreateContact;
    req.record = c;
    QuickAction.QuickActionResult res = QuickAction.performQuickAction(req);
    return c.id;
}
```

SEE ALSO:

QuickActionRequest Class

QuickActionResult Class

QuickAction Methods

The following are methods for `QuickAction`. All methods are static.

IN THIS SECTION:

describeAvailableQuickActions(parentType)

Returns metadata information for the available quick actions of the provided parent object.

describeQuickActions(sObjectNames)

Returns the metadata information for the provided quick actions.

performQuickAction(quickActionRequest)

Performs the quick action specified in the quick action request and returns the action result.

performQuickAction(quickActionRequest, allOrNothing)

Performs the quick action specified in the quick action request with the option for partial success, and returns the result.

performQuickActions(quickActionRequests)

Performs the quick actions specified in the quick action request list and returns action results.

performQuickActions(quickActionRequests, allOrNothing)

Performs the quick actions specified in the quick action request list with the option for partial success, and returns action results.

describeAvailableQuickActions(parentType)

Returns metadata information for the available quick actions of the provided parent object.

Signature

```
public static List<QuickAction.DescribeAvailableQuickActionResult>
describeAvailableQuickActions(String parentType)
```

Parameters

parentType

Type: String

The parent object type. This can be an object type name ('Account') or 'Global' (meaning that this method is called at a global level and not an entity level).

Return Value

Type: List<QuickAction.DescribeAvailableQuickActionResult>

The metadata information for the available quick actions of the parent object.

Example

```
// Called for Account entity.
List<QuickAction.DescribeAvailableQuickActionResult> result1 =
    QuickAction.DescribeAvailableQuickActions('Account');

// Called at global level, not entity level.
List<QuickAction.DescribeAvailableQuickActionResult> result2 =
    QuickAction.DescribeAvailableQuickActions('Global');
```

describeQuickActions(sObjectNames)

Returns the metadata information for the provided quick actions.

Signature

```
public static List<QuickAction.DescribeQuickActionResult>
describeQuickActions(List<String> sObjectNames)
```

Parameters

sObjectNames
Type: List<String>

The names of the quick actions. The quick action name can contain the entity name if it is at the entity level ('Account.QuickCreateContact'), or 'Global' if used for the action at the global level ('Global.CreateNewContact').

Return Value

Type: List<QuickAction.DescribeQuickActionResult>

The metadata information for the provided quick actions.

Example

```
// First 3 parameter values are for actions at the entity level.
// Last parameter is for an action at the global level.
List<QuickAction.DescribeQuickActionResult> result =
    QuickAction.DescribeQuickActions(new List<String> {
        'Account.QuickCreateContact', 'Opportunity.Update1',
        'Contact.Create1', 'Global.CreateNewContact' });
```

performQuickAction(quickActionRequest)

Performs the quick action specified in the quick action request and returns the action result.

Signature

```
public static QuickAction.QuickActionResult
performQuickAction(QuickAction.QuickActionRequest quickActionRequest)
```

Parameters

quickActionRequest
Type: QuickAction.QuickActionRequest

Return Value

Type: QuickAction.QuickActionResult

performQuickAction(quickActionRequest, allOrNothing)

Performs the quick action specified in the quick action request with the option for partial success, and returns the result.

Signature

```
public static QuickAction.QuickActionResult
performQuickAction(QuickAction.QuickActionRequest quickActionRequest, Boolean
allOrNothing)
```

Parameters

quickActionRequest
Type: QuickAction.QuickActionRequest

allOrNothing
Type: Boolean

Specifies whether this operation allows partial success. If you specify `false` for this argument and a record fails, the remainder of the DML operation can still succeed. This method returns a result object that can be used to verify which records succeeded, which failed, and why.

Return Value

Type: QuickAction.QuickActionResult

performQuickActions(quickActionRequests)

Performs the quick actions specified in the quick action request list and returns action results.

Signature

```
public static List<QuickAction.QuickActionResult>
performQuickActions(List<QuickAction.QuickActionRequest> quickActionRequests)
```

Parameters

quickActionRequests
Type: List<QuickAction.QuickActionRequest>

Return Value

Type: List<QuickAction.QuickActionResult>

performQuickActions(quickActionRequests, allOrNothing)

Performs the quick actions specified in the quick action request list with the option for partial success, and returns action results.

Signature

```
public static List<QuickAction.QuickActionResult>
performQuickActions(List<QuickAction.QuickActionRequest> quickActionRequests, Boolean
allOrNothing)
```

Parameters

quickActionRequests
Type: List<QuickAction.QuickActionRequest>

allOrNothing
Type: Boolean

Specifies whether this operation allows partial success. If you specify `false` for this argument and a record fails, the remainder of the DML operation can still succeed. This method returns a result object that can be used to verify which records succeeded, which failed, and why.

Return Value

Type: List<QuickAction.QuickActionResult>

Quiddity Enum

Specifies a Quiddity value used by the methods in the System.Request class

Enum Values

The following are the values of the `System.Quiddity` enum.

Value	Description
`ANONYMOUS`	Execution event is an anonymous Apex block.
`AURA`	Execution event is an Aura component.
`BATCH_ACS`	Execution event is an API Query Cursor driven batch Apex.
`BATCH_APEX`	Execution event is a batch Apex job.
`BATCH_CHUNK_PARALLEL`	Not used in API version 63.0 and later.
`BATCH_CHUNK_SERIAL`	Execution event is chunks of a batch Apex job running in serial.
`BULK_API`	Execution event is a bulk API request.
`COMMERCE_INTEGRATION`	Execution event is an Apex integration for B2B Commerce.
`DISCOVERABLE_LOGIN`	Execution event is Login Discoverable login page used by external users to log in to an Experience Cloud site.
`EXTERNAL_SERVICE_CALLBACK`	Execution event is an External Services asynchronous callback function.
`FUNCTION_CALLBACK`	Execution event is a callback function.
`FUTURE`	Execution event is a future method.
`INBOUND_EMAIL_SERVICE`	Execution event is an Apex inbound email service.
`INVOCABLE_ACTION`	Execution event is an invocable action.
`PLATFORM_EVENT_PUBLISH_CALLBACK`	Execution event is an Apex publish callback for platform events.

Value	Description
POST_INSTALL_SCRIPT	Execution event is a managed package install or upgrade.
QUEUEABLE	Execution event is a queueable Apex operation.
QUICK_ACTION	Execution event is a quick action.
REMOTE_ACTION	Execution event is a remote action.
REST	Execution event is an Apex RESTful Web service.
RUNTEST_ASYNC	Execution event is Apex tests running asynchronously.
RUNTEST_DEPLOY	Execution event is Apex tests run during deployment.
RUNTEST_SYNC	Execution event is Apex tests running synchronously.
SCHEDULED	Execution event is a scheduled Apex job.
SOAP	Execution event is an Apex SOAP Web service.
SYNCHRONOUS	Execution event is a synchronous Apex operation.
TRANSACTION_FINALIZER_QUEUEABLE	Execution event is a queueable job with transaction finalizers attached.
VF	Execution event is triggered by a Visualforce page.

RemoteObjectController

Use `RemoteObjectController` to access the standard Visualforce Remote Objects operations in your Remote Objects override methods.

Namespace

System

Usage

`RemoteObjectController` is supported only for use within Remote Objects methods. See *Overriding Default Remote Objects Operations* in the *Visualforce Developer's Guide* for examples of how to use `RemoteObjectController` with your Visualforce pages.

RemoteObjectController Methods

The following are methods for `RemoteObjectController`. All methods are static.

IN THIS SECTION:

create(type, fields)
Create a record in the database.

del(type, recordIds)
Delete records from the database.

retrieve(type, fields, criteria)
Retrieve records from the database.

update(type, recordIds, fields)
Update records in the database.

create(type, fields)

Create a record in the database.

Signature

```
public static Map<String,Object> create(String type, Map<String,Object> fields)
```

Parameters

type
Type: String

The sObject type on which create is being called.

fields
Type: Map<String,Object>

The fields and values to set on the new record.

Return Value

Type: Map<String,Object>

The return value is a map that represents the result of the Remote Objects operation. What is returned depends on the results of the call.

Success
A map that contains a single element with the ID of the record created. For example, `{ id: 'recordId' }`.

Failure
A map that contains a single element with the error message for the overall operation. For example, `{ error: 'errorMessage' }`.

del(type, recordIds)

Delete records from the database.

Signature

```
public static Map<String,Object> del(String type, List<String> recordIds)
```

Parameters

type
Type: String

The sObject type on which delete is being called.

recordIds
Type: List<String>

The IDs of the records to be deleted.

Return Value

Type: Map<String,Object>

The return value is a map that represents the result of the Remote Objects operation. What is returned depends on how the method was called and the results of the call.

Single Delete—Success
A map that contains a single element with the ID of the record that was deleted. For example, `{ id: 'recordId' }`.

Batch Delete—Success
A map that contains a single element, an array of Map<String,Object> elements. Each element contains the ID of a record that was deleted and an array of errors, if there were any, for that record's individual delete. For example, `{ results: [ { id: 'recordId', errors: ['errorMessage', ...]}, ...] }`.

Single and Batch Delete—Failure
A map that contains a single element with the error message for the overall operation. For example, `{ error: 'errorMessage' }`.

retrieve(type, fields, criteria)

Retrieve records from the database.

Signature

```
public static Map<String,Object> retrieve(String type, List<String> fields,
Map<String,Object> criteria)
```

Parameters

type
Type: String

The sObject type on which retrieve is being called.

fields
Type: List<String>

The fields to retrieve for each record.

criteria
Type: Map<String,Object>

The criteria to use when performing the query.

Return Value

Type: Map<String,Object>

The return value is a map that represents the result of the Remote Objects operation. What is returned depends on the results of the call.

Success
A map that contains the following elements.

- `records`: An array of records that match the query conditions.
- `type`: A string that indicates the type of the sObject that was retrieved.
- `size`: The number of records in the response.

Failure
A map that contains a single element with the error message for the overall operation. For example, `{ error: 'errorMessage' }`.

`update(type, recordIds, fields)`

Update records in the database.

Signature

```
public static Map<String,Object> update(String type, List<String> recordIds,
Map<String,Object> fields)
```

Parameters

type
Type: String

The sObject type on which update is being called.

recordIds
Type: List<String>

The IDs of the records to be updated.

fields
Type: Map<String,Object>

The fields to update, and the value to update each field with.

Return Value

Type: Map<String,Object>

The return value is a map that represents the result of the Remote Objects operation. What is returned depends on how the method was called and the results of the call.

Single Update—Success
A map that contains a single element with the ID of the record that was updated. For example, `{ id: 'recordId' }`.

Batch Update—Success
A map that contains a single element, an array of Map<String,Object> elements. Each element contains the ID of the record updated and an array of errors, if there were any, for that record's individual update. For example, `{ results: [ { id: 'recordId', errors: ['errorMessage', ...]}, ...] }`.

Single and Batch Update—Failure
A map that contains a single element with the error message for the overall operation. For example, `{ error: 'errorMessage' }`.

Request Class

Contains methods to obtain the request ID and Quiddity value of the current Salesforce request.

Namespace

System

Usage

Use the Request class to detect the current Apex context at runtime. The methods in the Request class obtain a unique request ID and the Quiddity value that represent the current Apex execution type. These values can also be used to correlate with debug and event logs:

- The request ID is universally unique and present in the debug logs that are triggered by the request.
- The request ID and Quiddity values are the same as in the event log files of the Apex Execution event type used in Event Monitoring.

Example

This example code shows how to obtain current Apex code context by retrieving the request ID and Quiddity value of the current request.

```
//Get info about the current request
Request reqInfo = Request.getCurrent();

//Get the identifier for this request, which is universally unique
//Same as REQUEST_ID in event monitoring
String currentRequestId = reqInfo.getRequestId();

//Enum representing how Apex is running. e.g. BULK_API vs LIGHTNING
Quiddity currentType = reqInfo.getQuiddity();
//Use this with a switch statement,
//instead of checking System.isFuture() || System.isQueueable() || ...
```

IN THIS SECTION:

Request Methods

Request Methods

The following are methods for `Request`.

IN THIS SECTION:

getCurrent()
Returns the current Request object that contains the request ID and Quiddity value.

getQuiddity()
Returns the Quiddity value of the current Request object.

getRequestId()
Returns the request ID of the current Request object.

`getCurrent()`

Returns the current Request object that contains the request ID and Quiddity value.

Signature

```
public static System.Request getCurrent()
```

Return Value

Type: System.Request

`getQuiddity()`

Returns the Quiddity value of the current Request object.

Signature

```
public System.Quiddity getQuiddity()
```

Return Value

Type: System.Quiddity

Uses the values from the Quiddity enum. This value identifies the type of execution event associated with the current request.

`getRequestId()`

Returns the request ID of the current Request object.

Signature

```
public String getRequestId()
```

Return Value

Type: String

ResetPasswordResult Class

Represents the result of a password reset.

Namespace

System

ResetPasswordResult Methods

The following are instance methods for `ResetPasswordResult`.

IN THIS SECTION:

getPassword()
Returns the password generated by the `System.resetPassword` method call.

getPassword()

Returns the password generated by the `System.resetPassword` method call.

Signature

```
public String getPassword()
```

Return Value

Type: String

RestContext Class

Contains the `RestRequest` and `RestResponse` objects.

Namespace

System

Usage

Use the `System.RestContext` class to access the `RestRequest` and `RestResponse` objects in your Apex REST methods.

Sample

The following example shows how to use `RestContext` to access the `RestRequest` and `RestResponse` objects in an Apex REST method.

```
@RestResource(urlMapping='/MyRestContextExample/*')
global with sharing class MyRestContextExample {

    @HttpGet
    global static Account doGet() {
        RestRequest req = RestContext.request;
        RestResponse res = RestContext.response;
        String accountId = req.requestURI.substring(req.requestURI.lastIndexOf('/')+1);
        Account result = [SELECT Id, Name, Phone, Website FROM Account WHERE Id =
:accountId];
        return result;
    }

}
```

RestContext Properties

The following are properties for `RestContext`.

IN THIS SECTION:

request
Returns the `RestRequest` for your Apex REST method.

response
Returns the `RestResponse` for your Apex REST method.

request

Returns the `RestRequest` for your Apex REST method.

Signature

```
public RestRequest request {get; set;}
```

Property Value

Type: System.RestRequest

response

Returns the `RestResponse` for your Apex REST method.

Signature

```
public RestResponse response {get; set;}
```

Property Value

Type: System.RestResponse

RestRequest Class

Use the `System.RestRequest` class to access and pass request data in a RESTful Apex method.

Namespace

System

Usage

An Apex RESTful Web service method is defined using one of the REST annotations. For more information about Apex RESTful Web service, see Exposing Apex Classes as REST Web Services.

Example: An Apex Class with REST Annotated Methods

The following example shows you how to implement the Apex REST API in Apex. This class exposes three methods that each handle a different HTTP request: GET, DELETE, and POST. You can call these annotated methods from a client by issuing HTTP requests.

```
@RestResource(urlMapping='/Account/*')
global with sharing class MyRestResource {

    @HttpDelete
    global static void doDelete() {
        RestRequest req = RestContext.request;
        RestResponse res = RestContext.response;
        String accountId = req.requestURI.substring(req.requestURI.lastIndexOf('/')+1);
        Account account = [SELECT Id FROM Account WHERE Id = :accountId];
        delete account;
    }

    @HttpGet
    global static Account doGet() {
        RestRequest req = RestContext.request;
        RestResponse res = RestContext.response;
        String accountId = req.requestURI.substring(req.requestURI.lastIndexOf('/')+1);
        Account result = [SELECT Id, Name, Phone, Website FROM Account WHERE Id =
:accountId];
        return result;
    }

  @HttpPost
    global static String doPost(String name,
        String phone, String website) {
        Account account = new Account();
        account.Name = name;
        account.phone = phone;
        account.website = website;
        insert account;
        return account.Id;
    }
}
```

IN THIS SECTION:

RestRequest Constructors

RestRequest Properties

RestRequest Methods

RestRequest Constructors

The following are constructors for `RestRequest`.

IN THIS SECTION:

RestRequest()
Creates a new instance of the `System.RestRequest` class.

`RestRequest()`

Creates a new instance of the `System.RestRequest` class.

Signature

```
public RestRequest()
```

RestRequest Properties

The following are properties for `RestRequest`.

Note: While the `RestRequest` List and Map properties are read-only, their contents are read-write. You can modify them by calling the collection methods directly or you can use of the associated `RestRequest` methods shown in the previous table.

IN THIS SECTION:

headers
Returns the headers that are received by the request.

httpMethod
Returns one of the supported HTTP request methods.

params
Returns the parameters that are received by the request.

remoteAddress
Returns the IP address of the client making the request.

requestBody
Returns or sets the body of the request.

requestURI
Returns or sets everything after the host in the HTTP request string.

resourcePath
Returns the REST resource path for the request.

`headers`

Returns the headers that are received by the request.

Signature

```
public Map<String, String> headers {get; set;}
```

Property Value

Type: Map<String, String>

`httpMethod`

Returns one of the supported HTTP request methods.

Signature

```
public String httpMethod {get; set;}
```

Property Value

Type: String

Possible values returned:

- DELETE
- GET
- HEAD
- PATCH
- POST
- PUT

`params`

Returns the parameters that are received by the request.

Signature

```
public Map <String, String> params {get; set;}
```

Property Value

Type: Map<String, String>

`remoteAddress`

Returns the IP address of the client making the request.

Signature

```
public String remoteAddress {get; set;}
```

Property Value

Type: String

`requestBody`

Returns or sets the body of the request.

Signature

```
public Blob requestBody {get; set;}
```

Property Value

Type: Blob

Usage

If the Apex method has no parameters, then Apex REST copies the HTTP request body into the `RestRequest.requestBody` property. If there are parameters, then Apex REST attempts to deserialize the data into those parameters and the data won't be deserialized into the `RestRequest.requestBody` property.

`requestURI`

Returns or sets everything after the host in the HTTP request string.

Signature

```
public String requestURI {get; set;}
```

Property Value

Type: String

Example

For example, if the request string is *`https://instance.salesforce.com/services/apexrest/Account/`* then the `requestURI` is */Account/*.

`resourcePath`

Returns the REST resource path for the request.

Signature

```
public String resourcePath {get; set;}
```

Property Value

Type: String

Example

For example, if the Apex REST class defines a `urlMapping` of `/MyResource/*`, the `resourcePath` property returns `/services/apexrest/MyResource/*`.

RestRequest Methods

The following are methods for `RestRequest`. All are instance methods.

Note: At runtime, you typically don't need to add a header or parameter to the `RestRequest` object because they are automatically deserialized into the corresponding properties. The following methods are intended for unit testing Apex REST classes. You can use them to add header or parameter values to the `RestRequest` object without having to recreate the REST method call.

IN THIS SECTION:

addHeader(name, value)
Adds a header to the request header map in an Apex test.

addParameter(name, value)
Adds a parameter to the request params map in an Apex test.

addHeader(name, value)

Adds a header to the request header map in an Apex test.

Signature

```
public Void addHeader(String name, String value)
```

Parameters

name
Type: String

value
Type: String

Return Value

Type: Void

Usage

This method is intended for unit testing of Apex REST classes.

The following headers aren't allowed:

- cookie
- set-cookie
- set-cookie2
- content-length
- authorization

If any of these headers are used, an Apex exception is thrown.

addParameter(name, value)

Adds a parameter to the request params map in an Apex test.

Signature

```
public Void addParameter(String name, String value)
```

Parameters

name
Type: String

value
Type: String

Return Value

Type: Void

Usage

This method is intended for unit testing of Apex REST classes.

RestResponse Class

Represents an object used to pass data from an Apex RESTful Web service method to an HTTP response.

Namespace

System

Usage

Use the `System.RestResponse` class to pass response data from an Apex RESTful web service method that is defined using one of the REST annotations.

IN THIS SECTION:

- RestResponse Constructors
- RestResponse Properties
- RestResponse Methods

RestResponse Constructors

The following are constructors for `RestResponse`.

IN THIS SECTION:

RestResponse()
Creates a new instance of the `System.RestResponse` class.

`RestResponse()`

Creates a new instance of the `System.RestResponse` class.

Signature

```
public RestResponse()
```

RestResponse Properties

The following are properties for `RestResponse`.

Note: While the `RestResponse` List and Map properties are read-only, their contents are read-write. You can modify them by calling the collection methods directly or you can use of the associated `RestResponse` methods shown in the previous table.

IN THIS SECTION:

responseBody
Returns or sets the body of the response.

headers
Returns the headers to be sent to the response.

statusCode
Returns or sets the response status code.

`responseBody`

Returns or sets the body of the response.

Signature

```
public Blob responseBody {get; set;}
```

Property Value

Type: Blob

Usage

The response is either the serialized form of the method return value or it's the value of the `responseBody` property based on the following rules:

- If the method returns void, then Apex REST returns the response in the `responseBody` property.
- If the method returns a value, then Apex REST serializes the return value as the response. If the return value contains fields with null value, those fields are not serialized in the response.

`headers`

Returns the headers to be sent to the response.

Signature

```
public Map<String, String> headers {get; set;}
```

Property Value

Type: Map<String, String>

statusCode

Returns or sets the response status code.

Signature

```
public Integer statuscode {get; set;}
```

Property Value

Type: Integer

Status Codes

The following are valid response status codes. The status code is returned by the `RestResponse.statusCode` property.

Note: If you set the `RestResponse.statusCode` property to a value that's not listed in the table, then an HTTP status of 500 is returned with the error message "Invalid status code for HTTP response: nnn" where nnn is the invalid status code value.

Status Code	Description
200	OK
201	CREATED
202	ACCEPTED
204	NO_CONTENT
206	PARTIAL_CONTENT
300	MULTIPLE_CHOICES
301	MOVED_PERMANENTLY
302	FOUND
304	NOT_MODIFIED
400	BAD_REQUEST
401	UNAUTHORIZED
403	FORBIDDEN
404	NOT_FOUND
405	METHOD_NOT_ALLOWED

Status Code	Description
406	NOT_ACCEPTABLE
409	CONFLICT
410	GONE
412	PRECONDITION_FAILED
413	REQUEST_ENTITY_TOO_LARGE
414	REQUEST_URI_TOO_LARGE
415	UNSUPPORTED_MEDIA_TYPE
417	EXPECTATION_FAILED
500	INTERNAL_SERVER_ERROR
503	SERVER_UNAVAILABLE

RestResponse Methods

The following are instance methods for `RestResponse`.

Note: At runtime, you typically don't need to add a header to the `RestResponse` object because it's automatically deserialized into the corresponding properties. The following methods are intended for unit testing Apex REST classes. You can use them to add header or parameter values to the `RestRequest` object without having to recreate the REST method call.

IN THIS SECTION:

addHeader(name, value)
Adds a header to the response header map.

`addHeader(name, value)`

Adds a header to the response header map.

Signature

```
public Void addHeader(String name, String value)
```

Parameters

`name`
Type: String

`value`
Type: String

Return Value

Type: Void

Usage

The following headers aren't allowed:

- cookie
- set-cookie
- set-cookie2
- content-length
- authorization
- Header names that aren't RFC 7230 compliant

If any of these headers are used, an Apex exception is thrown.

SandboxPostCopy Interface

To make your sandbox environment business ready, automate data manipulation or business logic tasks. Extend this interface and add methods to perform post-copy tasks, then specify the class during sandbox creation.

Namespace

System

Usage

Create an Apex class that implements this interface. Specify your class during sandbox creation. After your sandbox is created, the `runApexClass(context)` method in your class runs using the automated process user's permissions.

Important: The SandboxPostCopy Apex class is executed at the end of the sandbox copy using a special Automated Process user that isn't visible within the org. This user doesn't have access to all object and features; therefore, the Apex script cannot access all objects and features. If the script fails, run the script after sandbox activation as a user with appropriate permissions.

IN THIS SECTION:

SandboxPostCopy Methods

SandboxPostCopy Example Implementation

These examples show a simple implementation of the SandboxPostCopy interface and a test for that implementation. To test your SandboxPostCopy implementation, use the `System.Test.testSandboxPostCopyScript()` method.

SEE ALSO:

Tooling API: SandboxInfo

Tooling API: SandboxProcess

SandboxPostCopy Methods

The following method is for `SandboxPostCopy`.

IN THIS SECTION:

runApexClass(context)
Executes actions in a new sandbox to prepare it for use. For example, add logic to this method to create users, run sanitizing code on records, and perform other setup tasks.

`runApexClass(context)`

Executes actions in a new sandbox to prepare it for use. For example, add logic to this method to create users, run sanitizing code on records, and perform other setup tasks.

Signature

```
public void runApexClass(System.SandboxContext context)
```

Parameters

context
Type: System.SandboxContext

The org ID, sandbox ID, and sandbox name for your sandbox. To work with these values, reference `context.organizationId()`, `context.sandboxId()`, and `context.sandboxName()` in your code.

Return Value

Type: void

SandboxPostCopy Example Implementation

These examples show a simple implementation of the SandboxPostCopy interface and a test for that implementation. To test your SandboxPostCopy implementation, use the `System.Test.testSandboxPostCopyScript()` method.

Important: The SandboxPostCopy Apex class is executed at the end of the sandbox copy using a special Automated Process user that isn't visible within the org. This user doesn't have access to all objects and features; therefore, the Apex script can't access all objects and features. If the script fails, run the script after sandbox activation as a user with appropriate permissions.

This example implements the `System.SandboxPostCopy` interface.

```
global class PrepareMySandbox implements SandboxPostCopy {

    global PrepareMySandbox() {
        // Implementations of SandboxPostCopy must have a no-arg constructor.
        // This constructor is used during the sandbox copy process.
        // You can also implement constructors with arguments, but be aware that
        // they won't be used by the sandbox copy process (unless as part of the
        // no-arg constructor).
        this(some_args);
    }

    global PrepareMySandbox(String some_args) {
        // Logic for constructor.
    }
```

```
    global void runApexClass(SandboxContext context) {
        System.debug('Org ID: ' + context.organizationId());
        System.debug('Sandbox ID: ' + context.sandboxId());
        System.debug('Sandbox Name: ' + context.sandboxName());

        // Insert logic here to prepare the sandbox for use.
    }
}
```

The following example tests the implementation using the `System.Test.testSandboxPostCopyScript()` method. This method takes four parameters: a reference to a class that implements the SandboxPostCopy interface, and the three fields on the context object that you pass to the `runApexClass(context)` method. An overload on the method takes an optional Boolean parameter to indicate if the test must be performed as the Automated Process user.

```
@isTest
class PrepareMySandboxTest {

    @isTest
    static void testMySandboxPrep() {
       // Insert logic here to create records of the objects that the class you're testing

        // manipulates.

        Test.startTest();

        // Replace '00D000000000000' with your sandboxId and
        // execute test script with RunAsAutoProcUser set to true.

        Test.testSandboxPostCopyScript(
            new PrepareMySandbox(), UserInfo.getOrganizationId(),
                '00D000000000000', UserInfo.getOrganizationName(), true);

        Test.stopTest();

        // Insert assert statements here to check that the records you created above have

        // the values you expect.
    }
}
```

For more information on testing, see Testing Apex.

Schedulable Interface

The class that implements this interface can be scheduled to run at different intervals.

Namespace

System

SEE ALSO:

Apex Developer Guide: Scheduler

Schedulable Methods

The following are methods for `Schedulable`.

IN THIS SECTION:

execute(context)
Executes the scheduled Apex job.

`execute(context)`

Executes the scheduled Apex job.

Signature

```
public Void execute(SchedulableContext context)
```

Parameters

`context`
Type: System.SchedulableContext

Contains the job ID.

Return Value

Type: Void

SchedulableContext Interface

Represents the parameter type of a method in a class that implements the `Schedulable` interface and contains the scheduled job ID. This interface is implemented internally by Apex.

Namespace

System

SEE ALSO:

Schedulable Interface

SchedulableContext Methods

The following are methods for `SchedulableContext`.

IN THIS SECTION:

getTriggerId()
Returns the ID of the CronTrigger scheduled job.

getTriggerId()

Returns the ID of the CronTrigger scheduled job.

Signature

```
public Id getTriggerId()
```

Return Value

Type: ID

Schema Class

Contains methods for obtaining schema describe information.

Namespace

System

Schema Methods

The following are methods for `Schema`. All methods are static.

IN THIS SECTION:

getGlobalDescribe()
Returns a map of all sObject names (keys) to sObject tokens (values) for the standard and custom objects defined in your organization.

describeDataCategoryGroups(sObjectNames)
Returns a list of the category groups associated with the specified objects.

describeSObjects(sObjectTypes)
Describes metadata (field list and object properties) for the specified sObject or array of sObjects.

describeSObjects(SObjectTypes, SObjectDescribeOptions)
Describes metadata such as field list and object properties for the specified list of SObjects. The default describe option for this method is SObjectDescribeOptions.DEFERRED, which indicates lazy initialization of describe attributes on first use.

describeTabs()
Returns information about the standard and custom apps available to the running user.

describeDataCategoryGroupStructures(pairs,topCategoriesOnly)
Returns available category groups along with their data category structure for objects specified in the request.

getGlobalDescribe()

Returns a map of all sObject names (keys) to sObject tokens (values) for the standard and custom objects defined in your organization.

Signature

```
public static Map<String, Schema.SObjectType> getGlobalDescribe()
```

Return Value

Type: Map<String, Schema.SObjectType>

Usage

For more information on accessing SObjects, see Accessing All sObjects.

Example

```
Map<String, Schema.SObjectType> gd =
Schema.getGlobalDescribe();
```

describeDataCategoryGroups(sObjectNames)

Returns a list of the category groups associated with the specified objects.

Signature

```
public static List<Schema.DescribeDataCategoryGroupResult>
describeDataCategoryGroups(List<String> sObjectNames)
```

Parameters

sObjectNames
Type: List<String>

Return Value

Type: List<Schema.DescribeDataCategoryGroupResult>

Usage

You can specify one of the following sObject names:

- KnowledgeArticleVersion—to retrieve category groups associated with article types.
- Question—to retrieve category groups associated with questions.

For more information and code examples using describeDataCategoryGroups, see Accessing All Data Categories Associated with an sObject.

For additional information about articles and questions, see "Work with Articles and Translations" in the Salesforce online help.

describeSObjects(sObjectTypes)

Describes metadata (field list and object properties) for the specified sObject or array of sObjects.

Signature

```
public static List<Schema.DescribeSObjectResult> describeSObjects(List<String>
sObjectTypes)
```

Parameters

sObjectTypes
Type: List<String>

The *sObjectTypes* argument is a list of sObject type names you want to describe.

Return Value

Type: List<Schema.DescribeSObjectResult>

Usage

This method is similar to the `getDescribe` method on the `Schema.sObjectType` token. Unlike the `getDescribe` method, this method allows you to specify the sObject type dynamically and describe more than one sObject at a time.

You can first call `getGlobalDescribe` to retrieve a list of all objects for your organization, then iterate through the list and use `describeSObjects` to obtain metadata about individual objects.

Example

```
Schema.DescribeSObjectResult[] descResult = Schema.describeSObjects(
                                                                new
String[]{'Account','Contact'});
```

describeSObjects(SObjectTypes, SObjectDescribeOptions)

Describes metadata such as field list and object properties for the specified list of SObjects. The default describe option for this method is SObjectDescribeOptions.DEFERRED, which indicates lazy initialization of describe attributes on first use.

Signature

```
public static List<Schema.DescribeSObjectResult> describeSObjects(List<String>
SObjectTypes, Object SObjectDescribeOptions)
```

Parameters

SObjectTypes
Type: List<String>

The list of SObject types to describe.

SObjectDescribeOptions
Type: Object

The effective describe option used for the SObject.

Return Value

Type: List<Schema.DescribeSObjectResult>

describeTabs()

Returns information about the standard and custom apps available to the running user.

Signature

```
public static List<Schema.DescribeTabSetResult> describeTabs()
```

Return Value

Type: List<Schema.DescribeTabSetResult>

Usage

An app is a group of tabs that works as a unit to provide application functionality. For example, two of the standard Salesforce apps are "Sales" and "Service."

The `describeTabs` method returns the minimum required metadata that can be used to render apps in another user interface. Typically, this call is used by partner applications to render Salesforce data in another user interface, such as in a mobile or connected app.

In the Salesforce user interface, users have access to standard apps (and can also have access to custom apps) as listed in the Salesforce app menu at the top of the page. Selecting an app name in the menu allows the user to switch between the listed apps at any time.

Note: The "All Tabs" tab isn't included in the list of described tabs.

Example

This example shows how to call the `describeTabs` method.

```
Schema.DescribeTabSetResult[] tabSetDesc = Schema.describeTabs();
```

This longer example shows how to obtain describe metadata information for the Sales app. For each tab, the example gets describe information, such as the icon URL, whether the tab is custom or not, and colors. The describe information is written to the debug output.

```
// Get tab set describes for each app
List<Schema.DescribeTabSetResult> tabSetDesc = Schema.describeTabs();

// Iterate through each tab set describe for each app and display the info
for(DescribeTabSetResult tsr : tabSetDesc) {
    String appLabel = tsr.getLabel();
    System.debug('Label: ' + appLabel);
    System.debug('Logo URL: ' + tsr.getLogoUrl());
    System.debug('isSelected: ' + tsr.isSelected());
    String ns = tsr.getNamespace();
    if (ns == '') {
        System.debug('The ' + appLabel + ' app has no namespace defined.');
    }
    else {
        System.debug('Namespace: ' + ns);
    }

    // Display tab info for the Sales app
    if (appLabel == 'Sales') {
        List<Schema.DescribeTabResult> tabDesc = tsr.getTabs();
        System.debug('-- Tab information for the Sales app --');
        for(Schema.DescribeTabResult tr : tabDesc) {
            System.debug('getLabel: ' + tr.getLabel());
            System.debug('getColors: ' + tr.getColors());
```

```
            System.debug('getIconUrl: ' + tr.getIconUrl());
            System.debug('getIcons: ' + tr.getIcons());
            System.debug('getMiniIconUrl: ' + tr.getMiniIconUrl());
            System.debug('getSobjectName: ' + tr.getSobjectName());
            System.debug('getUrl: ' + tr.getUrl());
            System.debug('isCustom: ' + tr.isCustom());
        }
    }
}

// Example debug statement output
// DEBUG|Label: Sales
// DEBUG|Logo URL:
https://MyDomainName.my.salesforce.com/img/seasonLogos/2014_winter_aloha.png
// DEBUG|isSelected: true
// DEBUG|The Sales app has no namespace defined.// DEBUG|-- Tab information for the Sales
 app --
// (This is an example debug output for the Accounts tab.)
// DEBUG|getLabel: Accounts
// DEBUG|getColors:
(Schema.DescribeColorResult[getColor=236FBD;getContext=primary;getTheme=theme4;],
//       Schema.DescribeColorResult[getColor=236FBD;getContext=primary;getTheme=theme3;],

//       Schema.DescribeColorResult[getColor=236FBD;getContext=primary;getTheme=theme2;])
// DEBUG|getIconUrl: https://MyDomainName.my.salesforce.com/img/icon/accounts32.png
// DEBUG|getIcons:
(Schema.DescribeIconResult[getContentType=image/png;getHeight=32;getTheme=theme3;
//
getUrl=https://MyDomainName.my.salesforce.com/img/icon/accounts32.png;getWidth=32;],
//       Schema.DescribeIconResult[getContentType=image/png;getHeight=16;getTheme=theme3;
//
getUrl=https://MyDomainName.my.salesforce.com/img/icon/accounts16.png;getWidth=16;])
// DEBUG|getMiniIconUrl: https://MyDomainName.my.salesforce.com/img/icon/accounts16.png
// DEBUG|getSobjectName: Account
// DEBUG|getUrl: https://MyDomainName.my.salesforce.com/001/o
// DEBUG|isCustom: false
```

describeDataCategoryGroupStructures(pairs,topCategoriesOnly)

Returns available category groups along with their data category structure for objects specified in the request.

Signature

```
public static List<Schema.DescribeDataCategoryGroupStructureResult> describeDataCategory
GroupStructures(List<Schema.DataCategoryGroupSobjectTypePair> pairs,Boolean
topCategoriesOnly)
```

Parameters

pairs
Type: List<Schema.DataCategoryGroupSobjectTypePair>

The *pairs* argument is one or more category groups and objects to query Schema.DataCategoryGroupSobjectTypePairs. Visible data categories are retrieved for the specified object. For more information on data category group visibility, see "Data Category Visibility" in Salesforce Help.

topCategoriesOnly

Type: Boolean

Use `true` to return only the top visible category and `false` to return all the visible categories, depending on the user's data category group visibility settings. For more information on data category group visibility, see Data Category Visibility in Salesforce Help.

Return Value

Type: List<Schema.DescribeDataCategoryGroupStructureResult>

Search Class

Use the methods of the Search class to perform dynamic SOSL queries.

Namespace

System

Search Methods

The following are static methods for `Search`.

IN THIS SECTION:

find(searchQuery)

Performs a dynamic SOSL query that can include the SOSL `WITH SNIPPET` clause. Snippets provide more context for users in Salesforce Knowledge article search results.

find(searchQuery, accessLevel)

Performs a dynamic SOSL query that can include the SOSL `WITH SNIPPET` clause. Snippets provide more context for users in Salesforce Knowledge article search results.

query(query)

Performs a dynamic SOSL query.

query(query, accessLevel)

Performs a dynamic SOSL query.

suggest(searchQuery, sObjectType, suggestions)

Returns a list of records or Salesforce Knowledge articles whose names or titles match the user's search query string. Use this method to provide users with shortcuts to navigate to relevant records or articles before they perform a search.

suggest(searchQuery, sObjectType, suggestions, accessLevel)

Returns a list of records or Salesforce Knowledge articles whose names or titles match the user's search query string. Use this method to provide users with shortcuts to navigate to relevant records or articles before they perform a search.

`find(searchQuery)`

Performs a dynamic SOSL query that can include the SOSL `WITH SNIPPET` clause. Snippets provide more context for users in Salesforce Knowledge article search results.

Signature

```
public static Search.SearchResults find(String searchQuery)
```

Parameters

`searchQuery`
Type: String

A SOSL query string.

Return Value

Type: Search.SearchResults

Usage

Use this method wherever a static SOSL query can be used, such as in regular assignment statements and `for` loops.

See Use Dynamic SOSL to Return Snippets.

SEE ALSO:

get(sObjectType)

Apex Developer Guide: Dynamic SOSL

`find(searchQuery, accessLevel)`

Performs a dynamic SOSL query that can include the SOSL `WITH SNIPPET` clause. Snippets provide more context for users in Salesforce Knowledge article search results.

Signature

```
public static Search.SearchResults find(String searchQuery, System.AccessLevel
accessLevel)
```

Parameters

`searchQuery`
Type: String

A SOSL query string.

`accessLevel`
Type: System.AccessLevel

(Optional) The *`accessLevel`* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are

ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: Search.SearchResults

Usage

Use this method wherever a static SOSL query can be used, such as in regular assignment statements and `for` loops.

See Use Dynamic SOSL to Return Snippets.

query(query)

Performs a dynamic SOSL query.

Signature

```
public static sObject[sObject[]] query(String query)
```

Parameters

query

Type: String

A SOSL query string.

To create a SOSL query that includes the `WITH SNIPPET` clause, use the Search.find(String searchQuery) method instead.

Return Value

Type: sObject[sObject[]]

Usage

This method can be used wherever a static SOSL query can be used, such as in regular assignment statements and `for` loops.

For more information, see Dynamic SOSL.

query(query, accessLevel)

Performs a dynamic SOSL query.

Signature

```
public static List<List<SObject>> query(String query, System.AccessLevel accessLevel)
```

Parameters

query

Type: String

A SOSL query string.

To create a SOSL query that includes the `WITH SNIPPET` clause, use the Search.find(String searchQuery) method instead.

`accessLevel`
Type: System.AccessLevel

(Optional) The *`accessLevel`* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: sObject[sObject[]]

Usage

This method can be used wherever a static SOSL query can be used, such as in regular assignment statements and `for` loops.

For more information, see Dynamic SOSL.

`suggest(searchQuery, sObjectType, suggestions)`

Returns a list of records or Salesforce Knowledge articles whose names or titles match the user's search query string. Use this method to provide users with shortcuts to navigate to relevant records or articles before they perform a search.

Signature

```
public static Search.SuggestionResults suggest(String searchQuery, String sObjectType,
Search.SuggestionOption suggestions)
```

Parameters

`searchQuery`
Type: String

A SOSL query string.

`sObjectType`
Type: String

An sObject type.

`options`
Type: Search.SuggestionOption

This object contains options that change the suggestion results.

If the *`searchQuery`* returns KnowledgeArticleVersion objects, pass an *`options`* parameter with a Search.SuggestionOption object that contains a language KnowledgeSuggestionFilter and a publish status KnowledgeSuggestionFilter.

For suggestions for all other record types, the only supported option is a limit, which sets the maximum number of suggestions returned.

Return Value

Type: SuggestionResults

Usage

Use this method to return:

Suggestions for Salesforce Knowledge articles (KnowledgeArticleVersion)

Salesforce Knowledge must be enabled in your organization. The user must have the "View Articles" permission enabled.

The articles suggested include only the articles the user can access, based on the data categories and article types the user has permissions to view.

Suggestions for other record types

The records suggested include only the records the user can access.

This method returns a record if its name field starts with the text in the search string. This method automatically appends an asterisk wildcard (*) at the end of the search string. Records that contain the search string within a word aren't considered a match.

Records are suggested if the entire search string is found in the record name, in the same order as specified in the search string. For example, the text string `national u` is treated as `national u*` and returns "National Utility" and "National Urban Company" but not "National Company Utility" or "Urban National Company".

Note: If the user's search query contains quotation marks or wildcards, those symbols are automatically removed from the query string in the URI.

SEE ALSO:

Apex Developer Guide: Suggest Salesforce Knowledge Articles

`suggest(searchQuery, sObjectType, suggestions, accessLevel)`

Returns a list of records or Salesforce Knowledge articles whose names or titles match the user's search query string. Use this method to provide users with shortcuts to navigate to relevant records or articles before they perform a search.

Signature

```
public static Search.SuggestionResults suggest(String searchQuery, String sObjectType,
Search.SuggestionOption suggestions, System.AccessLevel accessLevel)
```

Parameters

`searchQuery`

Type: String

A SOSL query string.

`sObjectType`

Type: String

An sObject type.

`suggestions`

Type: Search.SuggestionOption

This object contains options that change the suggestion results.

If the *`searchQuery`* returns KnowledgeArticleVersion objects, pass an *`options`* parameter with a Search.SuggestionOption object that contains a language KnowledgeSuggestionFilter and a publish status KnowledgeSuggestionFilter.

For suggestions for all other record types, the only supported option is a limit, which sets the maximum number of suggestions returned.

accessLevel
Type: System.AccessLevel

(Optional) The *accessLevel* parameter specifies whether the method runs in system mode (`AccessLevel.SYSTEM_MODE`) or user mode (`AccessLevel.USER_MODE`). In system mode, the object and field-level permissions of the current user are ignored, and the record sharing rules are controlled by the class sharing keywords. In user mode, the object permissions, field-level security, and sharing rules of the current user are enforced. System mode is the default.

Return Value

Type: SuggestionResults

Usage

Use this method to return:

Suggestions for Salesforce Knowledge articles (KnowledgeArticleVersion)
Salesforce Knowledge must be enabled in your organization. The user must have the "View Articles" permission enabled.

The articles suggested include only the articles the user can access, based on the data categories and article types the user has permissions to view.

Suggestions for other record types
The records suggested include only the records the user can access.

This method returns a record if its name field starts with the text in the search string. This method automatically appends an asterisk wildcard (*) at the end of the search string. Records that contain the search string within a word aren't considered a match.

Records are suggested if the entire search string is found in the record name, in the same order as specified in the search string. For example, the text string *national u* is treated as *national u** and returns "National Utility" and "National Urban Company" but not "National Company Utility" or "Urban National Company".

Note: If the user's search query contains quotation marks or wildcards, those symbols are automatically removed from the query string in the URI.

Security Class

Contains methods to securely implement Apex applications.

Namespace

System

Usage

In the context of the current user's create, read, update, or upsert access permission, use the Security class methods to:

- Strip fields that aren't visible from query and subquery results
- Remove inaccessible fields before a DML operation without causing an exception
- Sanitize SObjects that have been deserialized from an untrusted source

IN THIS SECTION:

Security Methods

Security Methods

The following are methods for `Security`.

IN THIS SECTION:

stripInaccessible(accessCheckType, sourceRecords, enforceRootObjectCRUD)

Creates a list of sObjects from the source records, which are stripped of fields that fail the field-level security checks for the current user. The method also provides an option to enforce an object-level access check.

stripInaccessible(accessCheckType, sourceRecords)

Creates a list of sObjects from the source records, which are stripped of fields that fail the field-level security checks for the current user.

stripInaccessible(accessCheckType, sourceRecords, enforceRootObjectCRUD, permissionSetId)(Developer Preview)

Creates a list of sObjects from the source records, which are stripped of fields that fail field-level and object-level access checks. Apex enforces field-level security (FLS) and object permissions as per the specified permission set, in addition to the running user's permissions.

`stripInaccessible(accessCheckType, sourceRecords, enforceRootObjectCRUD)`

Creates a list of sObjects from the source records, which are stripped of fields that fail the field-level security checks for the current user. The method also provides an option to enforce an object-level access check.

Signature

```
public static System.SObjectAccessDecision stripInaccessible(System.AccessType
accessCheckType, List<SObject> sourceRecords, Boolean enforceRootObjectCRUD)
```

Parameters

`accessCheckType`
Type: System.AccessType

Uses values from the AccessType enum. This parameter determines the type of field-level access check to be performed. To check the current user's field-level access, use the Schema.DescribeFieldResult methods —`isCreatable()`, `isAccessible()`, or `isUpdatable()`.

`sourceRecords`
Type: List<SObject>

A list of sObjects to be checked for fields that aren't accessible in the context of the current user's operation.

`enforceRootObjectCRUD`
Type: Boolean

Indicates whether an object-level access check is performed. If this parameter is set to `true` and the access check fails, the method throws an exception. The default value of this optional parameter is `true`.

Return Value

Type: System.SObjectAccessDecision

Example

In this example, the user doesn't have permission to create the `Probability` field of an Opportunity.

```
List<Opportunity> opportunities = new List<Opportunity>{
    new Opportunity(Name='Opportunity1'),
    new Opportunity(Name='Opportunity2', Probability=95)
};

// Strip fields that are not creatable
SObjectAccessDecision decision = Security.stripInaccessible(
    AccessType.CREATABLE,
    opportunities);

// Print stripped records
for (SObject strippedOpportunity : decision.getRecords()) {
    System.debug(strippedOpportunity);
}

// Print modified indexes
System.debug(decision.getModifiedIndexes());

// Print removed fields
System.debug(decision.getRemovedFields());

//Lines from output log
//|DEBUG|Opportunity:{Name=Opportunity1}
//|DEBUG|Opportunity:{Name=Opportunity2}
//|DEBUG|{1}
//|DEBUG|{Opportunity={Probability}}
```

`stripInaccessible(accessCheckType, sourceRecords)`

Creates a list of sObjects from the source records, which are stripped of fields that fail the field-level security checks for the current user.

Signature

```
public static System.SObjectAccessDecision stripInaccessible(System.AccessType
accessCheckType, List<SObject> sourceRecords)
```

Parameters

`accessCheckType`

Type: System.AccessType

Uses values from the AccessType enum. This parameter determines the type of field-level access check to be performed. To check the current user's field-level access, use the Schema.DescribeFieldResult methods —`isCreatable()`, `isAccessible()`, or `isUpdatable()`.

sourceRecords
Type: List<SObject>

A list of sObjects to be checked for fields that aren't accessible in the context of the current user's operation.

Return Value

Type: System.SObjectAccessDecision

Example

In this example, the user doesn't have permission to read the `ActualCost` field of a Campaign.

```
List<Campaign> campaigns = new List<Campaign>{
    new Campaign(Name='Campaign1', BudgetedCost=1000, ActualCost=2000),
    new Campaign(Name='Campaign2', BudgetedCost=4000, ActualCost=1500)
};
insert campaigns;

// Strip fields that are not readable
SObjectAccessDecision decision = Security.stripInaccessible(
    AccessType.READABLE,
    [SELECT Name, BudgetedCost, ActualCost from Campaign]);

// Print stripped records
for (SObject strippedCampaign : decision.getRecords()) {
    System.debug(strippedCampaign); // Does not display ActualCost
}

// Print modified indexes
System.debug(decision.getModifiedIndexes());

// Print removed fields
System.debug(decision.getRemovedFields());

//Lines from output log
//|DEBUG|Campaign:{Name=Campaign1, BudgetedCost=1000, Id=701xx00000011nhAAA}
//|DEBUG|Campaign:{Name=Campaign2, BudgetedCost=4000, Id=701xx00000011niAAA}
//|DEBUG|{0, 1}
//|DEBUG|{Campaign={ActualCost}}
```

stripInaccessible(accessCheckType, sourceRecords, enforceRootObjectCRUD, permissionSetId) (Developer Preview)

Creates a list of sObjects from the source records, which are stripped of fields that fail field-level and object-level access checks. Apex enforces field-level security (FLS) and object permissions as per the specified permission set, in addition to the running user's permissions.

Note: Feature is available as a developer preview. Feature isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in a production environment. You can provide feedback and suggestions for the "Permission Sets with User Mode" feature in the Trailblazer Community.

This feature is available in scratch orgs where the `ApexUserModeWithPermset` feature is enabled. If the feature isn't enabled, Apex code with this feature can be compiled but not executed.

Signature

```
public static System.SObjectAccessDecision stripInaccessible(System.AccessType
accessCheckType, List<SObject> sourceRecords, Boolean enforceRootObjectCRUD, Id
permissionSetId)
```

Parameters

accessCheckType
Type: System.AccessType

Uses values from the AccessType enum. This parameter determines the type of field-level access check to be performed. To check the current user's field-level access, use the Schema.DescribeFieldResult methods —`isCreatable()`, `isAccessible()`, or `isUpdatable()`.

sourceRecords
Type: List<SObject>

A list of sObjects to be checked for fields that aren't accessible in the context of the current user's operation.

enforceRootObjectCRUD
Type: Boolean

Indicates whether an object-level access check is performed. If this parameter is set to `true` and the access check fails, the method throws an exception. The default value of this optional parameter is `true`.

permissionSetId
Type: Id

Permissions in the specified permission set are enforced in additon to the running user's permissions.

Return Value

Type: System.SObjectAccessDecision

SelectOption Class

A `SelectOption` object specifies one of the possible values for a Visualforce `selectCheckboxes`, `selectList`, or `selectRadio` component.

Namespace

System

`SelectOption` consists of a label that is displayed to the end user, and a value that is returned to the controller if the option is selected. A `SelectOption` can also be displayed in a disabled state, so that a user cannot select it as an option, but can still view it.

Instantiation

In a custom controller or controller extension, you can instantiate a SelectOption in one of the following ways:

- ```
  SelectOption option = new SelectOption(value, label, isDisabled);
  ```

  where *value* is the String that is returned to the controller if the option is selected by a user, *label* is the String that is displayed to the user as the option choice, and *isDisabled* is a Boolean that, if true, specifies that the user cannot select the option, but can still view it.

- ```
  SelectOption option = new SelectOption(value, label);
  ```

  where *value* is the String that is returned to the controller if the option is selected by a user, and *label* is the String that is displayed to the user as the option choice. Because a value for *isDisabled* is not specified, the user can both view and select the option.

Example

The following example shows how a list of SelectOptions objects can be used to provide possible values for a `selectCheckboxes` component on a Visualforce page. In the following custom controller, the `getItems` method defines and returns the list of possible SelectOption objects:

```
public class sampleCon {

  String[] countries = new String[]{};

  public PageReference test() {
    return null;
  }

  public List<SelectOption> getItems() {
    List<SelectOption> options = new List<SelectOption>();
    options.add(new SelectOption('US','US'));
    options.add(new SelectOption('CANADA','Canada'));
    options.add(new SelectOption('MEXICO','Mexico'));
    return options;
   }

  public String[] getCountries() {
     return countries;
  }

  public void setCountries(String[] countries) {
    this.countries = countries;
  }

}
```

In the following page markup, the `<apex:selectOptions>` tag uses the `getItems` method from the controller above to retrieve the list of possible values. Because `<apex:selectOptions>` is a child of the `<apex:selectCheckboxes>` tag, the options are displayed as checkboxes:

```
<apex:page controller="sampleCon">
 <apex:form>
   <apex:selectCheckboxes value="{!countries}">
     <apex:selectOptions value="{!items}"/>
   </apex:selectCheckboxes><br/>
   <apex:commandButton value="Test" action="{!test}" rerender="out" status="status"/>
```

```
  </apex:form>
  <apex:outputPanel id="out">
    <apex:actionstatus id="status" startText="testing...">
      <apex:facet name="stop">
        <apex:outputPanel>
          <p>You have selected:</p>
          <apex:dataList value="{!countries}" var="c">{!c}</apex:dataList>
        </apex:outputPanel>
      </apex:facet>
    </apex:actionstatus>
  </apex:outputPanel>
</apex:page>
```

IN THIS SECTION:

SelectOption Constructors

SelectOption Methods

SelectOption Constructors

The following are constructors for `SelectOption`.

IN THIS SECTION:

SelectOption(value, label)

Creates a new instance of the `SelectOption` class using the specified value and label.

SelectOption(value, label, isDisabled)

Creates a new instance of the `SelectOption` class using the specified value, label, and disabled setting.

`SelectOption(value, label)`

Creates a new instance of the `SelectOption` class using the specified value and label.

Signature

```
public SelectOption(String value, String label)
```

Parameters

value

Type: String

The string that is returned to the Visualforce controller if the option is selected by a user.

label

Type: String

The string that is displayed to the user as the option choice.

`SelectOption(value, label, isDisabled)`

Creates a new instance of the `SelectOption` class using the specified value, label, and disabled setting.

Signature

```
public SelectOption(String value, String label, Boolean isDisabled)
```

Parameters

value
Type: String

The string that is returned to the Visualforce controller if the option is selected by a user.

label
Type: String

The string that is displayed to the user as the option choice.

isDisabled
Type: Boolean

If set to true, the option can't be selected by the user but can still be viewed.

SelectOption Methods

The following are methods for `SelectOption`. All are instance methods.

IN THIS SECTION:

getDisabled()
Returns the current value of the SelectOption object's `isDisabled` attribute.

getEscapeItem()
Returns the current value of the SelectOption object's `itemEscaped` attribute.

getLabel()
Returns the option label that is displayed to the user.

getValue()
Returns the option value that is returned to the controller if a user selects the option.

setDisabled(isDisabled)
Sets the value of the SelectOption object's `isDisabled` attribute.

setEscapeItem(itemsEscaped)
Sets the value of the SelectOption object's `itemEscaped` attribute.

setLabel(label)
Sets the value of the option label that is displayed to the user.

setValue(value)
Sets the value of the option value that is returned to the controller if a user selects the option.

`getDisabled()`

Returns the current value of the SelectOption object's `isDisabled` attribute.

Signature

```
public Boolean getDisabled()
```

Return Value

Type: Boolean

Usage

If `isDisabled` is set to `true`, the user can view the option, but cannot select it. If `isDisabled` is set to `false`, the user can both view and select the option.

`getEscapeItem()`

Returns the current value of the SelectOption object's `itemEscaped` attribute.

Signature

```
public Boolean getEscapeItem()
```

Return Value

Type: Boolean

Usage

If `itemEscaped` is set to `true`, sensitive HTML and XML characters are escaped in the HTML output generated by this component. If `itemEscaped` is set to `false`, items are rendered as written.

`getLabel()`

Returns the option label that is displayed to the user.

Signature

```
public String getLabel()
```

Return Value

Type: String

`getValue()`

Returns the option value that is returned to the controller if a user selects the option.

Signature

```
public String getValue()
```

Return Value

Type: String

setDisabled(isDisabled)

Sets the value of the SelectOption object's `isDisabled` attribute.

Signature

```
public Void setDisabled(Boolean isDisabled)
```

Parameters

isDisabled
Type: Boolean

Return Value

Type: Void

Usage

If `isDisabled` is set to `true`, the user can view the option, but cannot select it. If `isDisabled` is set to `false`, the user can both view and select the option.

setEscapeItem(itemsEscaped)

Sets the value of the SelectOption object's `itemEscaped` attribute.

Signature

```
public Void setEscapeItem(Boolean itemsEscaped)
```

Parameters

itemsEscaped
Type: Boolean

Return Value

Type: Void

Usage

If `itemEscaped` is set to `true`, sensitive HTML and XML characters are escaped in the HTML output generated by this component. If `itemEscaped` is set to `false`, items are rendered as written.

setLabel(label)

Sets the value of the option label that is displayed to the user.

Signature

```
public Void setLabel(String label)
```

Parameters

label
Type: String

Return Value

Type: Void

setValue(value)

Sets the value of the option value that is returned to the controller if a user selects the option.

Signature

```
public Void setValue(String value)
```

Parameters

value
Type: String

Return Value

Type: Void

Set Class

Represents a collection of unique elements with no duplicate values.

Namespace

System

Usage

The Set methods work on a set, that is, an unordered collection of elements that was initialized using the `set` keyword. Set elements can be of any data type—primitive types, collections, sObjects, user-defined types, and built-in Apex types. Set methods are all instance methods, that is, they all operate on a particular instance of a Set. The following are the instance methods for sets.

Note:

- Uniqueness of set elements of user-defined types is determined by the `equals` and `hashCode` methods, which you provide in your classes. Uniqueness of all other non-primitive types is determined by comparing the objects' fields.
- If the set contains String elements, the elements are case-sensitive. Two set elements that differ only by case are considered distinct.

For more information on sets, see Sets.

IN THIS SECTION:

Set Constructors

Set Methods

Set Constructors

The following are constructors for `Set`.

IN THIS SECTION:

Set<T>()

Creates a new instance of the `Set` class. A set can hold elements of any data type T.

Set<T>(setToCopy)

Creates a new instance of the `Set` class by copying the elements of the specified set. T is the data type of the elements in both sets and can be any data type.

Set<T>(listToCopy)

Creates a new instance of the `Set` class by copying the list elements. T is the data type of the elements in the set and list and can be any data type.

`Set<T>()`

Creates a new instance of the `Set` class. A set can hold elements of any data type T.

Signature

```
public Set<T>()
```

Example

```
// Create a set of strings
Set<String> s1 = new Set<String>();
// Add two strings to it
s1.add('item1');
s1.add('item2');
```

`Set<T>(setToCopy)`

Creates a new instance of the `Set` class by copying the elements of the specified set. T is the data type of the elements in both sets and can be any data type.

Signature

```
public Set<T>(Set<T> setToCopy)
```

Parameters

setToCopy
Type: Set<T>

The set to initialize this set with.

Example

```
Set<String> s1 = new Set<String>();
s1.add('item1');
s1.add('item2');
Set<String> s2 = new Set<String>(s1);
// The set elements in s2 are copied from s1
System.debug(s2);
```

Set<T>(listToCopy)

Creates a new instance of the `Set` class by copying the list elements. T is the data type of the elements in the set and list and can be any data type.

Signature

```
public Set<T>(List<T> listToCopy)
```

Parameters

listToCopy
Type: Integer

The list to copy the elements of into this set.

Example

```
List<Integer> ls = new List<Integer>();
ls.add(1);
ls.add(2);
// Create a set based on a list
Set<Integer> s1 = new Set<Integer>(ls);
// Elements are copied from the list to this set
System.debug(s1);// DEBUG|{1, 2}
```

Set Methods

The following are methods for `Set`. All are instance methods.

IN THIS SECTION:

add(setElement)
Adds an element to the set if it is not already present.

addAll(fromList)
Adds all of the elements in the specified list to the set if they are not already present.

addAll(fromSet)

Adds all of the elements in the specified set to the set that calls the method if they are not already present.

clear()

Removes all of the elements from the set.

clone()

Makes a duplicate copy of the set.

contains(setElement)

Returns `true` if the set contains the specified element.

containsAll(listToCompare)

Returns `true` if the set contains all of the elements in the specified list. The list must be of the same type as the set that calls the method.

containsAll(setToCompare)

Returns `true` if the set contains all of the elements in the specified set. The specified set must be of the same type as the original set that calls the method.

equals(set2)

Compares this set with the specified set and returns `true` if both sets are equal; otherwise, returns `false`.

hashCode()

Returns the hashcode corresponding to this set and its contents.

isEmpty()

Returns `true` if the set has zero elements.

remove(setElement)

Removes the specified element from the set if it is present.

removeAll(listOfElementsToRemove)

Removes the elements in the specified list from the set if they are present.

removeAll(setOfElementsToRemove)

Removes the elements in the specified set from the original set if they are present.

retainAll(listOfElementsToRetain)

Retains only the elements in this set that are contained in the specified list.

retainAll(setOfElementsToRetain)

Retains only the elements in the original set that are contained in the specified set.

size()

Returns the number of elements in the set (its cardinality).

toString()

Returns the string representation of the set.

`add(setElement)`

Adds an element to the set if it is not already present.

Signature

```
public Boolean add(Object setElement)
```

Parameters

setElement
Type: Object

Return Value

Type: Boolean

Usage

This method returns true if the original set changed as a result of the call. For example:

```
Set<String> myString = new Set<String>{'a', 'b', 'c'};
Boolean result = myString.add('d');
System.assertEquals(true, result);
```

addAll(fromList)

Adds all of the elements in the specified list to the set if they are not already present.

Signature

```
public Boolean addAll(List<Object> fromList)
```

Parameters

fromList
Type: List

Return Value

Type: Boolean

Returns `true` if the original set changed as a result of the call.

Usage

This method results in the *union* of the list and the set. The list must be of the same type as the set that calls the method.

addAll(fromSet)

Adds all of the elements in the specified set to the set that calls the method if they are not already present.

Signature

```
public Boolean addAll(Set<Object> fromSet)
```

Parameters

fromSet
Type: Set<Object>

Return Value

Type: Boolean

This method returns `true` if the original set changed as a result of the call.

Usage

This method results in the *union* of the two sets. The specified set must be of the same type as the original set that calls the method.

Example

```
Set<String> myString = new Set<String>{'a', 'b'};
Set<String> sString = new Set<String>{'c'};

Boolean result1 = myString.addAll(sString);
System.assertEquals(true, result1);
```

clear()

Removes all of the elements from the set.

Signature

```
public Void clear()
```

Return Value

Type: Void

clone()

Makes a duplicate copy of the set.

Signature

```
public Set<Object> clone()
```

Return Value

Type: Set (of same type)

contains(setElement)

Returns `true` if the set contains the specified element.

Signature

```
public Boolean contains(Object setElement)
```

Parameters

setElement
Type: Object

Return Value

Type: Boolean

Example

```
Set<String> myString = new Set<String>{'a', 'b'};
Boolean result = myString.contains('z');
System.assertEquals(false, result);
```

containsAll(listToCompare)

Returns `true` if the set contains all of the elements in the specified list. The list must be of the same type as the set that calls the method.

Signature

```
public Boolean containsAll(List<Object> listToCompare)
```

Parameters

listToCompare
Type: List<Object>

Return Value

Type: Boolean

containsAll(setToCompare)

Returns `true` if the set contains all of the elements in the specified set. The specified set must be of the same type as the original set that calls the method.

Signature

```
public Boolean containsAll(Set<Object> setToCompare)
```

Parameters

setToCompare
Type: Set<Object>

Return Value

Type: Boolean

Example

```
Set<String> myString = new Set<String>{'a', 'b'};
Set<String> sString = new Set<String>{'c'};
Set<String> rString = new Set<String>{'a', 'b', 'c'};

Boolean result1, result2;
result1 = myString.addAll(sString);
system.assertEquals(true, result1);

result2 = myString.containsAll(rString);
System.assertEquals(true, result2);
```

equals(set2)

Compares this set with the specified set and returns `true` if both sets are equal; otherwise, returns `false`.

Signature

```
public Boolean equals(Set<Object> set2)
```

Parameters

set2

Type: Set<Object>

The *set2* argument is the set to compare this set with.

Return Value

Type: Boolean

Usage

Two sets are equal if their elements are equal, regardless of their order. The `==` operator is used to compare the elements of the sets.

The `==` operator is equivalent to calling the `equals` method, so you can call `set1.equals(set2);` instead of `set1 == set2;`.

hashCode()

Returns the hashcode corresponding to this set and its contents.

Signature

```
public Integer hashCode()
```

Return Value

Type: Integer

isEmpty()

Returns `true` if the set has zero elements.

Signature

```
public Boolean isEmpty()
```

Return Value

Type: Boolean

Example

```
Set<Integer> mySet = new Set<Integer>();
Boolean result = mySet.isEmpty();
System.assertEquals(true, result);
```

remove(setElement)

Removes the specified element from the set if it is present.

Signature

```
public Boolean remove(Object setElement)
```

Parameters

setElement
Type: Object

Return Value

Type: Boolean

Returns `true` if the original set changed as a result of the call.

removeAll(listOfElementsToRemove)

Removes the elements in the specified list from the set if they are present.

Signature

```
public Boolean removeAll(List<Object> listOfElementsToRemove)
```

Parameters

listOfElementsToRemove
Type: List<Object>

Return Value

Type: Boolean

Returns `true` if the original set changed as a result of the call.

Usage

This method results in the *relative complement* of the two sets. The list must be of the same type as the set that calls the method.

Example

```
Set<integer> mySet = new Set<integer>{1, 2, 3};
List<integer> myList = new List<integer>{1, 3};
Boolean result = mySet.removeAll(myList);
System.assertEquals(true, result);
Integer result2 = mySet.size();
System.assertEquals(1, result2);
```

removeAll(setOfElementsToRemove)

Removes the elements in the specified set from the original set if they are present.

Signature

```
public Boolean removeAll(Set<Object> setOfElementsToRemove)
```

Parameters

setOfElementsToRemove
Type: Set<Object>

Return Value

Type: Boolean

This method returns `true` if the original set changed as a result of the call.

Usage

This method results in the *relative complement* of the two sets. The specified set must be of the same type as the original set that calls the method.

retainAll(listOfElementsToRetain)

Retains only the elements in this set that are contained in the specified list.

Signature

```
public Boolean retainAll(List<Object> listOfElementsToRetain)
```

Parameters

listOfElementsToRetain
Type: List<Object>

Return Value

Type: Boolean

This method returns `true` if the original set changed as a result of the call.

Usage

This method results in the *intersection* of the list and the set. The list must be of the same type as the set that calls the method.

Example

```
Set<integer> mySet = new Set<integer>{1, 2, 3};
List<integer> myList = new List<integer>{1, 3};
Boolean result = mySet.retainAll(myList);
System.assertEquals(true, result);
```

retainAll(setOfElementsToRetain)

Retains only the elements in the original set that are contained in the specified set.

Signature

```
public Boolean retainAll(Set setOfElementsToRetain)
```

Parameters

setOfElementsToRetain
Type: Set

Return Value

Type: Boolean

Returns `true` if the original set changed as a result of the call.

Usage

This method results in the *intersection* of the two sets. The specified set must be of the same type as the original set that calls the method.

size()

Returns the number of elements in the set (its cardinality).

Signature

```
public Integer size()
```

Return Value

Type: Integer

Example

```
Set<Integer> mySet = new Set<Integer>{1, 2, 3};
Set<Integer> retainSet = new Set<Integer>{1, 3};
Boolean result = mySet.retainAll(retainSet);

Assert.isTrue(result, 'Expected to have changed mySet');

Integer retainedSetSize = mySet.size();
Assert.areEqual(2, retainedSetSize);
```

`toString()`

Returns the string representation of the set.

Signature

```
public String toString()
```

Return Value

Type: String

Usage

When used in cyclic references, the output is truncated to prevent infinite recursion. When used with large collections, the output is truncated to avoid exceeding total heap size and maximum CPU time.

- Up to 10 items per collection are included in the output, followed by an ellipsis (...).
- If the same object is included multiple times in a collection, it's shown in the output only once; subsequent references are shown as `(already output)`.

Site Class

Use the `Site` Class to manage your sites. Change, reset, validate, and check the expiration of passwords. Create site users, person accounts, and portal users. Get the admin email and ID. Get various URLs, the path prefix, the ID, the template, and the type of the site. Log in to the site.

Namespace

System

Site Methods

The following are methods for `Site`. All methods are static.

IN THIS SECTION:

changePassword(newPassword, verifyNewPassword, oldPassword)

Changes the password of the current user.

createExternalUser(user, accountId)

Creates a Salesforce Site or Experience Cloud site user for the given account and associates it with the site.

createExternalUser(user, accountId, password)

Creates a Salesforce Site or Experience Cloud site user for the given account and associates it with the site. This method sends an email with the specified password to the user.

createExternalUser(user, accountId, password, sendEmailConfirmation)

Creates a Salesforce Site or Experience Cloud site user and associates it with the given account. This method sends the user an email with the specified password and a new user confirmation email.

createPersonAccountPortalUser(user, ownerId, password)

Creates a person account using the default record type defined on the guest user's profile, then enables it for the site's portal.

createPersonAccountPortalUser(user, ownerId, recordTypeId, password)

Creates a person account using the specified *recordTypeID*, then enables it for the site's portal.

createPortalUser(user, accountId, password, sendEmailConfirmation)

Creates a portal user for the given account and associates it with the site's portal.

forgotPassword(username, emailTemplateName)

Resets the user's password and sends an email to the user with the user's new password. You can specify a custom email template or use the default email template. Returns a value indicating whether the password reset was successful.

forgotPassword(username)

Resets the user's password and sends an email to the user with the user's new password. Returns a value indicating whether the password reset was successful.

getAdminEmail()

Returns the email address of the site administrator.

getAdminId()

Returns the user ID of the site administrator.

getAnalyticsTrackingCode()

The tracking code associated with your site. Services such as Google Analytics can use this code to track page request data for your site.

getCurrentSiteUrl()

Deprecated. This method was replaced by `getBaseUrl()` in API version 30.0. Returns the base URL of the current site that references and links should use.

getBaseCustomUrl()

Returns a base URL for the current site that doesn't use a force.com subdomain. The returned URL uses the same protocol (HTTP or HTTPS) as the current request if at least one non-Force.com custom URL that supports HTTPS exists on the site. The returned value never ends with a `/` character. If all the custom URLs in this site end in Force.com or this site has no custom URLs, then this returns an empty string. If the current request is not a site request, then this method returns an empty string. This method replaced getCustomWebAddress and includes the custom URL's path prefix..

getBaseInsecureUrl()

Deprecated. Returns a base URL for the current site that uses HTTP instead of HTTPS. The current request's domain is used. The returned value includes the path prefix and never ends with a `/` character. If the current request is not a site request, then this method returns an empty string.

getBaseRequestUrl()

Returns the base URL of the current site for the requested URL. This isn't influenced by the referring page's URL. The returned URL uses the same protocol (HTTP or HTTPS) as the current request. The returned value includes the path prefix and never ends with a `/` character. If the current request is not a site request, then this method returns an empty string.

getBaseSecureUrl()

Returns a base URL for the current site that uses HTTPS instead of HTTP. The current request's domain is preferred if it supports HTTPS. Domains that are not Force.com subdomains are preferred over Force.com subdomains. A Force.com subdomain, if associated with the site, is used if no other HTTPS domains exist in the current site. If no HTTPS custom URLs exist in the site, then this method returns an empty string. The returned value includes the path prefix and never ends with a `/` character. If the current request is not a site request, then this method returns an empty string.

getBaseUrl()

Returns the base URL of the current site that references and links should use. Note that this field may return the referring page's URL instead of the current request's URL. The returned value includes the path prefix and never ends with a `/` character. If the current request is not a site request, then this field returns an empty string. This field replaces getCurrentSiteUrl.

getCustomWebAddress()

Deprecated. This method was replaced by `getBaseCustomUrl()` in API version 30.0.

getDomain()

Returns your Salesforce Sites based URL.

getErrorDescription()

Returns the error description for the current page if it's a designated error page for the site and an error exists; otherwise, returns an empty string.

getErrorMessage()

Returns an error message for the current page if it's a designated error page for the site and an error exists; otherwise, returns an empty string.

getExperienceId()

Returns the value of the experience ID (expid). This expid value comes from a cookie in the user's web browser.

getMasterLabel()

Returns the value of the Master Label field for the current site. If the current request is not a site request, then this field returns `null`.

getName()

Returns the API name of the current site.

getOriginalUrl()

Returns the original URL for this page if it's a designated error page for the site; otherwise, returns `null`.

getPasswordPolicyStatement()

Returns the password requirements for a Salesforce Site or Experience Cloud site created with the Customer Service template.

getPathPrefix()

Returns the URL path prefix of the current site or an empty string if none. For example, if the requested site URL is `https://myco.my.salesforce-sites.com/partners`, then `/partners` is the path prefix. If the current request is not a site request, then this method returns an empty string. This method replaced getPrefix in API version 30.0.

getPrefix()

Deprecated. This method was replaced by `getPathPrefix()` in API version 30.0.

getSiteId()

Returns the ID of the current site. If the current request is not a site request, then this field returns `null`.

getTemplate()

Returns the template name associated with the current site; returns the default template if no template has been designated.

getSiteType()

Returns the API value of the site type field for the current site. This can be Visualforce for a Salesforce site, Siteforce for a Site.com site, ChatterNetwork for an Experience Cloud site, or ChatterNetworkPicasso for an Experience Cloud site. If the current request is not a site request, then this method returns `null`.

getSiteTypeLabel()

Returns the value of the Site Type field's label for the current site. If the current request is not a site request, then this method returns `null`.

isLoginEnabled()

Returns `true` if the current site is associated with an active login-enabled portal; otherwise returns `false`.

isPasswordExpired()

For authenticated users, returns `true` if the currently logged-in user's password is expired. For non-authenticated users, returns `false`.

isRegistrationEnabled()

Returns `true` if the current site is associated with an active self-registration-enabled Customer Portal; otherwise returns `false`.

isValidUsername(username)

Returns `true` if the given username is valid; otherwise, returns `false`.

login(username, password, startUrl)

Allows users to log in to the current site with the given username and password, then takes them to the `startUrl`. If `startUrl` is not a relative path, it defaults to the site's designated index page.

passwordlessLogin(userId, methods, startUrl)

Logs in a user to a Salesforce Site or Experience Cloud site using an identity verification method, such as email or text, instead of a password. Passwordless login is a convenient, mobile-centric way to welcome users into your site. Let your users log in with something other than their password, like their email address or phone number.

setExperienceId(expIdValue)

Sets the experience ID for the current user. Use this method to populate the value of the experience ID (expid) cookie in the user's web browser.

setPortalUserAsAuthProvider(user, contactId)

Sets the specified user information within the site's portal via an authentication provider.

validatePassword(user, password, confirmPassword)

Indicates whether a given password meets the requirements specified by org-wide or profile-based password policies in the current user's org.

changePassword(newPassword, verifyNewPassword, oldPassword)

Changes the password of the current user.

Signature

```
public static System.PageReference changePassword(String newPassword, String
verifyNewPassword, String oldPassword)
```

Parameters

newPassword
Type: String

verifyNewPassword
Type: String

oldPassword
Type: String

Optional only if the current user's password has expired; otherwise, required.

Return Value

Type: System.PageReference

Usage

Calls to this method in API version 30.0 and later can't commit the transaction automatically. Calls to this method before API version 30.0 commit the transaction, making it impossible to roll back to a save point before the call.

createExternalUser(user, accountId)

Creates a Salesforce Site or Experience Cloud site user for the given account and associates it with the site.

Signature

```
public static Id createExternalUser(SObject user, String accountId)
```

Parameters

user
Type: SObject

Information required to create a user.

The email address of the user is used to look for matching contacts associated with the specified *accountId*. If a matching contact is found and is already used by an external user, self-registration isn't successful. If a matching contact is found but isn't used by an external user, it is used for the new external user. If there is no matching contact, a new contact is created for the new external user.

accountId
Type: String

The ID of the account you want to associate the user with.

Return Value

Type: Id

The ID of the user that this method creates.

Usage

This method throws `Site.ExternalUserCreateException` when user creation fails.

The `nickname` field is required for the `User` sObject when using the `createExternalUser` method.

Note: This method is only valid when a site is associated with a Customer Portal.

Calls to this method in API version 30.0 and later can't commit the transaction automatically. Calls to this method before API version 30.0 commit the transaction, making it impossible to roll back to a save point before the call.

createExternalUser(user, accountId, password)

Creates a Salesforce Site or Experience Cloud site user for the given account and associates it with the site. This method sends an email with the specified password to the user.

Signature

```
public static Id createExternalUser(SObject user, String accountId, String password)
```

Parameters

user
Type: SObject

Information required to create a user.

The email address of the user is used to look for matching contacts associated with the specified *accountId*. If a matching contact is found and is already used by an external user, self-registration isn't successful. If a matching contact is found but isn't used by an external user, it is used for the new external user. If there is no matching contact, a new contact is created for the new external user.

accountId
Type: String

The ID of the account you want to associate the user with.

password
Type: String

The password of the Salesforce Site or Experience Cloud site user. If not specified, or if set to `null` or an empty string, this method sends a new password email to the portal user.

Return Value

Type: Id

The ID of the user that this method creates.

Usage

This method throws `Site.ExternalUserCreateException` when user creation fails.

The `nickname` field is required for the `User` sObject when using the `createExternalUser` method.

Note: This method is only valid when a site is associated with a Customer Portal.

Calls to this method in API version 30.0 and later can't commit the transaction automatically. Calls to this method before API version 30.0 commit the transaction, making it impossible to roll back to a save point before the call.

createExternalUser(user, accountId, password, sendEmailConfirmation)

Creates a Salesforce Site or Experience Cloud site user and associates it with the given account. This method sends the user an email with the specified password and a new user confirmation email.

Signature

```
public static Id createExternalUser(SObject user, String accountId, String password,
Boolean sendEmailConfirmation)
```

Parameters

user
Type: SObject

Information required to create a user.

The email address of the user is used to look for matching contacts associated with the specified *accountId*. If a matching contact is found and is already used by an external user, self-registration isn't successful. If a matching contact is found but isn't used by an external user, it is used for the new external user. If there is no matching contact, a new contact is created for the new external user.

accountId
Type: String

The ID of the account you want to associate the user with.

password
Type: String

The password of the Salesforce Site or Experience Cloud site user. If not specified, or if set to `null` or an empty string, this method sends a new password email to the portal user.

sendEmailConfirmation
Type: Boolean

Determines whether a new user email is sent to the portal user. Set it to `true` to send a new user email to the portal user. The default is `false`, that is, the new user email isn't sent.

Return Value

Type: Id

The ID of the user that this method creates.

Usage

This method throws `Site.ExternalUserCreateException` when user creation fails.

The `nickname` field is required for the `User` sObject when using the `createExternalUser` method.

Note: This method is only valid when a site is associated with a Customer Portal.

Calls to this method in API version 30.0 and later can't commit the transaction automatically. Calls to this method before API version 30.0 commit the transaction, making it impossible to roll back to a save point before the call.

createPersonAccountPortalUser(user, ownerId, password)

Creates a person account using the default record type defined on the guest user's profile, then enables it for the site's portal.

Signature

```
public static ID createPersonAccountPortalUser(sObject user, String ownerId, String password)
```

Parameters

user
Type: sObject

ownerId
Type: String

password
Type: String

Return Value

Type: ID

Usage

Calls to this method in API version 30.0 and later can't commit the transaction automatically. Calls to this method before API version 30.0 commit the transaction, making it impossible to roll back to a save point before the call.

Note: This method is only valid when a site is associated with a Customer Portal, and when the user license for the default new user profile is a high-volume portal user.

createPersonAccountPortalUser(user, ownerId, recordTypeId, password)

Creates a person account using the specified *recordTypeID*, then enables it for the site's portal.

Signature

```
public static ID createPersonAccountPortalUser(sObject user, String ownerId, String recordTypeId, String password)
```

Parameters

user
Type: sObject

ownerId
Type: String

recordTypeId
Type: String

password
Type: String

Return Value

Type: ID

Usage

Calls to this method in API version 30.0 and later can't commit the transaction automatically. Calls to this method before API version 30.0 commit the transaction, making it impossible to roll back to a save point before the call.

Note: This method is only valid when a site is associated with a Customer Portal, and when the user license for the default new user profile is a high-volume portal user.

createPortalUser(user, accountId, password, sendEmailConfirmation)

Creates a portal user for the given account and associates it with the site's portal.

Signature

```
public static ID createPortalUser(sObject user, String accountId, String password,
Boolean sendEmailConfirmation)
```

Parameters

user
Type: sObject

accountId
Type: String

password
Type: String

(Optional) The password of the portal user. If not specified, or if set to `null` or an empty string, this method sends a new password email to the portal user.

sendEmailConfirmation
Type: Boolean

(Optional) Determines whether a new user email is sent to the portal user. Set it to `true` to send a new user email to the portal user. The default is `false`, that is, the new user email isn't sent.

Return Value

Type: ID

Usage

If you're using API version 34.0 or later, we recommend using the `createExternalUser()` methods because they offer better error handling than this method.

The `nickname` field is required for the user sObject when using the `createPortalUser` method.

Note: This method is only valid when a site is associated with a Customer Portal.

Calls to this method in API version 30.0 and later can't commit the transaction automatically. Calls to this method before API version 30.0 commit the transaction, making it impossible to roll back to a save point before the call.

forgotPassword(username, emailTemplateName)

Resets the user's password and sends an email to the user with the user's new password. You can specify a custom email template or use the default email template. Returns a value indicating whether the password reset was successful.

Signature

```
public static Boolean forgotPassword(String username,String emailTemplateName)
```

Parameters

username
Type: String

emailTemplateName
Type: String

If provided, the method applies the template to the email. Otherwise, the method applies the default system template. If an email template that doesn't exist is provided, the system logs an exception.

Return Value

Type: Boolean

Note: The return value is always true unless it's called outside of a Visualforce page.

Usage

Calls to this method in API version 30.0 and later can't commit the transaction automatically. Calls to this method before API version 30.0 commit the transaction, making it impossible to roll back to a save point before the call.

Calls to this method are subject to rate-limiting. If your rate of calls exceeds the limit, Salesforce doesn't send the password reset email. If you experience this issue, try waiting for an hour before you send another call.

Note: `Site.forgotPassword` cannot be used with the `@future` method, which enables asynchronous execution.

forgotPassword(username)

Resets the user's password and sends an email to the user with the user's new password. Returns a value indicating whether the password reset was successful.

Signature

```
public static Boolean forgotPassword(String username)
```

Parameters

username
Type: String

Return Value

Type: Boolean

Note: The return value is always true unless it's called outside of a Visualforce page.

Usage

Calls to this method in API version 30.0 and later can't commit the transaction automatically. Calls to this method before API version 30.0 commit the transaction, making it impossible to roll back to a save point before the call.

Calls to this method are subject to rate-limiting. If your rate of calls exceeds the limit, Salesforce doesn't send the password reset email. If you experience this issue, try waiting for an hour before you send another call.

Note: `Site.forgotPassword` cannot be used with the `@future` method, which enables asynchronous execution.

`getAdminEmail()`

Returns the email address of the site administrator.

Signature

```
public static String getAdminEmail()
```

Return Value

Type: String

`getAdminId()`

Returns the user ID of the site administrator.

Signature

```
public static ID getAdminId()
```

Return Value

Type: ID

`getAnalyticsTrackingCode()`

The tracking code associated with your site. Services such as Google Analytics can use this code to track page request data for your site.

Signature

```
public static String getAnalyticsTrackingCode()
```

Return Value

Type: String

getCurrentSiteUrl()

Deprecated. This method was replaced by `getBaseUrl()` in API version 30.0. Returns the base URL of the current site that references and links should use.

Note that this may return the referring page's URL instead of the current request's URL. The returned value includes the path prefix and always ends with a `/` character. If the current request is not a site request, then this method returns `null`. If the current request is not a site request, then this method returns `null`. This method was replaced by getBaseUrl in API version 30.0.

Signature

```
public static String getCurrentSiteUrl()
```

Return Value

Type: String

Usage

Use `getBaseUrl()` instead.

getBaseCustomUrl()

Returns a base URL for the current site that doesn't use a force.com subdomain. The returned URL uses the same protocol (HTTP or HTTPS) as the current request if at least one non-Force.com custom URL that supports HTTPS exists on the site. The returned value never ends with a `/` character. If all the custom URLs in this site end in Force.com or this site has no custom URLs, then this returns an empty string. If the current request is not a site request, then this method returns an empty string. This method replaced getCustomWebAddress and includes the custom URL's path prefix..

Signature

```
public static String getBaseCustomUrl()
```

Return Value

Type: String

Usage

This method replaces `getCustomWebAddress()` and includes the custom URL's path prefix.

getBaseInsecureUrl()

Deprecated. Returns a base URL for the current site that uses HTTP instead of HTTPS. The current request's domain is used. The returned value includes the path prefix and never ends with a `/` character. If the current request is not a site request, then this method returns an empty string.

Signature

```
public static String getBaseInsecureUrl()
```

Return Value

Type: String

getBaseRequestUrl()

Returns the base URL of the current site for the requested URL. This isn't influenced by the referring page's URL. The returned URL uses the same protocol (HTTP or HTTPS) as the current request. The returned value includes the path prefix and never ends with a / character. If the current request is not a site request, then this method returns an empty string.

Signature

```
public static String getBaseRequestUrl()
```

Return Value

Type: String

getBaseSecureUrl()

Returns a base URL for the current site that uses HTTPS instead of HTTP. The current request's domain is preferred if it supports HTTPS. Domains that are not Force.com subdomains are preferred over Force.com subdomains. A Force.com subdomain, if associated with the site, is used if no other HTTPS domains exist in the current site. If no HTTPS custom URLs exist in the site, then this method returns an empty string. The returned value includes the path prefix and never ends with a / character. If the current request is not a site request, then this method returns an empty string.

Signature

```
public static String getBaseSecureUrl()
```

Return Value

Type: String

getBaseUrl()

Returns the base URL of the current site that references and links should use. Note that this field may return the referring page's URL instead of the current request's URL. The returned value includes the path prefix and never ends with a / character. If the current request is not a site request, then this field returns an empty string. This field replaces getCurrentSiteUrl.

Signature

```
public static String getBaseUrl()
```

Return Value

Type: String

Usage

This method replaces `getCurrentSiteUrl()`.

getCustomWebAddress()

Deprecated. This method was replaced by `getBaseCustomUrl()` in API version 30.0.

Returns the request's custom URL if it doesn't end in Lightning Platform or returns the site's primary custom URL. If neither exist, then this returns null. Note that the URL's path is always the root, even if the request's custom URL has a path prefix. If the current request is not a site request, then this method returns null. The returned value always ends with a `/` character.

Signature

```
public static String getCustomWebAddress()
```

Return Value

Type: String

Usage

Use `getBaseCustomUrl()` instead.

getDomain()

Returns your Salesforce Sites based URL.

Signature

```
public static String getDomain()
```

Return Value

Type: String

getErrorDescription()

Returns the error description for the current page if it's a designated error page for the site and an error exists; otherwise, returns an empty string.

Signature

```
public static String getErrorDescription()
```

Return Value

Type: String

getErrorMessage()

Returns an error message for the current page if it's a designated error page for the site and an error exists; otherwise, returns an empty string.

Signature

```
public static String getErrorMessage()
```

Return Value

Type: String

getExperienceId()

Returns the value of the experience ID (expid). This expid value comes from a cookie in the user's web browser.

Signature

```
public static String getExperienceId()
```

Return Value

Type: String

Usage

Use the `getExperienceId` and `setExperienceId` methods to implement dynamic login experiences. You can set the experience ID with `setExperienceId` or by extending the following endpoints with `expid_`***`value`***.

- ***`community-url`***`/services/oauth2/authorize/expid_`***`value`***
- ***`community-url`***`/idp/endpoint/HttpPost/expid_`***`value`***
- ***`community-url`***`/idp/endpoint/HttpRedirect/expid_`***`value`***
- ***`community-url_login_page`***`/expid=`***`{value}`***
- ***`community-url`***`/CommunitiesSelfReg?expid=`***`{value}`***
- ***`secur`***`/forgotpassword.jsp?expid=`***`{value}`***

The cookie is set when the browser loads the URLs with the expid values.

getMasterLabel()

Returns the value of the Master Label field for the current site. If the current request is not a site request, then this field returns `null`.

Signature

```
public static String getMasterLabel()
```

Return Value

Type: String

getName()

Returns the API name of the current site.

Signature

```
public static String getName()
```

Return Value

Type: String

getOriginalUrl()

Returns the original URL for this page if it's a designated error page for the site; otherwise, returns `null`.

Signature

```
public static String getOriginalUrl()
```

Return Value

Type: String

getPasswordPolicyStatement()

Returns the password requirements for a Salesforce Site or Experience Cloud site created with the Customer Service template.

Signature

```
public static String getPasswordPolicyStatement()
```

Return Value

Type: String

getPathPrefix()

Returns the URL path prefix of the current site or an empty string if none. For example, if the requested site URL is `https://myco.my.salesforce-sites.com/partners`, then `/partners` is the path prefix. If the current request is not a site request, then this method returns an empty string. This method replaced getPrefix in API version 30.0.

Signature

```
public static String getPathPrefix()
```

Return Value

Type: String

getPrefix()

Deprecated. This method was replaced by `getPathPrefix()` in API version 30.0.

Returns the URL path prefix of the current site. For example, if your site URL is *MyDomainName*`.my.salesforce-sites.com/partners`, `/partners` is the path prefix. Returns `null` if the prefix isn't defined. If the current request is not a site request, then this method returns a `null`.

Signature

```
public static String getPrefix()
```

Return Value

Type: String

`getSiteId()`

Returns the ID of the current site. If the current request is not a site request, then this field returns `null`.

Signature

```
public static String getSiteId()
```

Return Value

Type: Id

`getTemplate()`

Returns the template name associated with the current site; returns the default template if no template has been designated.

Signature

```
public static System.PageReference getTemplate()
```

Return Value

Type: System.PageReference

`getSiteType()`

Returns the API value of the site type field for the current site. This can be Visualforce for a Salesforce site, Siteforce for a Site.com site, ChatterNetwork for an Experience Cloud site, or ChatterNetworkPicasso for an Experience Cloud site. If the current request is not a site request, then this method returns `null`.

Signature

```
public static String getSiteType()
```

Return Value

Type: String

`getSiteTypeLabel()`

Returns the value of the Site Type field's label for the current site. If the current request is not a site request, then this method returns `null`.

Signature

```
public static String getSiteTypeLabel()
```

Return Value

Type: String

`isLoginEnabled()`

Returns `true` if the current site is associated with an active login-enabled portal; otherwise returns `false`.

Signature

```
public static Boolean isLoginEnabled()
```

Return Value

Type: Boolean

`isPasswordExpired()`

For authenticated users, returns `true` if the currently logged-in user's password is expired. For non-authenticated users, returns `false`.

Signature

```
public static Boolean isPasswordExpired()
```

Return Value

Type: Boolean

`isRegistrationEnabled()`

Returns `true` if the current site is associated with an active self-registration-enabled Customer Portal; otherwise returns `false`.

Signature

```
public static Boolean isRegistrationEnabled()
```

Return Value

Type: Boolean

`isValidUsername(username)`

Returns `true` if the given username is valid; otherwise, returns `false`.

Signature

```
public static Boolean isValidUsername(String username)
```

Parameters

`username`
Type: String

The username to test for validity.

Return Value

Type: Boolean

`login(username, password, startUrl)`

Allows users to log in to the current site with the given username and password, then takes them to the `startUrl`. If `startUrl` is not a relative path, it defaults to the site's designated index page.

Signature

```
public static System.PageReference login(String username, String password, String
startUrl)
```

Parameters

`username`
Type: String

`password`
Type: String

`startUrl`
Type: String

Return Value

Type: System.PageReference

Usage

All DML statements before the call to `Site.login` get committed. It's not possible to roll back to a save point that was created before a call to `Site.login`.

Note: Do not include `http://` or `https://` in the `startURL`.

passwordlessLogin(userId, methods, startUrl)

Logs in a user to a Salesforce Site or Experience Cloud site using an identity verification method, such as email or text, instead of a password. Passwordless login is a convenient, mobile-centric way to welcome users into your site. Let your users log in with something other than their password, like their email address or phone number.

Signature

```
public static System.PageReference passwordlessLogin(Id userId,
List<Auth.VerificationMethod> methods, String startUrl)
```

Parameters

userId
Type: Id

ID of the user to log in.

methods
Type: List<Auth.VerificationMethod>

List of identity verification methods available to the user for passwordless login.

startUrl
Type: String

Path to the page that users see after they log in.

Return Value

Type: System.PageReference

Usage

Include this method in the Apex controller of a custom login page implementation.

PasswordlessLogin Example

This simple code example of an Apex controller contains the `passwordlessLogin` method. The `PageReference` returned by `passwordlessLogin` redirects the user to the Salesforce Verify page. When the user enters the correct code, the user is redirected to the site page specified by the start URL.

```
global with sharing class MFILoginController
{
   //Input variables
   global String input {get; set;}
   public String startURL  {get; set;}
   public List<Auth.VerificationMethod> methods;
   public String error;

   global MFILoginController()
   {
       // Add verification methods in priority order
       methods = new List<Auth.VerificationMethod>();
       methods.add(Auth.VerificationMethod.SMS);
```

```
        methods.add(Auth.VerificationMethod.EMAIL);
        methods.add(Auth.VerificationMethod.U2F);
        methods.add(Auth.VerificationMethod.SALESFORCE_AUTHENTICATOR);
        methods.add(Auth.VerificationMethod.TOTP);
    }

    global PageReference login() {
         List<User> users = null;

         // Empty input
         if(input == null || input == '')
         {
             error = 'Enter Username';
             return null;
         }

         users = [select name, id, email from User where username=:input];
         if(users == null || users.isEmpty())
         {
             error = 'Can\'t find a user';
             return null;
         }

         if (startURL == null) startURL = '/';
         return Site.passwordlessLogin(users[0].id, methods, startURL);
     }
}
```

setExperienceId(expIdValue)

Sets the experience ID for the current user. Use this method to populate the value of the experience ID (expid) cookie in the user's web browser.

Signature

```
public static void setExperienceId(String expIdValue)
```

Parameters

expIdValue

Type: String

A value that indicates the user's login experience.

The value must contain alphanumeric characters only, up to 30 characters.

Usage

Use `setExperienceId` when you're implementing dynamic login experiences. A login experience refers to a login page plus any secondary pages associated with the login page (such as multi-factor authentication (MFA) or a login flow). You define different login experiences depending on who users are or where they're logging in from. For example, you can require a different registration process based on the user's location. In this case, `expIdValue` includes a state or country code. When the user logs in, the URL contains the

experience ID parameter, `{expid}`. The `{expid}` parameter is replaced by the value stored in `expIdValue`, such as `.jp`. Then the user is redirected to the Japanese login experience.

Example

```
String expid = ApexPages.currentPage().getParameters().get('expid');
  if (expId != null) {
  Site.setExperienceId(expId);
  }
```

setPortalUserAsAuthProvider(user, contactId)

Sets the specified user information within the site's portal via an authentication provider.

Signature

```
public static Void setPortalUserAsAuthProvider(sObject user, String contactId)
```

Parameters

user
Type: sObject

contactId
Type: String

Return Value

Type: Void

Usage

- This method is only valid when a site is associated with a Customer Portal.
- Calls to this method in API version 30.0 and later can't commit the transaction automatically. Calls to this method before API version 30.0 commit the transaction, making it impossible to roll back to a save point before the call.
- For more information on an authentication provider, see RegistrationHandler.

validatePassword(user, password, confirmPassword)

Indicates whether a given password meets the requirements specified by org-wide or profile-based password policies in the current user's org.

Signature

```
public static void validatePassword(SObject user, String password, String
confirmPassword)
```

Parameters

user
Type: SObject

The user attempting to create a password during self-registration for a Salesforce Site or Experience Cloud site.

password
Type: String

The password entered by the user.

confirmPassword
Type: String

The password reentered by the user to confirm the password.

Return Value

Type: void

Usage

If validation fails when the method is run in a Lightning controller, this method throws an Apex exception describing the failed validation. If validation fails when the method is run in a Visualforce controller, the method provides Visualforce error messages.

SObject Class

Contains methods for the sObject data type.

Namespace

System

Usage

SObject methods are all instance methods: they are called by and operate on an sObject instance such as an account or contact. The following are the instance methods for sObjects.

For more information on sObjects, see Working with sObjects.

SObject Methods

The following are methods for `SObject`. All are instance methods.

IN THIS SECTION:

addError(errorMsg)
Marks a trigger record with a custom error message and prevents any DML operation from occurring.

addError(errorMsg, escape)
Marks a trigger record with a custom error message, specifies if the error message should be escaped, and prevents any DML operation from occurring.

addError(exceptionError)
Marks a trigger record with a custom error message and prevents any DML operation from occurring.

addError(exceptionError, escape)

Marks a trigger record with a custom exception error message, specifies whether or not the exception error message should be escaped, and prevents any DML operation from occurring.

addError(errorMsg)

Places the specified error message on a trigger record field in the Salesforce user interface and prevents any DML operation from occurring.

addError(errorMsg, escape)

Places the specified error message, which can be escaped or unescaped, on a trigger record field in the Salesforce user interface, and prevents any DML operation from occurring.

addError(fieldName, errorMsg)

Dynamically add errors to fields of an SObject associated with the specified field name.

addError(fieldToken, errorMsg)

Dynamically add errors to an SObject instance associated with the specified field.

addError(fieldName, errorMsg, escape)

Dynamically add errors to fields of an SObject associated with the specified field name.

addError(fieldToken, errorMsg, escape)

Dynamically add errors to an SObject instance associated with the specified field.

clear()

Clears all field values

clone(preserveId, isDeepClone, preserveReadonlyTimestamps, preserveAutonumber)

Creates a copy of the SObject record.

get(fieldName)

Returns the value for the field specified by *`fieldName`*, such as `AccountNumber`.

get(field)

Returns the value for the field specified by the field token `Schema.`***`sObjectField`***, such as, `Schema.Account.AccountNumber`.

getCloneSourceId()

Returns the ID of the entity from which an object was cloned. You can use it for objects cloned through the Salesforce user interface. You can also use it for objects created using the `System.SObject.clone(preserveId, isDeepClone, preserveReadonlyTimestamps, preserveAutonumber)` method, provided that the *`preserveId`* parameter wasn't used or was set to `false`. The `getCloneSourceId()` method can only be used within the transaction where the entity is cloned, as clone information doesn't persist in subsequent transactions.

getErrors()

Returns a list of `Database.Error` objects for an SObject instance. If the SObject has no errors, an empty list is returned.

getOptions()

Returns the database.DMLOptions object for the SObject.

getPopulatedFieldsAsMap()

Returns a map of populated field names and their corresponding values. The map contains only the fields that have been populated in memory for the SObject instance.

getSObject(fieldName)

Returns the value for the specified field. This method is primarily used with dynamic DML to access values for external IDs.

getSObject(field)

Returns the value for the field specified by the field token `Schema.sObjectField`, such as, `Schema.MyObj.MyExternalId`. This method is primarily used with dynamic DML to access values for external IDs.

getSObjects(fieldName)

Returns the values for the specified field. This method is primarily used with dynamic DML to access values for associated objects, such as child relationships.

getSObjects(fieldName)

Returns the value for the field specified by the field token `Schema.fieldName`, such as, `Schema.Account.Contact`. This method is primarily used with dynamic DML to access values for associated objects, such as child relationships.

getSObjectType()

Returns the token for this SObject. This method is primarily used with describe information.

getQuickActionName()

Retrieves the name of a quick action associated with this SObject. Typically used in triggers.

hasErrors()

Returns true if an SObject instance has associated errors. The error message can be associated to the SObject instance by using `SObject.addError()`, validation rules, or by other means.

isClone()

Returns `true` if an entity is cloned from something, even if the entity hasn't been saved. The method can only be used within the transaction where the entity is cloned, as clone information doesn't persist in subsequent transactions.

isSet(fieldName)

Returns information about the queried sObject field. Returns `true` if the sObject field is populated, either by direct assignment or by inclusion in a SOQL query. Returns `false` if the sObject field isn't set. If an invalid field is specified, an SObjectException is thrown.

isSet(field)

Returns information about the queried sObject field. Returns `true` if the sObject field is populated, either by direct assignment or by inclusion in a SOQL query. Returns `false` if the sObject field isn't set. If an invalid field is specified, an SObjectException is thrown.

put(fieldName, value)

Sets the value for the specified field and returns the previous value for the field.

put(field, value)

Sets the value for the field specified by the field token `Schema.sObjectField`, such as, `Schema.Account.AccountNumber` and returns the previous value for the field.

putSObject(fieldName, value)

Sets the value for the specified field. This method is primarily used with dynamic DML for setting external IDs. The method returns the previous value of the field.

putSObject(fieldName, value)

Sets the value for the field specified by the token `Schema.SObjectType`. This method is primarily used with dynamic DML for setting external IDs. The method returns the previous value of the field.

recalculateFormulas()

Deprecated as of API version 57.0. Use the `recalculateFormulas()` method in the `System.Formula` class instead.

setOptions(DMLOptions)

Sets the DMLOptions object for the SObject.

addError(errorMsg)

Marks a trigger record with a custom error message and prevents any DML operation from occurring.

Signature

```
public Void addError(String errorMsg)
```

Parameters

errorMsg
Type: String

The error message to mark the record with.

Return Value

Type: Void

Usage

When used on `Trigger.new` in `insert` and `update` triggers, and on `Trigger.old` in `delete` triggers, the error message is displayed in the application interface.

See Triggers and Trigger Exceptions.

Note: This method escapes any HTML markup in the specified error message. The escaped characters are: `\n`, `<`, `>`, `&`, `"`, `\`, `\u2028`, `\u2029`, and `\u00a9`. As a result, HTML markup is not rendered; instead, it is displayed as text in the Salesforce user interface.

When used in Visualforce controllers, the generated message is added to the collection of errors for the page. For more information, see Validation Rules and Standard Controllers in the *Visualforce Developer's Guide*.

Example

```
Trigger.new[0].addError('bad');
```

addError(errorMsg, escape)

Marks a trigger record with a custom error message, specifies if the error message should be escaped, and prevents any DML operation from occurring.

Signature

```
public Void addError(String errorMsg, Boolean escape)
```

Parameters

errorMsg
Type: String

The error message to mark the record with.

escape
Type: Boolean

Indicates whether any HTML markup in the custom error message should be escaped (`true`) or not (`false`). This parameter is ignored in both Lightning Experience and the Salesforce mobile app, and the HTML is always escaped. The escape parameter only applies in Salesforce Classic.

Return Value

Type: Void

Usage

The escaped characters are: `\n`, `<`, `>`, `&`, `"`, `\`, `\u2028`, `\u2029`, and `\u00a9`. As a result, HTML markup is not rendered; instead, it is displayed as text in the Salesforce user interface.

Warning: Be cautious if you specify `false` for the *escape* argument. Unescaped strings displayed in the Salesforce user interface can represent a vulnerability in the system because these strings might contain harmful code. If you want to include HTML markup in the error message, call this method with a `false` *escape* argument. Make sure that you escape any dynamic content, such as input field values. Otherwise, specify `true` for the *escape* argument or call `addError(String errorMsg)` instead.

Example

```
Trigger.new[0].addError('Fix & resubmit', false);
```

addError(exceptionError)

Marks a trigger record with a custom error message and prevents any DML operation from occurring.

Signature

```
public Void addError(Exception exceptionError)
```

Parameters

exceptionError
Type: System.Exception

An Exception object or a custom exception object that contains the error message to mark the record with.

Return Value

Type: Void

Usage

When used on `Trigger.new` in `insert` and `update` triggers, and on `Trigger.old` in `delete` triggers, the error message is displayed in the application interface.

See Triggers and Trigger Exceptions.

Note: This method escapes any HTML markup in the specified error message. The escaped characters are: `\n`, `<`, `>`, `&`, `"`, `\`, `\u2028`, `\u2029`, and `\u00a9`. As a result, HTML markup is not rendered; instead, it is displayed as text in the Salesforce user interface.

When used in Visualforce controllers, the generated message is added to the collection of errors for the page. For more information, see Validation Rules and Standard Controllers in the *Visualforce Developer's Guide*.

Example

```
public class MyException extends Exception {}
Trigger.new[0].addError(new myException('Invalid Id'));
```

addError(exceptionError, escape)

Marks a trigger record with a custom exception error message, specifies whether or not the exception error message should be escaped, and prevents any DML operation from occurring.

Signature

```
public Void addError(Exception exceptionError, Boolean escape)
```

Parameters

exceptionError
Type: System.Exception

An Exception object or a custom exception object that contains the error message to mark the record with.

escape
Type: Boolean

Indicates whether any HTML markup in the custom error message should be escaped (`true`) or not (`false`). This parameter is ignored in both Lightning Experience and the Salesforce mobile app, and the HTML is always escaped. The escape parameter only applies in Salesforce Classic.

Return Value

Type: Void

Usage

The escaped characters are: `\n`, `<`, `>`, `&`, `"`, `\`, `\u2028`, `\u2029`, and `\u00a9`. As a result, HTML markup is not rendered; instead, it is displayed as text in the Salesforce user interface.

Warning: Be cautious if you specify `false` for the *escape* argument. Unescaped strings displayed in the Salesforce user interface can represent a vulnerability in the system because these strings might contain harmful code. If you want to include HTML markup in the error message, call this method with a `false` *escape* argument. Make sure that you escape any dynamic content, such as input field values. Otherwise, specify `true` for the *escape* argument or call `addError(Exception e)` instead.

Example

```
public class MyException extends Exception {}
Trigger.new[0].addError(new myException('Invalid Id & other issues', false));
```

addError(errorMsg)

Places the specified error message on a trigger record field in the Salesforce user interface and prevents any DML operation from occurring.

Signature

```
public Void addError(String errorMsg)
```

Parameters

errorMsg
Type: String

Return Value

Type: Void

Usage

Note:

- When used on `Trigger.new` in `before insert` and `before update` triggers, and on `Trigger.old` in `before delete` triggers, the error appears in the application interface.
- When used in Visualforce controllers, if there is an `inputField` component bound to field, the message is attached to the component. For more information, see Validation Rules and Standard Controllers in the *Visualforce Developer's Guide*.
- This method is highly specialized because the field identifier is not actually the invoking object—the sObject record is the invoker. The field is simply used to identify the field that should be used to display the error.

See Triggers and Trigger Exceptions.

Note: This method escapes any HTML markup in the specified error message. The escaped characters are: `\n`, `<`, `>`, `&`, `"`, `\`, `\u2028`, `\u2029`, and `\u00a9`. As a result, HTML markup is not rendered; instead, it is displayed as text in the Salesforce user interface.

Example

```
Trigger.new[0].myField__c.addError('bad');
```

addError(errorMsg, escape)

Places the specified error message, which can be escaped or unescaped, on a trigger record field in the Salesforce user interface, and prevents any DML operation from occurring.

Signature

```
public Void addError(String errorMsg, Boolean escape)
```

Parameters

errorMsg
Type: String

The error message to mark the record with.

escape
Type: Boolean

Indicates whether any HTML markup in the custom error message should be escaped (`true`) or not (`false`). This parameter is ignored in both Lightning Experience and the Salesforce mobile app, and the HTML is always escaped. The escape parameter only applies in Salesforce Classic.

Return Value

Type:

Usage

The escaped characters are: `\n`, `<`, `>`, `&`, `"`, `\`, `\u2028`, `\u2029`, and `\u00a9`. As a result, HTML markup is not rendered; instead, it is displayed as text in the Salesforce user interface.

Warning: Be cautious if you specify `false` for the *escape* argument. Unescaped strings displayed in the Salesforce user interface can represent a vulnerability in the system because these strings might contain harmful code. If you want to include HTML markup in the error message, call this method with a `false` *escape* argument. Make sure that you escape any dynamic content, such as input field values. Otherwise, specify `true` for the *escape* argument or call *field*`.addError(String` ***errorMsg***`)` instead.

Example

```
Trigger.new[0].myField__c.addError('Fix & resubmit', false);
```

addError(fieldName, errorMsg)

Dynamically add errors to fields of an SObject associated with the specified field name.

Signature

```
public void addError(String fieldName, String errorMsg)
```

Parameters

fieldName
Type: String

The field name of the SObject .

errorMsg
Type: String

The error message to be added. HTML special characters in the error message string are always escaped.

Return Value

Type: void

Usage

If the field name is an empty string or null, the error is associated with the SObject and not with a specific field.

Example

```
// Add an error to an SObject field using the addError() method.
Account acct = new Account(name = 'TestAccount');
acct.addError('name', 'error in name field');
// Use the hasErrors() method to verify that the error is added, and then the getErrors()
 method to validate the error.
System.Assert(acct.hasErrors());
List<Database.Error> errors = acct.getErrors();
System.AssertEquals(1, errors.size());
```

addError(fieldToken, errorMsg)

Dynamically add errors to an SObject instance associated with the specified field.

Signature

```
public void addError(Schema.SObjectField fieldToken, String errorMsg
```

Parameters

fieldToken
Type: Schema.SObjectField

The field of the SObject instance.

errorMsg
Type: String

The error message to be added. HTML special characters in the error message string are always escaped.

Return Value

Type: void

Usage

Use this method to add errors to the specified field token of a standard or custom object. If `fieldToken`is null, the error is associated with the SObject and not with a specific field.

Example

```
// Add an error to a field of an SObject instance using the addError() method.
Account acct = new Account(name = 'TestAccount');
Schema.DescribeFieldResult nameDesc = Account.name.getDescribe();
```

```
Schema.sObjectField nameField = nameDesc.getSObjectField();
acct.addError(nameField, 'error is name field');
// Use the hasErrors() method to verify that the error is added, and then the getErrors()
 method to validate the error.
System.Assert(acct.hasErrors());
List<Database.Error> errors = acct.getErrors();
System.AssertEquals(1, errors.size());
```

addError(fieldName, errorMsg, escape)

Dynamically add errors to fields of an SObject associated with the specified field name.

Signature

```
public void addError(String fieldName, String errorMsg, Boolean escape)
```

Parameters

fieldName
Type: String

The field name of the SObject .

errorMsg
Type: String

The error message to be added.

escape
Type: Boolean

Indicates whether any HTML markup in the custom error message should be escaped (`true`) or not (`false`). This parameter is ignored in both Lightning Experience and the Salesforce mobile app, and the HTML is always escaped. The escape parameter only applies in Salesforce Classic.

Return Value

Type: void

Usage

If the field name is an empty string or null, the error is associated with the SObject and not with a specific field.

The escaped characters are: `\n`, `<`, `>`, `&`, `"`, `\`, `\u2028`, `\u2029`, and `\u00a9`. As a result, HTML markup is not rendered; instead, it is displayed as text in the Salesforce user interface.

Warning:

- The *escape* parameter cannot be disabled in Lightning Experience and in the Salesforce mobile app, and will be ignored.
- Be cautious if you specify `false` for the *escape* argument. Unescaped strings displayed in the Salesforce user interface can represent a vulnerability in the system because these strings might contain harmful code. If you want to include HTML markup in the error message, call this method with a `false` *escape* argument. Make sure that you escape any dynamic content, such as input field values. Otherwise, specify `true` for the *escape* argument or call `addError(String fieldName, String errorMsg)` instead.

Example

```
// Add an error to an SObject field using the addError() method.
Account acct = new Account(name = 'TestAccount');
acct.addError('name', 'error in name field', false);
// Use the hasErrors() method to verify that the error is added, and then the getErrors()
 method to validate the error.
System.Assert(acct.hasErrors());
List<Database.Error> errors = acct.getErrors();
System.AssertEquals(1, errors.size());
```

addError(fieldToken, errorMsg, escape)

Dynamically add errors to an SObject instance associated with the specified field.

Signature

```
public void addError(Schema.SObjectField fieldToken, String errorMsg, Boolean escape)
```

Parameters

fieldToken
Type: Schema.SObjectField

The field of the SObject instance.

errorMsg
Type: String

The error message to be added.

escape
Type: Boolean

Indicates whether any HTML markup in the custom error message should be escaped (`true`) or not (`false`). This parameter is ignored in both Lightning Experience and the Salesforce mobile app, and the HTML is always escaped. The escape parameter only applies in Salesforce Classic.

Return Value

Type: void

Usage

Use this method to add errors to the specified field token of a standard or custom object. If `fieldToken`is null, the error is associated with the SObject and not with a specific field.

The escaped characters are: `\n`, `<`, `>`, `&`, `"`, `\`, `\u2028`, `\u2029`, and `\u00a9`. As a result, HTML markup is not rendered; instead, it is displayed as text in the Salesforce user interface.

Warning:

- The *escape* parameter cannot be disabled in Lightning Experience and in the Salesforce mobile app, and will be ignored.
- Be cautious if you specify `false` for the *escape* argument. Unescaped strings displayed in the Salesforce user interface can represent a vulnerability in the system because these strings might contain harmful code. If you want to include HTML markup in the error message, call this method with a `false` *escape* argument. Make sure that you escape any dynamic

content, such as input field values. Otherwise, specify `true` for the *`escape`* argument or call `addError(Schema.SObjectField fieldToken, String errorMsg)` instead.

Example

```
// Add an error to a field of an SObject instance using the addError() method.
Account acct = new Account(name = 'TestAccount');
Schema.DescribeFieldResult nameDesc = Account.name.getDescribe();
Schema.sObjectField nameField = nameDesc.getSObjectField();
acct.addError(nameField, 'error is name field', false);
// Use the hasErrors() method to verify that the error is added, and then the getErrors()
 method to validate the error.
System.Assert(acct.hasErrors());
List<Database.Error> errors = acct.getErrors();
System.AssertEquals(1, errors.size());
```

clear()

Clears all field values

Signature

```
public Void clear()
```

Return Value

Type: Void

Example

```
Account acc = new account(Name = 'Acme');
acc.clear();
Account expected = new Account();
system.assertEquals(expected, acc);
```

clone(preserveId, isDeepClone, preserveReadonlyTimestamps, preserveAutonumber)

Creates a copy of the SObject record.

Signature

```
public SObject clone(Boolean preserveId, Boolean isDeepClone, Boolean
preserveReadonlyTimestamps, Boolean preserveAutonumber)
```

Parameters

`preserveId`
Type: Boolean

(Optional) Determines whether the ID of the original object is preserved or cleared in the duplicate. If set to `true`, the ID is copied to the duplicate. The default is `false`, that is, the ID is cleared.

isDeepClone
Type: Boolean

(Optional) Determines whether the method creates a full copy of the SObject field or just a reference:

- If set to `true`, the method creates a full copy of the SObject. All fields on the SObject are duplicated in memory, including relationship fields. Consequently, if you change a field on the cloned SObject, the original SObject isn't affected.
- If set to `false`, the method performs a shallow copy of the SObject fields. All copied relationship fields reference the original SObjects. Consequently, if you change a relationship field on the cloned SObject, the corresponding field on the original SObject is also affected, and vice versa. The default is `false`.

preserveReadonlyTimestamps
Type: Boolean

(Optional) Determines whether the read-only timestamp fields are preserved or cleared in the duplicate. If set to `true`, the read-only fields `CreatedById`, `CreatedDate`, `LastModifiedById`, and `LastModifiedDate` are copied to the duplicate. The default is `false`, that is, the values are cleared.

Note: Audit field values won't be persisted to the database via DML on the cloned SObject instance.

preserveAutonumber
Type: Boolean

(Optional) Determines whether auto number fields of the original object are preserved or cleared in the duplicate. If set to `true`, auto number fields are copied to the cloned object. The default is `false`, that is, auto number fields are cleared.

Return Value

Type: SObject (of the same type)

Usage

Note: For Apex saved using Salesforce API version 22.0 or earlier, the default value for the *preserveId* argument is `true`, that is, the ID is preserved.

Example

```
Account acc = new account(Name = 'Acme', Description = 'Acme Account');
Account clonedAcc = acc.clone(false, false, false, false);
System.assertEquals(acc, clonedAcc);
```

get(fieldName)

Returns the value for the field specified by *fieldName*, such as `AccountNumber`.

Signature

```
public Object get(String fieldName)
```

Parameters

fieldName
Type: String

Return Value

Type: Object

Usage

For more information, see Dynamic SOQL.

Example

```
Account acc = new account(Name = 'Acme', Description = 'Acme Account');
String description = (String)acc.get('Description');
System.assertEquals('Acme Account', description);
```

Versioned Behavior Changes

In API version 34.0 and later, you must include the namespace name to retrieve a field from a field Map using this method. For example, to get the *account__c* field in the *MyNamespace* namespace from a *fields* field Map, use: `fields.get('MyNamespace__account__c')`.

get(field)

Returns the value for the field specified by the field token `Schema.`***`sObjectField`***, such as, `Schema.Account.AccountNumber`.

Signature

```
public Object get(Schema.sObjectField field)
```

Parameters

field
Type: Schema.SObjectField

Return Value

Type: Object

Usage

For more information, see Dynamic SOQL.

Note: Field tokens aren't available for person accounts. If you access `Schema.Account.`***`fieldname`***, you get an exception error. Instead, specify the field name as a string.

Example

```
Account acc = new account(Name = 'Acme', Description = 'Acme Account');
String description = (String)acc.get(Schema.Account.Description);
System.assertEquals('Acme Account', description);
```

getCloneSourceId()

Returns the ID of the entity from which an object was cloned. You can use it for objects cloned through the Salesforce user interface. You can also use it for objects created using the `System.SObject.clone(preserveId, isDeepClone, preserveReadonlyTimestamps, preserveAutonumber)` method, provided that the *preserveId* parameter wasn't used or was set to `false`. The `getCloneSourceId()` method can only be used within the transaction where the entity is cloned, as clone information doesn't persist in subsequent transactions.

Signature

```
public Id getCloneSourceId()
```

Return Value

Type: Id

Usage

If A is cloned to B, B is cloned to C, and C is cloned to D, then B, C, and D all point back to A as their clone source.

Example

```
Account acc0 = new Account(Name = 'Acme');
insert acc0;
Account acc1 = acc0.clone();
Account acc2 = acc1.clone();
Account acc3 = acc2.clone();
Account acc4 = acc3.clone();
System.assert(acc0.Id != null);
System.assertEquals(acc0.Id, acc1.getCloneSourceId());
System.assertEquals(acc0.Id, acc2.getCloneSourceId());
System.assertEquals(acc0.Id, acc3.getCloneSourceId());
System.assertEquals(acc0.Id, acc4.getCloneSourceId());
System.assertEquals(null, acc0.getCloneSourceId());
```

getErrors()

Returns a list of `Database.Error` objects for an SObject instance. If the SObject has no errors, an empty list is returned.

Signature

```
public List<Database.Error> getErrors()
```

Return Value

Type: List<Database.Error>

getOptions()

Returns the database.DMLOptions object for the SObject.

Signature

```
public Database.DMLOptions getOptions()
```

Return Value

Type: Database.DMLOptions

Example

```
Database.DMLOptions dmo = new Database.dmlOptions();
dmo.assignmentRuleHeader.useDefaultRule = true;

Account acc = new Account(Name = 'Acme');
acc.setOptions(dmo);
Database.DMLOptions accDmo = acc.getOptions();
```

getPopulatedFieldsAsMap()

Returns a map of populated field names and their corresponding values. The map contains only the fields that have been populated in memory for the SObject instance.

Signature

```
public Map<String,Object> getPopulatedFieldsAsMap()
```

Return Value

Type: Map<String,Object>

A map of field names and their corresponding values.

Usage

The returned map contains only the fields that have been populated in memory for the SObject instance, which makes it easy to iterate over those fields. A field is populated in memory in the following cases.

- The field has been queried by a SOQL statement.
- The field has been explicitly set before the call to the `getPopulatedFieldsAsMap()` method.

Fields on related objects that are queried or set are also returned in the map.

The following example iterates over the map returned by the `getPopulatedFieldsAsMap()` method after a SOQL query.

```
Account a = new Account();
a.name = 'TestMapAccount1';
insert a;
a = [select Id,Name from Account where id=:a.Id];
Map<String, Object> fieldsToValue = a.getPopulatedFieldsAsMap();

for (String fieldName : fieldsToValue.keySet()){
    System.debug('field name is ' + fieldName + ', value is ' +
        fieldsToValue.get(fieldName));
}
```

```
// Example debug statement output:
// DEBUG|field name is Id, value is 001R0000003EPPkIAO
// DEBUG|field name is Name, value is TestMapAccount1
```

This example iterates over the map returned by the `getPopulatedFieldsAsMap()` method after fields on the SObject are explicitly set.

```
Account a = new Account();
a.name = 'TestMapAccount2';
a.phone = '123-4567';
insert a;
Map<String, Object> fieldsToValue = a.getPopulatedFieldsAsMap();

for (String fieldName : fieldsToValue.keySet()) {
    System.debug('field name is ' + fieldName + ', value is ' +
        fieldsToValue.get(fieldName));
}

// Example debug statement output:
// DEBUG|field name is Name, value is TestMapAccount2
// DEBUG|field name is Phone, value is 123-4567
// DEBUG|field name is Id, value is 001R0000003EPPpIAO
```

The following example shows how to use the `getPopulatedFieldsAsMap()` method with related objects.

```
Account a = new Account();
a.name='TestMapAccount3';
insert a;
Contact c = new Contact();
c.firstname='TestContactFirstName';
c.lastName ='TestContactLastName';
c.accountid = a.id;
insert c;

c = [SELECT id, Contact.Firstname, Contact.Account.Name FROM Contact
        where id=:c.id limit 1];
Map<String, Object> fieldsToValue = c.getPopulatedFieldsAsMap();

// To get the fields on Account, get the Account object
// and call getMapPopulatedFieldsAsMap() on that object.

a = (Account)fieldsToValue.get('Account');
fieldsToValue = a.getPopulatedFieldsAsMap();

for (String fieldName : fieldsToValue.keySet()) {
    System.debug('field name is ' + fieldName + ', value is ' +
        fieldsToValue.get(fieldName));
}

// Example debug statement output:
// DEBUG|field name is Id, value is 001R0000003EPPuIAO
// DEBUG|field name is Name, value is TestMapAccount3
```

Versioned Behavior Changes

In API version 39.0 and later, getPopulatedFieldsAsMap returns all values set on the SObject, even if values were set after the record was queried. This behavior is dependent on the version of the apex class calling this method and not on the version of the class that generated the SObject. If you query an SObject at API version 20.0, and then call this method in a class with API version 40.0, you will get the full set of fields.

`getSObject(fieldName)`

Returns the value for the specified field. This method is primarily used with dynamic DML to access values for external IDs.

Signature

```
public SObject getSObject(String fieldName)
```

Parameters

`fieldName`
Type: String

Return Value

Type: SObject

Example

```
Account acc = new account(Name = 'Acme', Description = 'Acme Account');
insert acc;
Contact con = new Contact(Lastname = 'AcmeCon', AccountId = acc.id);
insert con;

SObject contactDB =
    [SELECT Id, AccountId, Account.Name FROM Contact WHERE id = :con.id LIMIT 1];
Account a = (Account)contactDB.getSObject('Account');
System.assertEquals('Acme', a.name);
```

`getSObject(field)`

Returns the value for the field specified by the field token `Schema.`***`sObjectField`***, such as, `Schema.MyObj.MyExternalId`. This method is primarily used with dynamic DML to access values for external IDs.

Signature

```
public SObject getSObject(Schema.SObjectField field)
```

Parameters

`field`
Type: Schema.SObjectField

Return Value

Type: SObject

Usage

If the method references polymorphic fields, a Name object is returned. Use the `TYPEOF` clause in the SOQL SELECT statement to directly get results that depend on the runtime object type referenced by the polymorphic field. See Working with Polymorphic Relationships in SOQL Queries.

Example

```
Account acc = new account(name = 'Acme', description = 'Acme Account');
insert acc;
Contact con = new contact(lastname = 'AcmeCon', accountid = acc.id);
insert con;

Schema.DescribeFieldResult fieldResult = Contact.AccountId.getDescribe();
Schema.SObjectField field = fieldResult.getSObjectField();

SObject contactDB =
    [SELECT Id, AccountId, Account.Name FROM Contact WHERE id = :con.id LIMIT 1];
Account a = (Account)contactDB.getSObject(field);
System.assertEquals('Acme', a.name);
```

`getSObjects(fieldName)`

Returns the values for the specified field. This method is primarily used with dynamic DML to access values for associated objects, such as child relationships.

Signature

```
public SObject[] getSObjects(String fieldName)
```

Parameters

`fieldName`
Type: String

Return Value

Type: SObject[]

Usage

For more information, see Dynamic DML.

Example

```
Account acc = new account(name = 'Acme', description = 'Acme Account');
insert acc;
Contact con = new contact(lastname = 'AcmeCon', accountid = acc.id);
```

```
insert con;

SObject[] a = [SELECT id, (SELECT Name FROM Contacts LIMIT 1) FROM Account WHERE id =
:acc.id];
SObject[] contactsDB = a.get(0).getSObjects('Contacts');
String fieldValue = (String)contactsDB.get(0).get('Name');
System.assertEquals('AcmeCon', fieldValue);
```

getSObjects(fieldName)

Returns the value for the field specified by the field token `Schema.`***`fieldName`***, such as, `Schema.Account.Contact`. This method is primarily used with dynamic DML to access values for associated objects, such as child relationships.

Signature

```
public SObject[] getSObjects(Schema.SObjectType fieldName)
```

Parameters

`fieldName`
Type: Schema.SObjectType

Return Value

Type: SObject[]

getSObjectType()

Returns the token for this SObject. This method is primarily used with describe information.

Signature

```
public Schema.SObjectType getSObjectType()
```

Return Value

Type: Schema.SObjectType

Usage

For more information, see apex_dynamic_describe_objects_understanding.

Example

```
Account acc = new Account(name = 'Acme', description = 'Acme Account');
Schema.SObjectType expected = Schema.Account.getSObjectType();
System.assertEquals(expected, acc.getSObjectType());
```

getQuickActionName()

Retrieves the name of a quick action associated with this SObject. Typically used in triggers.

Signature

```
public String getQuickActionName()
```

Return Value

Type: String

Example

```
trigger accTrig2 on Contact (before insert) {
    for (Contact c : Trigger.new) {
        if (c.getQuickActionName() == QuickAction.CreateContact) {
            c.WhereFrom__c = 'GlobaActionl';
        } else if (c.getQuickActionName() == Schema.Account.QuickAction.CreateContact) {
            c.WhereFrom__c = 'AccountAction';
        } else if (c.getQuickActionName() == null) {
            c.WhereFrom__c = 'NoAction';
        } else {
            System.assert(false);
        }
    }
}
```

hasErrors()

Returns true if an SObject instance has associated errors. The error message can be associated to the SObject instance by using `SObject.addError()`, validation rules, or by other means.

Signature

```
public Boolean hasErrors()
```

Return Value

Type: Boolean

isClone()

Returns `true` if an entity is cloned from something, even if the entity hasn't been saved. The method can only be used within the transaction where the entity is cloned, as clone information doesn't persist in subsequent transactions.

Signature

```
public Boolean isClone()
```

Return Value

Type: Boolean

Example

```
Account acc = new Account(Name = 'Acme');
insert acc;
Account acc2 = acc.clone();
// Test before saving
System.assertEquals(true, acc2.isClone());
insert acc2;
// Test after saving
System.assertEquals(true, acc2.isClone());
```

isSet(fieldName)

Returns information about the queried sObject field. Returns `true` if the sObject field is populated, either by direct assignment or by inclusion in a SOQL query. Returns `false` if the sObject field isn't set. If an invalid field is specified, an SObjectException is thrown.

Signature

```
public Boolean isSet(String fieldName)
```

Parameters

fieldName
Type: String

Return Value

Type: Boolean

Usage

The `isSet` method doesn't check if a field is accessible to a specific user via org permissions or other specialized access permissions.

Example

```
Contact  c = new Contact(LastName = 'Joyce');
System.assertEquals(true, c.isSet('LastName'));
System.assertEquals(false, c.isSet('FirstName')); // FirstName field is not written to
c.firstName = null;
System.assertEquals(true, c.isSet('FirstName')); //FirstName field is written to
```

isSet(field)

Returns information about the queried sObject field. Returns `true` if the sObject field is populated, either by direct assignment or by inclusion in a SOQL query. Returns `false` if the sObject field isn't set. If an invalid field is specified, an SObjectException is thrown.

Signature

```
public Boolean isSet(Schema.SObjectField field)
```

Parameters

field
Type:SObjectField Class

Return Value

Type: Boolean

Usage

The `isSet` method doesn't check if a field is accessible to a specific user via org permissions or other specialized access permissions.

Example

```
Contact  newContact = new Contact(LastName = 'Joyce');
insert(newContact); //Insert a new contact with last name Joyce
Contact c = [SELECT FirstName FROM Contact WHERE Id = :newContact.Id];
System.assertEquals(true, c.isSet(Contact.FirstName)); //FirstName field in query
System.assertEquals(false, c.isSet(Contact.LastName)); //LastName field not in query
```

put(fieldName, value)

Sets the value for the specified field and returns the previous value for the field.

Signature

```
public Object put(String fieldName, Object value)
```

Parameters

fieldName
Type: String

value
Type: Object

Return Value

Type: Object

Example

```
Account acc = new Account(name = 'test', description = 'old desc');
String oldDesc = (String)acc.put('description', 'new desc');
System.assertEquals('old desc', oldDesc);
System.assertEquals('new desc', acc.description);
```

put(field, value)

Sets the value for the field specified by the field token `Schema.`***`sObjectField`***, such as, `Schema.Account.AccountNumber` and returns the previous value for the field.

Signature

```
public Object put(Schema.SObjectField field, Object value)
```

Parameters

field
Type: Schema.SObjectField

value
Type: Object

Return Value

Type: Object

Example

```
Account acc = new Account(name = 'test', description = 'old desc');
String oldDesc = (String)acc.put(Schema.Account.Description, 'new desc');
System.assertEquals('old desc', oldDesc);
System.assertEquals('new desc', acc.description);
```

Note: Field tokens aren't available for person accounts. If you access `Schema.Account.fieldname`, you get an exception error. Instead, specify the field name as a string.

putSObject(fieldName, value)

Sets the value for the specified field. This method is primarily used with dynamic DML for setting external IDs. The method returns the previous value of the field.

Signature

```
public SObject putSObject(String fieldName, SObject value)
```

Parameters

fieldName
Type: String

value
Type: SObject

Return Value

Type: SObject

Example

```
Account acc = new Account(name = 'Acme', description = 'Acme Account');
insert acc;
Contact con = new contact(lastname = 'AcmeCon', accountid = acc.id);
```

```
insert con;
Account acc2 = new account(name = 'Not Acme');

Contact contactDB =
    (Contact)[SELECT Id, AccountId, Account.Name FROM Contact WHERE id = :con.id LIMIT 1];
Account a = (Account)contactDB.putSObject('Account', acc2);
System.assertEquals('Acme', a.name);
System.assertEquals('Not Acme', contactDB.Account.name);
```

putSObject(fieldName, value)

Sets the value for the field specified by the token `Schema.SObjectType`. This method is primarily used with dynamic DML for setting external IDs. The method returns the previous value of the field.

Signature

```
public SObject putSObject(Schema.SObjectType fieldName, SObject value)
```

Parameters

fieldName
Type: Schema.SObjectType

value
Type: SObject

Return Value

Type: SObject

recalculateFormulas()

Deprecated as of API version 57.0. Use the `recalculateFormulas()` method in the `System.Formula` class instead.

Signature

```
public Void recalculateFormulas()
```

Return Value

Type: Void

Usage

This method doesn't recalculate cross-object formulas. If you call this method on objects that have both cross-object and non-cross-object formula fields, only the non-cross-object formula fields are recalculated.

Each `recalculateFormulas` call counts against the SOQL query limits. See Execution Governors and Limits.

SEE ALSO:

- recalculateFormulas(sobjects)
- What Is a Cross-Object Formula?

`setOptions(DMLOptions)`

Sets the DMLOptions object for the SObject.

Signature

```
public Void setOptions(database.DMLOptions DMLOptions)
```

Parameters

`DMLOptions`
Type: Database.DMLOptions

Return Value

Type: Void

Example

```
Database.DMLOptions dmo = new Database.dmlOptions();
dmo.assignmentRuleHeader.useDefaultRule = true;

Account acc = new Account(Name = 'Acme');
acc.setOptions(dmo);
```

SObjectAccessDecision Class

Contains the results of a call to the Security.stripInaccessible method and methods to retrieve those results.

Namespace

System

IN THIS SECTION:

- SObjectAccessDecision Methods

SObjectAccessDecision Methods

The following are methods for `SObjectAccessDecision`.

IN THIS SECTION:

getModifiedIndexes()
Returns the indexes of sObjects that are modified by the stripInaccessible method.

getRecords()
Returns a list of new sObjects that are identical to the source records, except that they are stripped of fields that fail the field-level security check for the current user.

getRemovedFields()
Returns a map of sObject types to their corresponding inaccessible fields. The map key is a string representation of the sObject type. The map value is a set of strings, which denote the fields names that are inaccessible.

getModifiedIndexes()

Returns the indexes of sObjects that are modified by the stripInaccessible method.

Signature

```
public Set<Integer> getModifiedIndexes()
```

Return Value

Type: Set<Integer>

A set of unsigned integers that represent the row indexes of the modified sObjects.

Example

In this example, the user doesn't have permission to update the `AnnualRevenue` field of an Account.

```
List<Account> accounts = new List<Account>{
    new Account(Name='Account1', AnnualRevenue=1000),
    new Account(Name='Account2')
};

// Strip fields that are not updatable
SObjectAccessDecision decision = Security.stripInaccessible(
    AccessType.UPDATABLE,
    accounts);

// Print stripped records
for (SObject strippedAccount : decision.getRecords()) {
    System.debug(strippedAccount);
}

// Print modified indexes
System.debug(decision.getModifiedIndexes());
```

getRecords()

Returns a list of new sObjects that are identical to the source records, except that they are stripped of fields that fail the field-level security check for the current user.

Usage

The stripInaccessible method performs field-level access check for the source records in the context of the current user's operation. The `getRecords()` method returns the new records that contain only the fields that the current user has access to.

Signature

```
public List<SObject> getRecords()
```

Return Value

Type: List<SObject>

Even if the result list contains only one sObject, the return type is still a list (of size one).

Example

In this example, the user doesn't have permission to update the `AnnualRevenue` field of an Account.

```
List<Account> accounts = new List<Account>{
    new Account(Name='Account1', AnnualRevenue=1000),
    new Account(Name='Account2')
};

// Strip fields that are not updatable
SObjectAccessDecision decision = Security.stripInaccessible(
    AccessType.UPDATABLE,
    accounts);

// Print stripped records
for (SObject strippedAccount : decision.getRecords()) {
    System.debug(strippedAccount);
}
```

`getRemovedFields()`

Returns a map of sObject types to their corresponding inaccessible fields. The map key is a string representation of the sObject type. The map value is a set of strings, which denote the fields names that are inaccessible.

Signature

```
public Map<String,Set<String>> getRemovedFields()
```

Return Value

Type: Map<String,Set<String>>

Example

In this example, the user doesn't have permission to update the `AnnualRevenue` field of an Account.

```
List<Account> accounts = new List<Account>{
    new Account(Name='Account1', AnnualRevenue=1000),
    new Account(Name='Account2')
```

```
};

// Strip fields that are not updatable
SObjectAccessDecision decision = Security.stripInaccessible(
    AccessType.UPDATABLE,
    accounts);

// Print stripped records
for (SObject strippedAccount : decision.getRecords()) {
    System.debug(strippedAccount);
}
// Print removed fields
System.debug(decision.getRemovedFields());
```

SoqlStubProvider Class

Contains a method to create a mock test class for handling SOQL query responses for Data Cloud data model objects (DMOs).

Namespace

System

Usage

To create mock test classes, extend the `SoqlStubProvider` class and override the `handleSoqlQuery()` class method.

Note: SOQL `For` Loops in Apex aren't supported for SOQL stubs in static or dynamic SOQL queries against DMOs.

Example

This example shows a mock test class for the *`SkyMilesForBusinessOptInController`* class.

```
@IsTest
public class SkyMilesForBusinessOptInController_Test {

    @IsTest
    public static void mockSoql() {

        SoqlStubProvider stub = new UnifiedIndividualSoqlStub();
        Test.createSoqlStub(UnifiedIndividualSoqlStub__dlm.sObjectType, stub);

       Assert.isTrue(Test.isSoqlStubDefined(UnifiedIndividualSoqlStub__dlm.sObjectType));


        Test.startTest();
        string companyId = 'SampleCompanyId';
        // Performs SOQL query against Data Model Object
        List<SkyMilesMember> members =
SkyMilesForBusinessOptInController.getSkyMilesProfilesFromDataCloud(companyId);

        Test.stopTest();
```

```
        Assert.areEqual(1, members.size());

        SkyMilesMember member = members[0];

        Assert.areEqual(companyId, member.CompanyId);
        Assert.areEqual(5000, member.SkyMilesBalance);
    }

    class UnifiedIndividualSoqlStub extends SoqlStubProvider {
       public override List<sObject> handleSoqlQuery(sObjectType sot, string stubbedQuery,
 Map<string, object> bindVars) {

            Assert.areEqual(UnifiedIndividual__dlm.sObjectType, sot);

           // Stub assumes that the SOQL query is searching for a single record by company
 id

            string companyId = 'Default';
            if(bindVars.containsKey('tmpVar1')) {
                companyId = (string)bindVars.get('tmpVar1');
            }

            UnifiedIndividual__dlm dmo = (UnifiedIndividual__dlm)Test.createStubQueryRow(

                sot,
                new Map<string, object> {
                    'ssot__FirstName__c' => 'Codey',
                    'ssot__LastName__c' => 'Bear',
                    'ssot__Email__c' => 'developer@salesforce.com',
                    'ssot__SkyMilesBalance__c' => 5000,
                    'ssot__MedallionStatus__c' => 'Gold',
                    'ssot__CompanyId__c' => companyId
                  }
            );
            return new List<sObject> { dmo };

        }
    }
}
```

```
public with sharing class SkyMilesForBusinessOptInController {
    public static List<SkyMilesMember> getSkyMilesProfilesFromDataCloud(String companyId)
 {
        List<UnifiedIndividual__dlm> unifiedIndividuals = [
          SELECT
            Id,
            ssot__FirstName__c,
            ssot__LastName__c,
            ssot__Email__c,
            ssot__SkyMilesBalance__c,
            ssot__MedallionStatus__c,
            ssot__CompanyId__c
          FROM UnifiedIndividual__dlm
          WHERE ssot__CompanyId__c = :companyId
```

```
        ];
        List<SkyMilesMember> skyMilesMembers = new List<SkyMilesMember>();
        for (UnifiedIndividual__dlm individual : unifiedIndividuals) {
          skyMilesMembers.add(
            new SkyMilesMember(
              individual.Id,
              individual.ssot__FirstName__c,
              individual.ssot__LastName__c,
              individual.ssot__Email__c,
              individual.ssot__SkyMilesBalance__c,
              individual.ssot__MedallionStatus__c,
              individual.ssot__CompanyId__c
            )
          );
        }
        return skyMilesMembers;
    }
}
```

IN THIS SECTION:

SoqlStubProvider Methods

SoqlStubProvider Methods

The following are methods for `SoqlStubProvider`.

IN THIS SECTION:

handleSoqlQuery(targetType, stubbedQuery, bindMap)
Defines a mocked response for a SOQL query executed against the specified SObject type.

handleSoqlQuery(targetType, stubbedQuery, bindMap)

Defines a mocked response for a SOQL query executed against the specified SObject type.

Signature

```
public List<SObject> handleSoqlQuery(Schema.SObjectType targetType, String stubbedQuery,
Map<String,Object> bindMap)
```

Parameters

targetType
Type: Schema.SObjectType

The SObject type to be stubbed. This parameter can't be null.

stubbedQuery
Type: String

The SOQL query whose response is to be stubbed. Bind variables are replaced with placeholders.

bindMap
Type: Map<String,Object>

A map that contains placeholder keys for each bind variable specified in the SOQL query string and its value.

Return Value

Type: List<SObject>

The list of stubbed SObjects resulting from the SOQL query.

StaticResourceCalloutMock Class

Utility class used to specify a fake response for testing HTTP callouts.

Namespace

System

Usage

Use the methods in this class to set the response properties for testing HTTP callouts.

IN THIS SECTION:

StaticResourceCalloutMock Constructors

StaticResourceCalloutMock Methods

StaticResourceCalloutMock Constructors

The following are constructors for `StaticResourceCalloutMock`.

IN THIS SECTION:

StaticResourceCalloutMock()
Creates a new instance of the `StaticResourceCalloutMock` class.

`StaticResourceCalloutMock()`

Creates a new instance of the `StaticResourceCalloutMock` class.

Signature

```
public StaticResourceCalloutMock()
```

StaticResourceCalloutMock Methods

The following are methods for `StaticResourceCalloutMock`. All are instance methods.

IN THIS SECTION:

setHeader(headerName, headerValue)
Sets the specified header name and value for the fake response.

setStaticResource(resourceName)
Sets the specified static resource, which contains the response body.

setStatus(httpStatus)
Sets the specified HTTP status for the response.

setStatusCode(httpStatusCode)
Sets the specified HTTP status for the response.

setHeader(headerName, headerValue)

Sets the specified header name and value for the fake response.

Signature

```
public Void setHeader(String headerName, String headerValue)
```

Parameters

headerName
Type: String

headerValue
Type: String

Return Value

Type: Void

setStaticResource(resourceName)

Sets the specified static resource, which contains the response body.

Signature

```
public Void setStaticResource(String resourceName)
```

Parameters

resourceName
Type: String

Return Value

Type: Void

`setStatus(httpStatus)`

Sets the specified HTTP status for the response.

Signature

```
public Void setStatus(String httpStatus)
```

Parameters

`httpStatus`
Type: String

Return Value

Type: Void

`setStatusCode(httpStatusCode)`

Sets the specified HTTP status for the response.

Signature

```
public Void setStatusCode(Integer httpStatusCode)
```

Parameters

`httpStatusCode`
Type: Integer

Return Value

Type: Void

String Class

Contains methods for the String primitive data type.

Namespace

System

Usage

For more information on Strings, see String Data Type.

String Methods

The following are methods for `String`.

IN THIS SECTION:

abbreviate(maxWidth)
Returns an abbreviated version of the String, of the specified length and with ellipses appended if the current String is longer than the specified length; otherwise, returns the original String without ellipses.

abbreviate(maxWidth, offset)
Returns an abbreviated version of the String, starting at the specified character offset and of the specified length. The returned String has ellipses appended at the start and the end if characters have been removed at these locations.

capitalize()
Returns the current String with the first letter changed to title case.

center(size)
Returns a version of the current String of the specified size padded with spaces on the left and right, so that it appears in the center. If the specified size is smaller than the current String size, the entire String is returned without added spaces.

center(size, paddingString)
Returns a version of the current String of the specified size padded with the specified String on the left and right, so that it appears in the center. If the specified size is smaller than the current String size, the entire String is returned without padding.

charAt(index)
Returns the value of the character at the specified index.

codePointAt(index)
Returns the Unicode code point value at the specified index.

codePointBefore(index)
Returns the Unicode code point value that occurs before the specified index.

codePointCount(beginIndex, endIndex)
Returns the number of Unicode code points within the specified text range.

compareTo(secondString)
Compares two strings lexicographically, based on the Unicode value of each character in the Strings.

contains(substring)
Returns `true` if and only if the String that called the method contains the specified sequence of characters in *`substring`*.

containsAny(inputString)
Returns `true` if the current String contains any of the characters in the specified String; otherwise, returns `false`.

containsIgnoreCase(substring)
Returns `true` if the current String contains the specified sequence of characters without regard to case; otherwise, returns `false`.

containsNone(inputString)
Returns `true` if the current String doesn't contain any of the characters in the specified String; otherwise, returns `false`.

containsOnly(inputString)
Returns `true` if the current String contains characters only from the specified sequence of characters and not any other characters; otherwise, returns `false`.

containsWhitespace()
Returns `true` if the current String contains any white space characters; otherwise, returns `false`.

countMatches(substring)
Returns the number of times the specified substring occurs in the current String.

deleteWhitespace()

Returns a version of the current String with all white space characters removed.

difference(secondString)

Returns the difference between the current String and the specified String.

endsWith(suffix)

Returns `true` if the String that called the method ends with the specified *`suffix`*.

endsWithIgnoreCase(suffix)

Returns `true` if the current String ends with the specified suffix; otherwise, returns `false`.

equals(secondString)

Deprecated. This method is replaced by `equals(stringOrId)`. Returns `true` if the passed-in string is not null and represents the same binary sequence of characters as the current string. Use this method to perform case-sensitive comparisons.

equals(stringOrId)

Returns `true` if the passed-in object is not null and represents the same binary sequence of characters as the current string. Use this method to compare a string to an object that represents a string or an ID.

equalsIgnoreCase(secondString)

Returns `true` if the *`secondString`* isn't null and represents the same sequence of characters as the String that called the method, ignoring case.

escapeCsv()

Returns a String for a CSV column enclosed in double quotes, if required.

escapeEcmaScript()

Escapes the characters in the String using EcmaScript String rules.

escapeHtml3()

Escapes the characters in a String using HTML 3.0 entities.

escapeHtml4()

Escapes the characters in a String using HTML 4.0 entities.

escapeJava()

Returns a String whose characters are escaped using Java String rules. Characters escaped include quotes and control characters, such as tab, backslash, and carriage return characters.

escapeSingleQuotes(stringToEscape)

Returns a String with the escape character (\) added before any single quotation marks in the String *`s`*.

escapeUnicode()

Returns a String whose Unicode characters are escaped to a Unicode escape sequence.

escapeXml()

Escapes the characters in a String using XML entities.

format(stringToFormat, formattingArguments)

Treat the first argument as a pattern and return a string using the second argument for substitution and formatting. The substitution and formatting are the same as `apex:outputText` and the Java `MessageFormat` class. Non-string types in the second argument's List are implicitly converted to strings, respecting the toString() method overrides that exist on the type.

fromCharArray(charArray)

Returns a String from the values of the list of integers.

getChars()

Returns an array of character values that represent the characters in this string.

getCommonPrefix(strings)

Returns the initial sequence of characters as a String that is common to all the specified Strings.

getLevenshteinDistance(stringToCompare)

Returns the Levenshtein distance between the current String and the specified String.

getLevenshteinDistance(stringToCompare, threshold)

Returns the Levenshtein distance between the current String and the specified String if it is less than or equal than the given threshold; otherwise, returns -1.

hashCode()

Returns a hash code value for this string.

indexOf(substring)

Returns the index of the first occurrence of the specified substring. If the substring does not occur, this method returns -1.

indexOf(substring, index)

Returns the zero-based index of the first occurrence of the specified substring from the point of the given index. If the substring does not occur, this method returns -1.

indexOfAny(substring)

Returns the zero-based index of the first occurrence of any character specified in the substring. If none of the characters occur, returns -1.

indexOfAnyBut(substring)

Returns the zero-based index of the first occurrence of a character that is not in the specified substring. Otherwise, returns -1.

indexOfChar(character)

Returns the index of the first occurrence of the character that corresponds to the specified character value.

indexOfChar(character, startIndex)

Returns the index of the first occurrence of the character that corresponds to the specified character value, starting from the specified index.

indexOfDifference(stringToCompare)

Returns the zero-based index of the character where the current String begins to differ from the specified String.

indexOfIgnoreCase(substring)

Returns the zero-based index of the first occurrence of the specified substring without regard to case. If the substring does not occur, this method returns -1.

indexOfIgnoreCase(substring, startPosition)

Returns the zero-based index of the first occurrence of the specified substring from the point of index `i`, without regard to case. If the substring does not occur, this method returns -1.

isAllLowerCase()

Returns `true` if all characters in the current String are lowercase; otherwise, returns `false`.

isAllUpperCase()

Returns `true` if all characters in the current String are uppercase; otherwise, returns `false`.

isAlpha()

Returns `true` if all characters in the current String are Unicode letters only; otherwise, returns `false`.

isAlphaSpace()

Returns `true` if all characters in the current String are Unicode letters or spaces only; otherwise, returns `false`.

isAlphanumeric()

Returns `true` if all characters in the current String are Unicode letters or numbers only; otherwise, returns `false`.

isAlphanumericSpace()

Returns `true` if all characters in the current String are Unicode letters, numbers, or spaces only; otherwise, returns `false`.

isAsciiPrintable()

Returns `true` if the current String contains only ASCII printable characters; otherwise, returns `false`.

isBlank(inputString)

Returns `true` if the specified String is white space, empty (''), or null; otherwise, returns `false`.

isEmpty(inputString)

Returns `true` if the specified String is empty ('') or null; otherwise, returns `false`.

isNotBlank(inputString)

Returns `true` if the specified String is not whitespace, not empty (''), and not null; otherwise, returns `false`.

isNotEmpty(inputString)

Returns `true` if the specified String is not empty ('') and not null; otherwise, returns `false`.

isNumeric()

Returns `true` if the current String contains only Unicode digits; otherwise, returns `false`.

isNumericSpace()

Returns `true` if the current String contains only Unicode digits or spaces; otherwise, returns `false`.

isWhitespace()

Returns `true` if the current String contains only white space characters or is empty; otherwise, returns `false`.

join(iterableObj, separator)

Joins the elements of the specified iterable object, such as a List, into a single String separated by the specified separator.

lastIndexOf(substring)

Returns the index of the last occurrence of the specified substring. If the substring does not occur, this method returns -1.

lastIndexOf(substring, endPosition)

Returns the index of the last occurrence of the specified substring, starting from the character at index 0 and ending at the specified index.

lastIndexOfChar(character)

Returns the index of the last occurrence of the character that corresponds to the specified character value.

lastIndexOfChar(character, endIndex)

Returns the index of the last occurrence of the character that corresponds to the specified character value, starting from the specified index.

lastIndexOfIgnoreCase(substring)

Returns the index of the last occurrence of the specified substring regardless of case.

lastIndexOfIgnoreCase(substring, endPosition)

Returns the index of the last occurrence of the specified substring regardless of case, starting from the character at index 0 and ending at the specified index.

left(length)

Returns the leftmost characters of the current String of the specified length.

leftPad(length)

Returns the current String padded with spaces on the left and of the specified length.

leftPad(length, padStr)

Returns the current String padded with String `padStr` on the left and of the specified length.

length()

Returns the number of 16-bit Unicode characters contained in the String.

mid(startIndex, length)

Returns a new String that begins with the character at the specified zero-based *`startIndex`* with the number of characters specified by *`length`*.

normalizeSpace()

Returns the current String with leading, trailing, and repeating white space characters removed.

offsetByCodePoints(index, codePointOffset)

Returns the index of the Unicode code point that is offset by the specified number of code points, starting from the given index.

remove(substring)

Removes all occurrences of the specified substring and returns the String result.

removeEnd(substring)

Removes the specified substring only if it occurs at the end of the String.

removeEndIgnoreCase(substring)

Removes the specified substring only if it occurs at the end of the String using a case-insensitive match.

removeStart(substring)

Removes the specified substring only if it occurs at the beginning of the String.

removeStartIgnoreCase(substring)

Removes the specified substring only if it occurs at the beginning of the String using a case-insensitive match.

repeat(numberOfTimes)

Returns the current String repeated the specified number of times.

repeat(separator, numberOfTimes)

Returns the current String repeated the specified number of times using the specified separator to separate the repeated Strings.

replace(target, replacement)

Replaces each substring of a string that matches the literal target sequence *`target`* with the specified literal replacement sequence *`replacement`*.

replaceAll(regExp, replacement)

Replaces each substring of a string that matches the regular expression *`regExp`* with the replacement sequence *`replacement`*.

replaceFirst(regExp, replacement)

Replaces the first substring of a string that matches the regular expression *`regExp`* with the replacement sequence *`replacement`*.

reverse()

Returns a String with all the characters reversed.

right(length)

Returns the rightmost characters of the current String of the specified length.

rightPad(length)

Returns the current String padded with spaces on the right and of the specified length.

rightPad(length, padStr)

Returns the current String padded with String `padStr` on the right and of the specified length.

split(regExp)

Returns a list that contains each substring of the String that is terminated by either the regular expression *`regExp`* or the end of the String.

split(regExp, limit)

Returns a list that contains each substring of the String that is terminated by either the regular expression *`regExp`* or the end of the String.

splitByCharacterType()

Splits the current String by character type and returns a list of contiguous character groups of the same type as complete tokens.

splitByCharacterTypeCamelCase()

Splits the current String by character type and returns a list of contiguous character groups of the same type as complete tokens, with the following exception: the uppercase character, if any, immediately preceding a lowercase character token belongs to the following character token rather than to the preceding.

startsWith(prefix)

Returns `true` if the String that called the method begins with the specified *`prefix`*.

startsWithIgnoreCase(prefix)

Returns `true` if the current String begins with the specified prefix regardless of the prefix case.

stripHtmlTags()

Removes HTML markup and returns plain text.

substring(startIndex)

Returns a new String that begins with the character at the specified zero-based *`startIndex`* and extends to the end of the String.

substring(startIndex, endIndex)

Returns a new String that begins with the character at the specified zero-based *`startIndex`* and extends to the character at *`endIndex`* - 1.

substringAfter(separator)

Returns the substring that occurs after the first occurrence of the specified separator.

substringAfterLast(separator)

Returns the substring that occurs after the last occurrence of the specified separator.

substringBefore(separator)

Returns the substring that occurs before the first occurrence of the specified separator.

substringBeforeLast(separator)

Returns the substring that occurs before the last occurrence of the specified separator.

substringBetween(tag)

Returns the substring that occurs between two instances of the specified *`tag`* String.

substringBetween(open, close)

Returns the substring that occurs between the two specified Strings.

swapCase()

Swaps the case of all characters and returns the resulting String by using the default (English US) locale.

toLowerCase()

Converts all of the characters in the String to lowercase using the rules of the default (English US) locale.

toLowerCase(locale)

Converts all of the characters in the String to lowercase using the rules of the specified locale.

toUpperCase()

Converts all of the characters in the String to uppercase using the rules of the default (English US) locale.

toUpperCase(locale)

Converts all of the characters in the String to the uppercase using the rules of the specified locale.

trim()

Returns a copy of the string that no longer contains any leading or trailing white space characters.

uncapitalize()

Returns the current String with the first letter in lowercase.

unescapeCsv()

Returns a String representing an unescaped CSV column.

unescapeEcmaScript()

Unescapes any EcmaScript literals found in the String.

unescapeHtml3()

Unescapes the characters in a String using HTML 3.0 entities.

unescapeHtml4()

Unescapes the characters in a String using HTML 4.0 entities.

unescapeJava()

Returns a String whose Java literals are unescaped. Literals unescaped include escape sequences for quotes (\\") and control characters, such as tab (\\t), and carriage return (\\n).

unescapeUnicode()

Returns a String whose escaped Unicode characters are unescaped.

unescapeXml()

Unescapes the characters in a String using XML entities.

valueOf(dateToConvert)

Returns a String that represents the specified Date in the standard "yyyy-MM-dd" format.

valueOf(datetimeToConvert)

Returns a String that represents the specified Datetime in the standard "yyyy-MM-dd HH:mm:ss" format for the local time zone.

valueOf(decimalToConvert)

Returns a String that represents the specified Decimal.

valueOf(doubleToConvert)

Returns a String that represents the specified Double.

valueOf(integerToConvert)

Returns a String that represents the specified Integer.

valueOf(longToConvert)

Returns a String that represents the specified Long.

valueOf(toConvert)

Returns a string representation of the specified object argument.

valueOfGmt(datetimeToConvert)

Returns a String that represents the specified Datetime in the standard "yyyy-MM-dd HH:mm:ss" format for the GMT time zone.

abbreviate(maxWidth)

Returns an abbreviated version of the String, of the specified length and with ellipses appended if the current String is longer than the specified length; otherwise, returns the original String without ellipses.

Signature

```
public String abbreviate(Integer maxWidth)
```

Parameters

maxWidth
Type: Integer

If *maxWidth* is less than four, this method throws a run-time exception.

Return Value

Type: String

Example

```
String s = 'Hello Maximillian';
String s2 = s.abbreviate(8);
System.assertEquals('Hello...', s2);
System.assertEquals(8, s2.length());
```

abbreviate(maxWidth, offset)

Returns an abbreviated version of the String, starting at the specified character offset and of the specified length. The returned String has ellipses appended at the start and the end if characters have been removed at these locations.

Signature

```
public String abbreviate(Integer maxWidth, Integer offset)
```

Parameters

maxWidth
Type: Integer

Note that the offset is not necessarily the leftmost character in the returned String or the first character following the ellipses, but it appears somewhere in the result. Regardless, `abbreviate` won't return a String of length greater than *maxWidth*.If *maxWidth* is too small, this method throws a run-time exception.

offset
Type: Integer

Return Value

Type: String

Example

```
String s = 'Hello Maximillian';
// Start at M
String s2 = s.abbreviate(9,6);
System.assertEquals('...Max...', s2);
System.assertEquals(9, s2.length());
```

capitalize()

Returns the current String with the first letter changed to title case.

Signature

```
public String capitalize()
```

Return Value

Type: String

Usage

This method is based on the `Character.toTitleCase(char)` Java method.

Example

```
String s = 'hello maximillian';
String s2 = s.capitalize();
System.assertEquals('Hello maximillian', s2);
```

center(size)

Returns a version of the current String of the specified size padded with spaces on the left and right, so that it appears in the center. If the specified size is smaller than the current String size, the entire String is returned without added spaces.

Signature

```
public String center(Integer size)
```

Parameters

size
Type: Integer

Return Value

Type: String

Example

```
String s = 'hello';
String s2 = s.center(9);
System.assertEquals(
   '  hello  ',
   s2);
```

center(size, paddingString)

Returns a version of the current String of the specified size padded with the specified String on the left and right, so that it appears in the center. If the specified size is smaller than the current String size, the entire String is returned without padding.

Signature

```
public String center(Integer size, String paddingString)
```

Parameters

size
Type: Integer

paddingString
Type: String

Return Value

Type: String

Example

```
String s = 'hello';
String s2 = s.center(9, '-');
System.assertEquals('--hello--', s2);
```

charAt(index)

Returns the value of the character at the specified index.

Signature

```
public Integer charAt(Integer index)
```

Parameters

index
Type: Integer

The index of the character to get the value of.

Return Value

Type: Integer

The integer value of the character.

Usage

The `charAt` method returns the value of the character pointed to by the specified index. If the index points to the beginning of a surrogate pair (the high-surrogate code point), this method returns only the high-surrogate code point. To return the supplementary code point corresponding to a surrogate pair, call `codePointAt` instead.

Example

This example gets the value of the first character at index 0.

```
String str = 'Ω is Omega.';
System.assertEquals(937, str.charAt(0));
```

This example shows the difference between `charAt` and `codePointAt`. The example calls these methods on escaped supplementary Unicode characters. `charAt(0)` returns the high surrogate value, which corresponds to `\uD835`. `codePointAt(0)` returns the value for the entire surrogate pair.

```
String str = '\uD835\uDD0A';
System.assertEquals(55349, str.charAt(0),
    'charAt(0) didn\'t return the high surrogate.');
System.assertEquals(120074, str.codePointAt(0),
    'codePointAt(0) didn\'t return the entire two-character supplementary value.');
```

codePointAt(index)

Returns the Unicode code point value at the specified index.

Signature

```
public Integer codePointAt(Integer index)
```

Parameters

index

Type: Integer

The index of the characters (Unicode code units) in the string. The index range is from zero to the string length minus one.

Return Value

Type: Integer

The Unicode code point value at the specified index.

Usage

If the *index* points to the beginning of a surrogate pair (the high-surrogate code point), and the character value at the following index points to the low-surrogate code point, this method returns the supplementary code point corresponding to this surrogate pair. Otherwise, this method returns the character value at the given index.

For more information on Unicode and surrogate pairs, see The Unicode Consortium.

Example

This example gets the code point value of the first character at index 0, which is the escaped Omega character. Also, the example gets the code point at index 20, which corresponds to the escaped supplementary Unicode characters (a pair of characters). Finally, it verifies that the escaped and unescaped forms of Omega have the same code point values.

The supplementary characters in this example (`\\uD835\\uDD0A`) correspond to mathematical fraktur capital G: 𝔊

```
String str = '\u03A9 is Ω (Omega), and \uD835\uDD0A ' +
    ' is Fraktur Capital G.';
System.assertEquals(937, str.codePointAt(0));
System.assertEquals(120074, str.codePointAt(20));
// Escaped or unescaped forms of the same character have the same code point
System.assertEquals(str.codePointAt(0), str.codePointAt(5));
```

codePointBefore(index)

Returns the Unicode code point value that occurs before the specified index.

Signature

```
public Integer codePointBefore(Integer index)
```

Parameters

index
Type: Integer

The index before the Unicode code point that is to be returned. The index range is from one to the string length.

Return Value

Type: Integer

The character or Unicode code point value that occurs before the specified index.

Usage

If the character value at ***index***`-1` is the low-surrogate code point, and ***index***`-2` is not negative and the character at this index location is the high-surrogate code point, this method returns the supplementary code point corresponding to this surrogate pair. If the character value at ***index***`-1` is an unpaired low-surrogate or high-surrogate code point, the surrogate value is returned.

For more information on Unicode and surrogate pairs, see The Unicode Consortium.

Example

This example gets the code point value of the first character (before index 1), which is the escaped Omega character. Also, the example gets the code point at index 20, which corresponds to the escaped supplementary characters (the two characters before index 22).

```
String str = '\u03A9 is Ω (Omega), and \uD835\uDD0A ' +
    ' is Fraktur Capital G.';
System.assertEquals(937, str.codePointBefore(1));
System.assertEquals(120074, str.codePointBefore(22));
```

codePointCount(beginIndex, endIndex)

Returns the number of Unicode code points within the specified text range.

Signature

```
public Integer codePointCount(Integer beginIndex, Integer endIndex)
```

Parameters

beginIndex
Type: Integer

The index of the first character in the range.

endIndex
Type: Integer

The index after the last character in the range.

Return Value

Type: Integer

The number of Unicode code points within the specified range.

Usage

The specified range begins at *beginIndex* and ends at ***endIndex***—1. Unpaired surrogates within the text range count as one code point each.

Example

This example writes the count of code points in a substring that contains an escaped Unicode character and another substring that contains Unicode supplementary characters, which count as one code point.

```
String str = '\u03A9 and \uD835\uDD0A characters.';
System.debug('Count of code points for ' + str.substring(0,1)
             + ': ' + str.codePointCount(0,1));
System.debug('Count of code points for ' + str.substring(6,8)
             + ': ' + str.codePointCount(6,8));

// Output:
// Count of code points for Ω: 1
// Count of code points for □□: 1
```

compareTo(secondString)

Compares two strings lexicographically, based on the Unicode value of each character in the Strings.

Signature

```
public Integer compareTo(String secondString)
```

Parameters

secondString
Type: String

Return Value

Type: Integer

Usage

The result is:

- A negative Integer if the String that called the method lexicographically precedes *secondString*
- A positive Integer if the String that called the method lexicographically follows *compsecondStringString*
- Zero if the Strings are equal

If there is no index position at which the Strings differ, then the shorter String lexicographically precedes the longer String.

Note that this method returns 0 whenever the `equals` method returns true.

Example

```
String myString1 = 'abcde';
String myString2 = 'abcd';
Integer result =
   myString1.compareTo(myString2);
System.assertEquals(result, 1);
```

contains(substring)

Returns `true` if and only if the String that called the method contains the specified sequence of characters in *substring*.

Signature

```
public Boolean contains(String substring)
```

Parameters

substring
Type: String

Return Value

Type: Boolean

Example

```
String myString1 = 'abcde';
String myString2 = 'abcd';
Boolean result =
   myString1.contains(myString2);
System.assertEquals(result, true);
```

containsAny(inputString)

Returns `true` if the current String contains any of the characters in the specified String; otherwise, returns `false`.

Signature

```
public Boolean containsAny(String inputString)
```

Parameters

inputString
Type: String

Return Value

Type: Boolean

Example

```
String s = 'hello';
Boolean b1 = s.containsAny('hx');
Boolean b2 = s.containsAny('x');
System.assertEquals(true, b1);
System.assertEquals(false, b2);
```

containsIgnoreCase(substring)

Returns `true` if the current String contains the specified sequence of characters without regard to case; otherwise, returns `false`.

Signature

```
public Boolean containsIgnoreCase(String substring)
```

Parameters

substring
Type: String

Return Value

Type: Boolean

Example

```
String s = 'hello';
Boolean b = s.containsIgnoreCase('HE');
System.assertEquals(
   true,
   b);
```

containsNone(inputString)

Returns `true` if the current String doesn't contain any of the characters in the specified String; otherwise, returns `false`.

Signature

```
public Boolean containsNone(String inputString)
```

Parameters

inputString
Type: String

If *inputString* is an empty string or the current String is empty, this method returns `true`. If *inputString* is null, this method returns a run-time exception.

Return Value

Type: Boolean

Example

```
String s1 = 'abcde';
System.assert(s1.containsNone('fg'));
```

containsOnly(inputString)

Returns `true` if the current String contains characters only from the specified sequence of characters and not any other characters; otherwise, returns `false`.

Signature

```
public Boolean containsOnly(String inputString)
```

Parameters

inputString
Type: String

Return Value

Type: Boolean

Example

```
String s1 = 'abba';
String s2 = 'abba xyz';
Boolean b1 =
   s1.containsOnly('abcd');
System.assertEquals(
   true,
   b1);
Boolean b2 =
   s2.containsOnly('abcd');
System.assertEquals(
   false,
   b2);
```

containsWhitespace()

Returns `true` if the current String contains any white space characters; otherwise, returns `false`.

Signature

```
public Boolean containsWhitespace()
```

Return Value

Type: Boolean

Example

```
String s = 'Hello Jane';
System.assert(s.containsWhitespace()); //true
s = 'HelloJane ';
System.assert(s.containsWhitespace()); //true
s = ' HelloJane';
System.assert(s.containsWhitespace()); //true
s = 'HelloJane';
System.assert(!s.containsWhitespace()); //false
```

countMatches(substring)

Returns the number of times the specified substring occurs in the current String.

Signature

```
public Integer countMatches(String substring)
```

Parameters

substring
Type: String

Return Value

Type: Integer

Example

```
String s = 'Hello Jane';
System.assertEquals(1, s.countMatches('Hello'));
s = 'Hello Hello';
System.assertEquals(2, s.countMatches('Hello'));
s = 'Hello hello';
System.assertEquals(1, s.countMatches('Hello'));
```

deleteWhitespace()

Returns a version of the current String with all white space characters removed.

Signature

```
public String deleteWhitespace()
```

Return Value

Type: String

Example

```
String s1 = ' Hello Jane ';
String s2 = 'HelloJane';
System.assertEquals(s2, s1.deleteWhitespace());
```

difference(secondString)

Returns the difference between the current String and the specified String.

Signature

```
public String difference(String secondString)
```

Parameters

secondString

Type: String

If *secondString* is an empty string, this method returns an empty string.If *secondString* is null, this method throws a run-time exception.

Return Value

Type: String

Example

```
String s = 'Hello Jane';
String d1 =
   s.difference('Hello Max');
System.assertEquals(
   'Max',
   d1);
String d2 =
   s.difference('Goodbye');
System.assertEquals(
   'Goodbye',
   d2);
```

endsWith(suffix)

Returns `true` if the String that called the method ends with the specified *`suffix`*.

Signature

```
public Boolean endsWith(String suffix)
```

Parameters

`suffix`
Type: String

Return Value

Type: Boolean

Example

```
String s = 'Hello Jason';
System.assert(s.endsWith('Jason'));
```

endsWithIgnoreCase(suffix)

Returns `true` if the current String ends with the specified suffix; otherwise, returns `false`.

Signature

```
public Boolean endsWithIgnoreCase(String suffix)
```

Parameters

`suffix`
Type: String

Return Value

Type: Boolean

Example

```
String s = 'Hello Jason';
System.assert(s.endsWithIgnoreCase('jason'));
```

equals(secondString)

Deprecated. This method is replaced by `equals(stringOrId)`. Returns `true` if the passed-in string is not null and represents the same binary sequence of characters as the current string. Use this method to perform case-sensitive comparisons.

Signature

```
public Boolean equals(String secondString)
```

Parameters

secondString
Type: String

Return Value

Type: Boolean

Usage

This method returns `true` when the `compareTo` method returns 0.

Use this method to perform case-sensitive comparisons. In contrast, the `==` operator performs case-insensitive string comparisons to match Apex semantics.

Example

```
String myString1 = 'abcde';
String myString2 = 'abcd';
Boolean result = myString1.equals(myString2);
System.assertEquals(result, false);
```

equals(stringOrId)

Returns `true` if the passed-in object is not null and represents the same binary sequence of characters as the current string. Use this method to compare a string to an object that represents a string or an ID.

Signature

```
public Boolean equals(Object stringOrId)
```

Parameters

stringOrId
Type: Object

Return Value

Type: Boolean

Usage

If you compare ID values, the lengths of IDs don't need to be equal. For example, if you compare a 15-character ID string to an object that represents the equivalent 18-character ID value, this method returns `true`. For more information about 15-character and 18-character IDs, see the ID Data Type.

Use this method to perform case-sensitive comparisons. In contrast, the `==` operator performs case-insensitive string comparisons to match Apex semantics.

Example

These examples show comparisons between different types of variables with both equal and unequal values. The examples also show how Apex automatically converts certain values before comparing them.

```
// Compare a string to an object containing a string
Object obj1 = 'abc';
String str = 'abc';
Boolean result1 = str.equals(obj1);
System.assertEquals(true, result1);

// Compare a string to an object containing a number
Integer obj2 = 100;
Boolean result2 = str.equals(obj2);
System.assertEquals(false, result2);

// Compare a string to an ID of the same length.
// 15-character ID
Id idValue15 = '001D000000Ju1zH';
// 15-character ID string value
String stringValue15 = '001D000000Ju1zH';
Boolean result3 = stringValue15.equals(IdValue15);
System.assertEquals(true, result3);

// Compare two equal ID values of different lengths:
//  15-character ID and 18-character ID
Id idValue18 = '001D000000Ju1zHIAR';
Boolean result4 = stringValue15.equals(IdValue18);
System.assertEquals(true, result4);
```

equalsIgnoreCase(secondString)

Returns `true` if the *secondString* isn't null and represents the same sequence of characters as the String that called the method, ignoring case.

Signature

```
public Boolean equalsIgnoreCase(String secondString)
```

Parameters

secondString
Type: String

Return Value

Type: Boolean

Usage

The `String.equalsIgnoreCase()` method ignores the locale of the context user. If you want the string comparison to be performed according to the locale, use the `==` operator instead. The `String.equalsIgnoreCase()` method typically executes faster than the operator because the method ignores the locale.

Example

```
String myString1 = 'abcd';
String myString2 = 'ABCD';
Boolean result =
myString1.equalsIgnoreCase(myString2);
System.assertEquals(result, true);
```

escapeCsv()

Returns a String for a CSV column enclosed in double quotes, if required.

Signature

```
public String escapeCsv()
```

Return Value

Type: String

Usage

If the String contains a comma, newline or double quote, the returned String is enclosed in double quotes. Also, any double quote characters in the String are escaped with another double quote.

If the String doesn't contain a comma, newline or double quote, it is returned unchanged.

Example

```
String s1 = 'Max1, "Max2"';
String s2 = s1.escapeCsv();
System.assertEquals('"Max1, ""Max2"""', s2);
```

escapeEcmaScript()

Escapes the characters in the String using EcmaScript String rules.

Signature

```
public String escapeEcmaScript()
```

Return Value

Type: String

Usage

The only difference between Apex strings and EcmaScript strings is that in EcmaScript, a single quote and forward-slash (/) are escaped.

Example

```
String s1 = '"grade": 3.9/4.0';
String s2 = s1.escapeEcmaScript();
System.debug(s2);
// Output is:
// \"grade\": 3.9\/4.0
System.assertEquals(
   '\\"grade\\": 3.9\\/4.0',
    s2);
```

escapeHtml3()

Escapes the characters in a String using HTML 3.0 entities.

Signature

```
public String escapeHtml3()
```

Return Value

Type: String

Example

```
String s1 =
   '"<Black&White>"';
String s2 =
   s1.escapeHtml3();
System.debug(s2);
// Output:
// &quot;&lt;Black&amp;
// White&gt;&quot;
```

escapeHtml4()

Escapes the characters in a String using HTML 4.0 entities.

Signature

```
public String escapeHtml4()
```

Return Value

Type: String

Example

```
String s1 =
   '"<Black&White>"';
String s2 =
   s1.escapeHtml4();
System.debug(s2);
// Output:
// &quot;&lt;Black&amp;
// White&gt;&quot;
```

escapeJava()

Returns a String whose characters are escaped using Java String rules. Characters escaped include quotes and control characters, such as tab, backslash, and carriage return characters.

Signature

```
public String escapeJava()
```

Return Value

Type: String

The escaped string.

Example

```
// Input string contains quotation marks
String s = 'Company: "Salesforce.com"';
String escapedStr = s.escapeJava();
// Output string has the quotes escaped
System.assertEquals('Company: \\"Salesforce.com\\"', escapedStr);
```

escapeSingleQuotes(stringToEscape)

Returns a String with the escape character (\) added before any single quotation marks in the String *s*.

Signature

```
public static String escapeSingleQuotes(String stringToEscape)
```

Parameters

stringToEscape
Type: String

Return Value

Type: String

Usage

This method is useful when creating a dynamic SOQL statement, to help prevent SOQL injection. For more information on dynamic SOQL, see Dynamic SOQL.

Example

```
String s = '\'Hello Jason\'';
system.debug(s); // Outputs 'Hello Jason'
String escapedStr = String.escapeSingleQuotes(s);
system.debug(escapedStr); // Outputs \'Hello Jason\'

// In the first string, \ and ' are escaped with a preceding \, so it is equal to escapedStr

system.assertEquals('\\\'Hello Jason\\\'', escapedStr);
```

escapeUnicode()

Returns a String whose Unicode characters are escaped to a Unicode escape sequence.

Signature

```
public String escapeUnicode()
```

Return Value

Type: String

The escaped string.

Example

```
String s = 'De onde você é?';
String escapedStr = s.escapeUnicode();
System.assertEquals('De onde voc\\u00EA \\u00E9?', escapedStr);
```

escapeXml()

Escapes the characters in a String using XML entities.

Signature

```
public String escapeXml()
```

Return Value

Type: String

Usage

Supports only the five basic XML entities (gt, lt, quot, amp, apos). Does not support DTDs or external entities. Unicode characters greater than 0x7f are not escaped.

Example

```
String s1 =
   '"<Black&White>"';
String s2 =
   s1.escapeXml();
System.debug(s2);
// Output:
// &quot;&lt;Black&amp;
// White&gt;&quot;
```

format(stringToFormat, formattingArguments)

Treat the first argument as a pattern and return a string using the second argument for substitution and formatting. The substitution and formatting are the same as `apex:outputText` and the Java `MessageFormat` class. Non-string types in the second argument's List are implicitly converted to strings, respecting the toString() method overrides that exist on the type.

Signature

```
public static String format(String stringToFormat, List<Object> formattingArguments)
```

Parameters

stringToFormat
Type: String

formattingArguments
Type: List<Object>

Return Value

Type: String

Versioned Behavior Changes

From version 51.0 and later, the `format()` method supports single quotes in the `stringToFormat` parameter and returns a formatted string using the `formattingArguments` parameter. In version 50.0 and earlier, single quotes weren't supported.

Example

```
String template = '{0} was last updated {1}';
List<Object> parameters = new List<Object> {'Universal Containers',
DateTime.newInstance(2018, 11, 15) };
String formatted = String.format(template, parameters);
System.debug ('Newly formatted string is:' + formatted);
```

fromCharArray(charArray)

Returns a String from the values of the list of integers.

Signature

```
public static String fromCharArray(List<Integer> charArray)
```

Parameters

charArray
Type: List<Integer>

Return Value

Type: String

Example

```
List<Integer> charArr= new Integer[]{74};
String convertedChar = String.fromCharArray(charArr);
System.assertEquals('J', convertedChar);
```

getChars()

Returns an array of character values that represent the characters in this string.

Signature

```
public List<Integer> getChars()
```

Return Value

Type: List<Integer>

A list of integers, each corresponding to a character value in the string.

Example

This sample converts a string to a character array and then gets the first array element, which corresponds to the value of 'J'.

```
String str = 'Jane goes fishing.';
Integer[] chars = str.getChars();
```

```
// Get the value of 'J'
System.assertEquals(74, chars[0]);
```

Usage

If a "/" (slash) character is present in the string, `String.getChars()` unescapes it in the returned character array. This example uses the `String.escapeJava()` method to generate the desired value of "\\" in the returned string.

```
String doubleSlash = '\\' + '\\'; //doubleSlash is set to "\\"
System.debug(String.fromCharArray(doubleSlash.getChars()));  //Returns "\"
System.debug(String.fromCharArray(doubleSlash.escapeJava().getChars())); //Returns "\\"
```

getCommonPrefix(strings)

Returns the initial sequence of characters as a String that is common to all the specified Strings.

Signature

```
public static String getCommonPrefix(List<String> strings)
```

Parameters

strings
Type: List<String>

Return Value

Type: String

Example

```
List<String> ls = new List<String>{'SFDCApex', 'SFDCVisualforce'};
String prefix = String.getCommonPrefix(ls);
System.assertEquals('SFDC', prefix);
```

getLevenshteinDistance(stringToCompare)

Returns the Levenshtein distance between the current String and the specified String.

Signature

```
public Integer getLevenshteinDistance(String stringToCompare)
```

Parameters

stringToCompare
Type: String

Return Value

Type: Integer

Usage

The Levenshtein distance is the number of changes needed to change one String into another. Each change is a single character modification (deletion, insertion or substitution).

Example

```
String s = 'Hello Joe';
Integer i = s.getLevenshteinDistance('Hello Max');
System.assertEquals(3, i);
```

getLevenshteinDistance(stringToCompare, threshold)

Returns the Levenshtein distance between the current String and the specified String if it is less than or equal than the given threshold; otherwise, returns -1.

Signature

```
public Integer getLevenshteinDistance(String stringToCompare, Integer threshold)
```

Parameters

stringToCompare
Type: String

threshold
Type: Integer

Return Value

Type: Integer

Usage

The Levenshtein distance is the number of changes needed to change one String into another. Each change is a single character modification (deletion, insertion or substitution).

Example:

In this example, the Levenshtein distance is 3, but the threshold argument is 2, which is less than the distance, so this method returns -1.

Example

```
String s = 'Hello Jane';
Integer i = s.getLevenshteinDistance('Hello Max', 2);
System.assertEquals(-1, i);
```

hashCode()

Returns a hash code value for this string.

Signature

```
public Integer hashCode()
```

Return Value

Type: Integer

Usage

This value is based on the hash code computed by the Java `String.hashCode` counterpart method.

You can use this method to simplify the computation of a hash code for a custom type that contains String member variables. You can compute your type's hash code value based on the hash code of each String variable. For example:

For more details about the use of hash code methods with custom types, see Using Custom Types in Map Keys and Sets.

Example

```
public class MyCustomClass {
   String x,y;
   // Provide a custom hash code
   public Integer hashCode() {
    return
    (31*x.hashCode())^(y.hashCode());
   }
}
```

indexOf(substring)

Returns the index of the first occurrence of the specified substring. If the substring does not occur, this method returns -1.

Signature

```
public Integer indexOf(String substring)
```

Parameters

substring
Type: String

Return Value

Type: Integer

Example

```
String myString1 = 'abcde';
String myString2 = 'cd';
Integer result = myString1.indexOf(mystring2);
System.assertEquals(2, result);
```

indexOf(substring, index)

Returns the zero-based index of the first occurrence of the specified substring from the point of the given index. If the substring does not occur, this method returns -1.

Signature

```
public Integer indexOf(String substring, Integer index)
```

Parameters

substring
Type: String

index
Type: Integer

Return Value

Type: Integer

Example

```
String myString1 = 'abcdabcd';
String myString2 = 'ab';
Integer result = myString1.indexOf(mystring2, 1);
System.assertEquals(4, result);
```

indexOfAny(substring)

Returns the zero-based index of the first occurrence of any character specified in the substring. If none of the characters occur, returns -1.

Signature

```
public Integer indexOfAny(String substring)
```

Parameters

substring
Type: String

Return Value

Type: Integer

Example

```
String s1 = 'abcd';
String s2 = 'xc';
Integer result = s1.indexOfAny(s2);
System.assertEquals(2, result);
```

indexOfAnyBut(substring)

Returns the zero-based index of the first occurrence of a character that is not in the specified substring. Otherwise, returns -1.

Signature

```
public Integer indexOfAnyBut(String substring)
```

Parameters

substring
Type: String

Return Value

Type: Integer

Example

```
String s1 = 'abcd';
String s2 = 'xc';
Integer result = s1.indexOfAnyBut(s2);
System.assertEquals(0, result);
```

indexOfChar(character)

Returns the index of the first occurrence of the character that corresponds to the specified character value.

Signature

```
public Integer indexOfChar(Integer character)
```

Parameters

character
Type: Integer

The integer value of the character in the string.

Return Value

Type: Integer

The index of the first occurrence of the specified character, -1 if the character is not found.

Usage

The index that this method returns is in Unicode code units.

Example

```
String str = '\\u03A9 is Ω (Omega)';
// Returns 0, which is the first character.
System.debug('indexOfChar(937)=' + str.indexOfChar(937));

// Output:
// indexOfChar(937)=0
```

indexOfChar(character, startIndex)

Returns the index of the first occurrence of the character that corresponds to the specified character value, starting from the specified index.

Signature

```
public Integer indexOfChar(Integer character, Integer startIndex)
```

Parameters

character
Type: Integer

The integer value of the character to look for.

startIndex
Type: Integer

The index to start the search from.

Return Value

Type: Integer

The index, starting from the specified start index, of the first occurrence of the specified character, -1 if the character is not found.

Usage

The index that this method returns is in Unicode code units.

Example

This example shows different ways of searching for the index of the Omega character. The first call to `indexOfChar` doesn't specify a start index and therefore the returned index is 0, which is the first occurrence of Omega in the entire string. The subsequent calls specify a start index to find the occurrence of Omega in substrings that start at the specified index.

```
String str = 'Ω and \\u03A9 and Ω';
System.debug('indexOfChar(937)=' + str.indexOfChar(937));
System.debug('indexOfChar(937,1)=' + str.indexOfChar(937,1));
System.debug('indexOfChar(937,10)=' + str.indexOfChar(937,10));

// Output:
// indexOfChar(937)=0
```

```
// indexOfChar(937,1)=6, (corresponds to the escaped form \\u03A9)
// indexOfChar(937,10)=12
```

indexOfDifference(stringToCompare)

Returns the zero-based index of the character where the current String begins to differ from the specified String.

Signature

```
public Integer indexOfDifference(String stringToCompare)
```

Parameters

stringToCompare
Type: String

Return Value

Type: Integer

Example

```
String s1 = 'abcd';
String s2 = 'abxc';
Integer result = s1.indexOfDifference(s2);
System.assertEquals(2, result);
```

indexOfIgnoreCase(substring)

Returns the zero-based index of the first occurrence of the specified substring without regard to case. If the substring does not occur, this method returns -1.

Signature

```
public Integer indexOfIgnoreCase(String substring)
```

Parameters

substring
Type: String

Return Value

Type: Integer

Example

```
String s1 = 'abcd';
String s2 = 'BC';
```

```
Integer result = s1.indexOfIgnoreCase(s2, 0);
System.assertEquals(1, result);
```

indexOfIgnoreCase(substring, startPosition)

Returns the zero-based index of the first occurrence of the specified substring from the point of index *i*, without regard to case. If the substring does not occur, this method returns -1.

Signature

```
public Integer indexOfIgnoreCase(String substring, Integer startPosition)
```

Parameters

substring
Type: String

startPosition
Type: Integer

Return Value

Type: Integer

isAllLowerCase()

Returns `true` if all characters in the current String are lowercase; otherwise, returns `false`.

Signature

```
public Boolean isAllLowerCase()
```

Return Value

Type: Boolean

Example

```
String allLower = 'abcde';
System.assert(allLower.isAllLowerCase());
```

isAllUpperCase()

Returns `true` if all characters in the current String are uppercase; otherwise, returns `false`.

Signature

```
public Boolean isAllUpperCase()
```

Return Value

Type: Boolean

Example

```
String allUpper = 'ABCDE';
System.assert(allUpper.isAllUpperCase());
```

isAlpha()

Returns `true` if all characters in the current String are Unicode letters only; otherwise, returns `false`.

Signature

```
public Boolean isAlpha()
```

Return Value

Type: Boolean

Example

```
// Letters only
String s1 = 'abc';
// Returns true
Boolean b1 =
   s1.isAlpha();
System.assertEquals(
   true, b1);

// Letters and numbers
String s2 = 'abc 21';
// Returns false
Boolean b2 =
   s2.isAlpha();
System.assertEquals(
   false, b2);
```

isAlphaSpace()

Returns `true` if all characters in the current String are Unicode letters or spaces only; otherwise, returns `false`.

Signature

```
public Boolean isAlphaSpace()
```

Return Value

Type: Boolean

Example

```
String alphaSpace = 'aA Bb';
System.assert(alphaSpace.isAlphaSpace());
String notAlphaSpace = 'ab 12';
System.assert(!notAlphaSpace.isAlphaSpace());
notAlphaSpace = 'aA$Bb';
System.assert(!notAlphaSpace.isAlphaSpace());
```

isAlphanumeric()

Returns `true` if all characters in the current String are Unicode letters or numbers only; otherwise, returns `false`.

Signature

```
public Boolean isAlphanumeric()
```

Return Value

Type: Boolean

Example

```
// Letters only
String s1 = 'abc';
// Returns true
Boolean b1 =
   s1.isAlphanumeric();
System.assertEquals(
   true, b1);

// Letters and numbers
String s2 = 'abc021';
// Returns true
Boolean b2 =
   s2.isAlphanumeric();
System.assertEquals(
   true, b2);
```

isAlphanumericSpace()

Returns `true` if all characters in the current String are Unicode letters, numbers, or spaces only; otherwise, returns `false`.

Signature

```
public Boolean isAlphanumericSpace()
```

Return Value

Type: Boolean

Example

```
String alphanumSpace = 'AE 86';
System.assert(alphanumSpace.isAlphanumericSpace());
String notAlphanumSpace = 'aA$12';
System.assert(!notAlphanumSpace.isAlphaSpace());
```

isAsciiPrintable()

Returns `true` if the current String contains only ASCII printable characters; otherwise, returns `false`.

Signature

```
public Boolean isAsciiPrintable()
```

Return Value

Type: Boolean

Example

```
String ascii = 'abcd1234!@#$%^&*()`~-_+={[}]|:<,>.?';
System.assert(ascii.isAsciiPrintable());
String notAscii = '√';
System.assert(!notAscii.isAsciiPrintable());
```

isBlank(inputString)

Returns `true` if the specified String is white space, empty (''), or null; otherwise, returns `false`.

Signature

```
public static Boolean isBlank(String inputString)
```

Parameters

inputString
Type: String

Return Value

Type: Boolean

Example

```
String blank = '';
String nullString = null;
String whitespace = '  ';
System.assert(String.isBlank(blank));
System.assert(String.isBlank(nullString));
System.assert(String.isBlank(whitespace));
```

```
String alpha = 'Hello';
System.assert(!String.isBlank(alpha));
```

isEmpty(inputString)

Returns `true` if the specified String is empty ('') or null; otherwise, returns `false`.

Signature

```
public static Boolean isEmpty(String inputString)
```

Parameters

inputString
Type: String

Return Value

Type: Boolean

Example

```
String empty = '';
String nullString = null;
System.assert(String.isEmpty(empty));
System.assert(String.isEmpty(nullString));
String whitespace = '  ';
String alpha = 'Hello';
System.assert(!String.isEmpty(whitespace));
System.assert(!String.isEmpty(alpha));
```

isNotBlank(inputString)

Returns `true` if the specified String is not whitespace, not empty (''), and not null; otherwise, returns `false`.

Signature

```
public static Boolean isNotBlank(String inputString)
```

Parameters

inputString
Type: String

Return Value

Type: Boolean

Example

```
String alpha = 'Hello world!';
System.assert(String.isNotBlank(alpha));
String blank = '';
String nullString = null;
String whitespace = '  ';
System.assert(!String.isNotBlank(blank));
System.assert(!String.isNotBlank(nullString));
System.assert(!String.isNotBlank(whitespace));
```

isNotEmpty(inputString)

Returns `true` if the specified String is not empty ('') and not null; otherwise, returns `false`.

Signature

```
public static Boolean isNotEmpty(String inputString)
```

Parameters

inputString
Type: String

Return Value

Type: Boolean

Example

```
String whitespace = '  ';
String alpha = 'Hello world!';
System.assert(String.isNotEmpty(whitespace));
System.assert(String.isNotEmpty(alpha));
String empty = '';
String nullString = null;
System.assert(!String.isNotEmpty(empty));
System.assert(!String.isNotEmpty(nullString));
```

isNumeric()

Returns `true` if the current String contains only Unicode digits; otherwise, returns `false`.

Signature

```
public Boolean isNumeric()
```

Return Value

Type: Boolean

Usage

A decimal point (1.2) is not a Unicode digit.

Example

```
String numeric = '1234567890';
System.assert(numeric.isNumeric());
String alphanumeric = 'R32';
String decimalPoint = '1.2';
System.assert(!alphanumeric.isNumeric());
System.assert(!decimalpoint.isNumeric());
```

isNumericSpace()

Returns `true` if the current String contains only Unicode digits or spaces; otherwise, returns `false`.

Signature

```
public Boolean isNumericSpace()
```

Return Value

Type: Boolean

Usage

A decimal point (1.2) is not a Unicode digit.

Example

```
String numericSpace = '1 2 3';
System.assert(numericSpace.isNumericspace());
String notNumericspace = 'FD3S FC3S';
System.assert(!notNumericspace.isNumericspace());
```

isWhitespace()

Returns `true` if the current String contains only white space characters or is empty; otherwise, returns `false`.

Signature

```
public Boolean isWhitespace()
```

Return Value

Type: Boolean

Example

```
String whitespace = ' ';
String blank = '';
System.assert(whitespace.isWhitespace());
System.assert(blank.isWhitespace());
String alphanum = 'SIL80';
System.assert(!alphanum.isWhitespace());
```

join(iterableObj, separator)

Joins the elements of the specified iterable object, such as a List, into a single String separated by the specified separator.

Signature

```
public static String join(Object iterableObj, String separator)
```

Parameters

iterableObj
Type: Object

separator
Type: String

Return Value

Type: String

Usage

```
List<Integer> li = new
   List<Integer>
   {10, 20, 30};
String s = String.join(
   li, '/');
System.assertEquals(
   '10/20/30', s);
```

lastIndexOf(substring)

Returns the index of the last occurrence of the specified substring. If the substring does not occur, this method returns -1.

Signature

```
public Integer lastIndexOf(String substring)
```

Parameters

substring
Type: String

Return Value

Type: Integer

Example

```
String s1 = 'abcdefgc';
Integer i1 = s1.lastIndexOf('c');
System.assertEquals(7, i1);
```

lastIndexOf(substring, endPosition)

Returns the index of the last occurrence of the specified substring, starting from the character at index 0 and ending at the specified index.

Signature

```
public Integer lastIndexOf(String substring, Integer endPosition)
```

Parameters

substring
Type: String

endPosition
Type: Integer

Return Value

Type: Integer

Usage

If the substring doesn't occur or *endPosition* is negative, this method returns -1. If *endPosition* is larger than the last index in the current String, the entire String is searched.

Example

```
String s1 = 'abcdaacd';
Integer i1 = s1.lastIndexOf('c', 7);
System.assertEquals(6, i1);
Integer i2 = s1.lastIndexOf('c', 3);
System.assertEquals(2, i2);
```

lastIndexOfChar(character)

Returns the index of the last occurrence of the character that corresponds to the specified character value.

Signature

```
public Integer lastIndexOfChar(Integer character)
```

Parameters

character
Type: Integer

The integer value of the character in the string.

Return Value

Type: Integer

The index of the last occurrence of the specified character, -1 if the character is not found.

Usage

The index that this method returns is in Unicode code units.

Example

```
String str = '\u03A9 is Ω (Omega)';
// Get the last occurrence of Omega.
System.assertEquals(5, str.lastIndexOfChar(937));
```

lastIndexOfChar(character, endIndex)

Returns the index of the last occurrence of the character that corresponds to the specified character value, starting from the specified index.

Signature

```
public Integer lastIndexOfChar(Integer character, Integer endIndex)
```

Parameters

character
Type: Integer

The integer value of the character to look for.

endIndex
Type: Integer

The index to end the search at.

Return Value

Type: Integer

The index, starting from the specified start index, of the last occurrence of the specified character. -1 if the character is not found.

Usage

The index that this method returns is in Unicode code units.

Example

This example shows different ways of searching for the index of the last occurrence of the Omega character. The first call to `lastIndexOfChar` doesn't specify an end index and therefore the returned index is 12, which is the last occurrence of Omega in the entire string. The subsequent calls specify an end index to find the last occurrence of Omega in substrings.

```
String str = 'Ω and \u03A9 and Ω';
System.assertEquals(12, str.lastIndexOfChar(937));
System.assertEquals(6, str.lastIndexOfChar(937,11));
System.assertEquals(0, str.lastIndexOfChar(937,5));
```

lastIndexOfIgnoreCase(substring)

Returns the index of the last occurrence of the specified substring regardless of case.

Signature

```
public Integer lastIndexOfIgnoreCase(String substring)
```

Parameters

substring
Type: String

Return Value

Type: Integer

Usage

If the substring doesn't occur, this method returns -1.

Example

```
String s1 = 'abcdaacd';
Integer i1 = s1.lastIndexOfIgnoreCase('DAAC');
System.assertEquals(3, i1);
```

lastIndexOfIgnoreCase(substring, endPosition)

Returns the index of the last occurrence of the specified substring regardless of case, starting from the character at index 0 and ending at the specified index.

Signature

```
public Integer lastIndexOfIgnoreCase(String substring, Integer endPosition)
```

Parameters

substring
Type: String

endPosition
Type: Integer

Return Value

Type: Integer

Usage

If the substring doesn't occur or *endPosition* is negative, this method returns -1. If *endPosition* is larger than the last index in the current String, the entire String is searched.

Example

```
String s1 = 'abcdaacd';
Integer i1 = s1.lastIndexOfIgnoreCase('C', 7);
System.assertEquals(6, i1);
```

left(length)

Returns the leftmost characters of the current String of the specified length.

Signature

```
public String left(Integer length)
```

Parameters

length
Type: Integer

Return Value

Type: String

Usage

If *length* is greater than the String size, the entire String is returned.

Example

```
String s1 = 'abcdaacd';
String s2 = s1.left(3);
System.assertEquals('abc', s2);
```

leftPad(length)

Returns the current String padded with spaces on the left and of the specified length.

Signature

```
public String leftPad(Integer length)
```

Parameters

length
Type: Integer

Usage

If *length* is less than or equal to the current String size, the entire String is returned without space padding.

Return Value

Type: String

Example

```
String s1 = 'abc';
String s2 = s1.leftPad(5);
System.assertEquals('  abc', s2);
```

leftPad(length, padStr)

Returns the current String padded with String `padStr` on the left and of the specified length.

Signature

```
public String leftPad(Integer length, String padStr)
```

Parameters

length
Type: Integer

padStr
Type: String

String to pad with; if null or empty treated as single blank.

Usage

If *length* is less than or equal to the current String size, the entire String is returned without space padding.

Return Value

Type: String

Example

```
String s1 = 'abc';
String s2 = 'xy';
String s3 = s1.leftPad(7,s2);
System.assertEquals('xyxyabc', s3);
```

`length()`

Returns the number of 16-bit Unicode characters contained in the String.

Signature

```
public Integer length()
```

Return Value

Type: Integer

Example

```
String myString = 'abcd';
Integer result = myString.length();
System.assertEquals(result, 4);
```

`mid(startIndex, length)`

Returns a new String that begins with the character at the specified zero-based *`startIndex`* with the number of characters specified by *`length`*.

Signature

```
public String mid(Integer startIndex, Integer length)
```

Parameters

`startIndex`
Type: Integer

If *`startIndex`* is negative, it is considered to be zero.

`length`
Type: Integer

If *`length`* is negative or zero, an empty String is returned. If *`length`* is greater than the remaining characters, the remainder of the String is returned.

Return Value

Type: String

Usage

This method is similar to the `substring(startIndex)` and `substring(startIndex, endIndex)` methods, except that the second argument is the number of characters to return.

Example

```
String s = 'abcde';
String s2 = s.mid(2, 3);
System.assertEquals(
   'cde', s2);
```

normalizeSpace()

Returns the current String with leading, trailing, and repeating white space characters removed.

Signature

```
public String normalizeSpace()
```

Return Value

Type: String

Usage

This method normalizes the following white space characters: space, tab (\t), new line (\n), carriage return (\r), and form feed (\f).

Example

```
String s1 =
   'Salesforce \t     force.com';
String s2 =
   s1.normalizeSpace();
System.assertEquals(
   'Salesforce force.com', s2);
```

offsetByCodePoints(index, codePointOffset)

Returns the index of the Unicode code point that is offset by the specified number of code points, starting from the given index.

Signature

```
public Integer offsetByCodePoints(Integer index, Integer codePointOffset)
```

Parameters

index

Type: Integer

The start index in the string.

codePointOffset
Type: Integer

The number of code points to be offset.

Return Value

Type: Integer

The index that corresponds to the start index that is added to the offset.

Usage

Unpaired surrogates within the text range that is specified by *index* and *codePointOffset* count as one code point each.

Example

This example calls `offsetByCodePoints` on a string with a start index of 0 (to start from the first character) and an offset of three code points. The string contains one sequence of supplementary characters in escaped form (a pair of characters). After an offset of three code points when counting from the beginning of the string, the returned code point index is four.

```
String str = 'A \uD835\uDD0A BC';
System.assertEquals(4, str.offsetByCodePoints(0,3));
```

remove(substring)

Removes all occurrences of the specified substring and returns the String result.

Signature

```
public String remove(String substring)
```

Parameters

substring
Type: String

Return Value

Type: String

Example

```
String s1 = 'Salesforce and force.com';
String s2 =
   s1.remove('force');
System.assertEquals(
   'Sales and .com', s2);
```

removeEnd(substring)

Removes the specified substring only if it occurs at the end of the String.

Signature

```
public String removeEnd(String substring)
```

Parameters

substring
Type: String

Return Value

Type: String

Example

```
String s1 = 'Salesforce and force.com';
String s2 =
   s1.removeEnd('.com');
System.assertEquals(
   'Salesforce and force', s2);
```

removeEndIgnoreCase(substring)

Removes the specified substring only if it occurs at the end of the String using a case-insensitive match.

Signature

```
public String removeEndIgnoreCase(String substring)
```

Parameters

substring
Type: String

Return Value

Type: String

Example

```
String s1 = 'Salesforce and force.com';
String s2 = s1.removeEndIgnoreCase('.COM');
System.assertEquals('Salesforce and force', s2);
```

removeStart(substring)

Removes the specified substring only if it occurs at the beginning of the String.

Signature

```
public String removeStart(String substring)
```

Parameters

substring
Type: String

Return Value

Type: String

Example

```
String s1 = 'Salesforce and force.com';
String s2 =
   s1.removeStart('Sales');
System.assertEquals(
   'force and force.com', s2);
```

removeStartIgnoreCase(substring)

Removes the specified substring only if it occurs at the beginning of the String using a case-insensitive match.

Signature

```
public String removeStartIgnoreCase(String substring)
```

Parameters

substring
Type: String

Return Value

Type: String

Example

```
String s1 = 'Salesforce and force.com';
String s2 =
   s1.removeStartIgnoreCase('SALES');
System.assertEquals(
   'force and force.com', s2);
```

repeat(numberOfTimes)

Returns the current String repeated the specified number of times.

Signature

```
public String repeat(Integer numberOfTimes)
```

Parameters

numberOfTimes
Type: Integer

Return Value

Type: String

Example

```
String s1 = 'SFDC';
String s2 = s1.repeat(2);
System.assertEquals('SFDCSFDC', s2);
```

repeat(separator, numberOfTimes)

Returns the current String repeated the specified number of times using the specified separator to separate the repeated Strings.

Signature

```
public String repeat(String separator, Integer numberOfTimes)
```

Parameters

separator
Type: String

numberOfTimes
Type: Integer

Return Value

Type: String

Example

```
String s1 = 'SFDC';
String s2 =
   s1.repeat('-', 2);
System.assertEquals(
   'SFDC-SFDC', s2);
```

replace(target, replacement)

Replaces each substring of a string that matches the literal target sequence *target* with the specified literal replacement sequence *replacement*.

Signature

```
public String replace(String target, String replacement)
```

Parameters

target
Type: String

replacement
Type: String

Return Value

Type: String

Example

```
String s1 = 'abcdbca';
String target = 'bc';
String replacement = 'xy';
String s2 = s1.replace(target, replacement);
System.assertEquals('axydxya', s2);
```

replaceAll(regExp, replacement)

Replaces each substring of a string that matches the regular expression *regExp* with the replacement sequence *replacement*.

Signature

```
public String replaceAll(String regExp, String replacement)
```

Parameters

regExp
Type: String

replacement
Type: String

Return Value

Type: String

Usage

See the Java `Pattern` class for information on regular expressions.

Example

```
String s1 = 'a b c 5 xyz';
String regExp = '[a-zA-Z]';
String replacement = '1';
String s2 = s1.replaceAll(regExp, replacement);
System.assertEquals('1 1 1 5 111', s2);
```

replaceFirst(regExp, replacement)

Replaces the first substring of a string that matches the regular expression *regExp* with the replacement sequence *replacement*.

Signature

```
public String replaceFirst(String regExp, String replacement)
```

Parameters

regExp
Type: String

replacement
Type: String

Return Value

Type: String

Usage

See the Java `Pattern` class for information on regular expressions.

Example

```
String s1 = 'a b c 11 xyz';
String regExp = '[a-zA-Z]{2}';
String replacement = '2';
String s2 = s1.replaceFirst(regExp, replacement);
System.assertEquals('a b c 11 2z', s2);
```

reverse()

Returns a String with all the characters reversed.

Signature

```
public String reverse()
```

Return Value

Type: String

right(length)

Returns the rightmost characters of the current String of the specified length.

Signature

```
public String right(Integer length)
```

Parameters

length
Type: Integer

If *length* is greater than the String size, the entire String is returned.

Return Value

Type: String

Example

```
String s1 = 'Hello Max';
String s2 =
   s1.right(3);
System.assertEquals(
   'Max', s2);
```

rightPad(length)

Returns the current String padded with spaces on the right and of the specified length.

Signature

```
public String rightPad(Integer length)
```

Parameters

length
Type: Integer

If *length* is less than or equal to the current String size, the entire String is returned without space padding.

Return Value

Type: String

Example

```
String s1 = 'abc';
String s2 =
   s1.rightPad(5);
System.assertEquals(
   'abc  ', s2);
```

rightPad(length, padStr)

Returns the current String padded with String `padStr` on the right and of the specified length.

Signature

```
public String rightPad(Integer length, String padStr)
```

Parameters

length
Type: Integer

padStr
Type: String

String to pad with; if null or empty treated as single blank.

Usage

If *length* is less than or equal to the current String size, the entire String is returned without space padding.

Return Value

Type: String

Example

```
String s1 = 'abc';
String s2 = 'xy';
String s3 = s1.rightPad(7, s2);
System.assertEquals('abcxyxy', s3);
```

split(regExp)

Returns a list that contains each substring of the String that is terminated by either the regular expression *regExp* or the end of the String.

Signature

```
public String[] split(String regExp)
```

Parameters

regExp
Type: String

Return Value

Type: String[]

Note: In API version 34.0 and earlier, a zero-width *regExp* value produces an empty list item at the beginning of the method's output.

Usage

See the Java `Pattern` class for information on regular expressions.

The substrings are placed in the list in the order in which they occur in the String. If *regExp* does not match any part of the String, the resulting list has just one element containing the original String.

Example

In the following example, a string is split using a backslash as a delimiter.

```
public String splitPath(String filename) {
    if (filename == null)
        return null;
    List<String> parts = filename.split('\\\\');
    filename = parts[parts.size()-1];
    return filename;
}

// For example, if the file path is e:\\processed\\PPDSF100111.csv
// This method splits the path and returns the last part.
// Returned filename is PPDSF100111.csv
```

split(regExp, limit)

Returns a list that contains each substring of the String that is terminated by either the regular expression *regExp* or the end of the String.

Signature

```
public String[] split(String regExp, Integer limit)
```

Parameters

regExp
Type: String

A regular expression.

limit
Type: Integer

Return Value

Type: String[]

Note: In API version 34.0 and earlier, a zero-width *regExp* value produces an empty list item at the beginning of the method's output.

Usage

The optional *limit* parameter controls the number of times the pattern is applied and therefore affects the length of the list.

- If *limit* is greater than zero:
  - The pattern is applied a maximum of (*limit* – 1) times.
  - The list's length is no greater than *limit*.
  - The list's last entry contains all input beyond the last matched delimiter.

- If *limit* is non-positive, the pattern is applied as many times as possible, and the list can have any length.
- If *limit* is zero, the pattern is applied as many times as possible, the list can have any length, and trailing empty strings are discarded.

Example

For example, for `String s = 'boo:and:moo'`:

- `s.split(':', 2)` results in `{'boo', 'and:moo'}`
- `s.split(':', 5)` results in `{'boo', 'and', 'moo'}`
- `s.split(':', -2)` results in `{'boo', 'and', 'moo'}`
- `s.split('o', 5)` results in `{'b', '', ':and:m', '', ''}`
- `s.split('o', -2)` results in `{'b', '', ':and:m', '', ''}`
- `s.split('o', 0)` results in `{'b', '', ':and:m'}`

splitByCharacterType()

Splits the current String by character type and returns a list of contiguous character groups of the same type as complete tokens.

Signature

```
public List<String> splitByCharacterType()
```

Return Value

Type: List<String>

Usage

For more information about the character types used, see java.lang.Character.getType(char).

Example

```
String s1 = 'Lightning.platform';
List<String> ls =
   s1.splitByCharacterType();
System.debug(ls);
// Writes this output:
// (L, ightning, ., platform)
```

splitByCharacterTypeCamelCase()

Splits the current String by character type and returns a list of contiguous character groups of the same type as complete tokens, with the following exception: the uppercase character, if any, immediately preceding a lowercase character token belongs to the following character token rather than to the preceding.

Signature

```
public List<String> splitByCharacterTypeCamelCase()
```

Return Value

Type: List<String>

Usage

For more information about the character types used, see java.lang.Character.getType(char).

Example

```
String s1 = 'Lightning.platform';
List<String> ls =
   s1.splitByCharacterTypeCamelCase();
System.debug(ls);
// Writes this output:
// (Lightning, ., platform)
```

startsWith(prefix)

Returns `true` if the String that called the method begins with the specified *`prefix`*.

Signature

```
public Boolean startsWith(String prefix)
```

Parameters

`prefix`
Type: String

Return Value

Type: Boolean

Example

```
String s1 = 'AE86 vs EK9';
System.assert(s1.startsWith('AE86'));
```

startsWithIgnoreCase(prefix)

Returns `true` if the current String begins with the specified prefix regardless of the prefix case.

Signature

```
public Boolean startsWithIgnoreCase(String prefix)
```

Parameters

`prefix`
Type: String

Return Value

Type: Boolean

Example

```
String s1 = 'AE86 vs EK9';
System.assert(s1.startsWithIgnoreCase('ae86'));
```

stripHtmlTags()

Removes HTML markup and returns plain text.

Signature

```
public String stripHtmlTags()
```

Return Value

Type: String

Usage

Warning: The stripHtmlTags function doesn't recursively strip tags; therefore, tags can still exist in the returned string. Don't use the stripHtmlTags function to sanitize input for inclusion as a raw HTML page. The unescaped output isn't considered safe to include in an HTML document. The function will be deprecated in a future release.

Example

```
String s1 = '<b>hello world</b>';
String s2 = s1.stripHtmlTags();
System.assertEquals(
   'hello world', s2);
```

substring(startIndex)

Returns a new String that begins with the character at the specified zero-based *startIndex* and extends to the end of the String.

Signature

```
public String substring(Integer startIndex)
```

Parameters

startIndex
Type: Integer

Return Value

Type: String

Example

```
String s1 = 'hamburger';
System.assertEquals('burger', s1.substring(3));
```

substring(startIndex, endIndex)

Returns a new String that begins with the character at the specified zero-based *startIndex* and extends to the character at *endIndex* - 1.

Signature

```
public String substring(Integer startIndex, Integer endIndex)
```

Parameters

startIndex
Type: Integer

endIndex
Type: Integer

Return Value

Type: String

Example

```
'hamburger'.substring(4, 8);
// Returns "urge"

'smiles'.substring(1, 5);
// Returns "mile"
```

substringAfter(separator)

Returns the substring that occurs after the first occurrence of the specified separator.

Signature

```
public String substringAfter(String separator)
```

Parameters

separator
Type: String

Return Value

Type: String

Example

```
String s1 = 'Salesforce.Lightning.platform';
String s2 =
   s1.substringAfter('.');
System.assertEquals(
   'Lightning.platform', s2);
```

substringAfterLast(separator)

Returns the substring that occurs after the last occurrence of the specified separator.

Signature

```
public String substringAfterLast(String separator)
```

Parameters

separator
Type: String

Return Value

Type: String

Example

```
String s1 = 'Salesforce.Lightning.platform';
String s2 =
   s1.substringAfterLast('.');
System.assertEquals(
   'platform', s2);
```

substringBefore(separator)

Returns the substring that occurs before the first occurrence of the specified separator.

Signature

```
public String substringBefore(String separator)
```

Parameters

separator
Type: String

Return Value

Type: String

Example

```
String s1 = 'Salesforce.Lightning.platform';
String s2 =
   s1.substringBefore('.');
System.assertEquals(
   'Salesforce', s2);
```

substringBeforeLast(separator)

Returns the substring that occurs before the last occurrence of the specified separator.

Signature

```
public String substringBeforeLast(String separator)
```

Parameters

separator
Type: String

Return Value

Type: String

Example

```
String s1 = 'Salesforce.Lightning.platform';
String s2 =
   s1.substringBeforeLast('.');
System.assertEquals(
   'Salesforce.Lightning', s2);
```

substringBetween(tag)

Returns the substring that occurs between two instances of the specified *tag* String.

Signature

```
public String substringBetween(String tag)
```

Parameters

tag
Type: String

Return Value

Type: String

Example

```
String s1 = 'tagYellowtag';
String s2 = s1.substringBetween('tag');
System.assertEquals('Yellow', s2);
```

substringBetween(open, close)

Returns the substring that occurs between the two specified Strings.

Signature

```
public String substringBetween(String open, String close)
```

Parameters

open
Type: String

close
Type: String

Return Value

Type: String

Example

```
String s1 = 'xYellowy';
String s2 =
   s1.substringBetween('x','y');
System.assertEquals(
   'Yellow', s2);
```

swapCase()

Swaps the case of all characters and returns the resulting String by using the default (English US) locale.

Signature

```
public String swapCase()
```

Return Value

Type: String

Usage

Upper case and title case converts to lower case, and lower case converts to upper case.

Example

```
String s1 = 'Force.com';
String s2 = s1.swapCase();
System.assertEquals('fORCE.COM', s2);
```

toLowerCase()

Converts all of the characters in the String to lowercase using the rules of the default (English US) locale.

Signature

```
public String toLowerCase()
```

Return Value

Type: String

Example

```
String s1 = 'ThIs iS hArD tO rEaD';
System.assertEquals('this is hard to read',
   s1.toLowerCase());
```

toLowerCase(locale)

Converts all of the characters in the String to lowercase using the rules of the specified locale.

Signature

```
public String toLowerCase(String locale)
```

Parameters

locale
Type: String

Return Value

Type: String

Example

```
// Example in Turkish
// An uppercase dotted "i", \u0304, which is İ
// Note this contains both a İ as well as a I
String s1 = 'KIYMETLİ';
String s1Lower = s1.toLowerCase('tr');
// Dotless lowercase "i", \u0131, which is ı
// Note this has both a i and ı
String expected = 'kıymetli';
```

```
System.assertEquals(expected, s1Lower);
// Note if this was done in toLowerCase('en'), it would output 'kiymetli'
```

toUpperCase()

Converts all of the characters in the String to uppercase using the rules of the default (English US) locale.

Signature

```
public String toUpperCase()
```

Return Value

Type: String

Example

```
String myString1 = 'abcd';
String myString2 = 'ABCD';
myString1 =
   myString1.toUpperCase();
Boolean result =
   myString1.equals(myString2);
System.assertEquals(result, true);
```

toUpperCase(locale)

Converts all of the characters in the String to the uppercase using the rules of the specified locale.

Signature

```
public String toUpperCase(String locale)
```

Parameters

locale
Type: String

Return Value

Type: String

Example

```
// Example in Turkish
// Dotless lowercase "i", \u0131, which is ı
// Note this has both a i and ı
String s1 = 'imkansız';
String s1Upper = s1.toUpperCase('tr');
// An uppercase dotted "i", \u0304, which is İ
// Note this contains both a İ as well as a I
```

```
String expected = 'İMKANSIZ';
System.assertEquals(expected, s1Upper);
```

trim()

Returns a copy of the string that no longer contains any leading or trailing white space characters.

Signature

```
public String trim()
```

Return Value

Type: String

Usage

Leading and trailing ASCII control characters such as tabs and newline characters are also removed. White space and control characters that aren't at the beginning or end of the sentence aren't removed.

Example

```
String s1 = '   Hello!   ';
String trimmed = s1.trim();
system.assertEquals('Hello!', trimmed);
```

uncapitalize()

Returns the current String with the first letter in lowercase.

Signature

```
public String uncapitalize()
```

Return Value

Type: String

Example

```
String s1 =
   'Hello max';
String s2 =
   s1.uncapitalize();
System.assertEquals(
   'hello max',
    s2);
```

unescapeCsv()

Returns a String representing an unescaped CSV column.

Signature

```
public String unescapeCsv()
```

Return Value

Type: String

Usage

If the String is enclosed in double quotes and contains a comma, newline or double quote, quotes are removed. Also, any double quote escaped characters (a pair of double quotes) are unescaped to just one double quote.

If the String is not enclosed in double quotes, or is and does not contain a comma, newline or double quote, it is returned unchanged.

Example

```
String s1 =
   '"Max1, ""Max2"""';
String s2 =
   s1.unescapeCsv();
System.assertEquals(
   'Max1, "Max2"',
    s2);
```

unescapeEcmaScript()

Unescapes any EcmaScript literals found in the String.

Signature

```
public String unescapeEcmaScript()
```

Return Value

Type: String

Example

```
String s1 =
   '\"3.8\",\"3.9\"';
String s2 =
   s1.unescapeEcmaScript();
System.assertEquals(
   '"3.8","3.9"',
   s2);
```

unescapeHtml3()

Unescapes the characters in a String using HTML 3.0 entities.

Signature

```
public String unescapeHtml3()
```

Return Value

Type: String

Example

```
String s1 =
   '&quot;&lt;Black&amp;White&gt;&quot;';
String s2 =
   s1.unescapeHtml3();
System.assertEquals(
   '"<Black&White>"',
   s2);
```

unescapeHtml4()

Unescapes the characters in a String using HTML 4.0 entities.

Signature

```
public String unescapeHtml4()
```

Return Value

Type: String

Usage

If an entity isn't recognized, it is kept as is in the returned string.

Example

```
String s1 =
   '&quot;&lt;Black&amp;White&gt;&quot;';
String s2 =
   s1.unescapeHtml4();
System.assertEquals(
   '"<Black&White>"',
   s2);
```

unescapeJava()

Returns a String whose Java literals are unescaped. Literals unescaped include escape sequences for quotes (\\") and control characters, such as tab (\\t), and carriage return (\\n).

Signature

```
public String unescapeJava()
```

Return Value

Type: String

The unescaped string.

Example

```
String s = 'Company: \\"Salesforce.com\\"';
String unescapedStr = s.unescapeJava();
System.assertEquals('Company: "Salesforce.com"', unescapedStr);
```

unescapeUnicode()

Returns a String whose escaped Unicode characters are unescaped.

Signature

```
public String unescapeUnicode()
```

Return Value

Type: String

The unescaped string.

Example

```
String s = 'De onde voc\u00EA \u00E9?';
String unescapedStr = s.unescapeUnicode();
System.assertEquals('De onde você é?', unescapedStr);
```

unescapeXml()

Unescapes the characters in a String using XML entities.

Signature

```
public String unescapeXml()
```

Return Value

Type: String

Usage

Supports only the five basic XML entities (gt, lt, quot, amp, apos). Does not support DTDs or external entities.

Example

```
String s1 =
   '&quot;&lt;Black&amp;White&gt;&quot;';
String s2 =
   s1.unescapeXml();
System.assertEquals(
   '"<Black&White>"',
   s2);
```

valueOf(dateToConvert)

Returns a String that represents the specified Date in the standard "yyyy-MM-dd" format.

Signature

```
public static String valueOf(Date dateToConvert)
```

Parameters

dateToConvert
Type: Date

Return Value

Type: String

Example

```
Date myDate = Date.Today();
String sDate = String.valueOf(myDate);
```

valueOf(datetimeToConvert)

Returns a String that represents the specified Datetime in the standard "yyyy-MM-dd HH:mm:ss" format for the local time zone.

Signature

```
public static String valueOf(Datetime datetimeToConvert)
```

Parameters

datetimeToConvert
Type: Datetime

Return Value

Type: String

Example

```
DateTime dt = datetime.newInstance(1996, 6, 23);
String sDateTime = String.valueOf(dt);
System.assertEquals('1996-06-23 00:00:00', sDateTime);
```

valueOf(decimalToConvert)

Returns a String that represents the specified Decimal.

Signature

```
public static String valueOf(Decimal decimalToConvert)
```

Parameters

decimalToConvert
Type: Decimal

Return Value

Type: String

Example

```
Decimal dec = 3.14159265;
String sDecimal = String.valueOf(dec);
System.assertEquals('3.14159265', sDecimal);
```

valueOf(doubleToConvert)

Returns a String that represents the specified Double.

Signature

```
public static String valueOf(Double doubleToConvert)
```

Parameters

doubleToConvert
Type: Double

Return Value

Type: String

Example

```
Double myDouble = 12.34;
String myString =
    String.valueOf(myDouble);
System.assertEquals(
  '12.34', myString);
```

valueOf(integerToConvert)

Returns a String that represents the specified Integer.

Signature

```
public static String valueOf(Integer integerToConvert)
```

Parameters

integerToConvert
Type: Integer

Return Value

Type: String

Example

```
Integer myInteger = 22;
String sInteger = String.valueOf(myInteger);
System.assertEquals('22', sInteger);
```

valueOf(longToConvert)

Returns a String that represents the specified Long.

Signature

```
public static String valueOf(Long longToConvert)
```

Parameters

longToConvert
Type: Long

Return Value

Type: String

Example

```
Long myLong = 123456789;
String sLong = String.valueOf(myLong);
System.assertEquals('123456789', sLong);
```

valueOf(toConvert)

Returns a string representation of the specified object argument.

Signature

```
public static String valueOf(Object toConvert)
```

Parameters

toConvert
Type: Object

Return Value

Type: String

Usage

If the argument is not a String, the `valueOf` method converts it into a String by calling the `toString` method on the argument, if available, or any overridden `toString` method if the argument is a user-defined type. Otherwise, if no `toString` method is available, it returns a String representation of the argument.

Example

```
List<Integer> ls =
  new List<Integer>();
ls.add(10);
ls.add(20);
String strList =
   String.valueOf(ls);
System.assertEquals(
  '(10, 20)', strList);
```

valueOfGmt(datetimeToConvert)

Returns a String that represents the specified Datetime in the standard "yyyy-MM-dd HH:mm:ss" format for the GMT time zone.

Signature

```
public static String valueOfGmt(Datetime datetimeToConvert)
```

Parameters

datetimeToConvert
Type: Datetime

Return Value

Type: String

Example

```
// For a PST timezone:
DateTime dt = datetime.newInstance(2001, 9, 14);
String sDateTime = String.valueOfGmt(dt);
System.assertEquals('2001-09-14 07:00:00', sDateTime);
```

StubProvider Interface

`StubProvider` is a callback interface that you can use as part of the Apex stub API to implement a mocking framework. Use this interface with the `Test.createStub()` method to create stubbed Apex objects for testing.

Namespace

System

Usage

The `StubProvider` interface allows you to define the behavior of a stubbed Apex class. The interface specifies a single method that requires implementing: `handleMethodCall()`. You specify the behavior of each method of the stubbed class in the `handleMethodCall()` method.

In your Apex test, you create a stubbed object using the `Test.createStub()` method. When you invoke methods on the stubbed object, `StubProvider.handleMethodCall()` is called, which performs the behavior that you've specified for each method.

IN THIS SECTION:

StubProvider Methods

SEE ALSO:

Apex Developer Guide: Build a Mocking Framework with the Stub API

createStub(parentType, stubProvider)

StubProvider Methods

The following are methods for `StubProvider`.

IN THIS SECTION:

handleMethodCall(stubbedObject, stubbedMethodName, returnType, listOfParamTypes, listOfParamNames, listOfArgs)
Use this method to define the behavior of each method of a stubbed class.

handleMethodCall(stubbedObject, stubbedMethodName, returnType, listOfParamTypes, listOfParamNames, listOfArgs)

Use this method to define the behavior of each method of a stubbed class.

Signature

```
public Object handleMethodCall(Object stubbedObject, String stubbedMethodName,
System.Type returnType, List<System.Type> listOfParamTypes, List<String>
listOfParamNames, List<Object> listOfArgs)
```

Parameters

stubbedObject
Type: Object

The stubbed object.

stubbedMethodName
Type: String

The name of the invoked method.

returnType
Type: System.Type

The return type of the invoked method.

listOfParamTypes
Type: List<System.Type>

A list of the parameter types of the invoked method.

listOfParamNames
Type: List<String>

A list of the parameter names of the invoked method.

listOfArgs
Type: List<Object>

The actual argument values passed into this method at runtime.

Return Value

Type: Object

Usage

You can use the parameters passed into this method to identify which method on the stubbed object was invoked. Then you can define the behavior for each identified method.

SEE ALSO:

Apex Developer Guide: Build a Mocking Framework with the Stub API

System Class

Contains methods for system operations, such as writing debug messages and scheduling jobs.

Namespace

System

System Methods

The following are methods for `System`. All methods are static.

IN THIS SECTION:

abortJob(jobId)

Stops the specified job. If the job is currently executing, the stopped job is still visible in the job queue in the Salesforce user interface. The specified job is stopped, but any code that is in progress will continue to execute until it completes.

assert(condition, msg)

Asserts that the specified condition is true. If it isn't , a fatal error is returned that causes code execution to halt.

assertEquals(expected, actual, msg)

Asserts that the first two arguments are the same. If they aren't, a fatal error is returned that causes code execution to halt.

assertNotEquals(expected, actual, msg)

Asserts that the first two arguments are different. If they're the same, a fatal error is returned that causes code execution to halt.

currentPageReference()

Returns a reference to the current page. This is used with Visualforce pages.

currentTimeMillis()

Returns the current time in milliseconds, which is expressed as the difference between the current time and midnight, January 1, 1970 UTC.

debug(msg)

Writes the specified message, in string format, to the execution debug log. The `DEBUG` log level is used.

debug(logLevel, msg)

Writes the specified message, in string format, to the execution debug log with the specified log level.

enqueueJob(queueableObj)

Adds a job to the Apex job queue that corresponds to the specified queueable class and returns the job ID.

enqueueJob(queueable, delay)

Adds a job to the Apex job queue that corresponds to the specified queueable class and returns the job ID. The job is scheduled with a specified minimum delay (0–10 minutes). The delay is ignored during Apex testing.

enqueueJob(queueable, asyncOptions)

Adds a job to the Apex job queue that corresponds to the specified queueable class and returns the job ID. Specify a unique signature for your queueable job, the maximum stack depth or the minimum queue delay in the asyncOptions parameter.

equals(obj1, obj2)

Returns `true` if both arguments are equal. Otherwise, returns `false`.

getApplicationReadWriteMode()

Returns the read write mode set for an organization during Salesforce.com upgrades and downtimes.

getQuiddityShortCode(QuidditValue)

Returns the short code for the Quiddity value of the current Request object.

hashCode(obj)

Returns the hash code of the specified object.

isBatch()

Returns `true` if a batch Apex job invoked the executing code, or `false` if not. In API version 35.0 and earlier, also returns `true` if a queueable Apex job invoked the code.

isFunctionCallback()

Returns `true` if an asynchronous Salesforce Function callback invoked the executing code, or `false` if not. Available in API version 51.0 and later.

isFuture()

Returns `true` if the currently executing code is invoked by code contained in a method annotated with `future`; `false` otherwise.

isQueueable()

Returns `true` if a queueable Apex job invoked the executing code. Returns `false` if not, including if a batch Apex job or a future method invoked the code.

isRunningElasticCompute()

Reserved for future use.

isScheduled()

Returns `true` if the currently executing code is invoked by a scheduled Apex job; `false` otherwise.

movePassword(targetUserId,sourceUserId)

Moves the specified user's password to a different user.

now()

Returns the current date and time in the GMT time zone.

pauseJobById(cronTriggerId)

Pause a scheduled Apex job specified by its CronTrigger ID.

pauseJobByName(jobName)

Pause a scheduled Apex job specified by its name.

process(workItemIds, action, comments, nextApprover)

Processes the list of work item IDs.

purgeOldAsyncJobs(dt)

Deletes asynchronous Apex job records for jobs that have finished execution before the specified date with a Completed, Aborted, or Failed status, and returns the number of records deleted.

requestVersion()

Returns a two-part version that contains the major and minor version numbers of a package. Applies to first-generation managed packages.

resetPassword(userId, sendUserEmail)

Resets the password for the specified user.

resetPasswordWithEmailTemplate(userId, sendUserEmail, emailTemplateName)

Resets the user's password and sends an email to the user with their new password. You specify the email template that is sent to the specified user. Use this method for external users of Experience Cloud sites.

resumeJobById(cronTriggerId)

Resume a paused scheduled Apex job specified by its CronTrigger ID.

resumeJobByName(jobName)

Resumes a paused scheduled Apex job specified by its name.

runAs(version)

Changes the current package version to the package version specified in the argument.

runAs(userSObject)

Changes the current user to the specified user.

schedule(jobName, cronExpression, schedulableClass)

Use `schedule` with an Apex class that implements the `Schedulable` interface to schedule the class to run at the time specified by a Cron expression.

scheduleBatch(batchable, jobName, minutesFromNow)

Schedules a batch job to run once in the future after the specified time interval and with the specified job name.

scheduleBatch(batchable, jobName, minutesFromNow, scopeSize)

Schedules a batch job to run once in the future after the specified the time interval, with the specified job name and scope size. Returns the scheduled job ID (CronTrigger ID).

setPassword(userId, password)

Sets the password for the specified user.

submit(workItemIds, comments, nextApprover)

Submits the processed approvals. The current user is the submitter and the entry criteria is evaluated for all processes applicable to the current user.

today()

Returns the current date in the current user's time zone.

`abortJob(jobId)`

Stops the specified job. If the job is currently executing, the stopped job is still visible in the job queue in the Salesforce user interface. The specified job is stopped, but any code that is in progress will continue to execute until it completes.

Signature

```
public static Void abortJob(String jobId)
```

Parameters

jobId
Type: String

The *jobId* is the ID associated with an AsyncApexJob ID for batch or future Apex jobs, or a CronTrigger ID for scheduled Apex jobs. You can't abort a scheduled Apex job using an AsyncApexJob ID.

Return Value

Type: Void

Usage

The following methods return the job ID that can be passed to `abortJob`.

- `System.schedule` method—returns the CronTrigger object ID associated with the scheduled job as a string.
- `SchedulableContext.getTriggerId` method—returns the CronTrigger object ID associated with the scheduled job as a string.
- `getJobId` method—returns the AsyncApexJob object ID associated with the batch job as a string.
- Using Batch Apex`Database.executeBatch` method—returns the AsyncApexJob object ID associated with the batch job as a string.

assert(condition, msg)

Asserts that the specified condition is true. If it isn't , a fatal error is returned that causes code execution to halt.

Important: We recommend that you use the methods of the Assert Class rather than this method. The `System.Assert` class provides methods that handle all types of logical assertions and comparisons, which improve the clarity of your Apex code.

Signature

```
public static Void assert(Boolean condition, Object msg)
```

Parameters

condition
Type: Boolean

msg
Type: Object

(Optional) Custom message returned as part of the error message.

Return Value

Type: Void

Usage

You can't catch an assertion failure using a try/catch block even though it's logged as an exception.

assertEquals(expected, actual, msg)

Asserts that the first two arguments are the same. If they aren't, a fatal error is returned that causes code execution to halt.

Important: We recommend that you use the methods of the Assert Class rather than this method. The `System.Assert` class provides methods that handle all types of logical assertions and comparisons, which improve the clarity of your Apex code.

Signature

```
public static Void assertEquals(Object expected, Object actual, Object msg)
```

Parameters

expected
Type: Object

Specifies the expected value.

actual
Type: Object

Specifies the actual value.

msg
Type: Object

(Optional) Custom message returned as part of the error message.

Return Value

Type: Void

Usage

You can't catch an assertion failure using a try/catch block even though it's logged as an exception.

assertNotEquals(expected, actual, msg)

Asserts that the first two arguments are different. If they're the same, a fatal error is returned that causes code execution to halt.

Important: We recommend that you use the methods of the Assert Class rather than this method. The `System.Assert` class provides methods that handle all types of logical assertions and comparisons, which improve the clarity of your Apex code.

Signature

```
public static Void assertNotEquals(Object expected, Object actual, Object msg)
```

Parameters

expected
Type: Object

Specifies the expected value.

actual
Type: Object

Specifies the actual value.

msg
Type: Object

(Optional) Custom message returned as part of the error message.

Return Value

Type: Void

Usage

You can't catch an assertion failure using a try/catch block even though it's logged as an exception.

currentPageReference()

Returns a reference to the current page. This is used with Visualforce pages.

Signature

```
public static System.PageReference currentPageReference()
```

Return Value

Type: System.PageReference

Usage

For more information, see PageReference Class.

currentTimeMillis()

Returns the current time in milliseconds, which is expressed as the difference between the current time and midnight, January 1, 1970 UTC.

Signature

```
public static Long currentTimeMillis()
```

Return Value

Type: Long

debug(msg)

Writes the specified message, in string format, to the execution debug log. The `DEBUG` log level is used.

Signature

```
public static Void debug(Object msg)
```

Parameters

msg
Type: Object

Return Value

Type: Void

Usage

If the *msg* argument is not a string, the `debug` method calls `String.valueOf` to convert it into a string. The `String.valueOf` method calls the `toString` method on the argument, if available, or any overridden `toString` method if the argument is a user-defined type. Otherwise, if no `toString` method is available, it returns a string representation of the argument.

If the log level for Apex Code is set to `DEBUG` or higher, the message of this debug statement will be written to the debug log.

Note that when a map or set is printed, the output is sorted in key order and is surrounded with square brackets (`[]`). When an array or list is printed, the output is enclosed in parentheses (`()`).

Note: Calls to System.debug are not counted as part of Apex code coverage.Calls to `System.debug` are not counted as part of Apex code coverage.

For more information on log levels, see Debug Log Levels in the Salesforce online help.

`debug(logLevel, msg)`

Writes the specified message, in string format, to the execution debug log with the specified log level.

Signature

```
public static Void debug(LoggingLevel logLevel, Object msg)
```

Parameters

logLevel
Type: LoggingLevel Enum

The logging level to set for this method.

msg
Type: Object

The message or object to write in string format to the execution debug log.

Return Value

Type: Void

Usage

If the *msg* argument is not a string, the `debug` method calls `String.valueOf` to convert it into a string. The `String.valueOf` method calls the `toString` method on the argument, if available, or any overridden `toString` method if the argument is a user-defined type. Otherwise, if no `toString` method is available, it returns a string representation of the argument.

Note: Calls to `System.debug` are not counted as part of Apex code coverage.

For more information on log levels, see Debug Log Levels in the Salesforce online help.

enqueueJob(queueableObj)

Adds a job to the Apex job queue that corresponds to the specified queueable class and returns the job ID.

Signature

```
public static ID enqueueJob(Object queueableObj)
```

Parameters

queueableObj
Type: Object

An instance of the class that implements the Queueable Interface.

Return Value

Type: ID

The job ID, which corresponds to the ID of an AsyncApexJob record.

Usage

To add a job for asynchronous execution, call `System.enqueueJob` by passing in an instance of your class implementation of the `Queueable` interface for execution as follows:

```
ID jobID = System.enqueueJob(new MyQueueableClass());
```

For more information about Queueable Apex, including information about limits, see Queueable Apex.

enqueueJob(queueable, delay)

Adds a job to the Apex job queue that corresponds to the specified queueable class and returns the job ID. The job is scheduled with a specified minimum delay (0–10 minutes). The delay is ignored during Apex testing.

Signature

```
public static Id enqueueJob(Object queueable, Integer delay)
```

Parameters

queueable
Type: Object

An instance of the class that implements the Queueable Interface.

delay
Type: Integer

The minimum delay (0–10 minutes) before the queueable job is scheduled for execution.

The delay is ignored during Apex testing.



Warning: When you set the delay to 0 (zero), the Queueable job is run as quickly as possible. With chained queueable jobs, implement a mechanism to slow down or halt the job if necessary. Without such a fail-safe mechanism in place, you can rapidly reach the daily async Apex limit.

Return Value

Type: Id

The job ID, which corresponds to the ID of an AsyncApexJob record.

Example

This example adds a job for delayed asynchronous execution by passing in an instance of your class implementation of the `Queueable` interface for execution. There's a minimum delay of 5 minutes before the job is executed.

```
Integer delayInMinutes = 5;
ID jobID = System.enqueueJob(new MyQueueableClass(), delayInMinutes);
```

For more information about Queueable Apex, including information about limits, see Queueable Apex.

`enqueueJob(queueable, asyncOptions)`

Adds a job to the Apex job queue that corresponds to the specified queueable class and returns the job ID. Specify a unique signature for your queueable job, the maximum stack depth or the minimum queue delay in the asyncOptions parameter.

Signature

```
public static Id enqueueJob(Object queueable, Object asyncoptions)
```

Parameters

`queueable`
Type: Object

An instance of the class that implements the Queueable Interface.

`asyncoptions`
Type: AsyncOptions

Specify a unique signature for your queueable job, the maximum stack depth, or a minimum queue delay in the AsyncOptions class properties.

Return Value

Type: Id

The job ID, which corresponds to the ID of an AsyncApexJob record.

Usage

The `System.AsyncInfo` class methods help you determine if maximum stack depth is set in your Queueable request and get the stack depths and queue delay for queueables that are currently running. Use information about the current queueable execution to make decisions on adjusting delays on subsequent calls.

These are methods in the `System.AsyncInfo` class.

- `hasMaxStackDepth()`
- `getCurrentQueueableStackDepth()`
- `getMaximumQueueableStackDepth()`
- `getMinimumQueueableDelayInMinutes()`

For more information about Queueable Apex, including information about limits, see Queueable Apex.

`equals(obj1, obj2)`

Returns `true` if both arguments are equal. Otherwise, returns `false`.

Signature

```
public static Boolean equals(Object obj1, Object obj2)
```

Parameters

obj1
Type: Object

Object being compared.

obj2
Type: Object

Object to compare with the first argument.

Return Value

Type: Boolean

Usage

obj1 and *obj2* can be of any type. They can be values, or object references, such as sObjects and user-defined types.

The comparison rules for `System.equals` are identical to the ones for the `==` operator. For example, string comparison is case insensitive. For information about the comparison rules, see the == operator.

`getApplicationReadWriteMode()`

Returns the read write mode set for an organization during Salesforce.com upgrades and downtimes.

Signature

```
public static System.ApplicationReadWriteMode getApplicationReadWriteMode()
```

Return Value

Type: System.ApplicationReadWriteMode

Valid values are:

- `DEFAULT`

- READ_ONLY

Using the `System.ApplicationReadWriteMode` Enum

Use the `System.ApplicationReadWriteMode` enum returned by the `getApplicationReadWriteMode` to programmatically determine if the application is in read-only mode during Salesforce upgrades and downtimes.

Valid values for the enum are:

- DEFAULT
- READ_ONLY

Example:

```
public class myClass {
  public static void execute() {
    ApplicationReadWriteMode mode = System.getApplicationReadWriteMode();

    if (mode == ApplicationReadWriteMode.READ_ONLY) {
      // Do nothing. If DML operaton is attempted in readonly mode,
      // InvalidReadOnlyUserDmlException will be thrown.
    } else if (mode == ApplicationReadWriteMode.DEFAULT) {
      Account account = new Account(name = 'my account');
      insert account;
    }
  }
}
```

`getQuiddityShortCode(QuiddityValue)`

Returns the short code for the Quiddity value of the current Request object.

Signature

```
public String getQuiddityShortCode(System.Quiddity QuidditityValue)
```

Parameters

`QuidditityValue`
Type: System.Quiddity

The Quiddity enum value that has an associated short code. This short code is used in Event Monitoring logs. For more information, see Apex Execution Event Type.

Return Value

Type: String

`hashCode(obj)`

Returns the hash code of the specified object.

Signature

```
public static Integer hashCode(Object obj)
```

Parameters

obj
Type: Object

The object to get the hash code for. This parameter can be of any type, including values or object references, such as sObjects or user-defined types.

Return Value

Type: Integer

Versioned Behavior Changes

In API version 51.0 and later, the `hashCode()` method returns the same hashCode for identical Id values. In API version 50.0 and earlier, identical Id values didn't always generate the same hashCode value.

isBatch()

Returns `true` if a batch Apex job invoked the executing code, or `false` if not. In API version 35.0 and earlier, also returns `true` if a queueable Apex job invoked the code.

Signature

```
public static Boolean isBatch()
```

Return Value

Type: Boolean

Usage

A batch Apex job can't invoke a future method. Before invoking a future method, use `isBatch()` to check whether the executing code is a batch Apex job.

isFunctionCallback()

Returns `true` if an asynchronous Salesforce Function callback invoked the executing code, or `false` if not. Available in API version 51.0 and later.

Signature

```
public static Boolean isFunctionCallback()
```

Return Value

Type: Boolean

Usage

Use this method to determine if the Apex code is being invoked as part of a callback from an asynchronous Salesforce Functions invocation. For more details on invoking Salesforce Functions from Apex, see Functions Namespace

`isFuture()`

Returns `true` if the currently executing code is invoked by code contained in a method annotated with `future`; `false` otherwise.

Signature

```
public static Boolean isFuture()
```

Return Value

Type: Boolean

Usage

Since a future method can't be invoked from another future method, use this method to check if the current code is executing within the context of a future method before you invoke a future method.

`isQueueable()`

Returns `true` if a queueable Apex job invoked the executing code. Returns `false` if not, including if a batch Apex job or a future method invoked the code.

Signature

```
public static Boolean isQueueable()
```

Return Value

Type: Boolean

Usage

```
public class SimpleQueueable implements Queueable {

    String name;

    public SimpleQueueable(String name) {
        this.name = name;
        System.assert(!System.isQueueable());    //Should return false
    }

    public void execute(QueueableContext ctx) {
        Account testAccount  = new Account();
        testAccount.name = 'testAcc';
        insert(testAccount);
        System.assert(System.isQueueable());    //Should return true
```

```
    }
}
```

```
global class ComplexBatch implements Database.Batchable<SObject> {

    global Database.QueryLocator start(Database.BatchableContext info) {
        System.assert(!System.isQueueable());    //Should return false
        return Database.getQueryLocator([SELECT Id, Name FROM Account LIMIT 1]);
    }

    global void execute(Database.BatchableContext info, SObject[] scope) {
        System.assert(!System.isQueueable());    //Should return false
        System.enqueueJob(new SimpleQueueable('CallingFromComplexBatch'));
        System.assert(!System.isQueueable());    //Should return false
    }

    global void finish(Database.BatchableContext info) {
        System.assert(!System.isQueueable());    //Should return false
    }
}
```

isRunningElasticCompute()

Reserved for future use.

Signature

```
public static Boolean isRunningElasticCompute()
```

Return Value

Type: Boolean

isScheduled()

Returns `true` if the currently executing code is invoked by a scheduled Apex job; `false` otherwise.

Signature

```
public static Boolean isScheduled()
```

Return Value

Type: Boolean

movePassword(targetUserId,sourceUserId)

Moves the specified user's password to a different user.

Signature

```
public static Void movePassword(ID targetUserId, ID sourceUserId)
```

Parameters

targetUserId
Type: ID

The user that the password is moved to.

sourceUserId
Type: ID

The user that the password is moved from.

Return Value

Type: Void

Usage

Moving a password simplifies converting a user to another type of user, such as when converting an external user to a user with less restrictive access. If you require access to the `movePassword` method, contact Salesforce.

Keep in mind these requirements.

- The *targetUserId*, *sourceUserId*, and user performing the move operation must all belong to the same Salesforce org.
- The *targetUserId* and the *sourceUserId* cannot be the same as the user performing the move operation.
- A user without a password can't be specified as the *sourceUserId*. For example, a source user who has already had their password moved is left without a password. That user can't be a source user again.

After the password is moved:

- The target user can log in with the password.
- The source user no longer has a password. To enable logins for this user, a password reset is required.

now()

Returns the current date and time in the GMT time zone.

Signature

```
public static Datetime now()
```

Return Value

Type: Datetime

pauseJobById(cronTriggerId)

Pause a scheduled Apex job specified by its CronTrigger ID.

Signature

```
public static void pauseJobById(String cronTriggerId)
```

Parameters

cronTriggerId
Type: String

The scheduled job ID.

Return Value

Type: void

pauseJobByName(jobName)

Pause a scheduled Apex job specified by its name.

Signature

```
public static void pauseJobByName(String jobName)
```

Parameters

jobName
Type: String

Return Value

Type: void

process(workItemIds, action, comments, nextApprover)

Processes the list of work item IDs.

Signature

```
public static List<Id> process(List<Id> workItemIds, String action, String comments,
String nextApprover)
```

Parameters

workItemIds
Type: List<Id>

action
Type: String

comments
Type: String

nextApprover
Type: String

Return Value

Type: List<Id>

purgeOldAsyncJobs(dt)

Deletes asynchronous Apex job records for jobs that have finished execution before the specified date with a Completed, Aborted, or Failed status, and returns the number of records deleted.

Signature

```
public static Integer purgeOldAsyncJobs(Date dt)
```

Parameters

dt

Type: Date

Specifies the date up to which old records are deleted. The date comparison is based on the `CompletedDate` field of AsyncApexJob, which is in the GMT time zone.

Return Value

Type: Integer

Usage

Asynchronous Apex job records are records in AsyncApexJob.

The system cleans up asynchronous job records for jobs that have finished execution and are older than seven days. You can use this method to further reduce the size of AsyncApexJob by cleaning up more records.

Each execution of this method counts as a single row against the governor limit for DML statements.

Example

This example shows how to delete all job records for jobs that have finished before today's date.

```
Integer count = System.purgeOldAsyncJobs
   (Date.today());
System.debug('Deleted ' +
   count + ' old jobs.');
```

requestVersion()

Returns a two-part version that contains the major and minor version numbers of a package. Applies to first-generation managed packages.

Signature

```
public static System.Version requestVersion()
```

Return Value

Type: System.Version

Usage

Using this method, you can determine the version of an installed instance of your package from which the calling code is referencing your package. Based on the version that the calling code has, you can customize the behavior of your package code.

The `requestVersion` method isn't supported for unmanaged packages. If you call it from an unmanaged package, an exception will be thrown.

resetPassword(userId, sendUserEmail)

Resets the password for the specified user.

Signature

```
public static System.ResetPasswordResult resetPassword(ID userId, Boolean sendUserEmail)
```

Parameters

userId
Type: ID

sendUserEmail
Type: Boolean

Return Value

Type: System.ResetPasswordResult

Usage

When the user logs in with the new password, they are prompted to enter a new password, and to select a security question and answer if they haven't already. If you specify `true` for *sendUserEmail*, the user is sent an email notifying them that their password was reset. A link to sign onto Salesforce using the new password is included in the email. Use `setPassword(userId, password)` if you don't want the user to be prompted to enter a new password when they log in.

Warning: Be careful with this method, and do not expose this functionality to end-users.

resetPasswordWithEmailTemplate(userId, sendUserEmail, emailTemplateName)

Resets the user's password and sends an email to the user with their new password. You specify the email template that is sent to the specified user. Use this method for external users of Experience Cloud sites.

Signature

```
public static System.ResetPasswordResult resetPasswordWithEmailTemplate(Id userId,
Boolean sendUserEmail, String emailTemplateName)
```

Parameters

userId
Type: Id

The ID of the user whose password was reset.

sendUserEmail
Type: Boolean

emailTemplateName
Type: String

Name of the email template.

Return Value

Type: System.ResetPasswordResult

Usage

If you specify `true` for *sendUserEmail*, specify the email template that is sent to the user notifying them that their password was reset. When the user logs in with the new password in the email, they are prompted to enter a new password. A link to sign onto Salesforce using the new password is included in the email. Use `setPassword(userId, password)` if you don't want the user to be prompted to enter a new password when they log in.

Warning: Be careful with this method, and do not expose this functionality to end-users.

resumeJobById(cronTriggerId)

Resume a paused scheduled Apex job specified by its CronTrigger ID.

Signature

```
public static void resumeJobById(String cronTriggerId)
```

Parameters

cronTriggerId
Type: String

The scheduled job ID.

Return Value

Type: void

Usage

If you resume a paused scheduled job, the job immediately runs one time. Subsequent executions of the job run according to the established schedule. Any scheduled executions that were missed while the job was paused don't run.

resumeJobByName(jobName)

Resumes a paused scheduled Apex job specified by its name.

Signature

```
public static void resumeJobByName(String jobName)
```

Parameters

jobName
Type: String

Return Value

Type: void

Usage

If you resume a paused scheduled job, the job immediately runs one time. Subsequent executions of the job run according to the established schedule. Any scheduled executions that were missed while the job was paused don't run.

runAs(version)

Changes the current package version to the package version specified in the argument.

Signature

```
public static Void runAs(System.Version version)
```

Parameters

version
Type: System.Version

Return Value

Type: Void

Usage

A package developer can use Version methods to continue to support existing behavior in classes and triggers in previous package versions while continuing to evolve the code. Apex classes and triggers are saved with the version settings for each installed managed package that the class or trigger references.

This method is used for testing your component behavior in different package versions that you upload to the AppExchange. This method effectively sets a two-part version consisting of major and minor numbers in a test method so that you can test the behavior for different package versions.

You can only use `runAs` in a test method. There is no limitation to the number of calls to this method in a transaction. For sample usage of this method, see Testing Behavior in Package Versions.

runAs(userSObject)

Changes the current user to the specified user.

Signature

```
public static Void runAs(User userSObject)
```

Parameters

userSObject
Type: User

Return Value

Type: Void

Usage

All of the specified user's record sharing is enforced during the execution of `runAs`. You can only use `runAs` in a test method. For more information, see Using the runAs() Method.

Note: The `runAs` method ignores user license limits. You can create new users with `runAs` even if your organization has no additional user licenses.

The `runAs` method implicitly inserts the user that is passed in as parameter if the user has been instantiated, but not inserted yet.

You can also use `runAs` to perform mixed DML operations in your test by enclosing the DML operations within the `runAs` block. In this way, you bypass the mixed DML error that is otherwise returned when inserting or updating setup objects together with other sObjects. See sObjects That Cannot Be Used Together in DML Operations.

Note: Every call to `runAs` counts against the total number of DML statements issued in the process.

schedule(jobName, cronExpression, schedulableClass)

Use `schedule` with an Apex class that implements the `Schedulable` interface to schedule the class to run at the time specified by a Cron expression.

Signature

```
public static String schedule(String jobName, String cronExpression, Object
schedulableClass)
```

Parameters

jobName
Type: String

cronExpression
Type: String

schedulableClass
Type: Object

Return Value

Type: String

Returns the scheduled job ID (CronTrigger ID).

Usage

Use extreme care if you're planning to schedule a class from a trigger. You must be able to guarantee that the trigger won't add more scheduled classes than the limit. In particular, consider API bulk updates, import wizards, mass record changes through the user interface, and all cases where more than one record can be updated at a time. Use the `abortJob` method to stop the job after it has been scheduled.

Note: Salesforce schedules the class for execution at the specified time. Actual execution may be delayed based on service availability.

Using the `System.Schedule` Method

After you implement a class with the `Schedulable` interface, use the `System.Schedule` method to execute it. The scheduler runs as system—all classes are executed, whether or not the user has permission to execute the class.

Note: Use extreme care if you're planning to schedule a class from a trigger. You must be able to guarantee that the trigger won't add more scheduled classes than the limit. In particular, consider API bulk updates, import wizards, mass record changes through the user interface, and all cases where more than one record can be updated at a time.

The `System.Schedule` method takes three arguments: a name for the job, an expression used to represent the time and date the job is scheduled to run, and the name of the class. This expression has the following syntax:

```
Seconds Minutes Hours Day_of_month Month Day_of_week Optional_year
```

Note: Salesforce schedules the class for execution at the specified time. Actual execution may be delayed based on service availability.

The `System.Schedule` method uses the user's timezone for the basis of all schedules.

The following are the values for the expression:

Name	Values	Special Characters
Seconds	0–59	None
Minutes	0–59	None
Hours	0–23	`, - * /`
Day_of_month	1–31	`, - * ? / L W`
Month	1–12 or the following: • `JAN` • `FEB` • `MAR` • `APR` • `MAY` • `JUN` • `JUL` • `AUG` • `SEP` • `OCT`	`, - * /`

Name	Values	Special Characters
	• NOV • DEC	
Day_of_week	1–7 or the following: • SUN • MON • TUE • WED • THU • FRI • SAT	, - * ? / L #
optional_year	null or 1970–2099	, - * /

The special characters are defined as follows:

Special Character	Description
,	Delimits values. For example, use JAN, MAR, APR to specify more than one month.
-	Specifies a range. For example, use JAN-MAR to specify more than one month.
*	Specifies all values. For example, if *Month* is specified as *, the job is scheduled for every month.
?	Specifies no specific value. This is only available for *Day_of_month* and *Day_of_week*, and is generally used when specifying a value for one and not the other.
/	Specifies increments. The number before the slash specifies when the intervals will begin, and the number after the slash is the interval amount. For example, if you specify 1/5 for *Day_of_month*, the Apex class runs every fifth day of the month, starting on the first of the month.
L	Specifies the end of a range (last). This is only available for *Day_of_month* and *Day_of_week*. When used with *Day of month*, L always means the last day of the month, such as January 31, February 29 for leap years, and so on. When used with *Day_of_week* by itself, it always means 7 or SAT. When used with a *Day_of_week* value, it means the last of that type of day in the month. For example, if you specify 2L, you are specifying the last Monday of the month. Do not use a range of values with L as the results might be unexpected.
W	Specifies the nearest weekday (Monday-Friday) of the given day. This is only available for *Day_of_month*. For example, if you specify 20W, and the 20th is a Saturday, the class runs on the 19th. If you specify 1W, and the first is a Saturday, the class does not run in the previous month, but on the third, which is the following Monday.

Special Character	Description
	Tip: Use the `L` and `W` together to specify the last weekday of the month.
`#`	Specifies the *`nth`* day of the month, in the format ***`weekday`*`#`*`day_of_month`***. This is only available for *`Day_of_week`*. The number before the `#` specifies weekday (`SUN-SAT`). The number after the `#` specifies the day of the month. For example, specifying `2#1` means the class runs on the first Monday of every month.

The following are some examples of how to use the expression.

Expression	Description
`0 0 13 * * ?`	Class runs every day at 1 PM.
`0 0 22 ? * 6L`	Class runs the last Friday of every month at 10 PM.
`0 0 10 ? * MON-FRI`	Class runs Monday through Friday at 10 AM.
`0 0 20 * * ? 2010`	Class runs every day at 8 PM during the year 2010.

In the following example, the class `proschedule` implements the `Schedulable` interface. The class is scheduled to run at 8 AM, on the 13 February.

```
proschedule p = new proschedule();
        String sch = '0 0 8 13 2 ?';
        system.schedule('One Time Pro', sch, p);
```

`scheduleBatch(batchable, jobName, minutesFromNow)`

Schedules a batch job to run once in the future after the specified time interval and with the specified job name.

Signature

```
public static String scheduleBatch(Database.Batchable batchable, String jobName, Integer minutesFromNow)
```

Parameters

`batchable`
Type: Database.Batchable

An instance of a class that implements the `Database.Batchable` interface.

`jobName`
Type: String

The name if the job that this method will start.

`minutesFromNow`
Type: Integer

The time interval in minutes after which the job should start executing. This argument must be greater than zero.

Return Value

Type: String

The scheduled job ID (CronTrigger ID).

Usage

Note: Some things to note about `System.scheduleBatch`:

- When you call `System.scheduleBatch`, Salesforce schedules the job for execution at the specified time. Actual execution occurs at or after that time, depending on service availability.
- The scheduler runs as system—all classes are executed, whether the user has permission to execute the class or not.
- When the job's schedule is triggered, the system queues the batch job for processing. If Apex flex queue is enabled in your org, the batch job is added at the end of the flex queue. For more information, see Holding Batch Jobs in the Apex Flex Queue.
- All scheduled Apex limits apply for batch jobs scheduled using `System.scheduleBatch`. After the batch job is queued (with a status of `Holding` or `Queued`), all batch job limits apply and the job no longer counts toward scheduled Apex limits.
- After calling this method and before the batch job starts, you can use the returned scheduled job ID to abort the scheduled job using the `System.abortJob` method.

For an example, see Using Batch Apex.

`scheduleBatch(batchable, jobName, minutesFromNow, scopeSize)`

Schedules a batch job to run once in the future after the specified the time interval, with the specified job name and scope size. Returns the scheduled job ID (CronTrigger ID).

Signature

```
public static String scheduleBatch(Database.Batchable batchable, String jobName, Integer minutesFromNow, Integer scopeSize)
```

Parameters

batchable
Type: Database.Batchable

The batch class that implements the `Database.Batchable` interface.

jobName
Type: String

The name of the job that this method will start.

minutesFromNow
Type: Integer

The time interval in minutes after which the job should start executing.

scopeSize
Type: Integer

The number of records that should be passed to the batch `execute` method.

Return Value

Type: String

Usage

Note: Some things to note about `System.scheduleBatch`:

- When you call `System.scheduleBatch`, Salesforce schedules the job for execution at the specified time. Actual execution occurs at or after that time, depending on service availability.
- The scheduler runs as system—all classes are executed, whether the user has permission to execute the class or not.
- When the job's schedule is triggered, the system queues the batch job for processing. If Apex flex queue is enabled in your org, the batch job is added at the end of the flex queue. For more information, see Holding Batch Jobs in the Apex Flex Queue.
- All scheduled Apex limits apply for batch jobs scheduled using `System.scheduleBatch`. After the batch job is queued (with a status of `Holding` or `Queued`), all batch job limits apply and the job no longer counts toward scheduled Apex limits.
- After calling this method and before the batch job starts, you can use the returned scheduled job ID to abort the scheduled job using the `System.abortJob` method.

For an example, see Using the `System.scheduleBatch` Method.

`setPassword(userId, password)`

Sets the password for the specified user.

Signature

```
public static Void setPassword(ID userId, String password)
```

Parameters

`userId`
Type: ID

`password`
Type: String

Return Value

Type: Void

Usage

- If a security question hasn't been previously configured, a user who logs in with a new password that was set using `setPassword()` is redirected to the "Change Your Password" page.
- Use `resetPassword(userId, sendUserEmail)` if you want the user to go through the reset process and create their own password.

Warning: Be careful with this method, and don't expose this functionality to end users.

`submit(workItemIds, comments, nextApprover)`

Submits the processed approvals. The current user is the submitter and the entry criteria is evaluated for all processes applicable to the current user.

Signature

```
public static List<ID> submit(List<ID> workItemIds, String comments, String nextApprover)
```

Parameters

`workItemIds`
Type: List<ID>

`comments`
Type: String

`nextApprover`
Type: String

Return Value

Type: List<ID>

Usage

For enhanced submit and evaluation features, see the ProcessSubmitRequest class.

`today()`

Returns the current date in the current user's time zone.

Signature

```
public static Date today()
```

Return Value

Type: Date

Test Class

Contains methods related to Apex tests.

Namespace

System

Test Methods

The following are methods for `Test`. All methods are static.

IN THIS SECTION:

calculatePermissionSetGroup(psgIds)

Calculates aggregate permissions in specified permission set groups for testing.

calculatePermissionSetGroup(psgId)

Calculates aggregate permissions in a specified permission set group for testing.

clearApexPageMessages()

Clear the messages on a Visualforce page while executing Apex test methods.

createSoqlStub(targetType, soqlStub)

Creates a stub that will respond to SOQL queries against the specified SObject type you can use during testing.

createStub(parentType, stubProvider)

Creates a stubbed version of an Apex class that you can use for testing. This method is part of the Apex stub API. You can use it with the `System.StubProvider` interface to create a mocking framework.

createStubQueryRow(targetType, fieldMapWithRelationshipKeys)

Creates an instance of a stubbed SObject type that you can use to provide testing results in the extended `System.SoqlStubProvider` class.

createStubQueryRows(targetType, fieldMapWithRelationshipKeysForMultipleRows)

Creates instances of stubbed SObject types that you can use to provide testing results in the extended `System.SoqlStubProvider` class.

enableChangeDataCapture()

Use this method in an Apex test so that change event notifications are generated for all supported Change Data Capture entities. Call this method at the beginning of your test before performing DML operations and calling `Test.getEventBus().deliver();`.

enqueueBatchJobs(numberOfJobs)

Adds the specified number of jobs with no-operation contents to the test-context queue. It first fills the test batch queue, up to the maximum 5 jobs, and then places jobs in the test flex queue. It throws a limit exception when the number of jobs in the test flex queue exceeds the allowed limit of 100 jobs.

getEventBus()

Returns an instance of the test event bus broker, which lets you operate on platform event or change event messages in an Apex test. For example, you can call `Test.getEventBus().deliver()` to deliver event messages.

getFlexQueueOrder()

Returns an ordered list of job IDs for jobs in the test-context flex queue. The job at index `0` is the next job slated to run. This method returns only test-context results, even if it's annotated with `@IsTest(SeeAllData=true)`.

getStandardPricebookId()

Returns the ID of the standard price book in the organization.

invokeContinuationMethod(controller, request)

Invokes the callback method for the specified controller and continuation in a test method.

isRunningTest()

Returns `true` if the currently executing code was called by code contained in a test method, `false` otherwise. Use this method if you need to run different code depending on whether it was being called from a test.

isSoqlStubDefined(targetType)

Returns `true` if a SOQL stub is defined for an SObject type; otherwise returns `false`.

loadData(sObjectToken, resourceName)

Inserts test records from the specified static resource .csv file and for the specified sObject type, and returns a list of the inserted sObjects.

newSendEmailQuickActionDefaults(contextId, replyToId)

Creates a new QuickAction.SendEmailQuickActionDefaults instance for testing a class implementing the QuickAction.QuickActionDefaultsHandler interface.

setContinuationResponse(requestLabel, mockResponse)

Sets a mock response for a continuation HTTP request in a test method.

setCreatedDate(recordId, createdDatetime)

Sets `CreatedDate` for a test-context sObject.

setCurrentPage(page)

A Visualforce test method that sets the current PageReference for the controller.

setCurrentPageReference(page)

A Visualforce test method that sets the current PageReference for the controller.

setFixedSearchResults(fixedSearchResults)

Defines a list of fixed search results to be returned by all subsequent SOSL statements in a test method.

setMock(interfaceType, instance)

Sets the response mock mode and instructs the Apex runtime to send a mock response whenever a callout is made through the HTTP classes or the auto-generated code from WSDLs.

setReadOnlyApplicationMode(applicationMode)

Sets the application mode for an organization to read-only in an Apex test to simulate read-only mode during Salesforce upgrades and downtimes. The application mode is reset to the default mode at the end of each Apex test run.

startTest()

Marks the point in your test code when your test actually begins. Use this method when you are testing governor limits.

stopTest()

Marks the point in your test code when your test ends. Use this method in conjunction with the `startTest` method.

testInstall(installImplementation, version, isPush)

Tests the implementation of the InstallHandler interface, which is used for specifying a post install script in packages. Tests run as the test initiator in the development environment.

testSandboxPostCopyScript(script, organizationId, sandboxId, sandboxName)

Tests the implementation of the SandboxPostCopy Interface, which is used for specifying a script to run at the completion of a Sandbox copy. Tests run as the test initiator in the development environment.

testSandboxPostCopyScript(script, organizationId, sandboxId, sandboxName, RunAsAutoProcUser)

Tests the implementation of the SandboxPostCopy Interface, which is used for specifying a script to run at the completion of a Sandbox copy. When `RunAsAutoProcUser` is `true`, tests run as Automated Process user in the development environment.

testUninstall(uninstallImplementation)

Tests the implementation of the UninstallHandler interface, which is used for specifying an uninstall script in packages. Tests run as the test initiator in the development environment.

`calculatePermissionSetGroup(psgIds)`

Calculates aggregate permissions in specified permission set groups for testing.

Signature

```
public static void calculatePermissionSetGroup(List<String> psgIds)
```

Parameters

psgIds
Type: List<String>

A list of IDs for permission set groups.

Return Value

Type: void

calculatePermissionSetGroup(psgId)

Calculates aggregate permissions in a specified permission set group for testing.

Signature

```
public static void calculatePermissionSetGroup(String psgId)
```

Parameters

psgId
Type: String

A single ID for a specified permission set group.

Return Value

Type: void

clearApexPageMessages()

Clear the messages on a Visualforce page while executing Apex test methods.

Signature

```
public static void clearApexPageMessages()
```

Return Value

Type: void

Usage

This method may only be used in tests.

Example:

```
@isTest
static void clearMessagesTest() {
    Test.setCurrentPage(new PageReference('/'));
    ApexPages.addMessage(
        new ApexPages.Message(ApexPages.Severity.WARNING, 'Sample Warning')
    );
    System.assertEquals(1, ApexPages.getMessages().size());
    Test.clearApexPageMessages();
    System.assertEquals(0, ApexPages.getMessages().size());
}
```

`createSoqlStub(targetType, soqlStub)`

Creates a stub that will respond to SOQL queries against the specified SObject type you can use during testing.

Signature

```
public static void createSoqlStub(Schema.SObjectType targetType, System.SoqlStubProvider soqlStub)
```

Parameters

`targetType`
Type: Schema.SObjectType

The SObject type to be stubbed. This parameter can't be null.

`soqlStub`
Type: System.SoqlStubProvider

An implementation of the `SoqlStubProvider` abstract class.

Return Value

Type: void

`createStub(parentType, stubProvider)`

Creates a stubbed version of an Apex class that you can use for testing. This method is part of the Apex stub API. You can use it with the `System.StubProvider` interface to create a mocking framework.

Signature

```
public static Object createStub(System.Type parentType, System.StubProvider stubProvider)
```

Parameters

`parentType`
Type: System.Type

The type of the Apex class to be stubbed.

stubProvider
System.StubProvider

An implementation of the `StubProvider` interface.

Return Value

Type: Object

Returns the stubbed object to use in testing.

Usage

The `createStub()` method works together with the `System.StubProvider` interface. You define the behavior of the stubbed object by implementing the `StubProvider` interface. Then you create a stubbed object using the `createStub()` method. When you invoke methods on the stubbed object, the `handleMethodCall()` method of the `StubProvider` interface is called to perform the behavior of the stubbed method.

SEE ALSO:

Apex Developer Guide: Build a Mocking Framework with the Stub API

`createStubQueryRow(targetType, fieldMapWithRelationshipKeys)`

Creates an instance of a stubbed SObject type that you can use to provide testing results in the extended `System.SoqlStubProvider` class.

Signature

```
public static SObject createStubQueryRow(Schema.SObjectType targetType,
Map<String,Object> fieldMapWithRelationshipKeys)
```

Parameters

targetType
Type: Schema.SObjectType

The SObject type to be stubbed. This parameter can't be null.

fieldMapWithRelationshipKeys
Type: Map<String,Object>

The map contains the fields for a parent entity, keyed by the field name with a value for each field. Key and value pairs can also be used for an aggregate relationship. The key holds the name of the aggregate relationship and the value is a list of SObjects.

Return Value

Type: SObject

Returns the stubbed SObject to use in testing.

Example

```
Map<String, Object> engagement1 = new Map<String, Object> {
                'id' => '0gj000000000XYZ'
            };
Map<String, Object> engagement2 = new Map<String, Object> {
                'id' => '0gj000000001XYZ'
            };

engagementMaps.add(engagement1);
engagementMaps.add(engagement2);

            List<Engagement__dlm> engagements = (List<Engagement__dlm>)
            Test.createStubQueryRows(Engagement__dlm.SObjectType,
                                                    engagementMaps);

            Map<String, Object> contactMap = new Map<String, Object> {
                'id' => '003000000000000',
                'LastName' => 'Bear',
                'engagements__r' => engagements
            };

            Contact obj = (Contact) Test.createStubQueryRow(sobjectType, contactMap);
```

`createStubQueryRows(targetType, fieldMapWithRelationshipKeysForMultipleRows)`

Creates instances of stubbed SObject types that you can use to provide testing results in the extended `System.SoqlStubProvider` class.

Signature

```
public static List<SObject> createStubQueryRows(Schema.SObjectType targetType,
List<Map<String,Object>> fieldMapWithRelationshipKeysForMultipleRows)
```

Parameters

`targetType`
Type: Schema.SObjectType

The SObject type to be stubbed. This parameter can't be null.

`fieldMapWithRelationshipKeysForMultipleRows`
Type: List<Map<String,Object>>

The list of maps containing the fields for a parent entity, keyed by the field name with a value for each field. Key and value pairs can also be used for an aggregate relationship used in the query. The key holds the name of the aggregate relationship and the value is a list of SObjects.

Return Value

Type: List<SObject>

Returns a list of stubbed SObject types to use in testing.

Example

```
Map<String, Object> engagement = new Map<String, Object> {
                'id' => '0gjxx0000000XYZ'
            };
            Map<String, Object> engagement2 = new Map<String, Object> {
                'id' => '0gjxx0000001XYZ'
            };

            engagementMaps.add(engagement);
            engagementMaps.add(engagement2);

            List<~~ENGAGEMENT_DMO_NAME~~> engagements = (List<~~ENGAGEMENT_DMO_NAME~~>)
            Test.createStubQueryRows(~~ENGAGEMENT_DMO_NAME~~.SObjectType,
                                                    engagementMaps);

            Map<String, Object> contactMap = new Map<String, Object> {
                'id' => '003000000000000',
                'LastName' => 'Bear',
                '~~DMO_REL_NAME~~' => engagements
            };

            Contact obj = (Contact) Test.createStubQueryRows(sobjectType, contactMap);
```

`enableChangeDataCapture()`

Use this method in an Apex test so that change event notifications are generated for all supported Change Data Capture entities. Call this method at the beginning of your test before performing DML operations and calling `Test.getEventBus().deliver();`.

Signature

```
public static void enableChangeDataCapture()
```

Return Value

Type: void

Usage

The `enableChangeDataCapture()` method ensures that Apex tests can fire change event triggers regardless of the entities selected in Setup in the Change Data Capture page. The `enableChangeDataCapture()` method doesn't affect the entities selected in Setup.

SEE ALSO:

Change Data Capture Developer Guide

`enqueueBatchJobs(numberOfJobs)`

Adds the specified number of jobs with no-operation contents to the test-context queue. It first fills the test batch queue, up to the maximum 5 jobs, and then places jobs in the test flex queue. It throws a limit exception when the number of jobs in the test flex queue exceeds the allowed limit of 100 jobs.

Signature

```
public static List<Id> enqueueBatchJobs(Integer numberOfJobs)
```

Parameters

numberOfJobs
Type: Integer

Number of test jobs to enqueue.

Return Value

Type: List<Id>

A list of IDs of enqueued test jobs.

Usage

Use this method to reduce testing time. Instead of using your org's real batch jobs for testing, you can use this method to simulate batch-job enqueueing. Using `enqueueBatchJobs(numberOfJobs)` is faster than enqueuing real batch jobs.

`getEventBus()`

Returns an instance of the test event bus broker, which lets you operate on platform event or change event messages in an Apex test. For example, you can call `Test.getEventBus().deliver()` to deliver event messages.

Signature

```
public static EventBus.TestBroker getEventBus()
```

Return Value

Type: EventBus.TestBroker

A broker for the test event bus.

Usage

Enclose `Test.getEventBus().deliver()` within the `Test.startTest()` and `Test.stopTest()` statement block.

```
Test.startTest();
// Create test events
// ...
// Publish test events with EventBus.publish()
// ...
// Deliver test events
Test.getEventBus().deliver();
// Perform validation
```

```
// ...
Test.stopTest();
```

SEE ALSO:

Platform Events Developer Guide

`getFlexQueueOrder()`

Returns an ordered list of job IDs for jobs in the test-context flex queue. The job at index `0` is the next job slated to run. This method returns only test-context results, even if it's annotated with `@IsTest(SeeAllData=true)`.

Signature

```
public static List<Id> getFlexQueueOrder()
```

Return Value

Type: List<Id>

An ordered list of IDs of the jobs in the test's flex queue.

`getStandardPricebookId()`

Returns the ID of the standard price book in the organization.

Signature

```
public static Id getStandardPricebookId()
```

Return Value

Type: Id

The ID of the standard price book.

Usage

This method returns the ID of the standard price book in your organization regardless of whether the test can query organization data. By default, tests can't query organization data unless they're annotated with `@isTest(SeeAllData=true)`.

Creating price book entries with a standard price requires the ID of the standard price book. Use this method to get the standard price book ID so that you can create price book entries in your tests.

Example

This example creates some test data for price book entries. The test method in this example gets the standard price book ID and uses this ID to create a price book entry for a product with a standard price. Next, the test creates a custom price book and uses the ID of this custom price book to add a price book entry with a custom price.

```
@isTest
public class PriceBookTest {
    // Utility method that can be called by Apex tests to create price book entries.
```

```
    static testmethod void addPricebookEntries() {
        // First, set up test price book entries.
        // Insert a test product.
        Product2 prod = new Product2(Name = 'Laptop X200',
            Family = 'Hardware');
        insert prod;

        // Get standard price book ID.
        // This is available irrespective of the state of SeeAllData.
        Id pricebookId = Test.getStandardPricebookId();

        // 1. Insert a price book entry for the standard price book.
        // Standard price book entries require the standard price book ID we got earlier.

        PricebookEntry standardPrice = new PricebookEntry(
            Pricebook2Id = pricebookId, Product2Id = prod.Id,
            UnitPrice = 10000, IsActive = true);
        insert standardPrice;

        // Create a custom price book
        Pricebook2 customPB = new Pricebook2(Name='Custom Pricebook', isActive=true);
        insert customPB;

        // 2. Insert a price book entry with a custom price.
        PricebookEntry customPrice = new PricebookEntry(
            Pricebook2Id = customPB.Id, Product2Id = prod.Id,
            UnitPrice = 12000, IsActive = true);
        insert customPrice;

        // Next, perform some tests with your test price book entries.
    }
}
```

`invokeContinuationMethod(controller, request)`

Invokes the callback method for the specified controller and continuation in a test method.

Signature

```
public static Object invokeContinuationMethod(Object controller, Continuation request)
```

Parameters

`controller`
Type: Object

An instance of the controller class that invokes the continuation request.

`request`
Type: Continuation

The continuation that is returned by an action method in the controller class.

Return Value

Type: Object

The response of the continuation callback method.

Usage

Use the `Test.setContinuationResponse` and `Test.invokeContinuationMethod` methods to test continuations. In test context, callouts of continuations aren't sent to the external service. By using these methods, you can set a mock response and cause the runtime to call the continuation callback method to process the mock response.

Call `Test.setContinuationResponse` before you call `Test.invokeContinuationMethod`. When you call `Test.invokeContinuationMethod`, the runtime executes the callback method that is associated with the continuation. The callback method processes the mock response that is set by `Test.setContinuationResponse`.

`isRunningTest()`

Returns `true` if the currently executing code was called by code contained in a test method, `false` otherwise. Use this method if you need to run different code depending on whether it was being called from a test.

Signature

```
public static Boolean isRunningTest()
```

Return Value

Type: Boolean

`isSoqlStubDefined(targetType)`

Returns `true` if a SOQL stub is defined for an SObject type; otherwise returns `false`.

Signature

```
public static Boolean isSoqlStubDefined(Schema.SObjectType targetType)
```

Parameters

`targetType`
Type: Schema.SObjectType

The SObject type to check. This parameter can't be null.

Return Value

Type: Boolean

`loadData(sObjectToken, resourceName)`

Inserts test records from the specified static resource .csv file and for the specified sObject type, and returns a list of the inserted sObjects.

Signature

```
public static List<sObject> loadData(Schema.SObjectType sObjectToken, String
resourceName)
```

Parameters

sObjectToken
Type: Schema.SObjectType

The sObject type for which to insert test records.

resourceName
Type: String

The static resource that corresponds to the .csv file containing the test records to load. The name is case insensitive.

Return Value

Type: List<sObject>

Usage

You must create the static resource prior to calling this method. The static resource is a comma-delimited file ending with a .csv extension. The file contains field names and values for the test records. The first line of the file must contain the field names and subsequent lines are the field values. To learn more about static resources, see "Defining Static Resources" in the Salesforce online help.

Once you create a static resource for your .csv file, the static resource will be assigned a MIME type. Supported MIME types are:

- text/csv
- application/vnd.ms-excel
- application/octet-stream
- text/plain

newSendEmailQuickActionDefaults(contextId, replyToId)

Creates a new QuickAction.SendEmailQuickActionDefaults instance for testing a class implementing the QuickAction.QuickActionDefaultsHandler interface.

Signature

```
public static QuickAction.SendEmailQuickActionDefaults newSendEmailQuickActionDefaults(ID
contextId, ID replyToId)
```

Parameters

contextId
Type: Id

Parent record of the email message.

replyToId
Type: Id

Previous email message ID if this email message is a reply.

Return Value

Type: SendEmailQuickActionDefaults Class

The default values used for an email message quick action.

setContinuationResponse(requestLabel, mockResponse)

Sets a mock response for a continuation HTTP request in a test method.

Signature

```
public static void setContinuationResponse(String requestLabel, System.HttpResponse mockResponse)
```

Parameters

requestLabel
Type: String

The unique label that corresponds to the continuation HTTP request. This label is returned by `Continuation.addHttpRequest`.

mockResponse
Type: HttpResponse

The fake response to be returned by `Test.invokeContinuationMethod`.

Return Value

Type: void

Usage

Use the `Test.setContinuationResponse` and `Test.invokeContinuationMethod` methods to test continuations. In test context, callouts of continuations aren't sent to the external service. By using these methods, you can set a mock response and cause the runtime to call the continuation callback method to process the mock response.

Call `Test.setContinuationResponse` before you call `Test.invokeContinuationMethod`. When you call `Test.invokeContinuationMethod`, the runtime executes the callback method that is associated with the continuation. The callback method processes the mock response that is set by `Test.setContinuationResponse`.

setCreatedDate(recordId, createdDatetime)

Sets `CreatedDate` for a test-context sObject.

Signature

```
public static void setCreatedDate(Id recordId, Datetime createdDatetime)
```

Parameters

recordId
Type: Id

The ID of an sObject.

createdDatetime
Type: Datetime

The value to assign to the sObject's `CreatedDate` field.

Return Value

Type: void

Usage

All database changes are rolled back at the end of a test. You can't use this method on records that existed before your test executed. You also can't use `setCreatedDate` in methods annotated with `@isTest(SeeAllData=true)`, because those methods have access to all data in your org. If you set `CreatedDate` to a future value, it can cause unexpected results. This method takes two parameters—an sObject ID and a Datetime value—neither of which can be null.

Insert your test record before you set its `CreatedDate`, as shown in this example.

```
@isTest
private class SetCreatedDateTest {
    static testMethod void testSetCreatedDate() {
        Account a = new Account(name='myAccount');
        insert a;
        Test.setCreatedDate(a.Id, DateTime.newInstance(2012,12,12));
        Test.startTest();
        Account myAccount = [SELECT Id, Name, CreatedDate FROM Account
                             WHERE Name ='myAccount' limit 1];
        System.assertEquals(myAccount.CreatedDate, DateTime.newInstance(2012,12,12));
        Test.stopTest();
    }
}
```

setCurrentPage(page)

A Visualforce test method that sets the current PageReference for the controller.

Signature

```
public static Void setCurrentPage(PageReference page)
```

Parameters

page
Type: System.PageReference

Return Value

Type: Void

setCurrentPageReference(page)

A Visualforce test method that sets the current PageReference for the controller.

Signature

```
public static Void setCurrentPageReference(PageReference page)
```

Parameters

page
Type: System.PageReference

Return Value

Type: Void

setFixedSearchResults(fixedSearchResults)

Defines a list of fixed search results to be returned by all subsequent SOSL statements in a test method.

Signature

```
public static Void setFixedSearchResults(ID[] fixedSearchResults)
```

Parameters

fixedSearchResults
Type: ID[]

The list of record IDs specified by *opt_set_search_results* replaces the results that would normally be returned by the SOSL queries if they were not subject to any `WHERE` or `LIMIT` clauses. If these clauses exist in the SOSL queries, they are applied to the list of fixed search results.

Return Value

Type: Void

Usage

If *opt_set_search_results* is not specified, all subsequent SOSL queries return no results.

For more information, see Dynamic SOSL.

setMock(interfaceType, instance)

Sets the response mock mode and instructs the Apex runtime to send a mock response whenever a callout is made through the HTTP classes or the auto-generated code from WSDLs.

Signature

```
public static Void setMock(Type interfaceType, Object instance)
```

Parameters

interfaceType
Type: System.Type

instance
Type: Object

Return Value

Type: Void

Usage

Note: To mock a callout if the code that performs the callout is in a managed package, call `Test.setMock` from a test method in the same package with the same namespace.

setReadOnlyApplicationMode(applicationMode)

Sets the application mode for an organization to read-only in an Apex test to simulate read-only mode during Salesforce upgrades and downtimes. The application mode is reset to the default mode at the end of each Apex test run.

Signature

```
public static Void setReadOnlyApplicationMode(Boolean applicationMode)
```

Parameters

applicationMode
Type: Boolean

Return Value

Type: Void

Usage

Also see the `getApplicationReadWriteMode()` System method.

Do not use `setReadOnlyApplicationMode` for purposes unrelated to Read-Only Mode testing, such as simulating DML exceptions.

Example

The following example sets the application mode to read-only and attempts to insert a new account record, which results in the exception. It then resets the application mode and performs a successful insert.

```
@isTest
private class ApplicationReadOnlyModeTestClass {
  public static testmethod void test() {
    // Create a test account that is used for querying later.
    Account testAccount = new Account(Name = 'TestAccount');
    insert testAccount;

    // Set the application read only mode.
    Test.setReadOnlyApplicationMode(true);
```

```
    // Verify that the application is in read-only mode.
    System.assertEquals(
                ApplicationReadWriteMode.READ_ONLY,
                System.getApplicationReadWriteMode());

    // Create a new account object.
    Account testAccount2 = new Account(Name = 'TestAccount2');

    try {
      // Get the test account created earlier. Should be successful.
      Account testAccountFromDb =
        [SELECT Id, Name FROM Account WHERE Name = 'TestAccount'];
      System.assertEquals(testAccount.Id, testAccountFromDb.Id);

      // Inserts should result in the InvalidReadOnlyUserDmlException
      // being thrown.
      insert testAccount2;
      System.assertEquals(false, true);
    } catch (System.InvalidReadOnlyUserDmlException e) {
      // Expected
    }
    // Insertion should work after read only application mode gets disabled.
    Test.setReadOnlyApplicationMode(false);

    insert testAccount2;
    Account testAccount2FromDb =
       [SELECT Id, Name FROM Account WHERE Name = 'TestAccount2'];
    System.assertEquals(testAccount2.Id, testAccount2FromDb.Id);
  }
}
```

`startTest()`

Marks the point in your test code when your test actually begins. Use this method when you are testing governor limits.

Signature

```
public static Void startTest()
```

Return Value

Type: Void

Usage

You can also use this method with `stopTest` to ensure that all asynchronous calls that come after the `startTest` method are run before doing any assertions or testing. Each test method is allowed to call this method only once. All of the code before this method should be used to initialize variables, populate data structures, and so on, allowing you to set up everything you need to run your test. Any code that executes after the call to `startTest` and before `stopTest` is assigned a new set of governor limits.

stopTest()

Marks the point in your test code when your test ends. Use this method in conjunction with the `startTest` method.

Signature

```
public static Void stopTest()
```

Return Value

Type: Void

Usage

Each test method is allowed to call this method only once. Any code that executes after the `stopTest` method is assigned the original limits that were in effect before `startTest` was called. All asynchronous calls made after the `startTest` method are collected by the system. When `stopTest` is executed, all asynchronous processes are run synchronously.

Note: Asynchronous calls, such as `@future` or `executeBatch`, called in a `startTest`, `stopTest` block, do not count against your limits for the number of queued jobs.

testInstall(installImplementation, version, isPush)

Tests the implementation of the InstallHandler interface, which is used for specifying a post install script in packages. Tests run as the test initiator in the development environment.

Signature

```
public static Void testInstall(InstallHandler installImplementation, Version version,
Boolean isPush)
```

Parameters

installImplementation
Type: System.InstallHandler

A class that implements the `InstallHandler` interface.

version
Type: System.Version

The version number of the existing package installed in the subscriber organization.

isPush
Type: Boolean

(Optional) Specifies whether the upgrade is a push. The default value is `false`.

Return Value

Type: Void

Usage

This method throws a run-time exception if the test install fails.

Example

```
@isTest static void test() {
  PostInstallClass postinstall =
    new PostInstallClass();
    Test.testInstall(postinstall,
      new Version(1,0));
  }
```

testSandboxPostCopyScript(script, organizationId, sandboxId, sandboxName)

Tests the implementation of the SandboxPostCopy Interface, which is used for specifying a script to run at the completion of a Sandbox copy. Tests run as the test initiator in the development environment.

Signature

```
public static void testSandboxPostCopyScript(System.SandboxPostCopy script, Id
organizationId, Id sandboxId, String sandboxName)
```

Parameters

script
Type: System.SandboxPostCopy

A class that implements the `SandboxPostCopy` interface.

organizationId
Type: Id

The sandbox organization ID

sandboxId
Type: Id

The sandbox ID to be provided to the SandboxPostCopy script.

sandboxName
Type: String

The sandbox name to be provided to the SandboxPostCopy script.

Return Value

Type: void

Usage

This method throws a run-time exception if the test install fails.

Note: Salesforce recommends that you use the `testSandboxPostCopyScript(script, organizationId, sandboxId, sandboxName, isRunAsAutoProcUser)` overload instead of this method. When `isRunAsAutoProcUser` is `true`, the `SandboxPostCopy` script is tested with the same user access permissions as used by post-copy tasks during sandbox creation. Using the same permissions enables the test to better simulate the actual usage of the class, and to uncover potential issues.

Example

See SandboxPostCopy Example Implementation

testSandboxPostCopyScript(script, organizationId, sandboxId, sandboxName, RunAsAutoProcUser)

Tests the implementation of the SandboxPostCopy Interface, which is used for specifying a script to run at the completion of a Sandbox copy. When `RunAsAutoProcUser` is `true`, tests run as Automated Process user in the development environment.

Signature

```
public static void testSandboxPostCopyScript(System.SandboxPostCopy script, Id
organizationId, Id sandboxId, String sandboxName, Boolean RunAsAutoProcUser)
```

Parameters

`script`
Type: System.SandboxPostCopy

A class that implements the `SandboxPostCopy` interface.

`organizationId`
Type: Id

The sandbox organization ID.

`sandboxId`
Type: Id

The sandbox ID to be provided to the SandboxPostCopy script.

`sandboxName`
Type: String

The sandbox name to be provided to the SandboxPostCopy script.

`RunAsAutoProcUser`
Type: Boolean

When `true`, the `SandboxPostCopy` script is tested with the same user access permissions as used by post-copy tasks during sandbox creation. Using the same permissions enables the test to better simulate the actual usage of the class, and to uncover potential issues.

When `false`, the test runs as the test initiator. This option can alter the permissions with which the script is tested, such as the ability to access objects and features.

Return Value

Type: void

Usage

This method throws a run-time exception if the test install fails.

Example

See SandboxPostCopy Example Implementation

`testUninstall(uninstallImplementation)`

Tests the implementation of the UninstallHandler interface, which is used for specifying an uninstall script in packages. Tests run as the test initiator in the development environment.

Signature

```
public static Void testUninstall(UninstallHandler uninstallImplementation)
```

Parameters

`uninstallImplementation`
Type: System.UninstallHandler

A class that implements the `UninstallHandler` interface.

Return Value

Type: Void

Usage

This method throws a run-time exception if the test uninstall fails.

Example

```
@isTest static void test() {
  UninstallClass uninstall =
    new UninstallClass();
    Test.testUninstall(uninstall);
  }
```

Time Class

Contains methods for the Time primitive data type.

Namespace

System

Usage

For more information on time, see Time Data Type.

Time Methods

The following are methods for `Time`.

IN THIS SECTION:

addHours(additionalHours)
Adds the specified number of hours to a Time.

addMilliseconds(additionalMilliseconds)
Adds the specified number of milliseconds to a Time.

addMinutes(additionalMinutes)
Adds the specified number of minutes to a Time.

addSeconds(additionalSeconds)
Adds the specified number of seconds to a Time.

hour()
Returns the hour component of a Time.

millisecond()
Returns the millisecond component of a Time.

minute()
Returns the minute component of a Time.

newInstance(hour, minutes, seconds, milliseconds)
Constructs a Time from Integer representations of the specified hour, minutes, seconds, and milliseconds. (UTC is assumed.)

second()
Returns the second component of a Time.

addHours(additionalHours)

Adds the specified number of hours to a Time.

Signature

```
public Time addHours(Integer additionalHours)
```

Parameters

additionalHours
Type: Integer

Return Value

Type: Time

Example

```
Time myTime = Time.newInstance(1, 2, 3, 4);
Time expected = Time.newInstance(4, 2, 3, 4);
System.assertEquals(expected, myTime.addHours(3));
```

addMilliseconds(additionalMilliseconds)

Adds the specified number of milliseconds to a Time.

Signature

```
public Time addMilliseconds(Integer additionalMilliseconds)
```

Parameters

additionalMilliseconds
Type: Integer

Return Value

Type: Time

Example

```
Time myTime = Time.newInstance(1, 2, 3, 0);
Time expected = Time.newInstance(1, 2, 4, 400);
System.assertEquals(expected, myTime.addMilliseconds(1400));
```

addMinutes(additionalMinutes)

Adds the specified number of minutes to a Time.

Signature

```
public Time addMinutes(Integer additionalMinutes)
```

Parameters

additionalMinutes
Type: Integer

Return Value

Type: Time

Example

```
Time myTime = Time.newInstance(18, 30, 2, 20);
Integer myMinutes = myTime.minute();
myMinutes = myMinutes + 5;
System.assertEquals(myMinutes, 35);
```

addSeconds(additionalSeconds)

Adds the specified number of seconds to a Time.

Signature

```
public Time addSeconds(Integer additionalSeconds)
```

Parameters

additionalSeconds
Type: Integer

Return Value

Type: Time

Example

```
Time myTime = Time.newInstance(1, 2, 55, 0);
Time expected = Time.newInstance(1, 3, 5, 0);
System.assertEquals(expected, myTime.addSeconds(10));
```

hour()

Returns the hour component of a Time.

Signature

```
public Integer hour()
```

Return Value

Type: Integer

Example

```
Time myTime = Time.newInstance(18, 30, 2, 20);
myTime = myTime.addHours(2);
Integer myHour = myTime.hour();
System.assertEquals(myHour, 20);
```

millisecond()

Returns the millisecond component of a Time.

Signature

```
public Integer millisecond()
```

Return Value

Type: Integer

Example

```
Time myTime = Time.newInstance(3, 14, 15, 926);
System.assertEquals(926, myTime.millisecond());
```

minute()

Returns the minute component of a Time.

Signature

```
public Integer minute()
```

Return Value

Type: Integer

Example

```
Time myTime = Time.newInstance(3, 14, 15, 926);
System.assertEquals(14, myTime.minute());
```

newInstance(hour, minutes, seconds, milliseconds)

Constructs a Time from Integer representations of the specified hour, minutes, seconds, and milliseconds. (UTC is assumed.)

Signature

```
public static Time newInstance(Integer hour, Integer minutes, Integer seconds, Integer
milliseconds)
```

Parameters

hour
Type: Integer

minutes
Type: Integer

seconds
Type: Integer

milliseconds
Type: Integer

Return Value

Type: Time

Example

The following example creates a time of 18:30:2:20 (UTC).

```
Time myTime =
Time.newInstance(18, 30, 2, 20);
```

second()

Returns the second component of a Time.

Signature

```
public Integer second()
```

Return Value

Type: Integer

Example

```
Time myTime = Time.newInstance(3, 14, 15, 926);
System.assertEquals(15, myTime.second());
```

TimeZone Class

Represents a time zone. Contains methods for creating a new time zone and obtaining time zone properties, such as the time zone ID, offset, and display name.

Namespace

System

Usage

You can use the methods in this class to get properties of a time zone, such as the properties of the time zone returned by `UserInfo.getTimeZone`, or the time zone returned by `getTimeZone` of this class.

Example

This example shows how to get properties of the current user's time zone and display them to the debug log. The output of the sample varies based on the user's time zone.

```
TimeZone tz = UserInfo.getTimeZone();
System.debug('Display name: ' + tz.getDisplayName());
System.debug('ID: ' + tz.getID());
// During daylight saving time for the America/Los_Angeles time zone
System.debug('Offset: ' + tz.getOffset(DateTime.newInstance(2012,10,23,12,0,0)));
// Not during daylight saving time for the America/Los_Angeles time zone
System.debug('Offset: ' + tz.getOffset(DateTime.newInstance(2012,11,23,12,0,0)));
System.debug('String format: ' + tz.toString());
```

This second example shows how to create a time zone for the New York time zone and get the offset of this time zone to the GMT time zone. The example uses two dates to get the offset from. One date is before DST (Daylight Saving Time), and one is after DST. In 2000,

DST ended on Sunday, October 29 for the New York time zone. Because the date occurs after DST ends, the offset on the first date is –5 hours to GMT. In 2012, DST ended on Sunday, November 4. Because the date is within DST, the offset on the second date is –4 hours.

```
// Get the New York time zone
Timezone tz = Timezone.getTimeZone('America/New_York');

// Create a date before the 2007 shift of DST into November
DateTime dtpre = DateTime.newInstanceGMT(2000, 11, 1, 0, 0, 0);
system.debug(tz.getOffset(dtpre));   //-18000000 (= -5 hours = EST)

// Create a date after the 2007 shift of DST into November
DateTime dtpost = DateTime.newInstanceGMT(2012, 11, 1, 0, 0, 0);
system.debug(tz.getOffset(dtpost));   //-14400000 (= -4 hours = EDT)
```

This next example is similar to the previous one except that it gets the offset around the boundary of DST. In 2014, DST ended on Sunday, November 2 at 2:00 AM local time for the New York time zone. The first offset is obtained right before DST ends, and the second offset is obtained right after DST ends. The dates are created by using the `DateTime.newInstanceGMT` method. This method expects the passed-in date values to be based on the GMT time zone.

```
// Get the New York time zone
Timezone tz = Timezone.getTimeZone('America/New_York');

// Before DST ends
DateTime dtpre = DateTime.newInstanceGMT(2014, 11, 2, 5, 59, 59);  //1:59:59AM local EDT
system.debug(tz.getOffset(dtpre));   //-14400000 (= -4 hours = still on DST)

// After DST ends
DateTime dtpost = DateTime.newInstanceGMT(2014, 11, 2, 6, 0, 0); //1:00:00AM local EST
system.debug(tz.getOffset(dtpost));  //-18000000 (= -5 hours = back one hour)
```

TimeZone Methods

The following are methods for `TimeZone`.

IN THIS SECTION:

getDisplayName()
Returns this time zone's display name.

getID()
Returns this time zone's ID.

getOffset(date)
Returns the time zone offset, in milliseconds, of the specified date to the GMT time zone.

getTimeZone(timeZoneIdString)
Returns the time zone corresponding to the specified time zone ID.

toString()
Returns the string representation of this time zone.

`getDisplayName()`

Returns this time zone's display name.

Signature

```
public String getDisplayName()
```

Return Value

Type: String

Versioned Behavior Changes

In API version 45.0 and later, getDisplayName displays Daylight Savings Time appropriately when daylight savings are in effect. For example, British Summer Time is displayed for Europe/London and Pacific Daylight Time for America/Los_Angeles.

getID()

Returns this time zone's ID.

Signature

```
public String getID()
```

Return Value

Type: String

getOffset(date)

Returns the time zone offset, in milliseconds, of the specified date to the GMT time zone.

Signature

```
public Integer getOffset(Datetime date)
```

Parameters

date

Type: Datetime

The *date* argument is the date and time to evaluate.

Return Value

Type: Integer

Usage

Note: The returned offset is adjusted for daylight saving time if the *date* argument falls within daylight saving time for this time zone.

getTimeZone(timeZoneIdString)

Returns the time zone corresponding to the specified time zone ID.

Signature

```
public static TimeZone getTimeZone(String timeZoneIdString)
```

Parameters

timeZoneIdString
Type: String

The time zone values you can use for the *Id* argument are any valid time zone values that the Java TimeZone class supports.

Return Value

Type: TimeZone

Example

```
TimeZone tz = TimeZone.getTimeZone('America/Los_Angeles');
String tzName = tz.getDisplayName();
System.assert(tzName.equals('(GMT-08:00) Pacific Standard Time (America/Los_Angeles)') ||

              tzName.equals('(GMT-07:00) Pacific Daylight Time (America/Los_Angeles)'));
```

`toString()`

Returns the string representation of this time zone.

Signature

```
public String toString()
```

Return Value

Type: String

Trigger Class

Use the `Trigger` class to access run-time context information in a trigger, such as the type of trigger or the list of sObject records that the trigger operates on.

Namespace

System

Trigger Context Variables

The `Trigger` class provides the following context variables.

Variable	Usage
`isExecuting`	Returns true if the current context for the Apex code is a trigger, not a Visualforce page, a Web service, or an `executeanonymous()` API call.
`isInsert`	Returns `true` if this trigger was fired due to an insert operation, from the Salesforce user interface, Apex, or the API.
`isUpdate`	Returns `true` if this trigger was fired due to an update operation, from the Salesforce user interface, Apex, or the API.
`isDelete`	Returns `true` if this trigger was fired due to a delete operation, from the Salesforce user interface, Apex, or the API.
`isBefore`	Returns `true` if this trigger was fired before any record was saved.
`isAfter`	Returns `true` if this trigger was fired after all records were saved.
`isUndelete`	Returns `true` if this trigger was fired after a record is recovered from the Recycle Bin. This recovery can occur after an undelete operation from the Salesforce user interface, Apex, or the API.
`new`	Returns a list of the new versions of the sObject records. This sObject list is only available in `insert`, `update`, and `undelete` triggers, and the records can only be modified in `before` triggers.
`newMap`	A map of IDs to the new versions of the sObject records. This map is only available in `before update`, `after insert`, `after update`, and `after undelete` triggers.
`old`	Returns a list of the old versions of the sObject records. This sObject list is only available in `update` and `delete` triggers.
`oldMap`	A map of IDs to the old versions of the sObject records. This map is only available in `update` and `delete` triggers.
`operationType`	Returns an enum of type System.TriggerOperation corresponding to the current operation. Possible values of the System.TriggerOperation enum are: `BEFORE_INSERT`, `BEFORE_UPDATE`, `BEFORE_DELETE`,`AFTER_INSERT`, `AFTER_UPDATE`, `AFTER_DELETE`, and `AFTER_UNDELETE`. If you vary your programming logic based on different trigger types, consider using the `switch` statement with different permutations of unique trigger execution enum states.
`size`	The total number of records in a trigger invocation, both old and new.

Note: The record firing a trigger can include an invalid field value, such as a formula that divides by zero. In this case, the field value is set to `null` in these variables:

- `new`
- `newMap`
- `old`
- `oldMap`

Example

For example, in this simple trigger, `Trigger.new` is a list of sObjects and can be iterated over in a `for` loop. It can also be used as a bind variable in the `IN` clause of a SOQL query.

```
Trigger simpleTrigger on Account (after insert) {
    for (Account a : Trigger.new) {
        // Iterate over each sObject
    }

    // This single query finds every contact that is associated with any of the
    // triggering accounts. Note that although Trigger.new is a collection of
    // records, when used as a bind variable in a SOQL query, Apex automatically
    // transforms the list of records into a list of corresponding Ids.
    Contact[] cons = [SELECT LastName FROM Contact
                      WHERE AccountId IN :Trigger.new];
}
```

This trigger uses Boolean context variables like `Trigger.isBefore` and `Trigger.isDelete` to define code that only executes for specific trigger conditions:

```
trigger myAccountTrigger on Account(before delete, before insert, before update,
                                    after delete, after insert, after update) {
if (Trigger.isBefore) {
    if (Trigger.isDelete) {

        // In a before delete trigger, the trigger accesses the records that will be
        // deleted with the Trigger.old list.
        for (Account a : Trigger.old) {
            if (a.name != 'okToDelete') {
                a.addError('You can\'t delete this record!');
            }
        }
    } else {

    // In before insert or before update triggers, the trigger accesses the new records
    // with the Trigger.new list.
        for (Account a : Trigger.new) {
            if (a.name == 'bad') {
                a.name.addError('Bad name');
            }
    }
    if (Trigger.isInsert) {
        for (Account a : Trigger.new) {
            System.assertEquals('xxx', a.accountNumber);
            System.assertEquals('industry', a.industry);
            System.assertEquals(100, a.numberofemployees);
            System.assertEquals(100.0, a.annualrevenue);
            a.accountNumber = 'yyy';
        }

// If the trigger is not a before trigger, it must be an after trigger.
} else {
    if (Trigger.isInsert) {
        List<Contact> contacts = new List<Contact>();
```

```
        for (Account a : Trigger.new) {
            if(a.Name == 'makeContact') {
                contacts.add(new Contact (LastName = a.Name,
                                          AccountId = a.Id));
            }
        }
      insert contacts;
    }
  }
}}}
```

TriggerOperation Enum

System.TriggerOperation enum values are associated with trigger events.

Enum Values

The following are the values of the `System.TriggerOperation` enum:

- AFTER_DELETE
- AFTER_INSERT
- AFTER_UNDELETE
- AFTER_UPDATE
- BEFORE_DELETE
- BEFORE_INSERT
- BEFORE_UPDATE

Type Class

Contains methods for getting the Apex type that corresponds to an Apex class and for instantiating new types.

Namespace

System

Usage

Use the `forName` methods to retrieve the type of an Apex class, which can be a built-in or a user-defined class. You can use these methods to retrieve the type of public and global classes, and not private classes even if the context user has access. Also, use the `newInstance` method if you want to instantiate a Type that implements an interface and call its methods while letting someone else, such as a subscriber of your package, provide the methods' implementations.

Note: A call to `Type.forName()` can cause the class to be compiled.

Example: Instantiating a Type Based on Its Name

The following sample shows how to use the Type methods to instantiate a Type based on its name. A typical application of this scenario is when a package subscriber provides a custom implementation of an interface that is part of an installed package. The package can

get the name of the class that implements the interface through a custom setting in the subscriber's org. The package can then instantiate the type that corresponds to this class name and invoke the methods that the subscriber implemented.

In this sample, `Vehicle` represents the interface that the `VehicleImpl` class implements. The last class contains the code sample that invokes the methods implemented in `VehicleImpl`.

This is the `Vehicle` interface.

```
global interface Vehicle {
    Long getMaxSpeed();
    String getType();
}
```

This is the implementation of the `Vehicle` interface.

```
global class VehicleImpl implements Vehicle {
    global Long getMaxSpeed() { return 100; }
    global String getType() { return 'Sedan'; }
}
```

The method in this class gets the name of the class that implements the `Vehicle` interface through a custom setting value. It then instantiates this class by getting the corresponding type and calling the `newInstance` method. Next, it invokes the methods implemented in `VehicleImpl`. This sample requires that you create a public list custom setting named *`CustomImplementation`* with a text field named *`className`*. Create one record for this custom setting with a data set name of *`Vehicle`* and a class name value of *`VehicleImpl`*.

```
public class CustomerImplInvocationClass {

    public static void invokeCustomImpl() {
        // Get the class name from a custom setting.
        // This class implements the Vehicle interface.
        CustomImplementation__c cs = CustomImplementation__c.getInstance('Vehicle');

        // Get the Type corresponding to the class name
        Type t = Type.forName(cs.className__c);

        // Instantiate the type.
        // The type of the instantiated object
        //   is the interface.
        Vehicle v = (Vehicle)t.newInstance();

        // Call the methods that have a custom implementation
        System.debug('Max speed: ' + v.getMaxSpeed());
        System.debug('Vehicle type: ' + v.getType());
    }
}
```

Class Property

The `class` property returns the `System.Type` of the type it is called on. It's exposed on all Apex built-in types including primitive data types and collections, sObject types, and user-defined classes. This property can be used instead of `forName` methods.

Call this property on the type name. For example:

```
System.Type t = Integer.class;
```

You can use this property for the second argument of `JSON.deserialize`, `deserializeStrict`, `JSONParser.readValueAs`, and `readValueAsStrict` methods to get the type of the object to deserialize. For example:

```
Decimal n = (Decimal)JSON.deserialize('100.1', Decimal.class);
```

Type Methods

The following are methods for `Type`.

IN THIS SECTION:

equals(typeToCompare)
Returns `true` if the specified type is equal to the current type; otherwise, returns `false`.

forName(fullyQualifiedName)
Returns the type that corresponds to the specified fully qualified class name.

forName(namespace, name)
Returns the type that corresponds to the specified namespace and class name.

getName()
Returns the name of the current type.

hashCode()
Returns a hash code value for the current type.

isAssignableFrom(sourceType)
Returns `true` if an object reference of the specified type can be assigned from the child type; otherwise, returns `false`.

newInstance()
Creates an instance of the current type and returns this new instance.

toString()
Returns a string representation of the current type, which is the type name.

`equals(typeToCompare)`

Returns `true` if the specified type is equal to the current type; otherwise, returns `false`.

Signature

```
public Boolean equals(Object typeToCompare)
```

Parameters

`typeToCompare`
Type: Object

The type to compare with the current type.

Return Value

Type: Boolean

Example

```
Type t1 = Account.class;
Type t2 = Type.forName('Account');
System.assert(t1.equals(t2));
```

forName(fullyQualifiedName)

Returns the type that corresponds to the specified fully qualified class name.

Signature

```
public static System.Type forName(String fullyQualifiedName)
```

Parameters

fullyQualifiedName
Type: String

The fully qualified name of the class to get the type of. The fully qualified class name contains the namespace name, for example, `MyNamespace.ClassName`.

Return Value

Type: `System.Type`

Usage

Note:

- This method returns `null` if called outside a managed package to get the type of a non-global class in a managed package. This is because the non-global class isn't visible outside the managed package. For Apex saved using Salesforce API version 27.0 and earlier, this method does return the corresponding class type for the non-global managed package class.
- When called from an installed managed package to get the name of a local type in an organization with no defined namespace, the `forName(fullyQualifiedName)` method returns `null`. Instead, use the `forName(namespace, name)` method and specify an empty string or `null` for the namespace argument.
- A call to `Type.forName()` can cause the class to be compiled.

forName(namespace, name)

Returns the type that corresponds to the specified namespace and class name.

Signature

```
public static System.Type forName(String namespace, String name)
```

Parameters

namespace
Type: String

The namespace of the class. If the class doesn't have a namespace, set the *namespace* argument to `null` or an empty string.

name
Type: String

The name of the class.

Return Value

Type: `System.Type`

Usage

Note:

- This method returns `null` if called outside a managed package to get the type of a non-global class in a managed package. This is because the non-global class isn't visible outside the managed package. For Apex saved using Salesforce API version 27.0 and earlier, this method does return the corresponding class type for the non-global managed package class.
- Use this method instead of `forName(fullyQualifiedName)` if it's called from a managed package installed in an organization with no defined namespace. To get the name of a local type, set the namespace argument to an empty string or `null`. For example, `Type t = Type.forName('', 'ClassName');`.
- A call to `Type.forName()` can cause the class to be compiled.

Example

This example shows how to get the type that corresponds to the `ClassName` class and the `MyNamespace` namespace.

```
Type myType =
   Type.forName('MyNamespace', 'ClassName');
```

Versioned Behavior Changes

In API version 60.0 and later, using an invalid namespace while calling this method returns null. Previously, Apex allowed you to specify an invalid namespace such as `Type.forName('InvalidNamespace', 'OuterClass.InnerClass')` or use an outer class as a namespace such as `Type.forName('OuterClass', 'InnerClass')` with indeterminate results.

getName()

Returns the name of the current type.

Signature

```
public String getName()
```

Return Value

Type: String

Example

This example shows how to get a Type's name. It first obtains a Type by calling `forName`, then calls `getName` on the Type object.

```
Type t =
   Type.forName('MyClassName');

String typeName =
   t.getName();
System.assertEquals('MyClassName',
   typeName);
```

hashCode()

Returns a hash code value for the current type.

Signature

```
public Integer hashCode()
```

Return Value

Type: Integer

Usage

The returned hash code value corresponds to the type name hash code that `String.hashCode` returns.

isAssignableFrom(sourceType)

Returns `true` if an object reference of the specified type can be assigned from the child type; otherwise, returns `false`.

Signature

```
public Boolean isAssignableFrom(Type sourceType)
```

Parameters

sourceType
The type of the object with which you are checking compatibility.

Return Value

Type: Boolean

The method returns `true` when the method is invoked as parentType.isAssignableFrom(childType). When invoked in any of the following ways, the method returns `false`:

- childType.isAssignableFrom(parentType)
- typeA.isAssignableFrom(TypeB) where TypeB is a sibling of TypeA
- typeA.isAssignableFrom(TypeB) where TypeB and TypeA are unrelated

Note: A childType is the child of a parentType when it implements an interface, extends a virtual or abstract class, or is the same `System.Type` as the parentType.

Usage

Unlike the `instanceof` operator, this method allows you to check type compatibility without having to create a class instance. This method eliminates static compile-time dependencies that `instanceof` requires.

The following code demonstrates how a typical ISV customer can use `isAssignableFrom()` to check compatibility between a customer-defined type (customerProvidedPluginType) and a valid plugin type.

```
//Scenario: Managed package code loading a "plugin" class that implements a managed
interface; the implementation done outside of the package
String pluginNameStr = Config__c.getInstance().PluginApexType__c;
Type customerProvidedPluginType = Type.forName(pluginNameStr);
Type pluginInterface = ManagedPluginInterface.class;

// Constructors may have side-effects, including potentially unsafe DML/callouts.
// We want to make sure the class is really designed to be a valid plugin before we
instantiate it
Boolean validPlugin = pluginInterface.isAssignableFrom(customerProvidedPluginType); //
validate that it implements the right interface

if(!validPlugin){
    throw new SecurityException('Cannot create instance of '+customerProvidedPluginType+'.
 Does not implement ManagedPluginInterface');
}else{
    return Type.newInstance(validPlugin);
}
```

Example

The following code snippet first defines sibling classes A and B that both implement the Callable interface and an unrelated class C. Then, it explores several type comparisons using `isAssignableFrom()`.

```
//Define classes A, B, and C

global class A implements Database.Batchable<String>, Callable {
    global Iterable<String> start(Database.BatchableContext context) { return null; }
    global void execute(Database.BatchableContext context, String[] scope) { }
    global void finish(Database.BatchableContext context) { }
    global Object call(String action, Map<String, Object> args) { return null; }
}
```

```
global class B implements Callable {
    global Object call(String action, Map<String, Object> args) { return null; }
}
```

```
global class C { }
```

```
Type listOfStrings = Type.forName('List<String>');
```

```
Type listOfIntegers = Type.forName('List<Integer>');
boolean flagListTypes = listOfIntegers.isAssignableFrom(listOfStrings); // false
```

```
//Examples with stringType and idType
Type stringType = Type.forName('String');
Type idType = Type.forName('Id');
boolean isId_assignableFromString = idType.isAssignableFrom(stringType); // true
//isAssignableFrom respects that String can be assigned to Id without an explicit cast
```

```
//Examples with typeA, typeB, and typeC
Type typeA = Type.forName('A');
Type typeB = Type.forName('B');
Type typeC = Type.forName('C');
boolean isTypeB_ofTypeA = typeB.isAssignableFrom( typeA ); // false - siblings
boolean isTypeA_ofTypeC = typeA.isAssignableFrom( typeC ); // false - unrelated types
boolean isTypeA_ofTypeA = typeA.isAssignableFrom(typeA); // true - identity
```

```
//Examples with callableType and batchableType
Type callableType = Type.forName('Callable');
Type batchableType = Type.forName('Database.Batchable');
boolean isTypeA_Callable = callableType.isAssignableFrom( typeA ); // true - type A is a
child of Callable type
boolean isTypeA_Batchable = batchableType.isAssignableFrom( typeA ); // true - type A is
a child of Batchable type
boolean isCallableOfTypeA = typeA.isAssignableFrom( callableType ); // false - Callable
type is not a child of type A
boolean isBatchableOfTypeA = typeA.isAssignableFrom( batchableType ); // false - Batchable
 type is not a child of type A
```

`newInstance()`

Creates an instance of the current type and returns this new instance.

Signature

```
public Object newInstance()
```

Return Value

Type: Object

Usage

Because `newInstance` returns the generic object type, you should cast the return value to the type of the variable that will hold this value.

This method enables you to instantiate a Type that implements an interface and call its methods while letting someone else provide the methods' implementation. For example, a package developer can provide an interface that a subscriber who installs the package can implement. The code in the package calls the subscriber's implementation of the interface methods by instantiating the subscriber's Type.

Example

This example shows how to create an instance of a Type. It first gets a Type by calling `forName` with the name of a class (`ShapeImpl`), then calls `newInstance` on this Type object. The `newObj` instance is declared with the interface type (`Shape`) that the `ShapeImpl` class implements. The return value of the `newInstance` method is cast to the `Shape` type.

```
Type t =
   Type.forName('ShapeImpl');

Shape newObj =
   (Shape)t.newInstance();
```

`toString()`

Returns a string representation of the current type, which is the type name.

Signature

```
public String toString()
```

Return Value

Type: String

Usage

This method returns the same value as `getName`. `String.valueOf` and `System.debug` use this method to convert their Type argument into a String.

Example

This example calls `toString` on the Type corresponding to a list of Integers.

```
Type t = List<Integer>.class;
String s = t.toString();
System.assertEquals('List<Integer>', s);
```

UninstallHandler Interface

Enables custom code to run after a managed package is uninstalled.

Namespace

System

Usage

App developers can implement this interface to specify Apex code that runs automatically after a subscriber uninstalls a managed package. This makes it possible to perform cleanup and notification tasks based on details of the subscriber's organization.

The uninstall script is subject to default governor limits. It runs as a special system user that represents your package, so all operations performed by the script will appear to be done by your package. You can access this user by using UserInfo. You will only see this user at runtime, not while running tests.

If the script fails, the uninstall continues but none of the changes performed by the script are committed. Any errors in the script are emailed to the user specified in the **Notify on Apex Error** field of the package. If no user is specified, the uninstall details will be unavailable.

The uninstall script has the following restrictions. You can't use it to initiate batch, scheduled, and future jobs, to access Session IDs, or to perform callouts.

The `UninstallHandler` interface has a single method called `onUninstall`, which specifies the actions to be performed on uninstall.

```
global interface UninstallHandler {
  void onUninstall(UninstallContext context)};
```

The `onUninstall` method takes a context object as its argument, which provides the following information.

- The org ID of the organization in which the uninstall takes place.
- The user ID of the user who initiated the uninstall.

The context argument is an object whose type is the `UninstallContext` interface. This interface is automatically implemented by the system. The following definition of the `UninstallContext` interface shows the methods you can call on the context argument.

```
global interface UninstallContext {
  ID organizationId();
  ID uninstallerId();
}
```

IN THIS SECTION:

UninstallHandler Methods

UninstallHandler Example Implementation

UninstallHandler Methods

The following are methods for `UninstallHandler`.

IN THIS SECTION:

onUninstall(context)
Specifies the actions to be performed on uninstall.

`onUninstall(context)`

Specifies the actions to be performed on uninstall.

Signature

```
public Void onUninstall(UninstallContext context)
```

Parameters

context
Type: UninstallContext

Return Value

Type: Void

UninstallHandler Example Implementation

Example of an Uninstall Script

This sample uninstall script performs the following actions on package uninstall.

- Inserts an entry in the feed describing which user did the uninstall and in which organization
- Creates and sends an email message confirming the uninstall to that user

```
global class UninstallClass implements UninstallHandler {
  global void onUninstall(UninstallContext ctx) {
    FeedItem feedPost = new FeedItem();
    feedPost.parentId = ctx.uninstallerID();
    feedPost.body = 'Thank you for using our application!';
    insert feedPost;

    User u = [Select Id, Email from User where Id =:ctx.uninstallerID()];
    String toAddress= u.Email;
    String[] toAddresses = new String[] {toAddress};
    Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
    mail.setToAddresses(toAddresses);
    mail.setReplyTo('support@package.dev');
    mail.setSenderDisplayName('My Package Support');
    mail.setSubject('Package uninstall successful');
    mail.setPlainTextBody('Thanks for uninstalling the package.');
    Messaging.sendEmail(new Messaging.Email[] { mail });
  }
}
```

You can test an uninstall script using the `testUninstall` method of the `Test` class. This method takes as its argument a class that implements the `UninstallHandler` interface.

This sample shows how to test an uninstall script implemented in the `UninstallClass` Apex class.

```
@isTest
static void testUninstallScript() {
  Id UninstallerId = UserInfo.getUserId();
  List<FeedItem> feedPostsBefore =
    [SELECT Id FROM FeedItem WHERE parentId=:UninstallerId AND CreatedDate=TODAY];
  Test.testUninstall(new UninstallClass());
  List<FeedItem> feedPostsAfter =
    [SELECT Id FROM FeedItem WHERE parentId=:UninstallerId AND CreatedDate=TODAY];
  System.assertEquals(feedPostsBefore.size() + 1, feedPostsAfter.size(),
    'Post to uninstaller failed.');
}
```

URL Class

Represents a uniform resource locator (URL) and provides access to parts of the URL. Enables access to the base URL used to access your Salesforce org.

Namespace

System

Usage

Use the methods of the `System.URL` class to create links to objects in your organization. Such objects can be files, images, logos, or records that you want to include in external emails, in activities, or in Chatter posts. For example, you can create a link to a file uploaded as an attachment to a Chatter post by concatenating the Salesforce base URL with the file ID:

```
// Get a file uploaded through Chatter.
ContentDocument doc = [SELECT Id FROM ContentDocument
          WHERE Title = 'myfile'];
// Create a link to the file.
String fullFileURL = URL.getOrgDomainURL().toExternalForm() +
   '/' + doc.id;
system.debug(fullFileURL);
```

The following example creates a link to a Salesforce record. The full URL is created by concatenating the Salesforce base URL with the record ID.

```
Account acct = [SELECT Id FROM Account WHERE Name = 'Acme' LIMIT 1];
String fullRecordURL = URL.getOrgDomainURL().toExternalForm() + '/' + acct.Id;
```

Example

In this example, the base URL and the full request URL of the current Salesforce server instance are retrieved. Next, a URL pointing to a specific account object is created. Finally, components of the base and full URL are obtained. This example prints out all the results to the debug log output.

```
// Create a new account called Acme that we will create a link for later.
Account myAccount = new Account(Name='Acme');
insert myAccount;

// Get the base URL.
String sfdcBaseURL = URL.getOrgDomainURL().toExternalForm();
System.debug('Base URL: ' + sfdcBaseURL );

// Get the URL for the current request.
String currentRequestURL = URL.getCurrentRequestUrl().toExternalForm();
System.debug('Current request URL: ' + currentRequestURL);

// Create the account URL from the base URL.
String accountURL = URL.getOrgDomainURL().toExternalForm() +
                    '/' + myAccount.Id;
System.debug('URL of a particular account: ' + accountURL);

// Get some parts of the base URL.
```

```
System.debug('Host: ' + URL.getOrgDomainURL().getHost());
System.debug('Protocol: ' + URL.getOrgDomainURL().getProtocol());

// Get the query string of the current request.
System.debug('Query: ' + URL.getCurrentRequestUrl().getQuery());
```

Versioned Behavior Changes

In API version 41.0 and later, Apex URL objects are represented by the `java.net.URI` type, not the `java.net.URL` type. The API version in which the URL object was instantiated determines the behavior of subsequent method calls to the specific instance. Salesforce strongly encourages you to use API 41.0 and later versions for fully RFC-compliant URL parsing that includes proper handling of edge cases of complex URL structures. API 41.0 and later versions also enforce that inputs are valid, RFC-compliant URL or URI strings.

IN THIS SECTION:

URL Constructors

URL Methods

SEE ALSO:

DomainCreator Class

URL Constructors

The following are constructors for `URL`.

IN THIS SECTION:

Url(spec)
Creates a new instance of the `URL` class using the specified string representation of the URL.

Url(context, spec)
Creates a new instance of the `URL` class by parsing the specified spec within the specified context.

Url(protocol, host, file)
Creates a new instance of the `URL` class using the specified protocol, host, and file on the host. The default port for the specified protocol is used.

Url(protocol, host, port, file)
Creates a new instance of the `URL` class using the specified protocol, host, port, and file on the host.

`Url(spec)`

Creates a new instance of the `URL` class using the specified string representation of the URL.

Signature

```
public Url(String spec)
```

Parameters

spec
Type: String

The string to parse as a URL.

Url(context, spec)

Creates a new instance of the `URL` class by parsing the specified spec within the specified context.

Signature

```
public Url(Url context, String spec)
```

Parameters

context
Type: URL on page 3980

The context in which to parse the specification.

spec
Type: String

The string to parse as a URL.

Usage

The new URL is created from the given context URL and the spec argument as described in RFC2396 "Uniform Resource Identifiers : Generic * Syntax" :

```
<scheme>://<authority><path>?<query>#<fragment>
```

For more information about the arguments of this constructor, see the corresponding URL(java.net.URL, java.lang.String) constructor for Java.

Url(protocol, host, file)

Creates a new instance of the `URL` class using the specified protocol, host, and file on the host. The default port for the specified protocol is used.

Signature

```
public Url(String protocol, String host, String file)
```

Parameters

protocol
Type: String

The protocol name for this URL.

host
Type: String

The host name for this URL.

file
Type: String

The file name for this URL.

Url(protocol, host, port, file)

Creates a new instance of the `URL` class using the specified protocol, host, port, and file on the host.

Signature

```
public Url(String protocol, String host, Integer port, String file)
```

Parameters

protocol
Type: String

The protocol name for this URL.

host
Type: String

The host name for this URL.

port
Type: Integer

The port number for this URL.

file
Type: String

The file name for this URL.

URL Methods

The following are methods for `URL`.

IN THIS SECTION:

getAuthority()
Returns the authority portion of the current URL.

getCurrentRequestUrl()
Returns the URL of an entire request on a Salesforce instance.

getDefaultPort()
Returns the default port number of the protocol associated with the current URL.

getFile()
Returns the file name of the current URL.

getFileFieldURL(entityId, fieldName)
Returns the download URL for a file attachment.

getHost()
Returns the host name of the current URL.

getOrgDomainUrl()
Returns the canonical URL for your org. For example, `https://MyDomainName.my.salesforce.com`.

getPath()
Returns the path portion of the current URL.

getPort()
Returns the port of the current URL.

getProtocol()
Returns the protocol name of the current URL, such as, `https`.

getQuery()
Returns the query portion of the current URL.

getRef()
Returns the anchor of the current URL.

getSalesforceBaseUrl()
In API version 59.0 and later, this method is deprecated and versioned out. Use getOrgDomainUrl() to get the canonical URL for your org or use getCurrentRequestUrl() to get the URL of an entire request on a Salesforce instance. Returns the URL of the current connection to the Salesforce org.

getUserInfo()
Gets the UserInfo portion of the current URL.

sameFile(URLToCompare)
Compares the current URL with the specified URL object, excluding the fragment component.

toExternalForm()
Returns a string representation of the current URL.

`getAuthority()`

Returns the authority portion of the current URL.

Signature

```
public String getAuthority()
```

Return Value

Type: String

`getCurrentRequestUrl()`

Returns the URL of an entire request on a Salesforce instance.

Signature

```
public static System.URL getCurrentRequestUrl()
```

Return Value

Type: `System.URL`

Usage

An example of a URL for an entire request is `https://yourInstance.salesforce.com/apex/myVfPage.apexp`.

getDefaultPort()

Returns the default port number of the protocol associated with the current URL.

Signature

```
public Integer getDefaultPort()
```

Return Value

Type: Integer

Usage

Returns -1 if the URL scheme or the stream protocol handler for the URL doesn't define a default port number.

getFile()

Returns the file name of the current URL.

Signature

```
public String getFile()
```

Return Value

Type: String

getFileFieldURL(entityId, fieldName)

Returns the download URL for a file attachment.

Signature

```
public static String getFileFieldURL(String entityId, String fieldName)
```

Parameters

entityId
Type: String

Specifies the ID of the entity that holds the file data.

fieldName
Type: String

Specifies the API name of a file field component, such as `AttachmentBody`.

Return Value

Type: String

Usage

Example:

Example

```
String fileURL =
  URL.getFileFieldURL(
    '087000000000123' ,
    'AttachmentBody');
```

getHost()

Returns the host name of the current URL.

Signature

```
public String getHost()
```

Return Value

Type: String

getOrgDomainUrl()

Returns the canonical URL for your org. For example, `https://MyDomainName.my.salesforce.com`.

Signature

```
public static System.Url getOrgDomainUrl()
```

Return Value

Type: `System.URL`

`getOrgDomainUrl()` always returns the login URL for your org, regardless of context. Use that URL when making API calls to your org.

Usage

Use `getOrgDomainUrl()` to interact with Salesforce REST and SOAP APIs in Apex code. Get endpoints for User Interface API calls, for creating and customizing picklist value sets and custom fields, and more.

`getOrgDomainUrl()` can access the domain URL only for the org in which the Apex code is running.

You don't need a RemoteSiteSetting for your org to interact with the Salesforce APIs using domain URLs retrieved with this method.

Example

This example uses the Salesforce REST API to get organization limit values. For information on limits, see Limits in the *REST API Developer Guide*.

```
Http h = new Http();
HttpRequest req = new HttpRequest();
req.setEndpoint(Url.getOrgDomainUrl().toExternalForm()
   + '/services/data/v44.0/limits');
req.setMethod('GET');
req.setHeader('Authorization', 'Bearer ' + UserInfo.getSessionId());
HttpResponse res = h.send(req);
```

SEE ALSO:

Lightning Aura Components Developer Guide: Making API Calls from Apex

User Interface API Developer Guide: Get Default Values to Clone a Record

User Interface API Developer Guide: Get Values for a Picklist Field

User Interface API Developer Guide: User Inteface API Resources

`getPath()`

Returns the path portion of the current URL.

Signature

```
public String getPath()
```

Return Value

Type: String

`getPort()`

Returns the port of the current URL.

Signature

```
public Integer getPort()
```

Return Value

Type: Integer

`getProtocol()`

Returns the protocol name of the current URL, such as, `https`.

Signature

```
public String getProtocol()
```

Return Value

Type: String

`getQuery()`

Returns the query portion of the current URL.

Signature

```
public String getQuery()
```

Return Value

Type: String

Usage

Returns `null` if no query portion exists.

`getRef()`

Returns the anchor of the current URL.

Signature

```
public String getRef()
```

Return Value

Type: String

Usage

Returns `null` if no query portion exists.

`getSalesforceBaseUrl()`

In API version 59.0 and later, this method is deprecated and versioned out. Use getOrgDomainUrl() to get the canonical URL for your org or use getCurrentRequestUrl() to get the URL of an entire request on a Salesforce instance. Returns the URL of the current connection to the Salesforce org.

Signature

```
public static System.URL getSalesforceBaseUrl()
```

Return Value

Type: `System.URL`

Returns the URL for the current connection: for example, `https://MyDomainName.my.salesforce.com` or `https://MyDomainName.lightning.force.com`.

SEE ALSO:

getOrgDomainUrl()

getUserInfo()

Gets the UserInfo portion of the current URL.

Signature

```
public String getUserInfo()
```

Return Value

Type: String

Usage

Returns `null` if no UserInfo portion exists.

sameFile(URLToCompare)

Compares the current URL with the specified URL object, excluding the fragment component.

Signature

```
public Boolean sameFile(System.URL URLToCompare)
```

Parameters

URLToCompare
Type: System.URL

Return Value

Type: Boolean

Returns `true` if both URL objects reference the same remote resource; otherwise, returns `false`.

Usage

For more information about the syntax of URIs and fragment components, see RFC3986.

toExternalForm()

Returns a string representation of the current URL.

Signature

```
public String toExternalForm()
```

Return Value

Type: String

UserInfo Class

Contains methods for obtaining information about the context user.

Namespace

System

UserInfo Methods

The following are methods for `UserInfo`. All methods are static.

IN THIS SECTION:

getCurrentUvid()
Returns the context guest user's unique visitor ID (UVID).

getDefaultCurrency()
Returns the context user's default currency code for multiple currency organizations or the organization's currency code for single currency organizations.

getFirstName()
Returns the context user's first name

getLanguage()
Returns the context user's language

getLastName()
Returns the context user's last name

getLocale()
Returns the context user's locale.

getName()
Returns the context user's full name. The format of the name depends on the language preferences specified for the organization.

getOrganizationId()
Returns the context organization's ID.

getOrganizationName()
Returns the context organization's company name.

getProfileId()
Returns the context user's profile ID.

getSessionId()
Returns the session ID for the current session.

getTimeZone()
Returns the current user's local time zone.

getUiTheme()
Returns the preferred theme for the current user. Use `getUiThemeDisplayed` to determine the theme actually displayed to the current user.

getUiThemeDisplayed()
Returns the theme being displayed for the current user.

getUserEmail()
Returns the current user's email address.

getUserId()
Returns the context user's ID

getUserName()
Returns the context user's login name.

getUserRoleId()
Returns the context user's role ID.

getUserType()
Returns the context user's type.

hasPackageLicense(packageId)
Returns `true` if the context user has a license to the managed package via a package license only. Otherwise, returns `false`.

isCurrentUserLicensed(namespace)
Returns `true` if the context user has a license to any managed package denoted by the namespace. Otherwise, returns `false`.

isCurrentUserLicensedForPackage(packageID)
Returns `true` if the context user has a license to the managed package denoted by the package ID. Otherwise, returns `false`. If the context user has access, it's determined either via the package license or a namespace permission set license for the package namespace.

isMultiCurrencyOrganization()
Specifies whether the organization uses multiple currencies.

`getCurrentUvid()`

Returns the context guest user's unique visitor ID (UVID).

Signature

```
public static String getCurrentUvid()
```

Return Value

Type: String

If a UVID isn't available, returns `null`.

`getDefaultCurrency()`

Returns the context user's default currency code for multiple currency organizations or the organization's currency code for single currency organizations.

Signature

```
public static String getDefaultCurrency()
```

Return Value

Type: String

Usage

Note: For Apex saved using Salesforce API version 22.0 or earlier, `getDefaultCurrency` returns `null` for single currency organizations.

`getFirstName()`

Returns the context user's first name

Signature

```
public static String getFirstName()
```

Return Value

Type: String

`getLanguage()`

Returns the context user's language

Signature

```
public static String getLanguage()
```

Return Value

Type: String

`getLastName()`

Returns the context user's last name

Signature

```
public static String getLastName()
```

Return Value

Type: String

`getLocale()`

Returns the context user's locale.

Signature

```
public static String getLocale()
```

Return Value

Type: String

Example

```
String result = UserInfo.getLocale();
System.assertEquals('en_US', result);
```

`getName()`

Returns the context user's full name. The format of the name depends on the language preferences specified for the organization.

Signature

```
public static String getName()
```

Return Value

Type: String

Usage

The format is one of the following:

- FirstName LastName
- LastName, FirstName

`getOrganizationId()`

Returns the context organization's ID.

Signature

```
public static String getOrganizationId()
```

Return Value

Type: String

getOrganizationName()

Returns the context organization's company name.

Signature

```
public static String getOrganizationName()
```

Return Value

Type: String

getProfileId()

Returns the context user's profile ID.

Signature

```
public static String getProfileId()
```

Return Value

Type: String

getSessionId()

Returns the session ID for the current session.

Signature

```
public static String getSessionId()
```

Return Value

Type: String

Usage

You can use `getSessionId()` both synchronously and asynchronously. In asynchronous Apex (Batch, Future, Queueable, or Scheduled Apex), this method returns the session ID only when the code is run by an active, valid user. When the code is run by an internal user, such as the automated process user or a proxy user, the method returns `null`.

As a best practice, ensure that your code handles both cases: when a session ID is or is not available.

getTimeZone()

Returns the current user's local time zone.

Signature

```
public static System.TimeZone getTimeZone()
```

Return Value

Type: System.TimeZone

Example

```
TimeZone tz =
  UserInfo.getTimeZone();
System.debug(
  'Display name: ' +
  tz.getDisplayName());
System.debug(
  'ID: ' +
  tz.getID());
```

getUiTheme()

Returns the preferred theme for the current user. Use `getUiThemeDisplayed` to determine the theme actually displayed to the current user.

Signature

```
public static String getUiTheme()
```

Return Value

Type: String

The preferred theme for the current user.

Valid values include:

- `Theme1`—Obsolete Salesforce theme
- `Theme2`—Salesforce Classic 2005 user interface theme
- `Theme3`—Salesforce Classic 2010 user interface theme
- `Theme4d`—Modern "Lightning Experience" Salesforce theme
- `Theme4t`—Salesforce mobile app theme
- `Theme4u`—Lightning Console theme
- `PortalDefault`—Salesforce Customer Portal theme that applies to Customer Portals only and not to Experience Builder sites
- `Webstore`—AppExchange theme

getUiThemeDisplayed()

Returns the theme being displayed for the current user.

Signature

```
public static String getUiThemeDisplayed()
```

Return Value

Type: String

The theme being displayed for the current user

Valid values include:

- `Theme1`—Obsolete Salesforce theme
- `Theme2`—Salesforce Classic 2005 user interface theme
- `Theme3`—Salesforce Classic 2010 user interface theme
- `Theme4d`—Modern "Lightning Experience" Salesforce theme
- `Theme4t`—Salesforce mobile app theme
- `Theme4u`—Lightning Console theme
- `PortalDefault`—Salesforce Customer Portal theme that applies to Customer Portals only and not to Experience Builder sites
- `Webstore`—AppExchange theme

`getUserEmail()`

Returns the current user's email address.

Signature

```
public static String getUserEmail()
```

Return Value

Type: String

Example

```
String emailAddress =
  UserInfo.getUserEmail();
System.debug(
  'Email address: ' +
  emailAddress);
```

`getUserId()`

Returns the context user's ID

Signature

```
public static String getUserId()
```

Return Value

Type: String

getUserName()

Returns the context user's login name.

Signature

```
public static String getUserName()
```

Return Value

Type: String

getUserRoleId()

Returns the context user's role ID.

Signature

```
public static String getUserRoleId()
```

Return Value

Type: String

getUserType()

Returns the context user's type.

Signature

```
public static String getUserType()
```

Return Value

Type: String

hasPackageLicense(packageId)

Returns `true` if the context user has a license to the managed package via a package license only. Otherwise, returns `false`.

Signature

```
public static Boolean hasPackageLicense(ID packageID)
```

Parameters

packageID
Type: String

Return Value

Type: Boolean

`isCurrentUserLicensed(namespace)`

Returns `true` if the context user has a license to any managed package denoted by the namespace. Otherwise, returns `false`.

Signature

```
public static Boolean isCurrentUserLicensed(String namespace)
```

Parameters

`namespace`
Type: String

Return Value

Type: Boolean

Usage

A `TypeException` is thrown if *`namespace`* is an invalid type.

`isCurrentUserLicensedForPackage(packageID)`

Returns `true` if the context user has a license to the managed package denoted by the package ID. Otherwise, returns `false`. If the context user has access, it's determined either via the package license or a namespace permission set license for the package namespace.

Signature

```
public static Boolean isCurrentUserLicensedForPackage(ID packageID)
```

Parameters

`packageID`
Type: String

Return Value

Type: Boolean

Usage

Retrieve *`packageID`* at runtime, with the getCurrentPackageId() method. Then, use `packageId` to confirm that the contextual user is licensed to use that managed package.

A `TypeException` is thrown if `packageID` is an invalid type or is the ID of an unlocked or unmanaged package.

SEE ALSO:

Set Up and Maintain Your Salesforce Organization: Manage Licenses for Installed Packages

`isMultiCurrencyOrganization()`

Specifies whether the organization uses multiple currencies.

Signature

```
public static Boolean isMultiCurrencyOrganization()
```

Return Value

Type: Boolean

UserManagement Class

Contains methods to manage end users, for example, to register their verification methods, verify their identity, or remove their personal information.

Namespace

System

Usage

Let users register and deregister identity verification methods. Create custom Login and Verify pages for passwordless login and self-registration. Convert mobile phone numbers to the proper format before registering users. Scramble user data when users request that Salesforce remove their personal information.

This class is available in API version 43.0 and later.

IN THIS SECTION:

UserManagement Methods

UserManagement Methods

The following are methods for `UserManagement`.

IN THIS SECTION:

clone()

Makes a duplicate copy of the System.UserManagement object.

deregisterVerificationMethod(userId, method)

Deregisters an identity verification method. Use this method to let users delete an existing verification method.

formatPhoneNumber(countryCode, phoneNumber)

Formats a mobile phone number for a user. Call this method to ensure that the phone number is formatted properly before updating a user's mobile phone number.

initPasswordlessLogin(userId, method)

Invokes a verification challenge for passwordless login when creating custom (Visualforce) Login and Verify pages for customers and partners.

initRegisterVerificationMethod(method)

Invokes a verification challenge for registering identity verification methods with a custom (Visualforce) page. Users can register either their email address or phone number.

initSelfRegistration(method, user)

Invokes a verification challenge for self-registration when creating a custom (Visualforce) Verify page for Experience Cloud self-registration.

initVerificationMethod(method)

Initiates a verification service for email, phone (SMS), and the Salesforce Authenticator verification methods.

initVerificationMethod(method, actionName, extras)

Initiates a verification service for email, phone (SMS), and the Salesforce Authenticator verification methods.

obfuscateUser(userId, username)

Scrambles users' data on their request when they no longer want their personal data recognized in Salesforce. When you invoke the method for the user, the data becomes anonymous, and you can never recover it. Use this method to set the username to a specific value after it's scrambled.

obfuscateUser(userId)

Scrambles users' data on their request when they no longer want their personal data recognized in Salesforce. When you invoke the method for the user, the data becomes anonymous, and you can never recover it.

registerVerificationMethod(method, startUrl)

Registers an identity verification method. Verification methods can be a time-based one-time password (TOTP), email or text verification code, Salesforce Authenticator, or U2F-compatible security key. End users register verification methods for themselves.

sendAsyncEmailConfirmation(userId, emailTemplateId, networkId, startUrl)

Send an email message to a user's email address for verification. The message contains a verification link (URL) that the user clicks to verify the email address later on. You can send email verifications in bulk.

verifyPasswordlessLogin(userId, method, identifier, code, startUrl)

Completes a verification challenge during a passwordless login that uses a custom Verify page (Visualforce only). If the user who is trying to log in enters the verification code successfully, the user is logged in.

verifyRegisterVerificationMethod(code, method)

Completes registering a user's email address or phone number as a verification method when customizing the identity verification process.

verifySelfRegistration(method, identifier, code, startUrl)

Completes a verification challenge when creating a custom (Visualforce) Verify page for Experience Cloud site self-registration. If the person who is attempting to register enters the verification code successfully, the user is created and logged in.

verifyVerificationMethod(identifier, code, method)

Completes the verification service for email, phone (SMS), Salesforce Authenticator, password, or time-based one-time password (TOTP) verification methods.

clone()

Makes a duplicate copy of the System.UserManagement object.

Signature

```
public Object clone()
```

Return Value

Type: User Management

deregisterVerificationMethod(userId, method)

Deregisters an identity verification method. Use this method to let users delete an existing verification method.

Signature

```
public static void deregisterVerificationMethod(Id userId, Auth.VerificationMethod
method)
```

Parameters

userId
Type: Id

User ID of the user deregistering the verification method.

method
Type: Auth.VerificationMethod

Verification method used to verify the identity of the user.

Return Value

Type: void

Usage

Use this method to deregister an existing identity verification method. For example, your users can deregister a phone number when their phone number changes. While only end users can register an identity verification method, you and your users can deregister one. Keep this behavior in mind when you implement a custom registration page.

This method is available in API version 43.0 and later.

Note: This method doesn't support deregistering built-in authenticators.

formatPhoneNumber(countryCode, phoneNumber)

Formats a mobile phone number for a user. Call this method to ensure that the phone number is formatted properly before updating a user's mobile phone number.

Signature

```
global static String formatPhoneNumber(String countryCode, String phoneNumber)
```

Parameters

countryCode
Type: String

A valid country code.

phoneNumber
Type: String

A mobile number that contains from 3 through 49 numeric characters, without the country code. For example, (415) 555-1234.

Return Value

Type: String

Returns a user's mobile phone number in the proper format.

Usage

Use this method to ensure a user's mobile phone number is formatted as required by Salesforce. Then use the method's return value to update the `mobile` field of the user's record. This mobile number is used for SMS-based device activation. For example, mobile phone numbers are stored along with other identity verification methods in Auth.VerificationMethod enum. This method is introduced in API version 43.0. It isn't available in earlier versions.

Here are some acceptable ways that users can enter their mobile number:

- +1, (415) 555-1234 (with plus signs, parentheses, and dashes)
- 1, 4155551234 (only numbers, no symbols)
- 1 , 415-555-1234 (extra spaces)

Now, consider the following examples.

- Correct examples:
  - `formatPhoneNumber('1', '4155551234');`
  - `formatPhoneNumber('+1','(415) 555-1234');`
  - `formatPhoneNumber('1', '415-555-1234');`
- Incorrect example, because the country code and mobile number aren't separated:
  - `formatPhoneNumber(null, '+1 415-555-1234');`
- Example that doesn't generate an error, but likely won't work as intended:
  - `formatPhoneNumber('+1', '+1 (415) 555-1234');`

Format Phone Number Code Example

Here's a code example that uses the `formatPhoneNumber` method. It gets the mobile number from the user and converts it to the format required by Salesforce. Then it updates the user's record with the formatted mobile number.

```
global with sharing class PhoneRegistrationController {
    //Input variables
```

```
    global String countryCode {get; set;}
    global String phoneNumber {get; set;}

    global String addPhoneNumber()
    {
        if(countryCode == null) return 'Country code is required';
        if(phoneNumber == null) return 'Phone number is required';

        String userId = UserInfo.getUserId();
        User u = [SELECT Id FROM User WHERE Id=:userId LIMIT 1];
       String formatNum = System.UserManagement.formatPhoneNumber(countryCode, phoneNumber);

        u.MobilePhone = formatNum;
        update u;
        return null;
    }

}
```

As long as the country code and phone number are separated, `formatPhoneNumber` returns a value in the proper format.

`initPasswordlessLogin(userId, method)`

Invokes a verification challenge for passwordless login when creating custom (Visualforce) Login and Verify pages for customers and partners.

Signature

```
public static String initPasswordlessLogin(Id userId, Auth.VerificationMethod method)
```

Parameters

`userId`
Type: Id

ID of the user who's logging in.

`method`
Type: Auth.VerificationMethod

Method used to verify the user's identity, which can be EMAIL or SMS.

Return Value

Type: String

Identifier of the verification attempt.

Usage

Use this method along with its paired `verifyPasswordlessLogin` to customize the login experience with your own Visualforce Login and Verify pages. Invoke `initPasswordlessLogin` from the Login page where the user enters an email address or phone number.

Note: An alternative to using this combination of methods is to use `Site.passwordlessLogin`. Both approaches let you customize the Login page in Visualforce. With the paired methods, you can create custom Login and Verify pages. With `Site.passwordlessLogin`, Salesforce supplies the Verify page.

First call the `initPasswordlessLogin` method to initiate an authentication challenge. This method:

- Gets the user ID and verification method, such as EMAIL or SMS, from the Login page.
- Looks up the user and checks that the user is unique and active.
- Sends a verification code to the user.
- Adds an entry for the verification attempt to the Identity Verification History log, assigning an identifier to the verification attempt and setting the status to **User challenged, waiting for response**.
- Adds an entry for the Passwordless Login to the Login History log.
- Returns the identifier to `verifyPasswordlessLogin` to link the transactions.

Then call `verifyPasswordlessLogin`, which, if the user enters the verification code correctly, logs in the user.

Note: Users must verify their identity by email address or phone number before they can log in without a password. You can check whether the user is verified from the user's detail page in Setup. Or you can check programmatically with `TwoFactorMethodsInfo`.

`initRegisterVerificationMethod(method)`

Invokes a verification challenge for registering identity verification methods with a custom (Visualforce) page. Users can register either their email address or phone number.

Signature

```
public static String initRegisterVerificationMethod(Auth.VerificationMethod method)
```

Parameters

`method`
Type: Auth.VerificationMethod

Method used to verify the user's identity, which can be EMAIL or SMS.

Return Value

Type: String

The method returns an error message if the phone number is already registered, the user isn't a customer or partner, or if the context isn't an Experience Cloud site.

Usage

Use this method along with its paired `verifyRegisterVerificationMethod` on page 4014 to customize the process for registering a user's verification method using a Visualforce Verify page.

First call the `initRegisterVerificationMethod` method to get the verification code sent to the user as input, and validate it. If the verification code isn't valid, it returns an error message.

Example

Here's a code example that registers a user's phone number as a verification method. When the user enters a verification code on the Visualforce page, it invokes `registerUser()`. The method gets the User ID of the user who's registering the verification method and the user's phone number. It also gets the user's registration status to check whether the phone number is verified already. If the user is registered with a different phone number, the number is updated.

```
public void registerUser() {
        try {
            exceptionText='';
            String userId = UserInfo.getUserId();
            User u = [Select MobilePhone, Id from User Where Id=:userId];
             currPhone = u.MobilePhone;
             mobilePhone = getFormattedSms(mobilePhone);
           if (mobilePhone != null && mobilePhone != '') {
           u.MobilePhone = mobilePhone;
           update u;
            // We're updating the email and phone number before verifying. Roll back
           //  the change in the verify API if it is unsuccessful.
            exceptionText = System.
            UserManagement.initRegisterVerificationMethod(Auth.VerificationMethod.SMS);
            if(exceptionText!= null && exceptionText!=''){
                isInit = false;
                showInitException = true;
            } else {
                 isInit = false;
                 isVerify = true;
            }
            } else {
                showInitException = true;
            }
        } catch (Exception e) {
            exceptionText = e.getMessage();
            isInit = false;
            showInitException = true;
        }
    }

public void verifyUser() {
    // Take the user's input for the code sent to their phone number
    exceptionText = System.UserManagement.
        verifyRegisterVerificationMethod(code, Auth.VerificationMethod.SMS);
    if(exceptionText != null && exceptionText !=''){
    showInitException = true;
    } else {
            //Success
    }
}
```

`initSelfRegistration(method, user)`

Invokes a verification challenge for self-registration when creating a custom (Visualforce) Verify page for Experience Cloud self-registration.

Signature

```
public static String initSelfRegistration(Auth.VerificationMethod method, User user)
```

Parameters

method
Type: Auth.VerificationMethod

Method used to verify the identity of the user, which can be EMAIL or SMS.

user
Type: User

User object to insert after successful registration.

Return Value

Type: String

Identifier of the registration attempt.

Usage

By default, when users sign up for your Experience Cloud site with an email address or phone number, Salesforce sends them a verification code. At the same time, it generates a Verify page for users to confirm their identity. You can replace the default Salesforce Verify page with your own Visualforce page and then invoke the verification process.

Call this method to initiate the authentication challenge, and include a User object to insert if the registration is successful. The method returns the identifier for the self-registration attempt.

Note: If you specify a language in the `LanguageLocaleKey` field on the User object, Salesforce uses this language for verification email and SMS messages.

Then call verifySelfRegistration, which, if the user enters the verification code correctly, logs in the user.

Example

This code contains the result of a verification challenge that registers a new user.

```
String id = System.UserManagement.initSelfRegistration
          (Auth.VerificationMethod.SMS, user);
        Auth.VerificationResult res = System.UserManagement.verifySelfRegistration
          (Auth.VerificationMethod.SMS, id, '123456', null);
        if(res.success == true){
     //redirect
}
```

`initVerificationMethod(method)`

Initiates a verification service for email, phone (SMS), and the Salesforce Authenticator verification methods.

Signature

```
public static String initVerificationMethod(Auth.VerificationMethod method)
```

Parameters

method
Type: Auth.VerificationMethod

Method used to initiate a verification service for `EMAIL`, `SMS`, or `SALESFORCE_AUTHENTICATOR` verification methods.

Return Value

Type: String

The returned identifier must be passed into `verifyVerificationMethod`.

Usage

Use this method along with its paired `verifyVerificationMethod` to customize a verification service for `EMAIL`, `SMS`, or `SALESFORCE_AUTHENTICATOR` verification methods. The returned identifier from `initVerificationMethod` must be passed into `verifyVerificationMethod`.

First invoke the `initVerificationMethod` method to send a verification code to the user's email or phone number, or to send a push notification to the Salesforce Authenticator. The user then enters the code or approves the push notification. If the verification code isn't valid or the push notification isn't approved, the service returns an error message.

Email Example

This example shows multi-factor authentication using email.

```
public void initVerification() {
// user will receive code on their registered verified email
 identifier = UserManagement.initVerificationMethod(Auth.VerificationMethod.EMAIL);
}

public Auth.VerificationResult verifyVerification() {
// requiring identifier from the initVerification
// the code will need to be entered in this method
return UserManagement.verifyVerificationMethod(identifier, code ,
Auth.VerificationMethod.EMAIL);
}
```

`initVerificationMethod(method, actionName, extras)`

Initiates a verification service for email, phone (SMS), and the Salesforce Authenticator verification methods.

Signature

```
public static String initVerificationMethod(Auth.VerificationMethod method, String
actionName, Map<String,String> extras)
```

Parameters

method
Type: Auth.VerificationMethod

Method used to initiate a verification service for `EMAIL`, `SMS`, or `SALESFORCE_AUTHENTICATOR` verification methods.

actionName
Type: String

For the SALESFORCE_AUTHENTICATOR verification method only, the name of the action to display on the Salesforce Authenticator, such as Connect to My Salesforce Org. The default action name is Apex-Defined Activity.

extras
Type: Map<String,String>

For the SALESFORCE_AUTHENTICATOR verification method only, the following extra settings.

- secure_device_required–If set to true, the user's device must be secured. For example, the user must enter the device's passcode to approve the request. Default setting is false.
- challenge_required–If set to true, the user must complete a biometric challenge, such as face recognition, on the device to approve the request. Default setting is false.

Return Value

Type: String

The returned identifier must be passed into verifyVerificationMethod method.

Usage

Use this method along with its paired verifyVerificationMethod to customize a verification service for EMAIL, SMS, or SALESFORCE_AUTHENTICATOR verification methods. The returned identifier from initVerificationMethod must be passed into verifyVerificationMethod method.

First invoke the initVerificationMethod method to send a verification code to the user's email or phone number, or to send a push notification to the Salesforce Authenticator. The user then enters the code or approves the push notification. If the verification code isn't valid or the push notification isn't approved, the service returns an error message.

Salesforce Authenticator Example

This example shows multi-factor authentication (MFA) using the Salesforce Authenticator mobile app. In this example, the *actionName* parameter is set to the default setting and the *extra* parameter settings are set to false.

```
public void initVerification() {
// user will receive push notification on their registered MFA devices
identifier =
UserManagement.initVerificationMethod(Auth.VerificationMethod.SALESFORCE_AUTHENTICATOR);
}

public Auth.VerificationResult verifyVerification() {
// requiring identifier from the initVerification
// user will need to take the action on their registered MFA devices
return UserManagement.verifyVerificationMethod(identifier, '' ,
Auth.VerificationMethod.SALESFORCE_AUTHENTICATOR);
}
```

This example shows multi-factor authentication using Salesforce Authenticator. In this example, the *actionName* parameter is set to Connect to My Salesforce Org and the challenge_required *extra* parameter is set to true.

```
public void initVerification() {
Map<String,String> extras = new Map<String,String>();
```

```
extras.put('challenge_required','true');
// user will receive push notification in their registered MFA devices
identifier =
UserManagement.initVerificationMethod(Auth.VerificationMethod.SALESFORCE_AUTHENTICATOR,
'Connect to My Salesforce Org', extras);
}

public Auth.VerificationResult verifyVerification() {
// requiring identifier from the initVerification
// user will need to take the action on their registered MFA devices
return UserManagement.verifyVerificationMethod(identifier, '' ,
Auth.VerificationMethod.SALESFORCE_AUTHENTICATOR);
}
```

obfuscateUser(userId, username)

Scrambles users' data on their request when they no longer want their personal data recognized in Salesforce. When you invoke the method for the user, the data becomes anonymous, and you can never recover it. Use this method to set the username to a specific value after it's scrambled.

Signature

```
public static void obfuscateUser(Id userId, String username)
```

Parameters

userId
Type: Id

ID of the user whose data this method scrambles.

username
Type: String

The username after the user's data is scrambled. Sets the value of the scrambled username to a specific string.

Return Value

Type: void

Usage

This method is introduced in API version 43.0. It isn't available in earlier versions.

You can use the `obfuscateUser` method to protect the personal information of your org's users. When invoked, Salesforce permanently scrambles the user's object data and replaces it with random character strings. The user's detail page exists, but the fields contain meaningless strings of characters. Salesforce merely obfuscates (scrambles) personal data because you can't delete a user in Salesforce; you can only disable or deactivate a user. In other words, the user record remains in the database and this method performs a soft delete.

Note: Take care when using this method. The users' data becomes anonymous and can never be recovered.

Considerations

- This method requires that the org's User Management setting, **Scramble Specific Users' Data**, is enabled from Setup.

- This method affects the standard fields of the user object—excluding a few fields such as the user ID, timezone, locale, and profile.
- It is recommended that you note the user's ID and other attributes for post processing, such as the email address, if you want to send the user a confirmation.
- This method changes only the user object. The association between the user and other objects is removed, but no other objects are changed. For example, contact, ThirdPartyAccountLink (TPAL), and user password authentication (UPA) objects remain unchanged.

Note: Assure your admins that invoking this method doesn't trigger an email change notification.

This method is part of our effort to protect users' personal data and privacy. For more information on what you can do to actively protect user data, see Data Protection and Privacy in Salesforce Help.

`obfuscateUser(userId)`

Scrambles users' data on their request when they no longer want their personal data recognized in Salesforce. When you invoke the method for the user, the data becomes anonymous, and you can never recover it.

Signature

```
public static void obfuscateUser(Id userId)
```

Parameters

`userId`
Type: Id

ID of the user whose data this method scrambles.

Return Value

Type: void

Usage

This method is introduced in API version 43.0. It isn't available in earlier versions.

You can use the `obfuscateUser` method to protect the personal information of your org's users. When invoked, Salesforce permanently scrambles the user's object data and replaces it with random character strings. The user's detail page exists, but the fields contain meaningless strings of characters. Salesforce merely obfuscates (scrambles) personal data because you can't delete a user in Salesforce; you can only disable or deactivate a user. In other words, the user record remains in the database and this method performs a soft delete.

Note: Take care when using this method. The users' data becomes anonymous and can never be recovered.

Considerations

- This method requires that the org's User Management setting, **Scramble Specific Users' Data**, is enabled from Setup.
- This method affects the standard fields of the user object—excluding a few fields such as the user ID, timezone, locale, and profile.
- If you want to send the user a confirmation, it's recommended that you note the user's ID and other attributes for post processing, such as the email address.
- This method changes only the user object. The association between the user and other objects is removed, but no other objects are changed. For example, contact, ThirdPartyAccountLink (TPAL), and user password authentication (UPA) objects remain unchanged.

Note: Assure your admins that invoking this method doesn't trigger an email change notification.

This method is part of our effort to protect users' personal data and privacy. For more information on what you can do to actively protect user data, see Data Protection and Privacy in Salesforce Help.

ObfuscateUser Code Example

```
public class UserManagementController{
    public List <User> users {get; set;}

    public UserManagementController()
    {
        Profile p = [select id from profile where name = 'Customer Community User'];

        users = [select username, id from User where profileId=:p.id AND isactive=true];
    }

    //Use method with extreme caution. Data can't be recovered.
    @InvocableMethod(label='User Management' description='Obfuscate User data and more')
    static public void obfuscate(List<User> users)
    {
        String uid = ApexPages.currentPage().getParameters().get('uid');

        if(uid == null)
            return;

        User u = [select contactId from user where id=:uid];

        System.UserManagement.obfuscateUser(uid);
    }
}
```

registerVerificationMethod(method, startUrl)

Registers an identity verification method. Verification methods can be a time-based one-time password (TOTP), email or text verification code, Salesforce Authenticator, or U2F-compatible security key. End users register verification methods for themselves.

Signature

```
public static System.PageReference registerVerificationMethod(Auth.VerificationMethod
method, String startUrl)
```

Parameters

method
Type: Auth.VerificationMethod

Verification method used to verify the identity of the user.

startUrl
Type: String

Path to the page that users see after they log in.

Return Value

Type:System.PageReference

Usage

Use this method to enable users to complete identity verification, such as multi-factor authentication (MFA), or to log in to their Experience Cloud site without a password. Users register these methods to verify their identity when logging in. You create a custom registration page when implementing mobile-centric passwordless logins. See VerifyPasswordlessLogin.

The `PageReference` returned by `registerVerificationMethod` redirects the user to the Salesforce Verify page. If the user enters the correct code, the user is redirected to the Experience Cloud site page specified by the start URL. For example:

```
PageReference pr =
System.UserManagement.registerVerificationMethod(Auth.VerificationMethod.TOTP,startUrl);
PageReference p =
System.UserManagement.deregisterVerificationMethod(userId,Auth.VerificationMethod.SALESFORCE_AUTHENTICATOR);
```

This method is available in API version 43.0 and later.

Note: As a security measure, when users add or update mobile numbers in their detail page, they must log in again to verify their identity. As a result, unsaved changes in the app are lost. To disable this security measure, contact Salesforce Support.

`sendAsyncEmailConfirmation(userId, emailTemplateId, networkId, startUrl)`

Send an email message to a user's email address for verification. The message contains a verification link (URL) that the user clicks to verify the email address later on. You can send email verifications in bulk.

Signature

```
public static Boolean sendAsyncEmailConfirmation(String userId, String emailTemplateId,
String networkId, String startUrl)
```

Parameters

`userId`
Type: String

ID of the user to receive the email confirmation.

`emailTemplateId`
Type: String

ID of the email template in which the verification link is defined.

`networkId`
Type: String

ID of the Experience Cloud site.

`startUrl`
Type: String

The user is redirected to this page after verification, with a success or error message as the parameter. If null, the user is redirected to the login page.

Return Value

Type: Boolean

Indicates whether sending the email message succeeded or failed.

Usage

Sending an async email message is good practice to ensure that users are registered with a valid email address that they truly own. To determine which users receive an email with the verification link, check whether the User Verified Email field in the User detail page is set to true. You can also get this information from the `TwoFactorMethodsInfo` API.

Send async email verification to customers and partners to verify their email address. These users must verify their email address before they can log in with email OTP (passwordless login).

The error code and description are passed as query parameters so that you can process any errors when building a custom landing page.

Example

```
System.UserManagement.sendAsyncEmailConfirmation('005RM000001a0Ox',
'00XRM000000hxnG','0DBRM000000015i', '/s/contactsupport');
```

verifyPasswordlessLogin(userId, method, identifier, code, startUrl)

Completes a verification challenge during a passwordless login that uses a custom Verify page (Visualforce only). If the user who is trying to log in enters the verification code successfully, the user is logged in.

Signature

```
public static Auth.VerificationResult verifyPasswordlessLogin(Id userId,
Auth.VerificationMethod method, String identifier, String code, String startUrl)
```

Parameters

userId
Type: Id

ID of the user who's logging in.

method
Type: Auth.VerificationMethod

Method used to verify the identity of the user, which can be either EMAIL or SMS.

identifier
Type: String

ID of the verification attempt received from the `initPasswordlessLogin` method.

code
Type: String

Code used to verify the identity of the user.

startUrl
Type: String

The page where the user is directed after successful login.

Return Value

Type: Auth.VerificationResult

Result of the verification challenge, which includes the message displayed, and where the user is directed if they enter the verification code correctly.

Usage

Call this method to complete the passwordless login authentication process. It validates the verification method and verification code. It also checks that the identifier is the same as the one returned by `initPasswordlessLogin` on page 4003.

Example

For an example, see `Auth.VerificationResult`.

verifyRegisterVerificationMethod(code, method)

Completes registering a user's email address or phone number as a verification method when customizing the identity verification process.

Signature

```
public static String verifyRegisterVerificationMethod(String code,
Auth.VerificationMethod method)
```

Parameters

code
Type: String

Code used to verify the identity of the user.

method
Type: Auth.VerificationMethod

Method used to verify the identity of the user, which can be either EMAIL or SMS.

Return Value

Type: String

If the user enters an incorrect verification code, the method returns an error message.

Usage

Call `verifyRegisterVerificationMethod` to complete the process of registering a user's verification method. This method checks whether the user entered the correct verification code. If the verification code is correct, the method

- Confirms that the user entered the correct verification code
- From the user's detail page, updates the user's verification method status (sets the verification bit)
- Sends an email to the user confirming that a verification method has been added to their record

If the verification code is incorrect, an error message is returned.

Note: If users want to change their email address after registering one, don't use the `initRegisterVerificationMethod` and `verify RegisterVerificationMethod` methods. To enable automatic identity verification for email address changes, from the Identity Verification Setup page, select the field **Require email confirmations for email address changes (applies to users in Experience Builder sites)**.

Example

Here's a code example that registers a user's phone number as a verification method. When the user enters a verification code on the Visualforce page, it invokes `registerUser()`. The method gets the User ID of the user who's registering the verification method and the user's phone number. It also gets the user's registration status to check whether the phone number is verified already. If the user is registered with a different phone number, the number is updated.

```
public void registerUser() {
        try {
            exceptionText='';
            String userId = UserInfo.getUserId();
            User u = [Select MobilePhone, Id from User Where Id=:userId];
             currPhone = u.MobilePhone;
             mobilePhone = getFormattedSms(mobilePhone);
           if (mobilePhone != null && mobilePhone != '') {
           u.MobilePhone = mobilePhone;
           update u;
            // We're updating the email and phone number before verifying. Roll back
           //  the change in the verify API if it is unsuccessful.
            exceptionText = System.
            UserManagement.initRegisterVerificationMethod(Auth.VerificationMethod.SMS);
            if(exceptionText!= null && exceptionText!=''){
                isInit = false;
                showInitException = true;
            } else {
                 isInit = false;
                 isVerify = true;
            }
            } else {
                showInitException = true;
            }
        } catch (Exception e) {
            exceptionText = e.getMessage();
            isInit = false;
            showInitException = true;
        }
    }

public void verifyUser() {
    // Take the user's input for the code sent to their phone number
    exceptionText = System.UserManagement.
        verifyRegisterVerificationMethod(code, Auth.VerificationMethod.SMS);
    if(exceptionText != null && exceptionText !=''){
```

```
    showInitException = true;
    } else {
            //Success
    }
}
```

verifySelfRegistration(method, identifier, code, startUrl)

Completes a verification challenge when creating a custom (Visualforce) Verify page for Experience Cloud site self-registration. If the person who is attempting to register enters the verification code successfully, the user is created and logged in.

Signature

```
public static Auth.VerificationResult verifySelfRegistration(Auth.VerificationMethod
method, String identifier, String code, String startUrl)
```

Parameters

method
Type: Auth.VerificationMethod

Method used to verify the identity of the user, which can be either EMAIL or SMS.

identifier
Type: String

The unique identifier received from the `initSelfRegistration` method.

code
Type: String

Code used to verify the identity of the user.

startUrl
Type: String

The page where the user is directed after successful self-registration.

Return Value

Type: Auth.VerificationResult

Result of the verification challenge, which includes the message displayed, and where the user is directed when they enter the verification code correctly.

Usage

By default, when users sign up for your Experience Cloud site with an email address or phone number, Salesforce sends them a verification code and generates a Verify page. This Verify page is where users enter the verification code to confirm their identity. You can replace this Salesforce-generated Verify page with a custom Verify page that you create with Visualforce. Then you invoke the verification process with Apex methods.

First, call the `initSelfRegistration` method, which returns the identifier of the user to create. Then call this `verifySelfRegistration` method to complete the verification process. If the user enters the verification code correctly, the user is created and directed to the page specified in the `startURL`.

This method returns the verification result, which contains the verification status and, if the user is created, the session ID. If the verification method is SMS, the User object must contain a properly formatted mobile number, which is country code, space, and then phone number, for example, +1 1234567890. Use `System.UserManagement.formatPhoneNumber` to ensure that the phone number is formatted correctly.

Example

This code contains the result of a verification challenge that registers a new user.

```
String id = System.UserManagement.initSelfRegistration
          (Auth.VerificationMethod.SMS, user);
        Auth.VerificationResult res = System.UserManagement.verifySelfRegistration
          (Auth.VerificationMethod.SMS, id, '123456', null);
        if(res.success == true){
     //redirect
}
```

verifyVerificationMethod(identifier, code, method)

Completes the verification service for email, phone (SMS), Salesforce Authenticator, password, or time-based one-time password (TOTP) verification methods.

Signature

```
public static VerificationResult verifyVerificationMethod(String identifier, String
code, Auth.VerificationMethod method)
```

Parameters

identifier
Type: String

Identifier returned from `initVerificationMethod` for `EMAIL`, `SMS`, and `SALESFORCE_AUTHENTICATOR`.

code
Type: String

Code used to verify the user's identity for `EMAIL`, `SMS`, or `PASSWORD`.

method
Type: Auth.VerificationMethod

Method used to verify the user's identity, which can be `EMAIL`, `PASSWORD`, `SALESFORCE_AUTHENTICATOR`, `SMS`, or `TOTP`.

Return Value

Type: VerificationResult

Usage

Use this method along with its paired `initVerificationMethod` to customize a verification service for `EMAIL`, `SMS`, or `SALESFORCE_AUTHENTICATOR` verification methods. Or use this method alone to provide a complete verification service for `PASSWORD` and `TOTP` verification methods.

This method checks whether the user entered the correct verification code or password. If the verification code or password is correct, the method verifies the user's identity.

If the verification code or password isn't valid, the service returns an error message.

Examples

This example shows multi-factor authentication using email.

```
public void initVerification() {
// user will receive code on their registered verified email
 identifier = UserManagement.initVerificationMethod(Auth.VerificationMethod.EMAIL);
}

public Auth.VerificationResult verifyVerification() {
// requiring identifier from the initVerification
// the code will need to be entered in this method
return UserManagement.verifyVerificationMethod(identifier, code ,
Auth.VerificationMethod.EMAIL);
}
```

The next two examples show multi-factor authentication using only the `verifyVerificationMethod` for password and TOTP verifications.

```
public Auth.VerificationResult verifyVerification() {
// user will enter their password as a param in the verifyVerificationMethod for password
 verification method
return UserManagement.verifyVerificationMethod('', password ,
Auth.VerificationMethod.PASSWORD);
}
```

```
public Auth.VerificationResult verifyVerification() {
// user will enter their registered time-based one-time password (TOTP) code (token)
return UserManagement.verifyVerificationMethod('', code , Auth.VerificationMethod.TOTP);
}
```

UUID Class

Contains methods to randomly generate a version 4 universally unique identifier (UUID), compare UUIDs, and convert UUID instance to a string.

Namespace

System

Usage

The UUID is generated using a cryptographically strong pseudo-random number generator and is represented as 32 hexadecimal values.

IN THIS SECTION:

UUID Methods

UUID Methods

The following are methods for `UUID`.

IN THIS SECTION:

equals(obj)
Compares a UUID instance with the specified object and returns true if both are equal. Otherwise, returns false.

fromString(str)
Converts a 32 character hexadecimal string representation of a UUID to a UUID instance.

hashCode()
Returns the hashcode corresponding to the UUID instance.

randomUUID()
A static method that randomly generates a version 4 UUID.

toString()
Returns the string representation of the UUID instance.

`equals(obj)`

Compares a UUID instance with the specified object and returns true if both are equal. Otherwise, returns false.

Signature

```
public Boolean equals(Object obj)
```

Parameters

obj
Type: Object

The UUID object to be compared.

Return Value

Type: Boolean

Example

```
// UUIDs are equal when all the characters in the UUID are the same
String uuidStr = '707b2538-98bb-41e7-95e3-1d77bf42b102';
UUID fromStr = UUID.fromString(uuidStr);
UUID fromStr2 = UUID.fromString(uuidStr);
Assert.isTrue(fromStr.equals(fromStr2));

// A UUID is never equal to a String or any non-UUID object
Assert.isFalse(fromStr.equals(uuidStr));
```

fromString(str)

Converts a 32 character hexadecimal string representation of a UUID to a UUID instance.

Signature

```
public static System.UUID fromString(String str)
```

Parameters

str
Type: String

Return Value

Type: System.UUID

Example

```
String uuidStr = '707b2538-98bb-41e7-95e3-1d77bf42b102';
UUID fromStr = UUID.fromString(uuidStr);

UUID.fromString(null); // Throws NullPointerException

UUID.fromString('not a uuid'); // Throws IllegalArgumentException
```

hashCode()

Returns the hashcode corresponding to the UUID instance.

Signature

```
public Integer hashCode()
```

Return Value

Type: Integer

randomUUID()

A static method that randomly generates a version 4 UUID.

Signature

```
public static System.UUID randomUUID()
```

Return Value

Type: System.UUID

A 32 hexadecimal value of the UUID generated.

Example

```
UUID randomUUID = UUID.randomUUID();
system.debug(randomUUID); // Prints the UUID string that was randomly generated
```

`toString()`

Returns the string representation of the UUID instance.

Signature

```
public String toString()
```

Return Value

Type: String

Version Class

Use the Version methods to get the version of a first-generation managed package, and to compare package versions.

Namespace

System

Usage

A package version is a number that identifies the set of components uploaded in a package. The version number has the format *majorNumber.minorNumber.patchNumber* (for example, 2.1.3). The major and minor numbers increase to a chosen value during every major release. The *patchNumber* is generated and updated only for a patch release.

A called component can check the version against which the caller was compiled using the `System.requestVersion` method and behave differently depending on the caller's expectations. This allows you to continue to support existing behavior in classes and triggers in previous package versions while continuing to evolve the code.

The value returned by the `System.requestVersion` method is an instance of this class with a two-part version number containing a major and a minor number. Since the `System.requestVersion` method doesn't return a patch number, the patch number in the returned Version object is null.

The `System.Version` class can also hold also a three-part version number that includes a patch number.

Example

This example shows how to use the methods in this class, along with the `requestVersion` method, to determine the managed package version of the code that is calling your package.

```
if (System.requestVersion() == new Version(1,0))
{
    // Do something
}
if ((System.requestVersion().major() == 1)
     && (System.requestVersion().minor() > 0)
```

```
     && (System.requestVersion().minor() <=9))
{
    // Do something different for versions 1.1 to 1.9
}
else if (System.requestVersion().compareTo(new Version(2,0)) >= 0)
{
    // Do something completely different for versions 2.0 or greater
}
```

IN THIS SECTION:

Version Constructors

Version Methods

Version Constructors

The following are constructors for `Version`.

IN THIS SECTION:

Version(major, minor)

Creates a new instance of the `Version` class as a two-part package version using the specified major and minor version numbers.

Version(major, minor, patch)

Creates a new instance of the `Version` class as a three-part package version using the specified major, minor, and patch version numbers.

`Version(major, minor)`

Creates a new instance of the `Version` class as a two-part package version using the specified major and minor version numbers.

Signature

```
public Version(Integer major, Integer minor)
```

Parameters

major

Type: Integer

The major version number.

minor

Type: Integer

The minor version number.

`Version(major, minor, patch)`

Creates a new instance of the `Version` class as a three-part package version using the specified major, minor, and patch version numbers.

Signature

```
public Version(Integer major, Integer minor, Integer patch)
```

Parameters

major
Type: Integer

The major version number.

minor
Type: Integer

The minor version number.

patch
Type: Integer

The patch version number.

Version Methods

The following are methods for `Version`. All are instance methods.

IN THIS SECTION:

compareTo(version)
Compares the current version with the specified version.

major()
Returns the major package version of the of the calling code.

minor()
Returns the minor package version of the calling code.

patch()
Returns the patch package version of the calling code or `null` if there is no patch version.

`compareTo(version)`

Compares the current version with the specified version.

Signature

```
public Integer compareTo(System.Version version)
```

Parameters

version
Type: System.Version

Return Value

Type: Integer

Returns one of the following values:

- zero if the current package version is equal to the specified package version
- an Integer value greater than zero if the current package version is greater than the specified package version
- an Integer value less than zero if the current package version is less than the specified package version

Usage

If a two-part version is being compared to a three-part version, the patch number is ignored and the comparison is based only on the major and minor numbers.

`major()`

Returns the major package version of the of the calling code.

Signature

```
public Integer major()
```

Return Value

Type: Integer

`minor()`

Returns the minor package version of the calling code.

Signature

```
public Integer minor()
```

Return Value

Type: Integer

`patch()`

Returns the patch package version of the calling code or `null` if there is no patch version.

Signature

```
public Integer patch()
```

Return Value

Type: Integer

WebServiceCallout Class

Enables making callouts to SOAP operations on an external Web service. This class is used in the Apex stub class that is auto-generated from a WSDL.

Namespace

System

IN THIS SECTION:

WebServiceCallout Methods

SEE ALSO:

Apex Developer Guide: SOAP Services: Defining a Class from a WSDL Document

WebServiceCallout Methods

The following is the static method for `WebServiceCallout`.

IN THIS SECTION:

invoke(stub, request, response, infoArray)
Invokes an external SOAP web service operation based on an Apex class that is auto-generated from a WSDL.

`invoke(stub, request, response, infoArray)`

Invokes an external SOAP web service operation based on an Apex class that is auto-generated from a WSDL.

Signature

```
public static void invoke(Object stub, Object request, Map<String,Object> response,
List<String> infoArray)
```

Parameters

`stub`
Type: Object

An instance of the Apex class that is auto-generated from a WSDL (the stub class).

`request`
Type: Object

The request to the external service. The request is an instance of a type that is created as part of the auto-generated stub class.

`response`
Type: Map<String, Object>

A map of key-value pairs that represent the response that the external service sends after receiving the request. In each pair, the key is a response identifier. The value is the response object, which is an instance of a type that is created as part of the auto-generated stub class.

`infoArray`
Type: String[]

An array of strings that contains information about the callout—web service endpoint, SOAP action, request, and response. The order of the elements in the array matters.

- Element at index 0 (`[0]`): One of the following options for identifying the URL of the external web service.

  - Endpoint URL. For example: `'http://YourServer/YourService'`
  - Named credential URL, which contains the scheme `callout`, the name of the named credential, and optionally, an appended path. For example: `'callout:MyNamedCredential/some/path'`
- Element at index 1 (`[1]`): The SOAP action. For example: `'urn:dotnet.callouttest.soap.sforce.com/EchoString'`
- Element at index 2 (`[2]`): The request namespace. For example: `'http://doc.sample.com/docSample'`
- Element at index 3 (`[3]`): The request name. For example: `'EchoString'`
- Element at index 4 (`[4]`): The response namespace. For example: `'http://doc.sample.com/docSample'`
- Element at index 5 (`[5]`): The response name. For example: `'EchoStringResponse'`
- Element at index 6 (`[6]`): The response type. For example: `'docSample.EchoStringResponse_element'`

Return Value

Type: Void

SEE ALSO:

Apex Developer Guide: Named Credentials as Callout Endpoints

WebServiceMock Interface

Enables sending fake responses when testing Web service callouts of a class auto-generated from a WSDL.

Namespace

System

Usage

For an implementation example, see Test Web Service Callouts.

WebServiceMock Methods

The following are methods for `WebServiceMock`.

IN THIS SECTION:

doInvoke(stub, soapRequest, responseMap, endpoint, soapAction, requestName, responseNamespace, responseName, responseType)
The implementation of this method is called by the Apex runtime to send a fake response when a Web service callout is made after `Test.setMock` has been called.

doInvoke(stub, soapRequest, responseMap, endpoint, soapAction, requestName, responseNamespace, responseName, responseType)

The implementation of this method is called by the Apex runtime to send a fake response when a Web service callout is made after `Test.setMock` has been called.

Signature

```
public Void doInvoke(Object stub, Object soapRequest, Map<String,Object> responseMap,
String endpoint, String soapAction, String requestName, String responseNamespace, String
responseName, String responseType)
```

Parameters

stub
Type: Object

An instance of the auto-generated class.

soapRequest
Type: Object

The SOAP Web service request being invoked.

responseMap
Type: Map<String, Object>

A collection of key/value pairs representing the response to send for the request.

When implementing this interface, set the *responseMap* argument to a key/value pair representing the response desired.

endpoint
Type: String

The endpoint URL for the request.

soapAction
Type: String

The requested SOAP operation.

requestName
Type: String

The requested SOAP operation name.

responseNamespace
Type: String

The response namespace.

responseName
Type: String

The name of the response element as defined in the WSDL.

responseType
Type: String

The class for the response as defined in the auto-generated class.

Return Value

Type: Void

Usage

`XmlStreamReader` Class

The `XmlStreamReader` class provides methods for forward, read-only access to XML data. You can pull data from XML or skip unwanted events. You can parse nested XML content that's up to 50 nodes deep.

Namespace

System

Usage

The `XmlStreamReader` class is similar to the XMLStreamReader utility class from StAX (Streaming API for XML). StAX is an API to read and write XML documents, originating from the Java programming language community.

Note: The `XmlStreamReader` class in Apex is based on its counterpart in Java. See `Java XMLStreamReader class`.

IN THIS SECTION:

XmlStreamReader Constructors

XmlStreamReader Methods

SEE ALSO:

Apex Developer Guide: Reading XML Using Streams

XmlStreamReader Constructors

The following are constructors for `XmlStreamReader`.

IN THIS SECTION:

XmlStreamReader(xmlInput)

Creates a new instance of the `XmlStreamReader` class for the specified XML input.

`XmlStreamReader(xmlInput)`

Creates a new instance of the `XmlStreamReader` class for the specified XML input.

Signature

```
public XmlStreamReader(String xmlInput)
```

Parameters

`xmlInput`

Type: String

The XML string input.

XmlStreamReader Methods

The following are methods for `XmlStreamReader`. All are instance methods.

IN THIS SECTION:

getAttributeCount()

Returns the number of attributes on the start element, excluding namespace definitions.

getAttributeLocalName(index)

Returns the local name of the attribute at the specified index.

getAttributeNamespace(index)

Returns the namespace URI of the attribute at the specified index.

getAttributePrefix(index)

Returns the prefix of this attribute at the specified index.

getAttributeType(index)

Returns the XML type of the attribute at the specified index.

getAttributeValue(namespaceUri, localName)

Returns the value of the attribute in the specified *`localName`* at the specified URI.

getAttributeValueAt(index)

Returns the value of the attribute at the specified index.

getEventType()

Returns the type of XML event the cursor is pointing to.

getLocalName()

Returns the local name of the current event.

getLocation()

Return the current location of the cursor.

getNamespace()

If the current event is a start element or end element, this method returns the URI of the prefix or the default namespace.

getNamespaceCount()

Returns the number of namespaces declared on a start element or end element.

getNamespacePrefix(index)

Returns the prefix for the namespace declared at the index.

getNamespaceURI(prefix)

Return the URI for the given prefix.

getNamespaceURIAt(index)

Returns the URI for the namespace declared at the index.

getPIData()

Returns the data section of a processing instruction.

getPITarget()

Returns the target section of a processing instruction.

getPrefix()

Returns the prefix of the current XML event or `null` if the event does not have a prefix.

getText()

Returns the current value of the XML event as a string.

getVersion()

Returns the XML version specified on the XML declaration. Returns `null` if none was declared.

hasName()

Returns `true` if the current XML event has a name. Returns `false` otherwise.

hasNext()

Returns `true` if there are more XML events and `false` if there are no more XML events.

hasText()

Returns `true` if the current event has text, `false` otherwise.

isCharacters()

Returns `true` if the cursor points to a character data XML event. Otherwise, returns `false`.

isEndElement()

Returns `true` if the cursor points to an end tag. Otherwise, it returns `false`.

isStartElement()

Returns `true` if the cursor points to a start tag. Otherwise, it returns `false`.

isWhiteSpace()

Returns `true` if the cursor points to a character data XML event that consists of all white space. Otherwise it returns `false`.

next()

Reads the next XML event. A processor may return all contiguous character data in a single chunk, or it may split it into several chunks. Returns an integer which indicates the type of event.

nextTag()

Skips any white space (the `isWhiteSpace` method returns `true`), comment, or processing instruction XML events, until a start element or end element is reached. Returns the index for that XML event.

setCoalescing(returnAsSingleBlock)

If you specify `true` for *`returnAsSingleBlock`*, text is returned in a single block, from a start element to the first end element or the next start element, whichever comes first. If you specify it as `false`, the parser may return text in multiple blocks.

setNamespaceAware(isNamespaceAware)

If you specify `true` for *`isNamespaceAware`*, the parser recognizes namespace. If you specify it as `false`, the parser does not. The default value is `true`.

toString()

Returns a string containing the length of the input XML given to `XmlStreamReader` and the first 50 characters of the input XML.

`getAttributeCount()`

Returns the number of attributes on the start element, excluding namespace definitions.

Signature

```
public Integer getAttributeCount()
```

Return Value

Type: Integer

Usage

This method is only valid on a start element or attribute XML events. The count for the number of attributes for an attribute XML event starts with zero.

getAttributeLocalName(index)

Returns the local name of the attribute at the specified index.

Signature

```
public String getAttributeLocalName(Integer index)
```

Parameters

index
Type: Integer

Return Value

Type: String

Usage

If there is no name, an empty string is returned. This method is only valid with start element or attribute XML events.

getAttributeNamespace(index)

Returns the namespace URI of the attribute at the specified index.

Signature

```
public String getAttributeNamespace(Integer index)
```

Parameters

index
Type: Integer

Return Value

Type: String

Usage

If no namespace is specified, `null` is returned. This method is only valid with start element or attribute XML events.

getAttributePrefix(index)

Returns the prefix of this attribute at the specified index.

Signature

```
public String getAttributePrefix(Integer index)
```

Parameters

index
Type: Integer

Return Value

Type: String

Usage

If no prefix is specified, `null` is returned. This method is only valid with start element or attribute XML events.

getAttributeType(index)

Returns the XML type of the attribute at the specified index.

Signature

```
public String getAttributeType(Integer index)
```

Parameters

index
Type: Integer

Return Value

Type: String

Usage

For example, `id` is an attribute type. This method is only valid with start element or attribute XML events.

getAttributeValue(namespaceUri, localName)

Returns the value of the attribute in the specified *localName* at the specified URI.

Signature

```
public String getAttributeValue(String namespaceUri, String localName)
```

Parameters

namespaceUri
Type: String

localName
Type: String

Return Value

Type: String

Usage

Returns `null` if the value is not found. You must specify a value for *localName*. This method is only valid with start element or attribute XML events.

getAttributeValueAt(index)

Returns the value of the attribute at the specified index.

Signature

```
public String getAttributeValueAt(Integer index)
```

Parameters

index
Type: Integer

Return Value

Type: String

Usage

This method is only valid with start element or attribute XML events.

getEventType()

Returns the type of XML event the cursor is pointing to.

Signature

```
public System.XmlTag getEventType()
```

Return Value

Type: System.XmlTag

`XmlTag` Enum

The values for `XmlTag` are:

- `ATTRIBUTE`
- `CDATA`
- `CHARACTERS`
- `COMMENT`
- `DTD`
- `END_DOCUMENT`
- `END_ELEMENT`
- `ENTITY_DECLARATION`
- `ENTITY_REFERENCE`
- `NAMESPACE`
- `NOTATION_DECLARATION`
- `PROCESSING_INSTRUCTION`
- `SPACE`
- `START_DOCUMENT`
- `START_ELEMENT`

`getLocalName()`

Returns the local name of the current event.

Signature

```
public String getLocalName()
```

Return Value

Type: String

Usage

For start element or end element XML events, it returns the local name of the current element. For the entity reference XML event, it returns the entity name. The current XML event must be start element, end element, or entity reference.

`getLocation()`

Return the current location of the cursor.

Signature

```
public String getLocation()
```

Return Value

Type: String

Usage

If the location is unknown, returns -1. The location information is only valid until the `next` method is called.

`getNamespace()`

If the current event is a start element or end element, this method returns the URI of the prefix or the default namespace.

Signature

```
public String getNamespace()
```

Return Value

Type: String

Usage

Returns `null` if the XML event does not have a prefix.

`getNamespaceCount()`

Returns the number of namespaces declared on a start element or end element.

Signature

```
public Integer getNamespaceCount()
```

Return Value

Type: Integer

Usage

This method is only valid on a start element, end element, or namespace XML event.

`getNamespacePrefix(index)`

Returns the prefix for the namespace declared at the index.

Signature

```
public String getNamespacePrefix(Integer index)
```

Parameters

index
Type: Integer

Return Value

Type: String

Usage

Returns `null` if this is the default namespace declaration. This method is only valid on a start element, end element, or namespace XML event.

`getNamespaceURI(prefix)`

Return the URI for the given prefix.

Signature

```
public String getNamespaceURI(String prefix)
```

Parameters

`prefix`
Type: String

Return Value

Type: String

Usage

The returned URI depends on the current state of the processor.

`getNamespaceURIAt(index)`

Returns the URI for the namespace declared at the index.

Signature

```
public String getNamespaceURIAt(Integer index)
```

Parameters

`index`
Type: Integer

Return Value

Type: String

Usage

This method is only valid on a start element, end element, or namespace XML event.

`getPIData()`

Returns the data section of a processing instruction.

Signature

```
public String getPIData()
```

Return Value

Type: String

getPITarget()

Returns the target section of a processing instruction.

Signature

```
public String getPITarget()
```

Return Value

Type: String

getPrefix()

Returns the prefix of the current XML event or `null` if the event does not have a prefix.

Signature

```
public String getPrefix()
```

Return Value

Type: String

getText()

Returns the current value of the XML event as a string.

Signature

```
public String getText()
```

Return Value

Type: String

Usage

The valid values for the different events are:

- The string value of a character XML event
- The string value of a comment

- The replacement value for an entity reference. For example, assume `getText` reads the following XML snippet:

```
<!ENTITY
  Title "Salesforce For Dummies" >
     ]>
  <moo a=\"b\">Name &Title;</moo>';
```

  The `getText` method returns `Salesforce for Dummies`, not `&Title`.
- The string value of a CDATA section
- The string value for a space XML event
- The string value of the internal subset of the DTD

`getVersion()`

Returns the XML version specified on the XML declaration. Returns `null` if none was declared.

Signature

```
public String getVersion()
```

Return Value

Type: String

`hasName()`

Returns `true` if the current XML event has a name. Returns `false` otherwise.

Signature

```
public Boolean hasName()
```

Return Value

Type: Boolean

Usage

This method is only valid for start element and stop element XML events.

`hasNext()`

Returns `true` if there are more XML events and `false` if there are no more XML events.

Signature

```
public Boolean hasNext()
```

Return Value

Type: Boolean

Usage

This method returns `false` if the current XML event is end document.

`hasText()`

Returns `true` if the current event has text, `false` otherwise.

Signature

```
public Boolean hasText()
```

Return Value

Type: Boolean

Usage

The following XML events have text: characters, entity reference, comment and space.

`isCharacters()`

Returns `true` if the cursor points to a character data XML event. Otherwise, returns `false`.

Signature

```
public Boolean isCharacters()
```

Return Value

Type: Boolean

`isEndElement()`

Returns `true` if the cursor points to an end tag. Otherwise, it returns `false`.

Signature

```
public Boolean isEndElement()
```

Return Value

Type: Boolean

`isStartElement()`

Returns `true` if the cursor points to a start tag. Otherwise, it returns `false`.

Signature

```
public Boolean isStartElement()
```

Return Value

Type: Boolean

`isWhiteSpace()`

Returns `true` if the cursor points to a character data XML event that consists of all white space. Otherwise it returns `false`.

Signature

```
public Boolean isWhiteSpace()
```

Return Value

Type: Boolean

`next()`

Reads the next XML event. A processor may return all contiguous character data in a single chunk, or it may split it into several chunks. Returns an integer which indicates the type of event.

Signature

```
public Integer next()
```

Return Value

Type: Integer

`nextTag()`

Skips any white space (the `isWhiteSpace` method returns `true`), comment, or processing instruction XML events, until a start element or end element is reached. Returns the index for that XML event.

Signature

```
public Integer nextTag()
```

Return Value

Type: Integer

Usage

This method throws an error if elements other than white space, comments, processing instruction, start elements or stop elements are encountered.

`setCoalescing(returnAsSingleBlock)`

If you specify `true` for *`returnAsSingleBlock`*, text is returned in a single block, from a start element to the first end element or the next start element, whichever comes first. If you specify it as `false`, the parser may return text in multiple blocks.

Signature

```
public Void setCoalescing(Boolean returnAsSingleBlock)
```

Parameters

returnAsSingleBlock
Type: Boolean

Return Value

Type: Void

setNamespaceAware(isNamespaceAware)

If you specify `true` for *isNamespaceAware*, the parser recognizes namespace. If you specify it as `false`, the parser does not. The default value is `true`.

Signature

```
public Void setNamespaceAware(Boolean isNamespaceAware)
```

Parameters

isNamespaceAware
Type: Boolean

Return Value

Type: Void

toString()

Returns a string containing the length of the input XML given to `XmlStreamReader` and the first 50 characters of the input XML.

Signature

```
public String toString()
```

Return Value

Type: String

XmlStreamWriter Class

The `XmlStreamWriter` class provides methods for writing XML data.

Namespace

System

Usage

You can use the `XmlStreamWriter` class to programmatically construct an XML document, then use HTTP classes to send the document to an external server.

The `XmlStreamWriter` class is similar to the XMLStreamWriter utility class from StAX (Streaming API for XML). StAX is an API to read and write XML documents, originating from the Java programming language community.

Note: The `XmlStreamWriter` class in Apex is based on its counterpart in Java. See `Java XMLStreamWriter class`.

IN THIS SECTION:

XmlStreamWriter Constructors

XmlStreamWriter Methods

SEE ALSO:

Http Class

HttpRequest Class

HttpResponse Class

XmlStreamWriter Constructors

The following are constructors for `XmlStreamWriter`.

IN THIS SECTION:

XmlStreamWriter()
Creates a new instance of the `XmlStreamWriter` class.

`XmlStreamWriter()`

Creates a new instance of the `XmlStreamWriter` class.

Signature

```
public XmlStreamWriter()
```

XmlStreamWriter Methods

The following are methods for `XmlStreamWriter`. All are instance methods.

IN THIS SECTION:

close()
Closes this instance of an XmlStreamWriter and free any resources associated with it.

getXmlString()
Returns the XML written by the XmlStreamWriter instance.

setDefaultNamespace(uri)

Binds the specified URI to the default namespace. This URI is bound in the scope of the current START_ELEMENT – END_ELEMENT pair.

writeAttribute(prefix, namespaceUri, localName, value)

Writes an attribute to the output stream.

writeCData(data)

Writes the specified CData to the output stream.

writeCharacters(text)

Writes the specified text to the output stream.

writeComment(comment)

Writes the specified comment to the output stream.

writeDefaultNamespace(namespaceUri)

Writes the specified namespace to the output stream.

writeEmptyElement(prefix, localName, namespaceUri)

Writes an empty element tag to the output stream.

writeEndDocument()

Closes any start tags and writes corresponding end tags to the output stream.

writeEndElement()

Writes an end tag to the output stream, relying on the internal state of the writer to determine the prefix and local name.

writeNamespace(prefix, namespaceUri)

Writes the specified namespace to the output stream.

writeProcessingInstruction(target, data)

Writes the specified processing instruction.

writeStartDocument(encoding, version)

Writes the XML Declaration using the specified XML encoding and version.

writeStartElement(prefix, localName, namespaceUri)

Writes the start tag specified by *localName* to the output stream.

close()

Closes this instance of an XmlStreamWriter and free any resources associated with it.

Signature

```
public Void close()
```

Return Value

Type: Void

getXmlString()

Returns the XML written by the XmlStreamWriter instance.

Signature

```
public String getXmlString()
```

Return Value

Type: String

setDefaultNamespace(uri)

Binds the specified URI to the default namespace. This URI is bound in the scope of the current START_ELEMENT – END_ELEMENT pair.

Signature

```
public Void setDefaultNamespace(String uri)
```

Parameters

uri
Type: String

Return Value

Type: Void

writeAttribute(prefix, namespaceUri, localName, value)

Writes an attribute to the output stream.

Signature

```
public Void writeAttribute(String prefix, String namespaceUri, String localName, String value)
```

Parameters

prefix
Type: String

namespaceUri
Type: String

localName
Type: String

Specifies the name of the attribute.

value
Type: String

Return Value

Type: Void

writeCData(data)

Writes the specified CData to the output stream.

Signature

```
public Void writeCData(String data)
```

Parameters

data
Type: String

Return Value

Type: Void

writeCharacters(text)

Writes the specified text to the output stream.

Signature

```
public Void writeCharacters(String text)
```

Parameters

text
Type: String

Return Value

Type: Void

writeComment(comment)

Writes the specified comment to the output stream.

Signature

```
public Void writeComment(String comment)
```

Parameters

comment
Type: String

Return Value

Type: Void

writeDefaultNamespace(namespaceUri)

Writes the specified namespace to the output stream.

Signature

```
public Void writeDefaultNamespace(String namespaceUri)
```

Parameters

namespaceUri
Type: String

Return Value

Type: Void

writeEmptyElement(prefix, localName, namespaceUri)

Writes an empty element tag to the output stream.

Signature

```
public Void writeEmptyElement(String prefix, String localName, String namespaceUri)
```

Parameters

prefix
Type: String

localName
Type: String

Specifies the name of the tag to be written.

namespaceUri
Type: String

Return Value

Type: Void

writeEndDocument()

Closes any start tags and writes corresponding end tags to the output stream.

Signature

```
public Void writeEndDocument()
```

Return Value

Type: Void

writeEndElement()

Writes an end tag to the output stream, relying on the internal state of the writer to determine the prefix and local name.

Signature

```
public Void writeEndElement()
```

Return Value

Type: Void

writeNamespace(prefix, namespaceUri)

Writes the specified namespace to the output stream.

Signature

```
public Void writeNamespace(String prefix, String namespaceUri)
```

Parameters

prefix
Type: String

namespaceUri
Type: String

Return Value

Type: Void

writeProcessingInstruction(target, data)

Writes the specified processing instruction.

Signature

```
public Void writeProcessingInstruction(String target, String data)
```

Parameters

target
Type: String

data
Type: String

Return Value

Type: Void

`writeStartDocument(encoding, version)`

Writes the XML Declaration using the specified XML encoding and version.

Signature

```
public Void writeStartDocument(String encoding, String version)
```

Parameters

`encoding`
Type: String

`version`
Type: String

Return Value

Type: Void

`writeStartElement(prefix, localName, namespaceUri)`

Writes the start tag specified by *`localName`* to the output stream.

Signature

```
public Void writeStartElement(String prefix, String localName, String namespaceUri)
```

Parameters

`prefix`
Type: String

`localName`
Type: String

`namespaceUri`
Type: String

Return Value

Type: Void

TerritoryMgmt Namespace

The `TerritoryMgmt` namespace provides an interface used for territory management.

The following is the interface in the `TerritoryMgmt` namespace.

IN THIS SECTION:

OpportunityTerritory2AssignmentFilter Global Interface
Apex interface that allows an implementing class to assign a single territory to an opportunity.

OpportunityTerritory2AssignmentFilter Global Interface

Apex interface that allows an implementing class to assign a single territory to an opportunity.

Namespace

TerritoryMgmt

Usage

Method called by Opportunity Territory Assignment job to assign territory to opportunity. Input is a list of (up to 1000) opportunityIds that have IsExcludedFromTerritory2Filter=false. Returns a map of OpportunityId to Territory2Id, which is used to update the Territory2Id field on the Opportunity object.

IN THIS SECTION:

OpportunityTerritory2AssignmentFilter Methods

OpportunityTerritory2AssignmentFilter Example Implementation

OpportunityTerritory2AssignmentFilter Methods

The following are methods for `OpportunityTerritory2AssignmentFilter`.

IN THIS SECTION:

getOpportunityTerritory2Assignments(opportunityIds)

Returns the mapping of opportunities to territory IDs. When Salesforce invokes this method, it supplies the list of opportunity IDs, except for opportunities that have been excluded from territory assignment (IsExcludedFromTerritory2Filter=false).

`getOpportunityTerritory2Assignments(opportunityIds)`

Returns the mapping of opportunities to territory IDs. When Salesforce invokes this method, it supplies the list of opportunity IDs, except for opportunities that have been excluded from territory assignment (IsExcludedFromTerritory2Filter=false).

Signature

```
public Map<Id,Id> getOpportunityTerritory2Assignments(List<Id> opportunityIds)
```

Parameters

`opportunityIds`

Type: List<Id>

Opportunity IDs.

Return Value

Type: Map<Id,Id>

A key value pair associating each Territory ID to an Opportunity ID.

OpportunityTerritory2AssignmentFilter Example Implementation

This is an example implementation of the `TerritoryMgmt.OpportunityTerritory2AssignmentFilter` interface.

```
/*** Apex version of the default logic.
* If opportunity's assigned account is assigned to
*  Case 1: 0 territories in active model
*            then set territory2Id = null
*  Case 2: 1 territory in active model
*            then set territory2Id = account's territory2Id
*  Case 3: 2 or more territories in active model
*            then set territory2Id = account's territory2Id that is of highest priority.
*           But if multiple territories have same highest priority, then set territory2Id
 = null
*/
global class OppTerrAssignDefaultLogicFilter implements
TerritoryMgmt.OpportunityTerritory2AssignmentFilter {
    /**
     * No-arg constructor.
     */
     global OppTerrAssignDefaultLogicFilter() {}

     /**
      * Get mapping of opportunity to territory2Id. The incoming list of opportunityIds
contains only those with IsExcludedFromTerritory2Filter=false.
      * If territory2Id = null in result map, clear the opportunity.territory2Id if set.
      * If opportunity is not present in result map, its territory2Id remains intact.
      */
    global Map<Id,Id> getOpportunityTerritory2Assignments(List<Id> opportunityIds) {
        Map<Id, Id> OppIdTerritoryIdResult = new Map<Id, Id>();

        // Get the active territory model Id
        Id activeModelId = getActiveModelId();

        if(activeModelId != null){
            List<Opportunity> opportunities =
              [Select Id, AccountId, Territory2Id from Opportunity where Id IN
:opportunityIds];
            Set<Id> accountIds = new Set<Id>();
            // Create set of parent accountIds
            for(Opportunity opp:opportunities){
                if(opp.AccountId != null){
                    accountIds.add(opp.AccountId);
                    }
                }

                Map<Id,Territory2Priority> accountMaxPriorityTerritory =
getAccountMaxPriorityTerritory(activeModelId, accountIds);

            // For each opportunity, assign the highest priority territory if there is no
 conflict, else assign null.
            for(Opportunity opp: opportunities){
               Territory2Priority tp = accountMaxPriorityTerritory.get(opp.AccountId);
               // Assign highest priority territory if there is only 1.
            if((tp != null) && (tp.moreTerritoriesAtPriority == false) && (tp.territory2Id
```

```
 != opp.Territory2Id)){
                    OppIdTerritoryIdResult.put(opp.Id, tp.territory2Id);
                }else{
                    OppIdTerritoryIdResult.put(opp.Id, null);
                }
            }
        }
        return OppIdTerritoryIdResult;
    }

    /**
      * Query assigned territoryIds in active model for given accountIds.
      * Create a map of accountId to max priority territory.
      */
     private Map<Id,Territory2Priority> getAccountMaxPriorityTerritory(Id activeModelId,
Set<Id> accountIds){
        Map<Id,Territory2Priority> accountMaxPriorityTerritory = new
Map<Id,Territory2Priority>();
        for(ObjectTerritory2Association ota:[Select ObjectId, Territory2Id,
Territory2.Territory2Type.Priority from ObjectTerritory2Association where objectId IN
:accountIds and Territory2.Territory2ModelId = :activeModelId]){
            Territory2Priority tp = accountMaxPriorityTerritory.get(ota.ObjectId);

            if((tp == null) || (ota.Territory2.Territory2Type.Priority > tp.priority)){
                // If this is the first territory examined for account or it has greater
priority than current highest priority territory, then set this as new highest priority
territory.
                tp = new
Territory2Priority(ota.Territory2Id,ota.Territory2.Territory2Type.priority,false);
            }else if(ota.Territory2.Territory2Type.priority == tp.priority){
                // The priority of current highest territory is same as this, so set
moreTerritoriesAtPriority to indicate multiple highest priority territories seen so far.
                tp.moreTerritoriesAtPriority = true;
            }

            accountMaxPriorityTerritory.put(ota.ObjectId, tp);
        }
        return accountMaxPriorityTerritory;
    }


    /**
     * Get the Id of the Active Territory Model.
     * If none exists, return null.
     */
    private Id getActiveModelId() {
        List<Territory2Model> models = [Select Id from Territory2Model where State =
'Active'];
        Id activeModelId = null;
        if(models.size() == 1){
            activeModelId = models.get(0).Id;
        }

        return activeModelId;
```

```
    }

    /**
    * Helper class to help capture territory2Id, its priority, and whether there are more
 territories with same priority assigned to the account.
    */
    private class Territory2Priority {
        public Id territory2Id { get; set; }
        public Integer priority { get; set; }
        public Boolean moreTerritoriesAtPriority { get; set; }

        Territory2Priority(Id territory2Id, Integer priority, Boolean
moreTerritoriesAtPriority){
            this.territory2Id = territory2Id;
            this.priority = priority;
            this.moreTerritoriesAtPriority = moreTerritoriesAtPriority;
        }
    }
}
```

TxnSecurity Namespace

The `TxnSecurity` namespace provides an interface used for transaction security.

The following is the interface and its supporting class in the `TxnSecurity` namespace.

IN THIS SECTION:

Event Class
Contains event information that the `evaluate` method uses to evaluate a transaction security policy.

EventCondition Interface
Allows an implementing class to specify whether to take action when certain events occur based on a transaction security policy. This interface is only used for Apex policies created in Real-Time Event Monitoring.

AsyncCondition Interface
Allows an implementing class to make asynchronous Apex calls. This interface is used only for transaction security Apex policies created in Real-Time Event Monitoring.

PolicyCondition Interface
Apex interface that allows an implementing class to specify actions to take when certain events occur based on a transaction security policy.

Event Class

Contains event information that the `evaluate` method uses to evaluate a transaction security policy.

Namespace

TxnSecurity

Usage

The Event class contains the information needed to determine if the event triggers a Transaction Security policy. Not all class attributes are used for every type of event.

Tip: The `EventClass` interface applies only to Legacy Transaction Security, which is a retired feature as of Summer '20. Use the `EventCondition` interface instead of the `EventClass` interface.

IN THIS SECTION:

Event Constructors

Event Properties

Event Constructors

The following is the constructor for `Event`.

IN THIS SECTION:

Event()
Creates an instance of the `TxnSecurity.Event` class.

`Event()`

Creates an instance of the `TxnSecurity.Event` class.

Signature

```
public Event()
```

Event Properties

The following are properties for `Event`.

IN THIS SECTION:

action
Specifies the action being taken on the resource for an Entity event. For example, a Login IP resource for an Entity event could have an `action` of `create`. The `action` attribute is not used by any other event type.

data
Contains data used by actions. For example, `data` for a login event includes the login history ID. Returns a map whose keys are the type of event data, like `SourceIp`.

entityId
The ID of any entity associated with the event. For example, the `entityId` of a DataExport event for an Account object contains the Account ID.

entityName
The name of the object the event acts on.

organizationId

The ID of the Salesforce org where the event occurred.

resourceType

The type of resource for the event. For example, an AccessResource event could have a Connected Application as a resource type. Not all event types have resources.

timeStamp

The time the event occurred.

userId

Identifies the user that caused the event.

action

Specifies the action being taken on the resource for an Entity event. For example, a Login IP resource for an Entity event could have an `action` of `create`. The `action` attribute is not used by any other event type.

Signature

```
public String action {get; set;}
```

Property Value

Type: String

data

Contains data used by actions. For example, `data` for a login event includes the login history ID. Returns a map whose keys are the type of event data, like `SourceIp`.

Signature

```
public Map<String,String> data {get; set;}
```

Property Value

Type: Map<String, String>

The following table lists all the available data types. Not all types appear with all event types. The data type values are always string representations. For example, the `isApi` value is a string in the map, but is actually a Boolean value. Convert the value from a string to its true type this way: `Boolean.valueOf(event.data.get('isApi'));`

Key Name	True Value Type	Events Supported
ActionName	`String` Values are: • Convert • Delete • Insert • Undelete • Update	Entity

Key Name	True Value Type	Events Supported
	• Upsert	
ApiType	`String` (Enum manifested as a String)	DataExport, Login
Application	`String`	AccessResource, DataExport
ClientId	`String` (ID of the client)	DataExport
ConnectedAppId	`String` (ID of the Connected App)	AccessResource, DataExport
ExecutionTime	milliseconds	DataExport
IsApi	`Boolean`	DataExport
IsScheduled	`Boolean`	DataExport
LoginHistoryId	`String`	DataExport, Login
NumberOfRecords	`Integer`	DataExport
PolicyId	`String` (ID of the current policy)	All events
SessionLevel	`String` (Enum manifested as a String. Values include `STANDARD` and `HIGH_ASSURANCE`)	AccessResource
SourceIp	`String` (IPV4 Address)	AccessResource
UserName	`String`	Entity

`entityId`

The ID of any entity associated with the event. For example, the `entityId` of a DataExport event for an Account object contains the Account ID.

Signature

```
public String entityId {get; set;}
```

Property Value

Type: String

`entityName`

The name of the object the event acts on.

Signature

```
public String entityName {get; set;}
```

Property Value

Type: String

organizationId

The ID of the Salesforce org where the event occurred.

Signature

```
public String organizationId {get; set;}
```

Property Value

Type: String

resourceType

The type of resource for the event. For example, an AccessResource event could have a Connected Application as a resource type. Not all event types have resources.

Signature

```
public String resourceType {get; set;}
```

Property Value

Type: String

timeStamp

The time the event occurred.

Signature

```
public Datetime timeStamp {get; set;}
```

Property Value

Type: Datetime

userId

Identifies the user that caused the event.

Signature

```
public String userId {get; set;}
```

Property Value

Type: String

EventCondition Interface

Allows an implementing class to specify whether to take action when certain events occur based on a transaction security policy. This interface is only used for Apex policies created in Real-Time Event Monitoring.

Usage

The `evaluate` method is called upon the occurrence of a real-time event monitored by a transaction security policy. A typical implementation first selects the fields of interest from the event. Then the fields are tested to see if they meet the conditions being monitored. If the conditions are met, the method returns `true`.

For example, imagine a transaction security policy that triggers when a user queries more than 1,000 lead records. For each API event, the `evaluate` method checks whether the `RowsProcessed` value is greater than 1,000 and the `QueriedEntities` value contains "Lead". If so, `true` is returned.

We recommend having test classes for the policy condition interface to ensure it works correctly. Testing is required regardless of whether the policy is moved from a sandbox to production, with a change set, or some other way. For example, test your policies in your development environment before moving the policies to production.

For more information about testing Apex transaction security policies, read Transaction Security Apex Testing.

IN THIS SECTION:

EventCondition Methods
The EventCondition interface has one method, evaluate(event).

EventCondition Example Implementation
Use EventCondition to create a custom condition for Shield Platform Encryption.

EventCondition Methods

The EventCondition interface has one method, evaluate(event).

The following are methods for `EventCondition`.

IN THIS SECTION:

evaluate(event)
Evaluates an event against a transaction security policy created in Real-Time Event Monitoring. If the event triggers the policy, the method returns `true`.

evaluate(event)

Evaluates an event against a transaction security policy created in Real-Time Event Monitoring. If the event triggers the policy, the method returns `true`.

Signature

```
public Boolean evaluate(SObject event)
```

Parameters

var1
Type: SObject

The event to check against the transaction security policy.

Return Value

Type: Boolean

Returns `true` when the policy is triggered. For example, suppose that the policy is to limit users to a single login session. If a user tries to log in a second time, the policy blocks the attempted login, and updates the Status, PolicyId, and PolicyOutcome fields of that LoginEvent. The policy also sends an email notification to the Salesforce admin. The `evaluate` method only checks the login event, and returns `true` if it's the user's second login attempt.

The system performs the action and notification, not the `evaluate` method.

EventCondition Example Implementation

Use EventCondition to create a custom condition for Shield Platform Encryption.

This example shows an implementation of the `TxnSecurity.EventCondition` interface. The transaction security policy triggers when the user queries an Account object.

```
public boolean evaluate(ApiEvent event) {
      switch on event {
         when ApiEvent apiEvent {
            return handleApiEvent(apiEvent);
         }
         when null {
         // Trigger action if event is null
            return true;
         }
         when else {
         // Trigger action for unhandled events
            return true;
         }
      }
   }

   private boolean handleApiEvent(ApiEvent apiEvent){
      if(apiEvent.QueriedEntities.contains('Account')){
         return true;
      }
      return false;
   }
}
```

For more examples, see Enhanced Apex Transaction Security Implementation Examples.

AsyncCondition Interface

Allows an implementing class to make asynchronous Apex calls. This interface is used only for transaction security Apex policies created in Real-Time Event Monitoring.

Namespace

TxnSecurity

Usage

If you make an Asynchronous Apex call in the class that implements your transaction security policy condition, the class must implement the `TxnSecurity.AsyncCondition` interface in addition to `TxnSecurity.EventCondition`. Use Asynchronous Apex instead of Apex callouts and DML statements, neither of which is allowed in transaction security Apex policies.

Apex offers multiple ways to run your Apex code asynchronously and all are supported in the `TxnSecurity.AsyncCondition` interface.

This interface has no methods.

IN THIS SECTION:

AsyncCondition Example Implementation

SEE ALSO:

*Apex Developer Guide:*Asynchronous Apex

AsyncCondition Example Implementation

Here's an example implementation of the `TxnSecurity.AsyncCondition` interface. The transaction security policy triggers when a user logs in. In the example, ExternalValidation__c is a custom object that contains information from an external validation system. The result of the SOQL query on ExternalValidation__c determines whether to block the user from logging in. The policy then queues the `CalloutToExternalValidator` class for asynchronous execution. When it executes, the `CalloutToExternalValidator` class makes an external call to the validation system to update it with information about this log in event. Because `CalloutToExternalValidator` is triggered by Asynchronous Apex, you must implement the `TxnSecurity.AsyncCondition` interface in the `ExternallyValidatedLoginCondition` Apex class along with the usual `TxnSecurity.EventCondition` interface.

```
global class ExternallyValidatedLoginCondition implements TxnSecurity.EventCondition,
TxnSecurity.AsyncCondition {
    public boolean evaluate(SObject event) {
        LoginEvent loginEvent = (LoginEvent) event;
        Boolean userBlocked = [select blocked from ExternalValidation__c where loginId =
loginEvent.UserId][0].Blocked;

        System.enqueueJob(new CalloutToExternalValidator(loginEvent.SourceIp,
loginEvent.LoginUrl));
        return userBlocked;
    }
}

public class CalloutToExternalValidator implements Queueable {
    private String sourceIp;
    private String loginUrl;

    public CalloutToExternalValidator(String sourceIp, String loginUrl) {
        this.sourceIp = sourceIp;
```

```
        this.loginUrl = loginUrl;
    }

    public void execute(QueueableContext context) {
        // callout to external validation service
        // pass sourceIp, loginUrl

        // update ExternalValidation__c
    }
}
```

PolicyCondition Interface

Apex interface that allows an implementing class to specify actions to take when certain events occur based on a transaction security policy.

Namespace

TxnSecurity

Usage

Tip: The `PolicyCondition` interface applies only to Legacy Transaction Security, which is a retired feature as of Summer '20. Use the `EventCondition` interface instead of the `PolicyCondition` interface.

The `evaluate` method is called upon the occurrence of an event monitored by a transaction security policy. A typical implementation first selects the item of interest from the event. Then the item is tested to see if it meets the condition being monitored. If the condition is met, the method returns `true`.

For example, imagine a transaction security policy that checks for the same user logging in more than once. For each login event, the method would check if the user logging in already has a login session in progress, and if so, `true` is returned.

We recommend having test classes for the policy condition interface to ensure it works correctly. Testing is required regardless of whether the policy is moved from a sandbox to production, with a change set, or some other way. For example, test your policies in your development environment before moving the policies to production.

Don't include DML statements in your custom policies because they can cause errors. When you send a custom email via Apex during transaction policy evaluation, you get an error, even if the record isn't explicitly related to another record. For more information, see Apex DML Operations in the *Apex Reference Guide*.

IN THIS SECTION:

PolicyCondition Methods

PolicyCondition Methods

The following is the method for `PolicyCondition`.

IN THIS SECTION:

evaluate(event)

Evaluates an event against a transaction security policy. If the event triggers the policy, `true` is returned.

`evaluate(event)`

Evaluates an event against a transaction security policy. If the event triggers the policy, `true` is returned.

Signature

```
public Boolean evaluate(TxnSecurity.Event event)
```

Parameters

`event`
Type: TxnSecurity.Event

The event to check against the transaction security policy.

Return Value

Type: Boolean

When the policy is triggered, `True` is returned. For example, let's suppose the policy is to limit users to a single login session. If anyone tries to log in a second time, the policy's action requires that they end their current session. The policy also sends an email notification to the Salesforce admin. The `evaluate()` method only checks the login event, and returns `True` if it's the user's second login. The Transaction Security system performs the action and notification, and not the `evaluate()` method.

UserProvisioning Namespace

The `UserProvisioning` namespace provides methods for monitoring outbound user provisioning requests.

The following is the class in the `UserProvisioning` namespace.

IN THIS SECTION:

ConnectorTestUtil Class
Enables developers to write Apex test classes for connectors used by the connected app provisioning solution. This class simulates provisioning for the associated app.

UserProvisioningLog Class
Provides methods for writing messages to monitor outbound user provisioning requests.

UserProvisioningPlugin Class
The `UserProvisioningPlugin` base class implements `Process.Plugin` for programmatic customization of the user provisioning process for connected apps.

ConnectorTestUtil Class

Enables developers to write Apex test classes for connectors used by the connected app provisioning solution. This class simulates provisioning for the associated app.

Namespace

UserProvisioning

Usage

Use this class for connector-based test accelerators. You can invoke it only from within an Apex test.

Example

This example creates an instance of a connected app, gets a value, and checks whether the value is correct. The test is simply a row inserted in the database table.

```
        @isTest
        private class SCIMCreateUserPluginTest {
        public static void callPlugin(Boolean validInputParams) {

        //Create an instance of a connected app
        ConnectedApplication capp
=UserProvisioning.ConnectorTestUtil.createConnectedApp('TestApp');
        Profile p = [SELECT Id FROM Profile WHERE Name='Standard User'];
        //Create a user
        User user = new User(username='testuser1@scimuserprov.test', Firstname= 'Test',
Lastname='User1', email='testuser1@testemail.com',
        FederationIdentifier='testuser1@testemail.com', profileId= p.Id,
communityNickName='tuser1', alias='tuser', TimeZoneSidKey='GMT',
        LocaleSidKey='en_US', EmailEncodingKey='ISO-8859-1', LanguageLocaleKey='en_US');
        //insert user into a row in the database table
        insert user;
        //Create a UPR
        UserProvisioningRequest upr = new UserProvisioningRequest(appname = capp.name,
connectedAppId=capp.id, operation='Create',
        state='New', approvalStatus='NotRequired',salesforceUserId=user.id);

        //Insert the UPR to test the flow end to end
        insert upr;
       }}
```

IN THIS SECTION:

ConnectorTestUtil Method

SEE ALSO:

Salesforce Help: User Provisioning for Connected Apps

ConnectorTestUtil Method

The `ConnectorTestUtil` class has 1 method.

IN THIS SECTION:

createConnectedApp(connectedAppName)

Creates an instance of a connected app to simulate provisioning.

`createConnectedApp(connectedAppName)`

Creates an instance of a connected app to simulate provisioning.

Signature

```
public static ConnectedApplication createConnectedApp(String connectedAppName)
```

Parameters

`connectedAppName`
Type: String

Name of the connected app to test for provisioning.

Return Value

Type: ConnectedApplication

The instance of the connected app to test for provisioning.

UserProvisioningLog Class

Provides methods for writing messages to monitor outbound user provisioning requests.

Namespace

UserProvisioning

Example

This example writes the user account information sent to a third-party system for a provisioning request to the UserProvisioningLog object.

```
String inputParamsStr = 'Input parameters: uprId=' + uprId + ',
endpointURL=' + endpointURL + ', adminUsername=' + adminUsername + ',
email=' + email + ', username=' + username + ', defaultPassword=' + defaultPassword + ',
defaultRoles =' + defaultRoles;
UserProvisioning.UserProvisioningLog.log(uprId, inputParamsStr);
```

IN THIS SECTION:

UserProvisioningLog Methods

UserProvisioningLog Methods

The following are methods for `UserProvisioningLog`. All methods are static.

IN THIS SECTION:

log(userProvisioningRequestId, details)
Writes a specific message, such as an error message, to monitor the progress of a user provisioning request.

log(userProvisioningRequestId, status, details)

Writes a specific status and message, such a status and detailed error message, to monitor the progress of a user provisioning request.

log(userProvisioningRequestId, externalUserId, externalUserName, userId, details)

Writes a specific message, such as an error message, to monitor the progress of a user provisioning request associated with a specific user.

log(userProvisioningRequestId, details)

Writes a specific message, such as an error message, to monitor the progress of a user provisioning request.

Signature

```
public void log(String userProvisioningRequestId, String details)
```

Parameters

userProvisioningRequestId
Type: String

A unique identifier for the user provisioning request.

details
Type: String

The text for the message.

Return Value

Type: void

log(userProvisioningRequestId, status, details)

Writes a specific status and message, such a status and detailed error message, to monitor the progress of a user provisioning request.

Signature

```
public void log(String userProvisioningRequestId, String status, String details)
```

Parameters

userProvisioningRequestId
Type: String

A unique identifier for the user provisioning request.

status
Type: String

A description of the current state. For example, while invoking a third-party API, the status could be `invoke`.

details
Type: String

The text for the message.

Return Value

Type: void

`log(userProvisioningRequestId, externalUserId, externalUserName, userId, details)`

Writes a specific message, such as an error message, to monitor the progress of a user provisioning request associated with a specific user.

Signature

```
public void log(String userProvisioningRequestId, String externalUserId, String externalUserName, String userId, String details)
```

Parameters

`userProvisioningRequestId`
Type: String

A unique identifier for the user provisioning request.

`externalUserId`
Type: String

The unique identifier for the user in the target system.

`externalUserName`
Type: String

The username for the user in the target system.

`userId`
Type: String

Salesforce ID of the user making the request.

`details`
Type: String

The text for the message.

Return Value

Type: void

UserProvisioningPlugin Class

The `UserProvisioningPlugin` base class implements `Process.Plugin` for programmatic customization of the user provisioning process for connected apps.

Namespace

UserProvisioning

Usage

Extending this class gives you a plug-in that can be used Flow Builder as a legacy Apex action, with the following input and output parameters.

Input Parameter Name	Description
`userProvisioningRequestId`	The unique ID of the request for the plug-in to process.
`userId`	The ID of the associated user for the request.
`NamedCredDevName`	The unique API name for the named credential to use for a request. The named credential identifies the third-party system and the third-party authentication settings. When the named credential is set in the User Provisioning Wizard, Salesforce stores the value in the `UserProvisioningConfig.NamedCredentialId` field.
`reconFilter`	When collecting and analyzing users on a third-party system, the plug-in uses this filter to limit the scope of the collection. When the filter is set in the User Provisioning Wizard, Salesforce stores the value in the `UserProvisioningConfig.ReconFilter` field.
`reconOffset`	When collecting and analyzing users on a third-party system, the plug-in uses this value as the starting point for the collection.

Output Parameter Name	Description
`Status`	The vendor-specific status of the provisioning operation on the third-party system.
`Details`	The vendor-specific message related to the status of the provisioning operation on the third-party system.
`ExternalUserId`	The vendor-specific ID for the associated user on the third-party system.
`ExternalUsername`	The vendor-specific username for the associated user on the third-party system.
`ExternalEmail`	The email address assigned to the user on the third-party system.
`ExternalFirstName`	The first name assigned to the user on the third-party system.
`ExternalLastName`	The last name assigned to the user on the third-party system.
`reconState`	The state of the collecting and analyzing process on the third-party system. When the value is `complete`, the process is finished and a subsequent call to the plug-in is no longer needed, nor made.

Output Parameter Name	Description
`nextReconOffset`	When collecting and analyzing users on a third-party system, the process may encounter a transaction limit and have to stop before finishing. The value specified here initiates a call to the plug-in with a new quota limit.

If you want to add more custom parameters, use the `buildDescribeCall()` method.

Example

The following example uses the `buildDescribeCall()` method to add a new input parameter and a new output parameter. The example also demonstrates how to bypass the limit of the 10,000 records processed in DML statements in an Apex transaction.

```
global class SampleConnector extends UserProvisioning.UserProvisioningPlugin {

    // Example of adding more input and output parameters to those defined in the base
class
    global override Process.PluginDescribeResult buildDescribeCall() {
        Process.PluginDescribeResult describeResult = new Process.PluginDescribeResult();

        describeResult.inputParameters = new
            List<Process.PluginDescribeResult.InputParameter>{
               new Process.PluginDescribeResult.InputParameter('testInputParam',
                       Process.PluginDescribeResult.ParameterType.STRING, false)
            };

        describeResult.outputParameters = new
            List<Process.PluginDescribeResult.OutputParameter>{
               new Process.PluginDescribeResult.OutputParameter('testOutputParam',
                       Process.PluginDescribeResult.ParameterType.STRING)
            };

        return describeResult;
    }

    // Example Plugin that demonstrates how to leverage the
reconOffset/nextReconOffset/reconState
    // parameters to create more than 10,000 users. (i.e. go beyond the 10,000 DML limit
per transaction)

    global override Process.PluginResult invoke(Process.PluginRequest request) {
        Map<String,String> result = new Map<String,String>();
        String uprId = (String) request.inputParameters.get('userProvisioningRequestId');

        UserProvisioning.UserProvisioningLog.log(uprId, 'Inserting Log from test Apex
connector');
        UserProvisioningRequest upr = [SELECT id, operation, connectedAppId, state
                  FROM userprovisioningrequest WHERE id = :uprId];
        if (upr.operation.equals('Reconcile')) {
            String reconOffsetStr = (String) request.inputParameters.get('reconOffset');
            Integer reconOffset = 0;
```

```
            if (reconOffsetStr != null) {
                reconOffset = Integer.valueOf(reconOffsetStr);
            }

            if (reconOffset > 44999) {
                result.put('reconState', 'Completed');
            }

            Integer i = 0;
            List<UserProvAccountStaging> upasList = new List<UserProvAccountStaging>();
            for (i = 0; i < 5000; i++) {
                UserProvAccountStaging upas = new UserProvAccountStaging();
                upas.Name = i + reconOffset + '';
                upas.ExternalFirstName = upas.Name;
                upas.ExternalEmail = 'externaluser@externalsystem.com';
                upas.LinkState = 'Orphaned';
                upas.Status = 'Active';
                upas.connectedAppId = upr.connectedAppId;
                upasList.add(upas);
            }
            insert upasList;
            result.put('nextReconOffset', reconOffset + 5000 + '');
        }

        return new Process.PluginResult(result);
    }
}
```

IN THIS SECTION:

UserProvisioningPlugin Methods

UserProvisioningPlugin Methods

The following are methods for `UserProvisioningPlugin`.

IN THIS SECTION:

buildDescribeCall()

Use this method to add more input and output parameters to those defined in the base class.

describe()

Returns a `Process.PluginDescribeResult` object that describes this method call.

getPluginClassName()

Returns the name of the class implementing the plugin.

invoke(request)

Primary method that the system invokes when the class that implements the interface is instantiated.

buildDescribeCall()

Use this method to add more input and output parameters to those defined in the base class.

Signature

```
public Process.PluginDescribeResult buildDescribeCall()
```

Return Value

Type: Process.PluginDescribeResult

describe()

Returns a `Process.PluginDescribeResult` object that describes this method call.

Signature

```
public Process.PluginDescribeResult describe()
```

Return Value

Type: Process.PluginDescribeResult

getPluginClassName()

Returns the name of the class implementing the plugin.

Signature

```
public String getPluginClassName()
```

Return Value

Type: String

invoke(request)

Primary method that the system invokes when the class that implements the interface is instantiated.

Signature

```
public Process.PluginResult invoke(Process.PluginRequest request)
```

Parameters

request
Type: Process.PluginRequest

Return Value

Type: Process.PluginDescribeResult

VisualEditor Namespace

The `VisualEditor` namespace provides classes and methods for interacting with the Lightning App Builder. The classes and methods in this namespace operate on Lightning components, which include Lightning web components and Aura components.

As of Spring '19 (API version 45.0), you can build Lightning components using two programming models: the Lightning Web Components model, and the original Aura Components model. Lightning web components are custom HTML elements built using HTML and modern JavaScript. Lightning web components and Aura components can coexist and interoperate on a page.

Configure Lightning web components and Aura components to work in Lightning App Builder and Experience Builder. Admins and end users don't know which programming model was used to develop the components. To them, they're simply Lightning components.

The following are the classes in the `VisualEditor` namespace.

IN THIS SECTION:

DataRow Class

Contains information about one item in a picklist used in a Lightning component on a Lightning page.

DesignTimePageContext Class

A class that provides context information about a Lightning page. It can be used to help define the values of a picklist in a Lightning component on a Lightning page based on the page's type and the object with which it's associated.

DynamicPickList Class

An abstract class, used to display the values of a picklist in a Lightning component on a Lightning page.

DynamicPickListRows Class

Contains a list of picklist items in a Lightning component on a Lightning page.

DataRow Class

Contains information about one item in a picklist used in a Lightning component on a Lightning page.

Namespace

VisualEditor

IN THIS SECTION:

DataRow Constructors

DataRow Methods

DataRow Constructors

The following are constructors for `DataRow`.

IN THIS SECTION:

DataRow(label, value, selected)
Creates an instance of the `VisualEditor.DataRow` class using the specified label, value, and selected option.

DataRow(label, value)
Creates an instance of the `VisualEditor.DataRow` class using the specified label and value.

DataRow(label, value, selected)

Creates an instance of the `VisualEditor.DataRow` class using the specified label, value, and selected option.

Signature

```
public DataRow(String label, Object value, Boolean selected)
```

Parameters

label
Type: String

User-facing label for the picklist item.

value
Type: Object

The value of the picklist item.

selected
Type: Boolean

Specifies whether the picklist item is selected (`true`) or not (`false`).

DataRow(label, value)

Creates an instance of the `VisualEditor.DataRow` class using the specified label and value.

Signature

```
public DataRow(String label, Object value)
```

Parameters

label
Type: String

User-facing label for the picklist item.

value
Type: Object

The value of the picklist item.

DataRow Methods

The following are methods for `DataRow`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `VisualEditor.DataRow` object.

compareTo(o)
Compares the current `VisualEditor.DataRow` object to the specified one. Returns an integer value that is the result of the comparison.

getLabel()
Returns the user-facing label of the picklist item.

getValue()
Returns the value of the picklist item.

isSelected()
Returns the state of the picklist item, indicating whether it's selected or not.

clone()

Makes a duplicate copy of the `VisualEditor.DataRow` object.

Signature

```
public Object clone()
```

Return Value

Type: Object

compareTo(o)

Compares the current `VisualEditor.DataRow` object to the specified one. Returns an integer value that is the result of the comparison.

Signature

```
public Integer compareTo(VisualEditor.DataRow o)
```

Parameters

o

Type: VisualEditor.DataRow

A single item in a picklist.

Return Value

Type: Integer

Returns one of the following values:

- Zero if the current package version is equal to the specified package version
- An integer value greater than zero if the current package version is greater than the specified package version

- An integer value less than zero if the current package version is less than the specified package version

`getLabel()`

Returns the user-facing label of the picklist item.

Signature

```
public String getLabel()
```

Return Value

Type: String

`getValue()`

Returns the value of the picklist item.

Signature

```
public Object getValue()
```

Return Value

Type: Object

`isSelected()`

Returns the state of the picklist item, indicating whether it's selected or not.

Signature

```
public Boolean isSelected()
```

Return Value

Type: Boolean

DesignTimePageContext Class

A class that provides context information about a Lightning page. It can be used to help define the values of a picklist in a Lightning component on a Lightning page based on the page's type and the object with which it's associated.

Namespace

VisualEditor

Usage

To use this class, create a parameterized constructor in the custom Apex class that extends `VisualEditor.DynamicPickList`.

Example

Here's an example of a custom Apex class extending the `VisualEditor.DynamicPickList` class. It includes `VisualEditor.DesignTimePageContext` to define a picklist value that is available only if the page type is `HomePage`.

```
global class MyCustomPickList extends VisualEditor.DynamicPickList{
    VisualEditor.DesignTimePageContext context;

    global MyCustomPickList(VisualEditor.DesignTimePageContext context) {
       this.context = context;
    }

    global override VisualEditor.DataRow getDefaultValue(){
        VisualEditor.DataRow defaultValue = new VisualEditor.DataRow('red', 'RED');
        return defaultValue;
    }
    global override VisualEditor.DynamicPickListRows getValues() {
        VisualEditor.DataRow value1 = new VisualEditor.DataRow('red', 'RED');
        VisualEditor.DataRow value2 = new VisualEditor.DataRow('yellow', 'YELLOW');
       VisualEditor.DynamicPickListRows  myValues = new VisualEditor.DynamicPickListRows();

        myValues.addRow(value1);
        myValues.addRow(value2);

        if (context.pageType == 'HomePage') {
            VisualEditor.DataRow value3 = new VisualEditor.DataRow('purple', 'PURPLE');
            myValues.addRow(value3);
        }

        return myValues;
    }
}
```

IN THIS SECTION:

DesignTimePageContext Properties

DesignTimePageContext Methods

DesignTimePageContext Properties

The following are properties for `DesignTimePageContext`.

IN THIS SECTION:

entityName

The API name of the sObject that a Lightning page is associated with, such as Account, Contact, or Custom_object__c. `entityName` is available only for object pages, and not all Lightning pages are associated with objects.

pageType

The type of Lightning page, such as `HomePage`, `AppPage`, or `RecordPage`.

`entityName`

The API name of the sObject that a Lightning page is associated with, such as Account, Contact, or Custom_object__c. `entityName` is available only for object pages, and not all Lightning pages are associated with objects.

Signature

```
public String entityName {get; set;}
```

Property Value

Type: String

`pageType`

The type of Lightning page, such as `HomePage`, `AppPage`, or `RecordPage`.

Signature

```
public String pageType {get; set;}
```

Property Value

Type: String

DesignTimePageContext Methods

The following are methods for `DesignTimePageContext`.

IN THIS SECTION:

clone()
Makes a duplicate copy of the `VisualEditor.DesignTimePageContext` object.

`clone()`

Makes a duplicate copy of the `VisualEditor.DesignTimePageContext` object.

Signature

```
public Object clone()
```

Return Value

Type: Object

DynamicPickList Class

An abstract class, used to display the values of a picklist in a Lightning component on a Lightning page.

Namespace

VisualEditor

Usage

To use this class as the datasource of a picklist in a Lightning component, it must be extended by a custom Apex class and then that class must be called in the component's design file.

Example

Here's an example of a custom Apex class extending the `VisualEditor.DynamicPickList` class.

```
global class MyCustomPickList extends VisualEditor.DynamicPickList{

    global override VisualEditor.DataRow getDefaultValue(){
        VisualEditor.DataRow defaultValue = new VisualEditor.DataRow('red', 'RED');
        return defaultValue;
    }
    global override VisualEditor.DynamicPickListRows getValues() {
        VisualEditor.DataRow value1 = new VisualEditor.DataRow('red', 'RED');
        VisualEditor.DataRow value2 = new VisualEditor.DataRow('yellow', 'YELLOW');
       VisualEditor.DynamicPickListRows  myValues = new VisualEditor.DynamicPickListRows();

        myValues.addRow(value1);
        myValues.addRow(value2);
        return myValues;
    }
}
```

Here's an example of how the custom Apex class gets called in a design file so that the picklist appears in the Lightning component.

```
<design:component>
        <design:attribute name="property1" datasource="apex://MyCustomPickList"/>
</design:component>
```

IN THIS SECTION:

DynamicPickList Methods

DynamicPickList Methods

The following are methods for `DynamicPickList`.

IN THIS SECTION:

clone()

Makes a duplicate copy of the `VisualEditor.DynamicPicklist` object.

getDefaultValue()

Returns the picklist item that is set as the default value for the picklist.

getLabel(attributeValue)
Returns the user-facing label for a specified picklist value.

getValues()
Returns the list of picklist item values.

isValid(attributeValue)
Returns the valid state of the picklist item's value. A picklist value is considered valid if it's a part of any `VisualEditor.DataRow` in the `VisualEditor.DynamicPickListRows` returned by `getValues()`.

clone()

Makes a duplicate copy of the `VisualEditor.DynamicPicklist` object.

Signature

```
public Object clone()
```

Return Value

Type: Object

getDefaultValue()

Returns the picklist item that is set as the default value for the picklist.

Signature

```
public VisualEditor.DataRow getDefaultValue()
```

Return Value

Type: VisualEditor.DataRow

getLabel(attributeValue)

Returns the user-facing label for a specified picklist value.

Signature

```
public String getLabel(Object attributeValue)
```

Parameters

attributeValue
Type: Object

The value of the picklist item.

Return Value

Type: String

`getValues()`

Returns the list of picklist item values.

Signature

```
public VisualEditor.DynamicPickListRows getValues()
```

Return Value

Type: VisualEditor.DynamicPickListRows

`isValid(attributeValue)`

Returns the valid state of the picklist item's value. A picklist value is considered valid if it's a part of any `VisualEditor.DataRow` in the `VisualEditor.DynamicPickListRows` returned by `getValues()`.

Signature

```
public Boolean isValid(Object attributeValue)
```

Parameters

`attributeValue`
Type: Object

The value of the picklist item.

Return Value

Type: Boolean

DynamicPickListRows Class

Contains a list of picklist items in a Lightning component on a Lightning page.

Namespace

VisualEditor

IN THIS SECTION:

DynamicPickListRows Constructors

DynamicPickListRows Methods

DynamicPickListRows Constructors

The following are constructors for `DynamicPickListRows`.

IN THIS SECTION:

DynamicPickListRows(rows, containsAllRows)
Creates an instance of the `VisualEditor.DynamicPickListRows` class using the specified parameters.

DynamicPickListRows(rows)
Creates an instance of the `VisualEditor.DynamicPickListRows` class using the specified parameter.

DynamicPickListRows()
Creates an instance of the `VisualEditor.DynamicPickListRows` class. You can then add rows by using the class's `addRow` or `addAllRows` methods.

DynamicPickListRows(rows, containsAllRows)

Creates an instance of the `VisualEditor.DynamicPickListRows` class using the specified parameters.

Signature

```
public DynamicPickListRows(List<VisualEditor.DataRow> rows, Boolean containsAllRows)
```

Parameters

rows
Type: List VisualEditor.DataRow

List of picklist items.

containsAllRows
Type: Boolean

Indicates if all values of the picklist are included in a type-ahead search query (true) or only those values initially displayed when the list is clicked on (false).

A picklist in a Lightning component can display only the first 200 values of a list. If *containsAllRows* is set to false, when a user does a type-ahead search to find values in the picklist, the search will only look at those first 200 values that were displayed, not the complete set of picklist values.

DynamicPickListRows(rows)

Creates an instance of the `VisualEditor.DynamicPickListRows` class using the specified parameter.

Signature

```
public DynamicPickListRows(List<VisualEditor.DataRow> rows)
```

Parameters

rows
Type: List VisualEditor.DataRow

List of picklist rows.

`DynamicPickListRows()`

Creates an instance of the `VisualEditor.DynamicPickListRows` class. You can then add rows by using the class's `addRow` or `addAllRows` methods.

Signature

```
public DynamicPickListRows()
```

DynamicPickListRows Methods

The following are methods for `DynamicPickListRows`.

IN THIS SECTION:

addAllRows(rows)
Adds a list of picklist items to a dynamic picklist rendered in a Lightning component on a Lightning page.

addRow(row)
Adds a single picklist item to a dynamic picklist rendered in a Lightning component on a Lightning page.

clone()
Makes a duplicate copy of the `VisualEditor.DynamicPickListRows` object.

containsAllRows()
Returns a Boolean value indicating whether all values of the picklist are included when a user does a type-ahead search query (true) or only those values initially displayed when the list is clicked on (false).

get(i)
Returns a picklist element stored at the specified index.

getDataRows()
Returns a list of picklist items.

setContainsAllRows(containsAllRows)
Sets the value indicating whether all values of the picklist are included when a user does a type-ahead search query (true) or only those values initially displayed when the list is clicked on (false).

size()
Returns the size of the list of `VisualEditor.DynamicPickListRows`.

sort()
Sorts the list of `VisualEditor.DynamicPickListRows`.

`addAllRows(rows)`

Adds a list of picklist items to a dynamic picklist rendered in a Lightning component on a Lightning page.

Signature

```
public void addAllRows(List<VisualEditor.DataRow> rows)
```

Parameters

rows
Type: List VisualEditor.DataRow

List of picklist items.

Return Value

Type: void

addRow(row)

Adds a single picklist item to a dynamic picklist rendered in a Lightning component on a Lightning page.

Signature

```
public void addRow(VisualEditor.DataRow row)
```

Parameters

row
Type: VisualEditor.DataRow

A single picklist item.

Return Value

Type: void

clone()

Makes a duplicate copy of the `VisualEditor.DynamicPickListRows` object.

Signature

```
public Object clone()
```

Return Value

Type: Object

containsAllRows()

Returns a Boolean value indicating whether all values of the picklist are included when a user does a type-ahead search query (true) or only those values initially displayed when the list is clicked on (false).

Signature

```
public Boolean containsAllRows()
```

Return Value

Type: Boolean

A picklist in a Lightning component can display only the first 200 values of a list. If *containsAllRows* is set to false, when a user does a type-ahead search to find values in the picklist, the search will only look at those first 200 values that were displayed, not the complete set of picklist values.

get(i)

Returns a picklist element stored at the specified index.

Signature

```
public VisualEditor.DataRow get(Integer i)
```

Parameters

i

Type: Integer

The index.

Return Value

Type: VisualEditor.DataRow

getDataRows()

Returns a list of picklist items.

Signature

```
public List<VisualEditor.DataRow> getDataRows()
```

Return Value

Type: List VisualEditor.DataRow

setContainsAllRows(containsAllRows)

Sets the value indicating whether all values of the picklist are included when a user does a type-ahead search query (true) or only those values initially displayed when the list is clicked on (false).

Signature

```
public void setContainsAllRows(Boolean containsAllRows)
```

Parameters

containsAllRows

Type: Boolean

Indicates if all values of the picklist are included in a type-ahead search query (true) or only those values initially displayed when the list is clicked on (false).

A picklist in a Lightning component can display only the first 200 values of a list. If `containsAllRows` is set to false, when a user does a type-ahead search to find values in the picklist, the search will only look at those first 200 values that were displayed, not the complete set of picklist values.

Return Value

Type: void

`size()`

Returns the size of the list of `VisualEditor.DynamicPickListRows`.

Signature

```
public Integer size()
```

Return Value

Type: Integer

`sort()`

Sorts the list of `VisualEditor.DynamicPickListRows`.

Signature

```
public void sort()
```

Return Value

Type: void

`Wave` Namespace

The classes in the `Wave` namespace are part of the CRM Analytics Analytics SDK, designed to facilitate querying CRM Analytics data from Apex code.

The following are the classes in the `Wave` namespace.

IN THIS SECTION:

QueryBuilder Class
The QueryBuilder class provides methods for constructing well-formed SAQL queries to pass to CRM Analytics.

QueryNode Class
Define each node of the query - such as projection, groups, order, filters. Execute the query.

ProjectionNode Class
Add aggregate functions to the query, or define an alias.

Templates Class

The Templates class provides methods for retrieving CRM Analytics template collections, individual templates, and template configurations.

TemplatesSearchOptions Class

The TemplatesSearchOptions class provides optional properties to filter the template collection.

QueryBuilder Class

The QueryBuilder class provides methods for constructing well-formed SAQL queries to pass to CRM Analytics.

Namespace

wave

Usage

Use QueryBuilder and its associated classes, `Wave.ProjectionNode` and `Wave.QueryNode`, to incrementally build your SAQL statement. For example:

```
public static void executeApexQuery(String name){
  Wave.ProjectionNode[] projs = new Wave.ProjectionNode[]{
    Wave.QueryBuilder.get('State').alias('State'),
    Wave.QueryBuilder.get('City').alias('City'),
    Wave.QueryBuilder.get('Revenue').avg().alias('avg_Revenue'),
    Wave.QueryBuilder.get('Revenue').sum().alias('sum_Revenue'),

    Wave.QueryBuilder.count().alias('count')};

  ConnectApi.LiteralJson result = Wave.QueryBuilder.load('0FbD00000004DSzKAM',
'0FcD00000004FEZKA2')
    .group(new String[]{'State', 'City'})
    .foreach(projs)
    .execute('q');
  String response = result.json;
}
```

Examples

QueryBuilder is the core of this first phase of the CRM Analytics Apex SDK, so let's take a closer look. Here's a simple count query.

```
Wave.ProjectionNode[] projs = new
Wave.ProjectionNode[]{Wave.QueryBuilder.count().alias('c')};
String query = Wave.QueryBuilder.load('datasetId',
'datasetVersionId').group().foreach(projs).build('q');
```

The resulting SAQL query looks like this:

```
q = load "datasetId/datasetVersionId";
q = group q by all;
q = foreach q generate count as c;
```

Here's a more complex example that uses a union statement.

```
Wave.ProjectionNode[] projs = new Wave.ProjectionNode[]{Wave.QueryBuilder.get('Name'),
Wave.QueryBuilder.get('AnnualRevenue').alias('Revenue')};
Wave.QueryNode nodeOne =
Wave.QueryBuilder.load('datasetOne','datasetVersionOne').foreach(projs);
Wave.QueryNode nodeTwo = Wave.QueryBuilder.load('datasetTwo',
'datasetVersionTwo').foreach(projs);
String query = Wave.QueryBuilder.union(new List<Wave.QueryNode>{nodeOne,
nodeTwo}).build('q');
```

The resulting SAQL query has two projection streams, *qa* and *qb*.

```
qa = load "datasetOne/datasetVersionOne";
qa = foreach q generate Name,AnnualRevenue as Revenue;
qb = load "datasetTwo/datasetVersionTwo";
qb = foreach q generate Name,AnnualRevenue as Revenue;
q = union qa, qb;
```

IN THIS SECTION:

QueryBuilder Methods

QueryBuilder Methods

The following are methods for `QueryBuilder`.

IN THIS SECTION:

load(datasetID, datasetVersionID)
Load a stream from a dataset.

count()
Calculate the number of rows that match the query criteria.

get(projection)
Query by selecting specific attributes.

union(unionNodes)
Combine multiple result sets into one result set.

cogroup(cogroupNodes, groups)
Cogrouping means that two input streams are grouped independently and arranged side by side. Only data that exists in both groups appears in the results.

`load(datasetID, datasetVersionID)`

Load a stream from a dataset.

Signature

```
public static wave.QueryNode load(String datasetID, String datasetVersionID)
```

Parameters

datasetID
Type: String

The ID of the dataset.

datasetVersionID
Type: String

The ID identifying the version of the dataset.

Return Value

Type: wave.QueryNode

count()

Calculate the number of rows that match the query criteria.

Signature

```
public static wave.ProjectionNode count()
```

Return Value

Type: wave.ProjectionNode

get(projection)

Query by selecting specific attributes.

Signature

```
public static wave.ProjectionNode get(String proj)
```

Parameters

proj
Type: String

The name of the column to query.

Return Value

Type: wave.ProjectionNode

union(unionNodes)

Combine multiple result sets into one result set.

Signature

```
global static Wave.QueryNode union(List<Wave.QueryNode> unionNodes)
```

Parameters

unionNodes
Type: List<wave.QueryNode>

List of nodes to combine.

Return Value

Type: wave.QueryNode

cogroup(cogroupNodes, groups)

Cogrouping means that two input streams are grouped independently and arranged side by side. Only data that exists in both groups appears in the results.

Signature

```
global static Wave.QueryNode cogroup(List<Wave.QueryNode> cogroupNodes,
List<List<String>> groups)
```

Parameters

cogroupNodes
Type: wave.QueryNode

List of nodes to group.

groups
Type: String

The type of grouping.

Return Value

Type: wave.QueryNode

QueryNode Class

Define each node of the query - such as projection, groups, order, filters. Execute the query.

Namespace

wave

Usage

Refer to the QueryBuilder example.

IN THIS SECTION:

QueryNode Methods

QueryNode Methods

The following are methods for `QueryNode`.

IN THIS SECTION:

build(streamName)
Build the query string represented by this QueryNode and assign it to a stream name.

foreach(projections)
Applies a set of expressions to every row in a dataset. This action is often referred to as projection.

group(groups)
Groups matched records (group by specific dataset attributes).

group()
Groups matched records (group by all).

order(orders)
Sorts in ascending or descending order on one or more fields.

cap(cap)
Limits the number of results that are returned.

filter(filterCondition)
Selects rows from a dataset based on a filter condition (a predicate).

filter(filterConditions)
Selects rows from a dataset based on multiple filter conditions (predicates).

execute(streamName)
Execute the query and return rows as JSON.

`build(streamName)`

Build the query string represented by this QueryNode and assign it to a stream name.

Signature

```
public String build(String streamName)
```

Parameters

`streamName`
Type: String

The identifier for the stream - for example, "q".

Return Value

Type: String

The SAQL query string represented by the QueryNode.

foreach(projections)

Applies a set of expressions to every row in a dataset. This action is often referred to as projection.

Signature

```
public wave.QueryNode foreach(List<wave.ProjectionNode> projections)
```

Parameters

projections
Type: List<wave.ProjectionNode>

A list of ProjectionNodes to be added to this QueryNode.

Return Value

Type: wave.QueryNode

group(groups)

Groups matched records (group by specific dataset attributes).

Signature

```
public wave.QueryNode group(List<String> groups)
```

Parameters

groups
Type: List<String>

A list of expressions.

Return Value

Type: wave.QueryNode

Example

```
Wave.ProjectionNode[] projs = new Wave.ProjectionNode[]{Wave.QueryBuilder.get('Name'),
Wave.QueryBuilder.get('Revenue').sum().alias('REVENUE_SUM')};
ConnectApi.LiteralJson result = Wave.QueryBuilder.load('datasetId',
'datasetVersionId').group(new String[]{'Name'}).foreach(projs).build('q');
```

group()

Groups matched records (group by all).

Signature

```
public wave.QueryNode group()
```

Return Value

Type: wave.QueryNode

Example

```
String query = Wave.QueryBuilder.load('datasetId',
'datasetVersionId').group().foreach(projs).build('q');
```

order(orders)

Sorts in ascending or descending order on one or more fields.

Signature

```
public wave.QueryNode group(List<String> groups)
```

Parameters

groups
Type: List<String>

A list of column names and associated ascending or descending keywords, for example

```
List<List<String>>{new List<String>{'Name', 'asc'}, new List<String>{'Revenue', 'desc'}}
```

Return Value

Type: wave.QueryNode

cap(cap)

Limits the number of results that are returned.

Signature

```
global Wave.QueryNode cap(Integer cap)
```

Parameters

cap
Type: Integer

The maximum number of rows to return.

Return Value

Type: wave.QueryNode

filter(filterCondition)

Selects rows from a dataset based on a filter condition (a predicate).

Signature

```
public wave.QueryNode filter(String filterCondition)
```

Parameters

filterCondition
Type: String

For example: `filter('Name != \'My Name\'')`

Return Value

Type: wave.QueryNode

filter(filterConditions)

Selects rows from a dataset based on multiple filter conditions (predicates).

Signature

```
public wave.QueryNode filter(List<String> filterCondition)
```

Parameters

filterCondition
Type: List<String>

A list of filter conditions.

Return Value

Type: wave.QueryNode

execute(streamName)

Execute the query and return rows as JSON.

Signature

```
global ConnectApi.LiteralJson execute(String streamName)
```

Parameters

streamName
Type: String

The query stream to execute. For example:

```
ConnectApi.LiteralJson result = Wave.QueryBuilder.load('datasetId',
      'datasetVersionId').group().foreach(projs).execute('q');
```

Return Value

Type: ConnectApi.LiteralJson

ProjectionNode Class

Add aggregate functions to the query, or define an alias.

Namespace

wave on page 4083

Usage

Refer to the QueryBuilder example.

IN THIS SECTION:

ProjectionNode Methods

ProjectionNode Methods

The following are methods for `ProjectionNode`.

IN THIS SECTION:

sum()
Returns the sum of a numeric field.

avg()
Returns the average value of a numeric field.

min()
Returns the minimum value of a field.

max()
Returns the maximum value of a field.

count()
Returns the number of rows that match the query criteria.

unique()
Returns the count of unique values.

alias(name)
Define output column names.

`sum()`

Returns the sum of a numeric field.

Signature

```
public wave.ProjectionNode sum()
```

Return Value

Type: wave.ProjectionNode

`avg()`

Returns the average value of a numeric field.

Signature

```
public wave.ProjectionNode avg()
```

Return Value

Type: wave.ProjectionNode

`min()`

Returns the minimum value of a field.

Signature

```
public wave.ProjectionNode min()
```

Return Value

Type: wave.ProjectionNode

`max()`

Returns the maximum value of a field.

Signature

```
public wave.ProjectionNode max()
```

Return Value

Type: wave.ProjectionNode

`count()`

Returns the number of rows that match the query criteria.

Signature

```
public wave.ProjectionNode count()
```

Return Value

Type: wave.ProjectionNode

`unique()`

Returns the count of unique values.

Signature

```
public wave.ProjectionNode unique()
```

Return Value

Type: wave.ProjectionNode

`alias(name)`

Define output column names.

Signature

```
public wave.ProjectionNode alias(String name)
```

Parameters

name
Type: String

The name to use for this column. For example, this code defines the alias `c`:

```
Wave.ProjectionNode[] projs = new
Wave.ProjectionNode[]{Wave.QueryBuilder.count().alias('c')};
```

Return Value

Type: wave.ProjectionNode

Templates Class

The Templates class provides methods for retrieving CRM Analytics template collections, individual templates, and template configurations.

Namespace

Wave

Usage

Use Templates and its associated class `Wave.TemplatesSearchOptions` to get CRM Analytics template information.

Examples

This code sample declares a method that returns a list of the template names.

```
@AuraEnabled(cacheable=true)
public static void List<String> getTemplateNames() {
  Map<String, Object> o = Wave.Templates.getTemplates(new Wave.TemplatesSearchOptions());

  List<Object> templates = (List<Object>) o.get('templates');
  List<String> names = new List<String>();
  for (Object templateObj : templates) {
    names.add((String) ((Map<String, Object>) templateObj.get('name'));
  }
  return names;
}
```

Adding the `@AuraEnabled` annotation allows Lightning Web Components to access Templates methods directly.

For example, in the lwc.js file:

```
import getTemplates from '@salesforce/apex/Wave.Templates.getTemplates';
export default class Templates extends LightningElement {
  @wire(getTemplates, {
    // specifying 'options' is optional
    options: {
      // values in TemplatesSearchOptions go here; all optional
      type: 'app'
    }
  })
  onTemplates({ data, error }) {
    if (data) {
      console.log('template names=' + data.templates.map(l => l.name).join(', '));
    }
  }
}
```

IN THIS SECTION:

Templates Methods

Templates Methods

The following are methods for `Templates`.

IN THIS SECTION:

getTemplate(templateIdOrApiName)

Gets a CRM Analytics template by the specified ID or API name. The returned template is a map of the template JSON attributes as name/value pairs.

getTemplateConfig(templateIdOrApiName)

Gets the CRM Analytics template configuration by the specified ID or API name. The returned template configuration is a map of the JSON attributes as name/value pairs.

getTemplates(options)

Get a filtered collection of CRM Analytics templates using search options.

getTemplates()

Gets all CRM Analytics templates.

getTemplate(templateIdOrApiName)

Gets a CRM Analytics template by the specified ID or API name. The returned template is a map of the template JSON attributes as name/value pairs.

Signature

```
public static Map<String,Object> getTemplate(String templateIdOrApiName)
```

Parameters

templateIdOrApiName

Type: String

The template ID or API name of the template to retrieve.

Return Value

Type: Map<String,Object>

A map of the template JSON attribute name/value pairs, where the name is a string with an object value. For attributes details, see TemplateRepresentation.

Example

```
String templateName = (String) Wave.Templates.getTemplate(templateId).get('name');
```

getTemplateConfig(templateIdOrApiName)

Gets the CRM Analytics template configuration by the specified ID or API name. The returned template configuration is a map of the JSON attributes as name/value pairs.

Signature

```
public static Map<String,Object> getTemplateConfig(String templateIdOrApiName)
```

Parameters

templateIdOrApiName

Type: String

The template ID or developer name to retrieve the template configuration for.

Return Value

Type: Map<String,Object>

A map of template configuration JSON attribute names and the object values. For attribute details, see TemplateConfigurationRepresentation.

Example

```
Map<String, Object> templateVariables = (Map<String, Object>)
Wave.Templates.getTemplateConfig(templateId).get('variables');
```

getTemplates(options)

Get a filtered collection of CRM Analytics templates using search options.

Signature

```
public static Map<String,Object> getTemplates(Wave.TemplatesSearchOptions options)
```

Parameters

options

Type: Wave.TemplatesSearchOptions on page 4098

The search options to use for filtering the template collection.

Return Value

Type: Map<String,Object>

A map of template names and the template object values. For template collection details, see TemplateCollectionRepresentation.

Example

```
Map<String,Object> templatesMap = Wave.Templates.getTemplates(new
Wave.TemplatesSearchOptions());
```

getTemplates()

Gets all CRM Analytics templates.

Signature

```
public static Map<String,Object> getTemplates()
```

Return Value

Type: Map<String,Object>

A map of template names and the template object values. For template collection details, see TemplateCollectionRepresentation.

Example

```
Map<String,Object> templatesMap = Wave.Templates.getTemplates();
```

TemplatesSearchOptions Class

The TemplatesSearchOptions class provides optional properties to filter the template collection.

Namespace

Wave

Usage

Use TemplatesSearchOptions with `Wave.Templates` class to filter the CRM Analytics template collection returned. For example:

```
public static void List<String> getAppTemplates() {
  Wave.TemplateSearchOptions tsOptions = new Wave.TemplatesSearchOptions();
  tsOptions.type = 'app';

  Map<String, Object> o = Wave.Templates.getTemplates(tsOptions);
  List<Object> appTemplates = (List<Object>) o.get('templates');
  List<String> names = new List<String>();
  for (Object templateObj : appTemplates) {
    names.add((String) ((Map<String, Object>) templateObj.get('name'));
  }
  return names;
}
```

IN THIS SECTION:

TemplatesSearchOptions Properties

TemplatesSearchOptions Properties

The following are properties for `TemplatesSearchOptions`.

IN THIS SECTION:

filterGroup

Specifies the Connect API filter group for CRM Analytics template search options.

options

Specifies the template visibility option to filter the CRM Analytics template collection by.

type

Sets the template type to filter the CRM Analytics template collection by.

`filterGroup`

Specifies the Connect API filter group for CRM Analytics template search options.

Signature

```
public String filterGroup {get; set;}
```

Property Value

Type: String

Uses the ConnectFilterGroupEnum values.

Example

```
Wave.TemplateSearchOptions tsOptions = new Wave.TemplatesSearchOptions();
tsOptions.filterGroup = 'small';
```

options

Specifies the template visibility option to filter the CRM Analytics template collection by.

Signature

```
public String options {get; set;}
```

Property Value

Type: String

Uses the `ConnectWaveTemplateVisibilityOptionsEnum` values. Valid values are `CreateApp`, `ViewOnly`, and `ManageableOnly`.

Example

```
Wave.TemplateSearchOptions tsOptions = new Wave.TemplatesSearchOptions();
tsOptions.options = 'ViewOnly';
```

type

Sets the template type to filter the CRM Analytics template collection by.

Signature

```
public String type {get; set;}
```

Property Value

Type: String

Uses the `ConnectWaveTemplateTypeEnum` values. Valid values are `app`, `dashboard`, `embedded`, and `lens`.

Example

```
Wave.TemplateSearchOptions tsOptions = new Wave.TemplatesSearchOptions();
tsOptions.type = 'app';
```

Appendices

IN THIS SECTION:

Shipping Invoice Example

Reserved Keywords

These words can be used only as keywords.

Documentation Typographical Conventions

Apex and Visualforce documentation uses these typographical conventions.

Shipping Invoice Example

This appendix provides an example of an Apex application. This is a more complex example than the Hello World example.

- Shipping Invoice Walk-Through
- Shipping Invoice Example Code

IN THIS SECTION:

1. Shipping Invoice Example Walk-Through
2. Shipping Invoice Example Code

Shipping Invoice Example Walk-Through

The sample application in this section includes traditional Salesforce functionality blended with Apex. Many of the syntactic and semantic features of Apex, along with common idioms, are illustrated in this application.

Note: The Shipping Invoice sample requires custom objects. You can either create these on your own, or download the objects and Apex code as an unmanaged package from the Salesforce AppExchange. To obtain the sample assets in your org, install the Apex Tutorials Package. This package also contains sample code and objects for the Apex Quick Start.

Scenario

In this sample application, the user creates a new shipping invoice, or order, and then adds items to the invoice. The total amount for the order, including shipping cost, is automatically calculated and updated based on the items added or deleted from the invoice.

Data and Code Models

This sample application uses two new objects: Item and Shipping_invoice.

The following assumptions are made:

- Item A cannot be in both orders shipping_invoice1 and shipping_invoice2. Two customers cannot obtain the same (physical) product.
- The tax rate is 9.25%.
- The shipping rate is 75 cents per pound.
- Once an order is over $100, the shipping discount is applied (shipping becomes free).

The fields in the Item custom object include:

Name	Type	Description
Name	String	The name of the item
Price	Currency	The price of the item
Quantity	Number	The number of items in the order
Weight	Number	The weight of the item, used to calculate shipping costs
Shipping_invoice	Master-Detail (shipping_invoice)	The order this item is associated with

The fields in the Shipping_invoice custom object include:

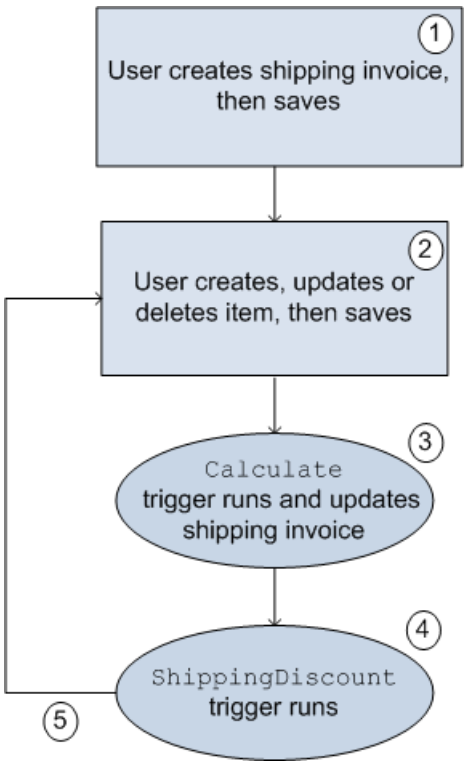
Name	Type	Description
Name	String	The name of the shipping invoice/order
Subtotal	Currency	The subtotal
GrandTotal	Currency	The total amount, including tax and shipping
Shipping	Currency	The amount charged for shipping (assumes $0.75 per pound)
ShippingDiscount	Currency	Only applied once when subtotal amount reaches $100
Tax	Currency	The amount of tax (assumes 9.25%)
TotalWeight	Number	The total weight of all items

All of the Apex for this application is contained in triggers. This application has the following triggers:

Object	Trigger Name	When Runs	Description
Item	Calculate	after insert, after update, after delete	Updates the shipping invoice, calculates the totals and shipping
Shipping_invoice	ShippingDiscount	after update	Updates the shipping invoice, calculating if there is a shipping discount

The following is the general flow of user actions and when triggers run:

Flow of user action and triggers for the shopping cart application



1. User clicks **Orders** > **New**, names the shipping invoice and clicks **Save**.
2. User clicks **New Item**, fills out information, and clicks **Save**.
3. Calculate trigger runs. Part of the Calculate trigger updates the shipping invoice.
4. ShippingDiscount trigger runs.
5. User can then add, delete or change items in the invoice.

In Shipping Invoice Example Code both of the triggers and the test class are listed. The comments in the code explain the functionality.

Testing the Shipping Invoice Application

Before an application can be included as part of a package, 75% of the code must be covered by unit tests. Therefore, one piece of the shipping invoice application is a class used for testing the triggers.

The test class verifies the following actions are completed successfully:

- Inserting items
- Updating items
- Deleting items
- Applying shipping discount
- Negative test for bad input

Shipping Invoice Example Code

The following triggers and test class make up the shipping invoice example application:

- Calculate trigger
- ShippingDiscount trigger
- Test class

Calculate Trigger

```
trigger calculate on Item__c (after insert, after update, after delete) {

// Use a map because it doesn't allow duplicate values

Map<ID, Shipping_Invoice__C> updateMap = new Map<ID, Shipping_Invoice__C>();

// Set this integer to -1 if we are deleting
Integer subtract ;

// Populate the list of items based on trigger type
List<Item__c> itemList;
    if(trigger.isInsert || trigger.isUpdate){
        itemList = Trigger.new;
        subtract = 1;
    }
    else if(trigger.isDelete)
    {
        // Note -- there is no trigger.new in delete
        itemList = trigger.old;
        subtract = -1;
    }

// Access all the information we need in a single query
// rather than querying when we need it.
// This is a best practice for bulkifying requests

set<Id> AllItems = new set<id>();

for(item__c i :itemList){
// Assert numbers are not negative.
// None of the fields would make sense with a negative value

System.assert(i.quantity__c > 0, 'Quantity must be positive');
System.assert(i.weight__c >= 0, 'Weight must be non-negative');
System.assert(i.price__c >= 0, 'Price must be non-negative');

// If there is a duplicate Id, it won't get added to a set
AllItems.add(i.Shipping_Invoice__C);
}

// Accessing all shipping invoices associated with the items in the trigger
List<Shipping_Invoice__C> AllShippingInvoices = [SELECT Id, ShippingDiscount__c,
                    SubTotal__c, TotalWeight__c, Tax__c, GrandTotal__c
                    FROM Shipping_Invoice__C WHERE Id IN :AllItems];

// Take the list we just populated and put it into a Map.
// This will make it easier to look up a shipping invoice
```

```
// because you must iterate a list, but you can use lookup for a map,
Map<ID, Shipping_Invoice__C> SIMap = new Map<ID, Shipping_Invoice__C>();

for(Shipping_Invoice__C sc : AllShippingInvoices)
{
    SIMap.put(sc.id, sc);
}

// Process the list of items
    if(Trigger.isUpdate)
    {
        // Treat updates like a removal of the old item and addition of the
        // revised item rather than figuring out the differences of each field
        // and acting accordingly.
        // Note updates have both trigger.new and trigger.old
        for(Integer x = 0; x < Trigger.old.size(); x++)
        {
            Shipping_Invoice__C myOrder;
            myOrder = SIMap.get(trigger.old[x].Shipping_Invoice__C);

            // Decrement the previous value from the subtotal and weight.
            myOrder.SubTotal__c -= (trigger.old[x].price__c *
                                    trigger.old[x].quantity__c);
            myOrder.TotalWeight__c -= (trigger.old[x].weight__c *
                                       trigger.old[x].quantity__c);

            // Increment the new subtotal and weight.
            myOrder.SubTotal__c += (trigger.new[x].price__c *
                                    trigger.new[x].quantity__c);
            myOrder.TotalWeight__c += (trigger.new[x].weight__c *
                                       trigger.new[x].quantity__c);
        }

        for(Shipping_Invoice__C myOrder : AllShippingInvoices)
        {

            // Set tax rate to 9.25%  Please note, this is a simple example.
            // Generally, you would never hard code values.
            // Leveraging Custom Settings for tax rates is a best practice.
            // See Custom Settings in the Apex Developer Guide
            // for more information.
            myOrder.Tax__c = myOrder.Subtotal__c * .0925;

            // Reset the shipping discount
            myOrder.ShippingDiscount__c = 0;

            // Set shipping rate to 75 cents per pound.
            // Generally, you would never hard code values.
            // Leveraging Custom Settings for the shipping rate is a best practice.
            // See Custom Settings in the Apex Developer Guide
            // for more information.
            myOrder.Shipping__c = (myOrder.totalWeight__c * .75);
            myOrder.GrandTotal__c = myOrder.SubTotal__c + myOrder.tax__c +
                                    myOrder.Shipping__c;
```

```
            updateMap.put(myOrder.id, myOrder);
         }
    }
    else
    {
        for(Item__c itemToProcess : itemList)
        {
            Shipping_Invoice__C myOrder;

            // Look up the correct shipping invoice from the ones we got earlier
            myOrder = SIMap.get(itemToProcess.Shipping_Invoice__C);
            myOrder.SubTotal__c += (itemToProcess.price__c *
                                    itemToProcess.quantity__c * subtract);
            myOrder.TotalWeight__c += (itemToProcess.weight__c *
                                       itemToProcess.quantity__c * subtract);
        }

        for(Shipping_Invoice__C myOrder : AllShippingInvoices)
        {

             // Set tax rate to 9.25%  Please note, this is a simple example.
             // Generally, you would never hard code values.
             // Leveraging Custom Settings for tax rates is a best practice.
             // See Custom Settings in the Apex Developer Guide
             // for more information.
             myOrder.Tax__c = myOrder.Subtotal__c * .0925;

             // Reset shipping discount
             myOrder.ShippingDiscount__c = 0;

            // Set shipping rate to 75 cents per pound.
            // Generally, you would never hard code values.
            // Leveraging Custom Settings for the shipping rate is a best practice.
            // See Custom Settings in the Apex Developer Guide
            // for more information.
            myOrder.Shipping__c = (myOrder.totalWeight__c * .75);
            myOrder.GrandTotal__c = myOrder.SubTotal__c + myOrder.tax__c +
                                    myOrder.Shipping__c;

            updateMap.put(myOrder.id, myOrder);

         }
     }

     // Only use one DML update at the end.
     // This minimizes the number of DML requests generated from this trigger.
     update updateMap.values();
}
```

ShippingDiscount Trigger

```
trigger ShippingDiscount on Shipping_Invoice__C (before update) {
    // Free shipping on all orders greater than $100
```

```
    for(Shipping_Invoice__C myShippingInvoice : Trigger.new)
    {
        if((myShippingInvoice.subtotal__c >= 100.00) &&
           (myShippingInvoice.ShippingDiscount__c == 0))
        {
            myShippingInvoice.ShippingDiscount__c =
                        myShippingInvoice.Shipping__c * -1;
            myShippingInvoice.GrandTotal__c += myShippingInvoice.ShippingDiscount__c;
        }
    }
}
```

Shipping Invoice Test

```
@IsTest
private class TestShippingInvoice{

    // Test for inserting three items at once
    public static testmethod void testBulkItemInsert(){
        // Create the shipping invoice. It's a best practice to either use defaults
        // or to explicitly set all values to zero so as to avoid having
        // extraneous data in your test.
        Shipping_Invoice__C order1 = new Shipping_Invoice__C(subtotal__c = 0,
                          totalweight__c = 0, grandtotal__c = 0,
                          ShippingDiscount__c = 0, Shipping__c = 0, tax__c = 0);

        // Insert the order and populate with items
        insert Order1;
        List<Item__c> list1 = new List<Item__c>();
        Item__c item1 = new Item__C(Price__c = 10, weight__c = 1, quantity__c = 1,
                                    Shipping_Invoice__C = order1.id);
        Item__c item2 = new Item__C(Price__c = 25, weight__c = 2, quantity__c = 1,
                                    Shipping_Invoice__C = order1.id);
        Item__c item3 = new Item__C(Price__c = 40, weight__c = 3, quantity__c = 1,
                                    Shipping_Invoice__C = order1.id);
        list1.add(item1);
        list1.add(item2);
        list1.add(item3);
        insert list1;

        // Retrieve the order, then do assertions
        order1 = [SELECT id, subtotal__c, tax__c, shipping__c, totalweight__c,
                  grandtotal__c, shippingdiscount__c
                  FROM Shipping_Invoice__C
                  WHERE id = :order1.id];

        System.assert(order1.subtotal__c == 75,
                'Order subtotal was not $75, but was '+ order1.subtotal__c);
        System.assert(order1.tax__c == 6.9375,
                'Order tax was not $6.9375, but was ' + order1.tax__c);
        System.assert(order1.shipping__c == 4.50,
                'Order shipping was not $4.50, but was ' + order1.shipping__c);
```

```
        System.assert(order1.totalweight__c == 6.00,
                'Order weight was not 6 but was ' + order1.totalweight__c);
        System.assert(order1.grandtotal__c == 86.4375,
                'Order grand total was not $86.4375 but was '
                 + order1.grandtotal__c);
        System.assert(order1.shippingdiscount__c == 0,
                'Order shipping discount was not $0 but was '
                + order1.shippingdiscount__c);
    }

    // Test for updating three items at once
    public static testmethod void testBulkItemUpdate(){

        // Create the shipping invoice. It's a best practice to either use defaults
        // or to explicitly set all values to zero so as to avoid having
        // extraneous data in your test.
        Shipping_Invoice__C order1 = new Shipping_Invoice__C(subtotal__c = 0,
                          totalweight__c = 0, grandtotal__c = 0,
                          ShippingDiscount__c = 0, Shipping__c = 0, tax__c = 0);

        // Insert the order and populate with items.
        insert Order1;
        List<Item__c> list1 = new List<Item__c>();
        Item__c item1 = new Item__C(Price__c = 1, weight__c = 1, quantity__c = 1,
                                    Shipping_Invoice__C = order1.id);
        Item__c item2 = new Item__C(Price__c = 2, weight__c = 2, quantity__c = 1,
                                    Shipping_Invoice__C = order1.id);
        Item__c item3 = new Item__C(Price__c = 4, weight__c = 3, quantity__c = 1,
                                    Shipping_Invoice__C = order1.id);
        list1.add(item1);
        list1.add(item2);
        list1.add(item3);
        insert list1;

        // Update the prices on the 3 items
        list1[0].price__c = 10;
        list1[1].price__c = 25;
        list1[2].price__c = 40;
        update list1;

        // Access the order and assert items updated
        order1 = [SELECT id, subtotal__c, tax__c, shipping__c, totalweight__c,
                  grandtotal__c, shippingdiscount__c
                  FROM Shipping_Invoice__C
                  WHERE Id = :order1.Id];

        System.assert(order1.subtotal__c == 75,
                        'Order subtotal was not $75, but was '+ order1.subtotal__c);
        System.assert(order1.tax__c == 6.9375,
                        'Order tax was not $6.9375, but was ' + order1.tax__c);
        System.assert(order1.shipping__c == 4.50,
                        'Order shipping was not $4.50, but was '
                        + order1.shipping__c);
        System.assert(order1.totalweight__c == 6.00,
```

```
                    'Order weight was not 6 but was ' + order1.totalweight__c);
        System.assert(order1.grandtotal__c == 86.4375,
                    'Order grand total was not $86.4375 but was '
                    + order1.grandtotal__c);
        System.assert(order1.shippingdiscount__c == 0,
                    'Order shipping discount was not $0 but was '
                    + order1.shippingdiscount__c);

    }

    // Test for deleting items
    public static testmethod void testBulkItemDelete(){

        // Create the shipping invoice. It's a best practice to either use defaults
        // or to explicitly set all values to zero so as to avoid having
        // extraneous data in your test.
        Shipping_Invoice__C order1 = new Shipping_Invoice__C(subtotal__c = 0,
                        totalweight__c = 0, grandtotal__c = 0,
                        ShippingDiscount__c = 0, Shipping__c = 0, tax__c = 0);

        // Insert the order and populate with items
        insert Order1;
        List<Item__c> list1 = new List<Item__c>();
        Item__c item1 = new Item__C(Price__c = 10, weight__c = 1, quantity__c = 1,
                                    Shipping_Invoice__C = order1.id);
        Item__c item2 = new Item__C(Price__c = 25, weight__c = 2, quantity__c = 1,
                                    Shipping_Invoice__C = order1.id);
        Item__c item3 = new Item__C(Price__c = 40, weight__c = 3, quantity__c = 1,
                                    Shipping_Invoice__C = order1.id);
        Item__c itemA = new Item__C(Price__c = 1, weight__c = 3, quantity__c = 1,
                                    Shipping_Invoice__C = order1.id);
        Item__c itemB = new Item__C(Price__c = 1, weight__c = 3, quantity__c = 1,
                                    Shipping_Invoice__C = order1.id);
        Item__c itemC = new Item__C(Price__c = 1, weight__c = 3, quantity__c = 1,
                                    Shipping_Invoice__C = order1.id);
        Item__c itemD = new Item__C(Price__c = 1, weight__c = 3, quantity__c = 1,
                                    Shipping_Invoice__C = order1.id);
        list1.add(item1);
        list1.add(item2);
        list1.add(item3);
        list1.add(itemA);
        list1.add(itemB);
        list1.add(itemC);
        list1.add(itemD);
        insert list1;

        // Seven items are now in the shipping invoice.
       // The following deletes four of them.
        List<Item__c> list2 = new List<Item__c>();
        list2.add(itemA);
        list2.add(itemB);
        list2.add(itemC);
        list2.add(itemD);
        delete list2;
```

```
        // Retrieve the order and verify the deletion
        order1 = [SELECT id, subtotal__c, tax__c, shipping__c, totalweight__c,
                  grandtotal__c, shippingdiscount__c
                  FROM Shipping_Invoice__C
                  WHERE Id = :order1.Id];

        System.assert(order1.subtotal__c == 75,
                      'Order subtotal was not $75, but was '+ order1.subtotal__c);
        System.assert(order1.tax__c == 6.9375,
                      'Order tax was not $6.9375, but was ' + order1.tax__c);
        System.assert(order1.shipping__c == 4.50,
                      'Order shipping was not $4.50, but was ' + order1.shipping__c);
        System.assert(order1.totalweight__c == 6.00,
                      'Order weight was not 6 but was ' + order1.totalweight__c);
        System.assert(order1.grandtotal__c == 86.4375,
                      'Order grand total was not $86.4375 but was '
                      + order1.grandtotal__c);
        System.assert(order1.shippingdiscount__c == 0,
                      'Order shipping discount was not $0 but was '
                      + order1.shippingdiscount__c);
    }
    // Testing free shipping
    public static testmethod void testFreeShipping(){

        // Create the shipping invoice. It's a best practice to either use defaults
        // or to explicitly set all values to zero so as to avoid having
        // extraneous data in your test.
        Shipping_Invoice__C order1 = new Shipping_Invoice__C(subtotal__c = 0,
                         totalweight__c = 0, grandtotal__c = 0,
                         ShippingDiscount__c = 0, Shipping__c = 0, tax__c = 0);

        // Insert the order and populate with items.
        insert Order1;
        List<Item__c> list1 = new List<Item__c>();
        Item__c item1 = new Item__C(Price__c = 10, weight__c = 1,
                                    quantity__c = 1, Shipping_Invoice__C = order1.id);
        Item__c item2 = new Item__C(Price__c = 25, weight__c = 2,
                                    quantity__c = 1, Shipping_Invoice__C = order1.id);
        Item__c item3 = new Item__C(Price__c = 40, weight__c = 3,
                                    quantity__c = 1, Shipping_Invoice__C = order1.id);
        list1.add(item1);
        list1.add(item2);
        list1.add(item3);
        insert list1;

        // Retrieve the order and verify free shipping not applicable
        order1 = [SELECT id, subtotal__c, tax__c, shipping__c, totalweight__c,
                  grandtotal__c, shippingdiscount__c
                  FROM Shipping_Invoice__C
                  WHERE Id = :order1.Id];

        // Free shipping not available on $75 orders
        System.assert(order1.subtotal__c == 75,
```

```
                    'Order subtotal was not $75, but was '+ order1.subtotal__c);
        System.assert(order1.tax__c == 6.9375,
                    'Order tax was not $6.9375, but was ' + order1.tax__c);
        System.assert(order1.shipping__c == 4.50,
                    'Order shipping was not $4.50, but was ' + order1.shipping__c);
        System.assert(order1.totalweight__c == 6.00,
                    'Order weight was not 6 but was ' + order1.totalweight__c);
        System.assert(order1.grandtotal__c == 86.4375,
                    'Order grand total was not $86.4375 but was '
                    + order1.grandtotal__c);
        System.assert(order1.shippingdiscount__c == 0,
                    'Order shipping discount was not $0 but was '
                    + order1.shippingdiscount__c);

        // Add items to increase subtotal
        item1 = new Item__C(Price__c = 25, weight__c = 20, quantity__c = 1,
                            Shipping_Invoice__C = order1.id);
        insert item1;

        // Retrieve the order and verify free shipping is applicable
        order1 = [SELECT id, subtotal__c, tax__c, shipping__c, totalweight__c,
                  grandtotal__c, shippingdiscount__c
                  FROM Shipping_Invoice__C
                  WHERE Id = :order1.Id];

        // Order total is now at $100, so free shipping should be enabled
        System.assert(order1.subtotal__c == 100,
                    'Order subtotal was not $100, but was '+ order1.subtotal__c);
        System.assert(order1.tax__c == 9.25,
                    'Order tax was not $9.25, but was ' + order1.tax__c);
        System.assert(order1.shipping__c == 19.50,
                    'Order shipping was not $19.50, but was '
                    + order1.shipping__c);
        System.assert(order1.totalweight__c == 26.00,
                    'Order weight was not 26 but was ' + order1.totalweight__c);
        System.assert(order1.grandtotal__c == 109.25,
                    'Order grand total was not $86.4375 but was '
                    + order1.grandtotal__c);
        System.assert(order1.shippingdiscount__c == -19.50,
                    'Order shipping discount was not -$19.50 but was '
                    + order1.shippingdiscount__c);
    }

     // Negative testing for inserting bad input
    public static testmethod void testNegativeTests(){

        // Create the shipping invoice. It's a best practice to either use defaults
        // or to explicitly set all values to zero so as to avoid having
        // extraneous data in your test.
        Shipping_Invoice__C order1 = new Shipping_Invoice__C(subtotal__c = 0,
                          totalweight__c = 0, grandtotal__c = 0,
                          ShippingDiscount__c = 0, Shipping__c = 0, tax__c = 0);

        // Insert the order and populate with items.
```

```
        insert Order1;
        Item__c item1 = new Item__C(Price__c = -10, weight__c = 1, quantity__c = 1,
                                    Shipping_Invoice__C = order1.id);
        Item__c item2 = new Item__C(Price__c = 25, weight__c = -2, quantity__c = 1,
                                    Shipping_Invoice__C = order1.id);
        Item__c item3 = new Item__C(Price__c = 40, weight__c = 3, quantity__c = -1,
                                    Shipping_Invoice__C = order1.id);
        Item__c item4 = new Item__C(Price__c = 40, weight__c = 3, quantity__c = 0,
                                    Shipping_Invoice__C = order1.id);

        try{
            insert item1;
        }
        catch(Exception e)
        {
            system.assert(e.getMessage().contains('Price must be non-negative'),
                          'Price was negative but was not caught');
        }

        try{
            insert item2;
        }
        catch(Exception e)
        {
            system.assert(e.getMessage().contains('Weight must be non-negative'),
                          'Weight was negative but was not caught');
        }

        try{
            insert item3;
        }
        catch(Exception e)
        {
            system.assert(e.getMessage().contains('Quantity must be positive'),
                          'Quantity was negative but was not caught');
        }

        try{
            insert item4;
        }
        catch(Exception e)
        {
            system.assert(e.getMessage().contains('Quantity must be positive'),
                          'Quantity was zero but was not caught');
        }
    }
}
```

Reserved Keywords

These words can be used only as keywords.

Table 3: Reserved Keywords

abstract	false	package
activate	final	parallel
and	finally	pragma
any	float	private
array	for	protected
as	from	public
asc	global	retrieve
autonomous	goto	return
begin	group	rollback
bigdecimal	having	select
blob	hint	set
boolean	if	short
break	implements	sObject
bulk	import	sort
by	in	static
byte	inner	string
case	insert	super
cast	instanceof	switch
catch	int	synchronized
char	integer	system
class	interface	testmethod
collect	into	then
commit	join	this
const	like	throw
continue	limit	time
currency	list	transaction
date	long	trigger
datetime	loop	true
decimal	map	try
default	merge	undelete
delete	new	update
desc	not	upsert
do	null	using
double	nulls	virtual
else	number	void

end	object	webservice
enum	of	when
exception	on	where
exit	or	while
export	outer	
extends	override	

These words are special types of keywords that aren't reserved words and can be used as identifiers.

- after
- before
- count
- excludes
- first
- includes
- last
- order
- sharing
- with

Documentation Typographical Conventions

Apex and Visualforce documentation uses these typographical conventions.

Convention	Description
`Courier font`	In descriptions of syntax, a monospace font indicates items that you should type as shown, except for brackets. For example: `Public class HelloWorld`
`Italics`	In descriptions of syntax, italics represent variables. You supply the actual value. In the following example, three values must be supplied: *`datatype variable_name`* [`=` *`value`*]; If the syntax is bold and italic, the text represents a code element that needs a value supplied by you, such as a class name or variable value: `public static class` ***`YourClassHere`*** `{ ... }`
`Bold Courier font`	In code samples and syntax descriptions, a bold courier font emphasizes a portion of the code or syntax.
< >	In descriptions of syntax, less-than and greater-than symbols (< >) are typed exactly as shown. `<apex:pageBlockTable value="{!account.Contacts}" var="contact">`

Convention	Description
	<pre> <apex:column value="{!contact.Name}"/>\n <apex:column value="{!contact.MailingCity}"/>\n <apex:column value="{!contact.Phone}"/>\n</apex:pageBlockTable></pre>
{ }	In descriptions of syntax, braces ({ }) are typed exactly as shown. <pre><apex:page>\n Hello {!$User.FirstName}!\n</apex:page></pre>
[ ]	In descriptions of syntax, anything included in brackets is optional. In the following example, specifying ***value*** is optional: <pre>***data_type variable_name*** [= ***value***];</pre>
|	In descriptions of syntax, the pipe sign means "or". You can do one of the following (not all). In the following example, you can create a new unpopulated set in one of two ways, or you can populate the set: <pre>Set<***data_type***> ***set_name***\n [= new Set<***data_type***>();] |\n [= new Set<***data_type***{***value*** [, ***value2***. . .] };] |\n ;</pre>