
Loyalty Management Developer Guide

Version 66.0, Spring '26



CONTENTS

Chapter 1: Loyalty Management Standard Objects	1
Loyalty Management Data Model	4
Loyalty Clubs Data Model	4
AnalyticsDatasetDefinition	5
Benefit	7
BenefitParameterValue	10
BenefitType	12
DigitalPass	14
DigitalPassTemplate	19
DigitalPassTplParameter	22
GameDefinition	24
GameParticipant	27
GameParticipantReward	31
GameReward	34
JournalReason	37
JournalSubType	38
JournalType	40
LoyaltyAggrPointExprLedger	42
LoyaltyLedger	44
LoyaltyLedgerTraceability	48
LoyaltyMemberCurrency	52
LoyaltyMembershipLifecycle	57
LoyaltyMemberTier	61
LoyaltyPartnerProduct	65
LoyaltyPgmCurrencySubtype	68
LoyaltyPgmEngmtAttribute	70
LoyaltyPgmEngmtAttrProm	74
LoyaltyPgmGroupMbrRInsp	76
LoyaltyPgmMbrAttributeVal	79
LoyaltyPgmMbrLinkedPtnr	82
LoyaltyPgmMbrPromEligView	85
LoyaltyPgmPartnerCurrency	92
LoyaltyPgmPartnerPromotion	94
LoyaltyPgmPtnrLdgrSummary	96
LoyaltyPgmPtnrPrepaidPack	99
LoyaltyProgram	102
LoyaltyProgramBadge	104
LoyaltyProgramCurrency	107
LoyaltyProgramCurrencyTier	112

Contents

LoyaltyProgramMbrPromotion	113
LoyaltyProgramMember	116
LoyaltyProgramMemberBadge	123
LoyaltyProgramMemberCase	126
LoyaltyProgramMemberMerge	128
LoyaltyProgramPartner	133
LoyaltyProgramPartnerLedger	140
LoyaltyProgramProcess	143
LoyaltyProgramWidget	147
LoyaltyTier	150
LoyaltyTierBenefit	153
LoyaltyTierEligibilitySrc	154
LoyaltyTierGroup	156
LoyaltyTierMshpFeeOption	161
LoyaltyTierPromotion	163
MemberBenefit	165
NetworkReferencedObject	169
Promotion	170
PromotionActionableList	181
PromotionChannel	183
PromotionExecutionEvalGrp	186
PromotionExecEvalGrpItem	189
PromotionLimit	191
PromotionLimitUsage	194
PromotionLoyaltyPtnrProdt	195
PromotionMarketSegment	198
PromotionPartyUsage	201
PromotionPartyUsageDetail	205
PromotionProduct	209
PromotionProductCategory	210
PromotionRuleTemplate	212
PromotionStage	214
PromotionStageEmailTemplate	217
TransactionJournal	218
Voucher	233
VoucherDefinition	239
Chapter 2: Loyalty Management Fields on Standard Objects	247
Loyalty Management Fields on EngagementChannelType	248
Loyalty Management Fields on InternalOrganizationUnit	248
Chapter 3: Coupon Objects in Loyalty Management	250
Chapter 4: Loyalty Management Associated Objects	251
StandardObjectNameChangeEvent	252

StandardObjectNameFeed	255
StandardObjectNameHistory	262
StandardObjectName OwnerSharingRule OwnerSharingRule	264
StandardObjectNameShare	266
Event	267
Task	288
Chapter 5: Apex Reference	303
ConnectApi Namespace	304
LoyaltyConnect Methods	304
LoyaltyManagementConnect Methods	305
LoyaltyVoucherConnect Methods	309
Input Classes	313
Output Classes	318
ConnectApi Enums Enums	325
LoyaltyManagement Namespace	326
CdpBasedLoyaltyPromotionInput Class	328
CdpBasedLoyaltyPromotionInputBuilder Class	329
CdpBasedLoyaltyPromotionOutput Class	330
ChangeTierInput Class	331
ChangeTierInputBuilder Class	332
ChangeTierOutput Class	335
CreditPointsOutput Class	337
DebitPointsOutput Class	339
GetMemberActiveSegmentsInput Class	341
GetMemberActiveSegmentsInputBuilder Class	341
GetMemberActiveSegmentsOutput Class	343
IssueVoucherInput Class	344
IssueVoucherInputBuilder Class	344
IssueVoucherOutput Class	350
LoyaltyActionError Class	352
LoyaltyActionResult Class	353
LoyaltyPromotionInput Class	354
LoyaltyPromotionInputBuilder Class	355
LoyaltyPromotionOutput Class	356
LoyaltyResources Class	357
MemberPointBalanceInput Class	362
MemberPointBalanceInputBuilder Class	363
MemberPointBalanceOutput Class	365
MemberTierInput Class	367
MemberTierInputBuilder Class	368
MemberTierOutput Class	369
PointsInput Class	372
PointsInputBuilder Class	373

TransferMemberPointsToGroupsInput Class	378
TransferMemberPointsToGroupsInputBuilder Class	378
UpdateCumulativeUsageCompletedInput Class	379
UpdateCumulativeUsageCompletedOutput Class	381
Chapter 6: Loyalty Management Business APIs	383
REST Reference	384
Resources	384
Request Bodies	456
Response Bodies	500
Loyalty Management Connect API Interchangeable Request Fields	604
Chapter 7: Loyalty Management Standard Invocable Actions	606
Adjust Points	608
Assign Member Tier Benefits	610
Accrual Cancellation	611
Credit Points	613
Change Tier	615
Change All Tiers or None	618
Debit Points	620
Generate Referral Code	623
Get Member's Active Segments	624
Get Points Balance	626
Get Loyalty Promotions Based on Member's Data Cloud	628
Get Promotions Based on Transaction Journal	630
Get Tier	632
Issue Voucher Action	634
Map Traceable Points for Redemption Transactions Action	637
Merge Memberships Action	639
Process Member Benefit	641
Redemption Cancellation	642
Run Program Process for Transaction Journal	644
Transfer Member Points to Groups	645
Transfer Points	646
Unmerge Memberships Action	649
Update Member Progress for Cumulative Promotion	651
Chapter 8: Loyalty Management Custom Invocable Actions	654
Issue Digital Pass	655
Refresh Digital Pass	658
Run Program Process	662
Chapter 9: Loyalty Management Metadata API Types	665
AnalyticsDatasetDefinition	666
BenefitAction	667

Flow for Loyalty Management	670
LoyaltyProgramSetup	672
Settings	688
IndustriesGamificationSettings	688
IndustriesLoyaltySettings	689
IndustriesUnifiedPromotionsSettings	693
Chapter 10: Loyalty Management Tooling API Objects	695
BenefitAction	696
Chapter 11: Loyalty Management Mobile SDK	701
What to Expect from Loyalty Management Mobile SDK	702
Supported Versions of Tools and Components	702
Supported DateTime Format	703
Getting Started with Loyalty Management Mobile SDK	703
Prepare to Install Loyalty Management Mobile SDK	704
Create an External Client App to Integrate with Loyalty Management	704
Prepare Experience Cloud Users for Loyalty Management Mobile SDK	705
Loyalty Management Mobile SDK for iOS Development	707
Clone and Import Loyalty Management Mobile SDK for iOS	708
Loyalty Management Mobile SDK for iOS Classes	709
Loyalty Management Mobile SDK API References for iOS	722
Loyalty Management Mobile SDK for Android Development	751
Clone and Import Loyalty Management Mobile SDK for Android	752
Loyalty Management Mobile SDK for Android Classes	753
Loyalty Management Mobile SDK API References for Android	764
Loyalty Management Sample App for Mobile SDK	764
Use the Loyalty Management Mobile SDK Sample App	765
Walkthrough of the Sample App for Loyalty Management Mobile SDK	766
Customize the Android SDK	766
Customize the iOS SDK	767
Receipt Scanning with Loyalty Management Mobile SDK	769
How Receipt Scanning Works	769
Loyalty Management Receipt Scanner Kit Unlocked Package	773
Set Up Amazon Web Services	776
Set Up Salesforce Org With Loyalty Management Receipt Scanner Kit Unlocked Package	781
Receipt Scan and Upload on iOS and Android	787
Loyalty Management Mobile SDK and Sample App Support	790
Upgrade Loyalty Management Mobile SDK for iOS	790
Chapter 12: Gamification Mobile SDK	791
Supported Versions of Tools and Components	792
Gamification Mobile SDK for iOS	792

Clone and Import Gamification Mobile SDK for iOS	793
Integrate Gamification Mobile iOS SDK into Loyalty Management Mobile iOS SDK	793
Gamification Mobile SDK iOS Classes	794
Gamification Mobile SDK API Reference for iOS	797
Game Zone UI Customization for iOS	805
Gamification Mobile SDK for Android	810
Clone and Import Gamification Mobile SDK for Android	811
Integrate Gamification Mobile Android SDK into Loyalty Management Mobile Android SDK	812
Gamification Mobile SDK Android Classes	812
Gamification Mobile Android SDK API Reference	815
Game Zone UI Customization for Android	815
Chapter 13: Data Processing Engine, Batch Management, and Monitor Workflow Services	817
Data Model	818
Common Tooling API Object	819
BatchJobDefinition	819
Common Platform Event	825
BatchJobStatusChangedEvent	825
Common Business APIs	827
REST Reference	827
Data Processing Engine	829
Data Processing Engine Tooling API Objects	830
Data Processing Engine Standard Object	836
Data Processing Engine Metadata API	841
Data Processing Engine Invocable Actions	868
Batch Management	871
Batch Management Tooling API Objects	871
Batch Management Standard Objects	882
Batch Management Metadata API	884
Batch Management Invocable Actions	890
Monitor Workflow Services	893
BatchJob	894
BatchJobPart	899
BatchJobPartFailedRecord	905
Chapter 14: StandardValueSet Names and Standard Picklist Fields for Loyalty Management	909

CHAPTER 1 Loyalty Management Standard Objects

In this chapter ...

- [Loyalty Management Data Model](#)
- [Loyalty Clubs Data Model](#)
- [AnalyticsDatasetDefinition](#)
- [Benefit](#)
- [BenefitParameterValue](#)
- [BenefitType](#)
- [DigitalPass](#)
- [DigitalPassTemplate](#)
- [DigitalPassTplParameter](#)
- [GameDefinition](#)
- [GameParticipant](#)
- [GameParticipantReward](#)
- [GameReward](#)
- [JournalReason](#)
- [JournalSubType](#)
- [JournalType](#)
- [LoyaltyAggrPointExpLdger](#)
- [LoyaltyLedger](#)
- [LoyaltyLedgerTraceability](#)
- [LoyaltyMemberCurrency](#)
- [LoyaltyMembershipLifecycle](#)
- [LoyaltyMemberTier](#)
- [LoyaltyPartnerProduct](#)
- [LoyaltyPgmCurrencySubtype](#)
- [LoyaltyPgmEngmtAttribute](#)
- [LoyaltyPgmEngmtAttrProm](#)
- [LoyaltyPgmGroupMbrRhsP](#)
- [LoyaltyPgmMbrAttributeVal](#)
- [LoyaltyPgmMbrLinkedPtnr](#)
- [LoyaltyPgmMbrPromBgView](#)
- [LoyaltyPgmPartnerCurrency](#)

Loyalty Management data model provides objects and fields to create and manage loyalty programs for your organization. Use the objects to manage partners and members, track members' activities, reward members, and manage promotions.

In addition to the standard objects included with the Loyalty Management license, the standard objects included with the Salesforce platform licenses enabled in your Org are available out-of-the-box.



Warning: When you customize standard fields, we recommend that you don't change the API names of the default picklist values.

SEE ALSO:

[Object Reference for the Salesforce Platform: Overview of Salesforce Platform Objects and Fields](#)
[SOAP API Developer Guide: About SOAP API](#)

Loyalty Management Standard Objects

- [LoyaltyPgmPartnerPromotion](#)
- [LoyaltyPgmPntldgSummary](#)
- [LoyaltyPgmPntPrpaidPack](#)
- [LoyaltyProgram](#)
- [LoyaltyProgramBadge](#)
- [LoyaltyProgramCurrency](#)
- [LoyaltyProgramCurrencyTier](#)
- [LoyaltyProgramMktPromotion](#)
- [LoyaltyProgramMember](#)
- [LoyaltyProgramMemberBadge](#)
- [LoyaltyProgramMemberCase](#)
- [LoyaltyProgramMemberMerge](#)
- [LoyaltyProgramPartner](#)
- [LoyaltyProgramPartnerLedger](#)
- [LoyaltyProgramProcess](#)
- [LoyaltyProgramWidget](#)
- [LoyaltyTier](#)
- [LoyaltyTierBenefit](#)
- [LoyaltyTierEligibilitySrc](#)
- [LoyaltyTierGroup](#)
- [LoyaltyTierMshpFeeOption](#)
- [LoyaltyTierPromotion](#)
- [MemberBenefit](#)
- [NetworkReferencedObject](#)
- [Promotion](#)
- [PromotionActionableList](#)
- [PromotionChannel](#)
- [PromotionExecutionEvalGp](#)
- [PromotionExecEvalGpItem](#)
- [PromotionLimit](#)
- [PromotionLimitUsage](#)
- [PromotionLoyaltyPntPrdft](#)
- [PromotionMarketSegment](#)
- [PromotionPartyUsage](#)
- [PromotionPartyUsageDetail](#)
- [PromotionProduct](#)
- [PromotionProductCategory](#)
- [PromotionRuleTemplate](#)
- [PromotionStage](#)
- [PromotionStageEmailTemplate](#)
- [TransactionJournal](#)
- [Voucher](#)

- [VoucherDefinition](#)

Loyalty Management Data Model

Learn about the objects and relationships within the Loyalty Management data model.

- [Loyalty program data model](#)
- [Loyalty journal data model](#)

Loyalty Clubs Data Model

Clubs are interest-based membership initiatives designed to provide exclusive benefits and privileges to customers.

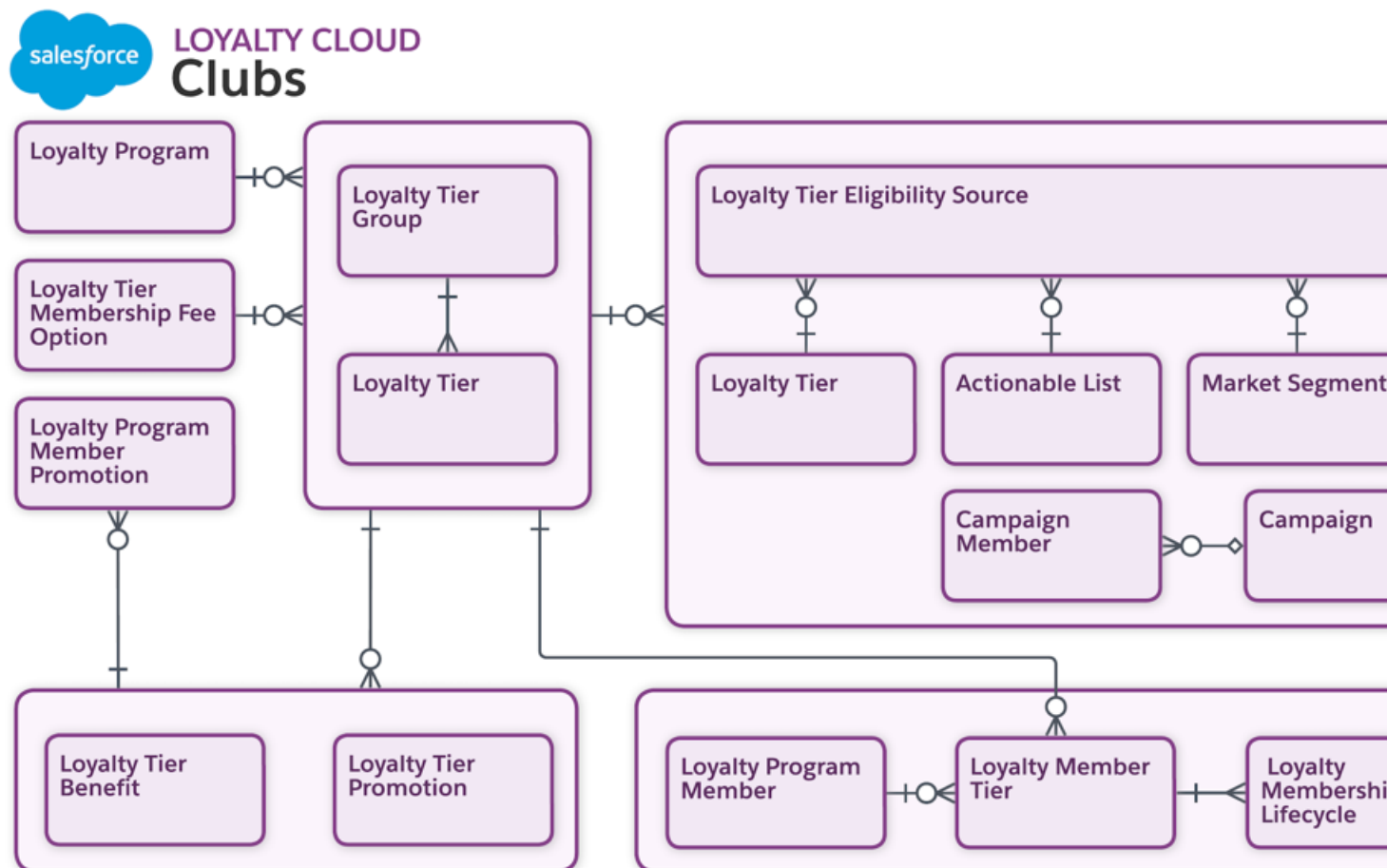
Clubs are modeled as loyalty tier groups. For every club, a loyalty tier is created with the same name.

This loyalty tier is then associated with various entities, such as eligibility criteria, promotions, and enrollment processes.

Clubs are either independent or associated with existing loyalty programs. For independent clubs, a dummy loyalty program is set up.

EDITIONS

Available in: Lightning Experience in **Enterprise, Performance, Developer,** and **Unlimited** Editions that have Loyalty Management enabled.



The following objects are associated with the clubs data model:

- [Loyalty Program](#) on page 102: Represents the loyalty program that the club is a part of.
- [Loyalty Tier Group](#) on page 156: Groups all your company's clubs together.

- [Loyalty Tier](#) on page 150: Stores detailed information about each club, including its name, description, and specific attributes.
- [Loyalty Tier Eligibility Source](#) on page 154: Determines the eligibility criteria for club membership. This can be based on:
 - Loyalty Tiers: Loyalty program members from specific tiers.
 - Segments: Customers who are part of specific Data Cloud segments.
 - Campaigns: Customers who are part of specific campaigns.
 - Actionable Lists: Customers who are part of specific actionable lists.
- [Loyalty Tier Membership Fee Option](#) on page 161: Stores information about the subscription fees for each club, including the amount and payment models, such as:
 - Cash
 - Loyalty Currency
 - Cash and Loyalty Currency
- [Loyalty Tier Benefit](#) on page 153: Associates club to a benefit that the club offers.
- [Loyalty Tier Promotion](#) on page 163: Associates exclusive promotions with the club, which can be offered to members.
- [Loyalty Program Member](#) on page 116: Stores information about a club member.
- [Loyalty Member Tier](#) on page 61: Tracks the membership tenure of each member, including the start and end dates of their club membership.
- [Loyalty Program Member Promotion](#) on page 113: Associates a club member with the promotion they are eligible for.
- [LoyaltyMembershipLifecycle](#) on page 57: Manages the stages of a member's subscription, from initial enrollment to ongoing engagement, and any changes in status.



Note: The objects and fields listed are shared between loyalty programs and clubs of your organization.

AnalyticsDatasetDefinition

Represents the datasets used for analytics, such as those in CRM Analytics and Tableau. This object is available in API version 65.0 and later.

Supported Calls

`create()`, `delete()`, `describeObjects()`, `query()`, `retrieve()`, `update()`, `upsert()`

Fields

Field	Details
DatasetCategory	Type picklist Properties Create, Defaulted on create, Filter, Group, Restricted picklist, Sort, Update Description The type of metric that the dataset is designed to support. Possible values are:

Field	Details
	- PromotionsByMembers—Members - PromotionsByRevenue—Revenue - PromotionsByTransactions—Transaction Journals The default value is PromotionsByRevenue.
DatasetName	Type string Properties Create, Filter, Group, Sort, Update Description The name of the dataset.
DeveloperName	Type string Properties Create, Filter, Group, Sort, Update Description The developer name of the dataset.
Language	Type picklist Properties Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The language of the dataset.
MasterLabel	Type string Properties Create, Filter, Group, Sort, Update Description The label of the dataset.
NamespacePrefix	Type string Properties Filter, Group, Nillable, Sort Description The namespace prefix that is associated with this object. Each Developer Edition org that creates a managed package has a unique namespace prefix. Limit: 15 characters. You can refer to a component in a managed package by using the <i>namespacePrefix__componentName</i> notation.

Field	Details
	The namespace prefix can have one of the following values: - In Developer Edition organizations, the namespace prefix is set to the namespace prefix of the organization for all objects that support it. There's an exception if an object is in an installed managed package. In that case, the object has the namespace prefix of the installed managed package. This field's value is the namespace prefix of the Developer Edition organization of the package developer. - In organizations that aren't Developer Edition organizations, NamespacePrefix is only set for objects that are part of an installed managed package. There's no namespace prefix for all other objects.
UsageCategory	Type picklist Properties Create, Defaulted on create, Filter, Group, Restricted picklist, Sort, Update Description The category of usage that uniquely identifies an agent topic. Possible values are: PromotionPerformanceInsights The default value is PromotionPerformanceInsights.

Benefit

Represents information about benefits associated with the loyalty program. Benefits are mapped to the members based on the tier or other program rules. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
BenefitActionId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description ID of the benefit action.

Field	Details
	This is a relationship field. Relationship Name BenefitAction Relationship Type Lookup Refers To BenefitAction
BenefitTypeId	Type reference Properties Create, Filter, Group, Sort Description ID of the benefit type. This is a relationship field. Relationship Name BenefitType Relationship Type Lookup Refers To BenefitType
Description	Type textarea Properties Create, Filter, Nillable, Sort, Update Description Detailed description of loyalty benefit.
ImageUrl	Type url Properties Create, Filter, Nillable, Sort, Update Description The URL for the image. This field is available in API version 57.0 and later.
IsActive	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update

Field	Details
	Description Required. Indicates whether the loyalty benefit is active.
IsBenefitSelectionAllowed	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether a member is allowed to select the benefit (<code>true</code>) or not (<code>false</code>). This field is available in API version 57.0 and later. The default value is <code>false</code> .
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
Name	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description The name of the loyalty benefit.
PrioritySequence	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description Priority rating of the loyalty benefit.
UnitId	Type reference

Field	Details
	Properties Create, Filter, Group, Nillable, Sort, Update Description The unit of measure for quantity. This is a relationship field. Relationship Name Unit Relationship Type Lookup Refers To UnitOfMeasure
Value	Type double Properties Create, Filter, Nillable, Sort, Update Description The quantity included in the loyalty benefit.

BenefitParameterValue

Represents the runtime value of a parameter that's used when executing a flow for benefit action. This object is available in API version 52.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
BenefitActionParameterId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The benefit action parameter value associated with the benefit action. This is a relationship field.

Field	Details
	Relationship Name BenefitActionParameter
	Relationship Type Lookup
	Refers To BenefitActionParameter
BenefitId	Type reference
	Properties Create, Filter, Group, Sort
	Description The benefit associated with the benefit action. This is a relationship field.
	Relationship Name Benefit
	Relationship Type Lookup
	Refers To Benefit
LastReferencedDate	Type dateTime
	Properties Filter, Nillable, Sort
	Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime
	Properties Filter, Nillable, Sort
	Description The most recent date on which a user viewed this record.
Name	Type string
	Properties Create, Filter, Group, idLookup, Sort, Update
	Description The name of the benefit parameter value.

Field	Details
ParameterValue	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The pre-defined values of the parameter.

BenefitType

Represents information about the type of benefits such as Support Benefits available to the loyalty program member. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
Category	Type picklist Properties Create, Filter, Group, Sort, Update Description Required. The category of the benefit type. Picklist values aren't provided for this field and must be added based on the requirements of the organization.
Description	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The description of benefit type.
ImageSource	Type string Properties Create, Filter, Nillable, Sort, Update

Field	Details
	Description The source of the image that represents the benefit type. This field is available in API version 62.0 and later.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyProgramId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description ID of the loyalty program. This is a relationship field. Relationship Name LoyaltyProgram Relationship Type Lookup Refers To LoyaltyProgram
Name	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description The name of the benefit type.
OwnerId	Type reference

Field	Details
	Properties Create, Defaulted on create, Filter, Group, Sort, Update Description The ID of the user who owns this record. This is a polymorphic relationship field. Relationship Name Owner Relationship Type Lookup Refers To Group, User
ProcessType	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description Required. The process type associated with the loyalty benefit type. Possible values are: • Loyalty • Rebate

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

BenefitTypeChangeEvent on page 252 (API Version 62.0)

Change events are available for the object.

BenefitTypeOwnerSharingRule

Sharing rules are available for the object.

BenefitTypeShare

Sharing is available for the object.

DigitalPass

Represents a user-facing representation generated from a digital pass template. It contains specific, relevant data, such as a membership number or voucher details, making each pass unique to the user. This pass can be optionally stored in a mobile or digital wallet. This object is available in API version 66.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
AccountId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The account ID associated with the digital pass. This field is a relationship field. Relationship Name Account Refers To Account
ContactId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The contact ID associated with the digital pass. This field is a relationship field. Relationship Name Contact Refers To Contact
DigitalPassTemplateId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The ID of the digital pass template. This field is a relationship field. Relationship Name DigitalPassTemplate

Field	Details
	Refers To DigitalPassTemplate
ExpiryDate	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description Date when the digital pass expires.
ExternalUniqueIdentifier	Type string Properties Create, Filter, Group, Sort, Update Description Required. The unique identifier of the digital pass at the service provider's end.
InputPayload	Type textarea Properties Create, Nillable, Update Description The specific data or parameters provided to issue a digital pass, such as a loyalty program member ID.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The timestamp when the current user last accessed this record indirectly, for example, through a list view or related record.
LastSyncDate	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The latest date when the digital pass was synchronized.
LastViewedDate	Type dateTime

Field	Details
	Properties Filter, Nillable, Sort Description The timestamp when the current user last viewed this record or list view. If this value is null, and <code>LastReferenceDate</code> is not null, the user accessed this record or list view indirectly.
LoyaltyProgramMemberId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The ID of the loyalty program member associated with the digital pass. This field is a relationship field. Relationship Name LoyaltyProgramMember Refers To LoyaltyProgramMember
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The name of the digital pass.
OwnerId	Type reference Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Required. The ID of the owner of this object. This field is a polymorphic relationship field. Relationship Name Owner Refers To Group, User
PartyId	Type reference Properties Create, Filter, Group, Sort, Update

Field	Details
	Description Required. The entity or individual to whom the digital pass is issued. It can be an account, or contact or a loyalty program member. This field is a polymorphic relationship field. Relationship Name Party Refers To Account, Contact, LoyaltyProgramMember
SourceRecordReferenceId	Type reference Properties Create, Filter, Group, Sort, Update Description Required. The activity or a transaction associated with the digital pass. A digital pass can be associated to a voucher, event or loyalty program enrollment. This field is a polymorphic relationship field. Relationship Name SourceRecordReference Refers To LoyaltyProgramMember, Voucher
Status	Type picklist Properties Create, Filter, Group, Restricted picklist, Sort, Update Description Required. Specifies the current state of the digital pass in the lifecycle of issuance and usage Possible values are: • Active • Expired • Inactive • SyncError
SyncErrorMessage	Type textarea Properties Create, Filter, Nillable, Sort, Update Description The message displayed when there is an error in synchronization.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

DigitalPassChangeEvent on page 252

Change events are available for the object.

DigitalPassHistory on page 262

History is available for tracked fields of the object.

DigitalPassOwnerSharingRule on page 264

Sharing rules are available for the object.

DigitalPassShare on page 266

Sharing is available for the object.

DigitalPassTemplate

Represents a template for creating digital passes, such as loyalty membership cards or coupons. It outlines the pass's design, data fields, and functionality, enabling the generation of multiple unique passes from a single, reusable template. This object is available in API version 66.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
Description	Type textarea Properties Create, Filter, Nillable, Sort, Update Description The description of the digital pass template.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The timestamp when the current user last accessed this record indirectly, for example, through a list view or related record.

Field	Details
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The timestamp when the current user last viewed this record or list view. If this value is null, and LastReferenceDate is not null, the user accessed this record or list view indirectly.
Name	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description Required. The name of the digital pass template.
OwnerId	Type reference Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Required. The ID of the owner of this object. This field is a polymorphic relationship field. Relationship Name Owner Refers To Group, User
ServiceProvider	Type picklist Properties Create, Filter, Group, Restricted picklist, Sort, Update Description Required. The organization that develops the digital pass. Possible values are: Google
ServicePrvdTplIdentifier	Type string Properties Create, Filter, Group, Nillable, Sort, Update

Field	Details
	Description The ID of the external template used by the digital pass.
Status	Type picklist Properties Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The status of the digital pass template. Possible values are: • ActivationInProgress • Active • Draft • Inactive The default value is Draft.
StatusReason	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The reason for the current status of the digital pass template. Displays the detailed error message from the system in case of digital wallet activation failure.
Type	Type picklist Properties Create, Filter, Group, Restricted picklist, Sort, Update Description Required. Specifies the type of the digital pass template. Possible values are: • GiftCard • Loyalty

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[DigitalPassTemplateChangeEvent](#) on page 252

Change events are available for the object.

DigitalPassTemplateHistory on page 262

History is available for tracked fields of the object.

DigitalPassTemplateOwnerSharingRule on page 264

Sharing rules are available for the object.

DigitalPassTemplateShare on page 266

Sharing is available for the object.

DigitalPassTmplParameter

Represents a specific, configurable data field within a digital pass template. It defines the content type such as text, date, or image, and properties such as label and visibility, for a particular section of the pass. This object is available in API version 66.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
DataType	Type picklist Properties Create, Filter, Group, Restricted picklist, Sort, Update Description Required. The data type of the digital pass template attribute. Possible values are: • Date • Datetime • Double • String
DefaultValue	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The default value of the digital pass template attribute.
DigitalPassTemplateId	Type reference

Field	Details
	Properties Create, Filter, Group, Sort Description Required. The ID of the digital pass template. This field is a relationship field. Relationship Name DigitalPassTemplate Relationship Type Master-detail Refers To DigitalPassTemplate (the master object)
IsRequired	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether the digital pass template attribute is required (<code>true</code>) or not (<code>false</code>). The default value is <code>false</code>.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The timestamp when the current user last accessed this record indirectly, for example, through a list view or related record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The timestamp when the current user last viewed this record or list view. If this value is null, and <code>LastReferenceDate</code> is not null, the user accessed this record or list view indirectly.
Name	Type string Properties Create, Filter, Group, idLookup, Sort, Update

Field	Details
	Description Required. The name of the digital pass template parameter.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[DigitalPassTplParameterChangeEvent](#) on page 252

Change events are available for the object.

[DigitalPassTplParameterHistory](#) on page 262

History is available for tracked fields of the object.

GameDefinition

Represents information about a game that the loyalty program offers to its members. This object is available in API version 60.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
Description	Type textarea Properties Create, Filter, Nillable, Sort, Update Description The description of the game.
EndDate	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The date till which the game can be offered to members.
ExpirationPeriod	Type int

Field	Details
	Properties Create, Filter, Group, Nillable, Sort, Update Description The number of days or months for which a game is valid for a member to play after the game is assigned to them.
ExpirationPeriodUnit	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The unit of measure for the ExpirationPeriod field. Possible values are: • Days • Months
IsGameAssignmentRequired	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether members are eligible to play the game only if they are assigned the game (<code>true</code>) or not (<code>false</code>). The default value is <code>false</code>.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
Name	Type string

Field	Details
	Properties Create, Filter, Group, idLookup, Sort, Update Description The name of the game.
OwnerId	Type reference Properties Create, Defaulted on create, Filter, Group, Sort, Update Description The owner of the record. This field is a polymorphic relationship field. Relationship Name Owner Relationship Type Lookup Refers To Group, User
StartDate	Type dateTime Properties Create, Filter, Sort, Update Description The date from which the game can be offered to members.
Status	Type picklist Properties Create, Filter, Group, Restricted picklist, Sort, Update Description The status of the game. Possible values are: • Active • Draft • Inactive
TimeoutDuration	Type int Properties Create, Filter, Group, Nillable, Sort, Update

Field	Details
	Description The duration the game is valid for after a participant starts to play it. The value is specified in seconds.
Type	Type picklist Properties Create, Filter, Group, Restricted picklist, Sort, Update Description The type of game. Possible values are: • Scratchcard • SpintheWheel—Spin The Wheel

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

GameDefinitionFeed

Feed tracking is available for the object.

GameDefinitionHistory

History is available for tracked fields of the object.

GameDefinitionShare

Sharing is available for the object.

GameParticipant

Represents a junction between a game definition and a participant who is either an account, a contact, or a loyalty program member. This object is available in API version 60.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
AccountId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description ID of the participant's account that's associated with the game definition. This field is a relationship field. Relationship Name Account Relationship Type Lookup Refers To Account
ContactId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description ID of the participant's contact that's associated with the game definition. This field is a relationship field. Relationship Name Contact Relationship Type Lookup Refers To Contact
GameDefinitionId	Type reference Properties Create, Filter, Group, Sort, Update Description ID of the game definition that has been assigned to the participant. This field is a relationship field. Relationship Name GameDefinition

Field	Details
	Relationship Type Lookup Refers To GameDefinition
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyProgramMemberId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description ID of the participant's loyalty program member record that's associated with the game definition. This field is a relationship field. Relationship Name LoyaltyProgramMember Relationship Type Lookup Refers To LoyaltyProgramMember
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The autogenerated unique number of the record.

Field	Details
OwnerId	Type reference Properties Create, Defaulted on create, Filter, Group, Sort, Update Description The owner of the record This field is a polymorphic relationship field. Relationship Name Owner Relationship Type Lookup Refers To Group, User
ParticipantId	Type reference Properties Create, Filter, Group, Sort, Update Description ID of the participant's contact, account, or loyalty program member record. This field is a polymorphic relationship field. Relationship Name Participant Relationship Type Lookup Refers To Account, Contact, LoyaltyProgramMember

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[GameParticipantChangeEvent](#) on page 252 (API Version 63.0)

Change events are available for the object.

[GameParticipantFeed](#)

Feed tracking is available for the object.

[GameParticipantHistory](#)

History is available for tracked fields of the object.

[GameParticipantShare](#)

Sharing is available for the object.

GameParticipantReward

Represents the reward that's provided to a participant after they've played a game. This object is available in API version 60.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
ExpirationDate	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The date till which the game is valid for the participant to play.
GameParticipantId	Type reference Properties Create, Filter, Group, Sort Description ID of the associated game participant. This field is a relationship field. Relationship Name GameParticipant Relationship Type Lookup Refers To GameParticipant
GameRewardId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description ID of the reward offered to the participant. This field is a relationship field. Relationship Name GameReward

Field	Details
	Relationship Type Lookup Refers To GameReward
IssuedRewardReferenceId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description ID of the voucher issued to the participant as a reward or the transaction journal that credits the points rewarded to the participant. This field is a polymorphic relationship field. Relationship Name IssuedRewardReference Relationship Type Lookup Refers To TransactionJournal, Voucher
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The autogenerated unique number of the record.

Field	Details
SourceActivityId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description ID of the record that represents the reason why the participant was eligible to play the game. This field is a polymorphic relationship field. Relationship Name SourceActivity Relationship Type Lookup Refers To GameDefinition, Promotion, TransactionJournal
Status	Type picklist Properties Create, Filter, Group, Restricted picklist, Sort, Update Description The participant's status for the game. Possible values are: - Expired - NoReward - Rewarded - YetToReward

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[GameParticipantRewardChangeEvent](#) on page 252 (API Version 63.0)

Change events are available for the object.

[GameParticipantRewardFeed](#)

Feed tracking is available for the object.

[GameParticipantRewardHistory](#)

History is available for tracked fields of the object.

GameReward

The reward that's offered to participants of a game. This object is available in API version 60.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
Color	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The color of the segment in the Spin the Wheel board that the represents the game reward.
CurrentRewardCount	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description The number of times the reward has been provided to game participants until the current date.
Description	Type textarea Properties Create, Filter, Nillable, Sort, Update Description The description of the reward.
GameDefinitionId	Type reference Properties Create, Filter, Group, Sort Description The game that the reward belongs to. This field is a relationship field.

Field	Details
	Relationship Name GameDefinition Relationship Type Lookup Refers To GameDefinition
ImageUrl	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The image URL of the segment in the Spin the Wheel board that the represents the game reward.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
MaximumRewardCount	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description The maximum number of times the reward can be provided to game participants.
Name	Type string Properties Create, Filter, Group, idLookup, Sort, Update

Field	Details
	Description The name of the reward.
RewardDefinitionId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description ID of the voucher definition that's used to issue vouchers or the loyalty program currency that's used to credit points to participants as game reward. This field is a polymorphic relationship field. Relationship Name RewardDefinition Relationship Type Lookup Refers To LoyaltyProgramCurrency, VoucherDefinition
RewardType	Type picklist Properties Create, Filter, Group, Restricted picklist, Sort, Update Description Possible values are: - CustomReward - LoyaltyPoints - NoReward - Raffle—Reserved for future use. - Voucher
RewardValue	Type double Properties Create, Filter, Nillable, Sort, Update Description The value of the currency or voucher that's issued to participants as game reward.
WinProbability	Type double

Field	Details
	Properties Create, Filter, Nillable, Sort, Update
	Description The probability of the reward being provided to a participant.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[GameRewardChangeEvent](#) on page 252 (API Version 62.0)

Change events are available for the object.

[GameRewardFeed](#)

Feed tracking is available for the object.

[GameRewardHistory](#)

History is available for tracked fields of the object.

JournalReason

Represents a view that specifies the reason for the action that created a transaction journal. This object is available in API version 51.0 and later.



Note: 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.

Supported Calls

`describeSObjects()`, `query()`, `retrieve()`

Fields

Field	Details
ApiName	Type string
	Properties Filter, Group, idLookup, Sort
	Description The unique name of the object in the API. This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. In managed packages, this field prevents naming conflicts on package installations. With

Field	Details
	this field, a developer can change the object's name in a managed package and the changes are reflected in a subscriber's organization.
IsActive	Type picklist Properties Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort Description Specifies if the Journal Reason picklist values in the Transaction Journal entity are active. Possible values are: • active • deleteInProgress • inactive
JournalReasonType	Type picklist Properties Filter, Group, Nillable, Restricted picklist, Sort Description Provides the reason for the action that created a transaction journal. Possible values are: • AccrualReversal—Accrual Reversal • ManualPointsAdjustment—Manual Points Adjustment • RedemptionReversal—Redemption Reversal
MasterLabel	Type string Properties Filter, Group, Nillable, Sort Description The label of the Journal Reason picklist.

SEE ALSO:

[Salesforce Help: Prerequisites for Managing Member Transactions](#)

JournalSubType

Represents the subcategory of transaction journal. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
Description	Type textarea Properties Create, Filter, Nillable, Sort, Update Description The description of journal subtype.
JournalTypeId	Type reference Properties Create, Filter, Group, Sort Description ID of the Journal Type. This is a relationship field. Relationship Name JournalType Relationship Type Lookup Refers To JournalType
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.

Field	Details
Name	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description The name of the journal subtype.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[JournalSubTypeChangeEvent](#) on page 252 (API Version 62.0)

Change events are available for the object.

[JournalSubTypeHistory](#)

History is available for tracked fields of the object.

SEE ALSO:

[Salesforce Help: Prerequisites for Managing Member Transactions](#)

JournalType

Represents the category of transactions. For example: accrual, redemption. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
Description	Type textarea Properties Create, Filter, Nillable, Sort, Update Description Detailed description of the journal type.

Field	Details
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
Name	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description The name of the journal type.
OwnerId	Type reference Properties Create, Defaulted on create, Filter, Group, Sort, Update Description The ID of the user who owns this record. This is a polymorphic relationship field. Relationship Name Owner Relationship Type Lookup Refers To Group, User
ProcessType	Type picklist Properties Create, Defaulted on create, Filter, Group, Restricted picklist, Sort, Update Description Required. The process type associated with the journal type.

Field	Details
	Possible value is:
	• <code>Loyalty</code>

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[JournalTypeChangeEvent](#) on page 252 (API Version 62.0)

Change events are available for the object.

[JournalTypeHistory](#)

History is available for tracked fields of the object.

[JournalTypeOwnerSharingRule](#)

Sharing rules are available for the object.

[JournalTypeShare](#)

Sharing is available for the object.

SEE ALSO:

[Salesforce Help: Prerequisites for Managing Member Transactions](#)

LoyaltyAggrPointExprLedger

Represents information about non-qualifying currency points that expire based on a fixed model tier group. This object is available in API version 54.0 and later.

Supported Calls

`create()`, `delete()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `undelete()`, `update()`, `upsert()`

LoyaltyAggrPointExprLedger isn't available in the Object Manager and can't be customized.

Fields

Field	Details
<code>ExpirationDate</code>	Type date Properties Create, Filter, Group, Sort, Update Description Required. The date on which the points are set to expire.

Field	Details
ExpirationPoints	Type double Properties Create, Filter, Sort, Update Description Required. The number of points that are set to expire on the expiration date.
LoyaltyProgramCurrencyId	Type reference Properties Create, Filter, Group, Sort, Update Description Required. The loyalty program currency for which points are set to expire. This is a relationship field. Relationship Name LoyaltyProgramCurrency Relationship Type Lookup Refers To LoyaltyProgramCurrency
LoyaltyProgramMemberId	Type reference Properties Create, Filter, Group, Sort, Update Description Required. The loyalty program member whose points are set to expire. This is a relationship field. Relationship Name LoyaltyProgramMember Relationship Type Lookup Refers To LoyaltyProgramMember
PointsStatus	Type picklist Properties Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The status of the points that are set to expire.

Field	Details
	Possible values are: • Active • Inactive The default value is 'Active'.
RedeemedPoints	Type double Properties Create, Filter, Nillable, Sort, Update Description The number of points that have been redeemed by the associated loyalty program member out of the total points set to expire.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[LoyaltyAggrPointExprLedgerChangeEvent](#)

Change events are available for the object.

LoyaltyLedger

Records the points that have been credited or debited for a member across the transactions. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `undelete()`

Fields

Field	Details
ActivityDate	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The date on which the transaction was performed by the loyalty program member.

Field	Details
AdditionalNotes	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The notes about the transactions performed by the loyalty program members.
Correlation	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The ID of any Salesforce or external record. The field lets you store IDs that are up to 15 characters. If you store IDs that are fewer than 15 characters and use the field to query the object, add spaces after your query term to ensure that the query term is 15 characters.
EscrowPointsCreditDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The date on which the escrow points are credited to the points balance for the loyalty ledger. This field is available in API version 53.0 and later.
EventType	Type picklist Properties Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The type of the loyalty transaction. Possible values are: • Credit • Debit • Expiry • Transfer The default value is 'Credit'.
ExpiryDate	Type date

Field	Details
	Properties Create, Filter, Group, Nillable, Sort, Update Description The date on which the transaction expires.
Flow	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The name of the flow.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyPgmCrcySubtypeId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description Loyalty program currency subtype associated with the loyalty ledger. This field is available in API version 61.0 and later. This field is a relationship field. Relationship Name LoyaltyPgmCrcySubtype Relationship Type Lookup Refers To LoyaltyPgmCurrencySubtype

Field	Details
LoyaltyProgramCurrencyId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description ID of the loyalty program currency. This is a relationship field. Relationship Name LoyaltyProgramCurrency Relationship Type Lookup Refers To LoyaltyProgramCurrency
LoyaltyProgramMemberId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description ID of the loyalty program member. This is a relationship field. Relationship Name LoyaltyProgramMember Relationship Type Lookup Refers To LoyaltyProgramMember
Points	Type double Properties Create, Filter, Nillable, Sort, Update Description The points available for the loyalty program member.
PromotionId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update

Field	Details
	Description The promotion that's applied to the associated transaction resulting in the credit or debit of points. This field is available in API version 52.0 and later This is a relationship field. Relationship Name Promotion Relationship Type Lookup Refers To Promotion
TransactionJournalId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description ID of the transaction journal. This is a relationship field. Relationship Name TransactionJournal Relationship Type Lookup Refers To TransactionJournal

LoyaltyLedgerTraceability

Represents a tracker for the loyalty ledger. The debit ledger consumes points from the linked credit ledger for a member redemption or an accrual cancellation. This object is available in API version 62.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `undelete()`, `update()`, `upsert()`

LoyaltyAggrPointExprLedger isn't available in the Object Manager and can't be customized.

Fields

Field	Details
ActionType	Type picklist Properties Create, Defaulted on create, Filter, Group, Restricted picklist, Sort Description Required. Specifies the action that creates an entry in the ledger. Possible values are: • CreditCancellation • CreditForArrears • Debit • DebitCancellation • DebitWithArrears • GroupTransfer The default value is Debit.
CreditLoyaltyLedgerId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description Required. The credit type loyalty ledger associated with the loyalty ledger tracker. Points are consumed from the credit type ledger for the loyalty program member's debit type ledger. This field is a relationship field. Relationship Name CreditLoyaltyLedger Refers To LoyaltyLedger
DebitLoyaltyLedgerId	Type reference Properties Create, Filter, Group, Sort Description Required. The debit ledger associated with the loyalty ledger tracker. Points are consumed by the debit ledger from the loyalty program member's point balance. This field is a relationship field. Relationship Name DebitLoyaltyLedger

Field	Details
	Refers To LoyaltyLedger
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyPgmCurrencySubtypeId	Type reference Properties Create, Filter, Group, Nillable, Sort Description The ID of the loyalty program currency subtype associated with the related loyalty ledger. This field is a relationship field. Relationship Name LoyaltyPgmCurrencySubtype Refers To LoyaltyPgmCurrencySubtype
LoyaltyProgramCurrencyId	Type reference Properties Create, Filter, Group, Nillable, Sort Description The ID of the loyalty program currency associated with the related loyalty ledger. This field is a relationship field. Relationship Name LoyaltyProgramCurrency Refers To LoyaltyProgramCurrency

Field	Details
LoyaltyProgramMemberId	Type reference Properties Create, Filter, Group, Nillable, Sort Description The ID of the loyalty program member associated with the related loyalty ledger. This field is a relationship field. Relationship Name LoyaltyProgramMember Refers To LoyaltyProgramMember
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The autogenerated number of the record.
OwnerId	Type reference Properties Create, Defaulted on create, Filter, Group, Sort, Update Description The ID of the user who owns this record. This field is a polymorphic relationship field. Relationship Name Owner Refers To Group, User
Points	Type double Properties Create, Filter, Sort Description Required. The number of points used from the credit ledger by the debit ledger.
ProcessingSequenceInfo	Type textarea

Field	Details
	Properties Create, Nillable, Update Description The order in which the currency subtypes are processed.
ProcessingStatus	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The processing status of the traceability records. Possible values are: • InProgress • Pending • Processed

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

LoyaltyLedgerTraceabilityShare

Sharing is available for the object.

LoyaltyMemberCurrency

Represents the information about point balance for a member and a particular currency. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
BalanceBeforeReset	Type double

Field	Details
	Properties Create, Filter, Nillable, Sort, Update Description The last balance of points before resetting the qualifying points of the member.
EscrowPointsBalance	Type double Properties Filter, Nillable, Sort Description The number of escrow points to be credited to the loyalty member's points balance. This field is available in API version 53.0 and later. This is a calculated field.
ExpirablePoints	Type double Properties Create, Filter, Nillable, Sort, Update Description Calculates the points to be expired for unredeemed points.
LastAccrualProcessedDate	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The date when the last accrual was processed. This field indicates the date and time at which the Point Balance calculation process updated the total accrued points for a member and its currency the last time.
LastEscrowProcessedDate	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The last date and time when the Roll Over Escrow Points to Members Data Processing Engine definition was run to calculate rollover of escrow points. This field is available in API version 53.0 and later.
LastExpirationProcessRunDate	Type date

Field	Details
	Properties Create, Filter, Group, Nillable, Sort, Update Description The most recent date when the expired points for a member were calculated.
LastExpiredPtTraceabilityRunDt	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The latest date on which a process was run to trace points that have expired. This field is available in API version 65.0 and later.
LastPointsAggregationDate	Type Properties Description The latest date on which a process was run to trace points that have expired.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastResetDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The most recent date when the qualifying points for members were reset.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyMemberId	Type reference

Field	Details
	Properties Create, Filter, Group, Sort Description ID of the loyalty program member. This is a relationship field. Relationship Name LoyaltyMember Relationship Type Lookup Refers To LoyaltyProgramMember
LoyaltyProgramCurrencyId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description ID of the loyalty program currency. This is a relationship field. Relationship Name LoyaltyProgramCurrency Relationship Type Lookup Refers To LoyaltyProgramCurrency
Name	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description The name of the loyalty member currency.
NextResetDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The next date when the qualifying points for members are reset.

Field	Details
PointsBalBeforeSecondLastReset	Type double Properties Create, Filter, Nillable, Sort, Update Description The points balance before the second to last points reset of the member's associated qualifying currency. This field is available in API version 55.0 and later.
PointsBalance	Type double Properties Filter, Nillable, Sort Description Represents information about Point balance for a member and a particular currency.
RedemStlPendFromDateTime	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description Date and time from when the member's redemption settlement is pending for the currency. This field is available in API version 61.0 and later.
TotalEscrowPointsAccrued	Type double Properties Create, Filter, Nillable, Sort, Update Description The number of escrow points that the associated member has accrued. This field is available in API version 53.0 and later.
TotalEscrowRolloverPoints	Type double Properties Create, Filter, Nillable, Sort, Update Description The total number of escrow points that are credited to the points balance until the current date. This field is available in API version 53.0 and later.
TotalPointsAccrued	Type double

Field	Details
	Properties Create, Filter, Nillable, Sort, Update Description The total points accrued by the member until today.
TotalPointsExpired	Type double Properties Create, Filter, Nillable, Sort, Update Description The total points expired for a member until today.
TotalPointsRedeemed	Type double Properties Create, Filter, Nillable, Sort, Update Description The total points redeemed by the member until today.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[LoyaltyMemberCurrencyChangeEvent](#) on page 252 (API Version 62.0)

Change events are available for the object.

[LoyaltyMemberCurrencyFeed](#) (API version 52.0)

Sharing rules are available for the object.

[LoyaltyMemberCurrencyHistory](#) (API version 52.0)

Sharing is available for the object.

LoyaltyMembershipLifecycle

Represents information about a member's lifecycle in the loyalty program. This object is available in API version 57.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
Comment	Type textarea Properties Create, Filter, Nillable, Sort, Update Description The comments for the member's lifecycle event.
EffectiveEndDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The date till when the member's lifecycle event is effective.
EffectiveEndTime	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The start date and time of the lifecycle event.
EffectiveStartDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The date from when the member's lifecycle event is effective.
EffectiveStartTime	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The start date and time of the lifecycle event.
ExternalIdentifier	Type string Properties Create, Filter, Group, Nillable, Sort, Update

Field	Details
	Description The external identifier of the member's lifecycle event.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyMemberTierId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The ID of the loyalty member tier associated with the loyalty membership lifecycle. This field is a relationship field. Relationship Name LoyaltyMemberTier Refers To LoyaltyMemberTier
LoyaltyProgramMemberId	Type reference Properties Create, Filter, Group, Sort Description The loyalty program member whose lifecycle is tracked. This field is a relationship field. Relationship Name LoyaltyProgramMember Relationship Type Lookup

Field	Details
	Refers To LoyaltyProgramMember
MembershipStatus	Type picklist Properties Create, Filter, Group, Sort, Update Description Required. Specifies the status of the member's lifecycle event.
MembershipType	Type picklist Properties Create, Filter, Group, Sort, Update Description Required. Specifies the type of lifecycle event of the member.
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description Name of the lifecycle event.
OrderItemId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The order associated with the lifecycle event. This field is a relationship field. Relationship Name OrderItem Relationship Type Lookup Refers To OrderItem

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[LoyaltyMembershipLifecycleHistory](#)

History is available for tracked fields of the object.

LoyaltyMemberTier

Represents information about the current tier of the member across tier groups. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
AreBenefitsAssigned	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether benefits are already assigned to a member tier.
ChangeReasonType	Type picklist Properties Create, Filter, Group, Nillable, Sort, Update Description Represents the information of the reason for changing a member tier. Possible values are: - Based on Past Activities or Tier - Resolve Tier Conflict - VIP Member
EffectiveDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update

Field	Details
	Description The date from which the tier is in effect.
EnrollmentStartDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The date on which the member has enrolled in the associated tier.
EffectiveTime	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The date and time on which a member is assigned the associated tier.
EnrollmentType	Type picklist Properties Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description Specifies the type of enrollment for the loyalty member tier. Possible values are: • Paid • Segment • Tier The default value is Segment.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort

Field	Details
	Description The most recent date on which a user viewed this record.
LoyaltyMemberId	Type reference Properties Create, Filter, Group, Sort Description ID of the loyalty program member. This is a relationship field. Relationship Name LoyaltyMember Relationship Type Lookup Refers To LoyaltyProgramMember
LoyaltyTierGroupId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description ID of the loyalty tier group. This is a relationship field. Relationship Name LoyaltyTierGroup Relationship Type Lookup Refers To LoyaltyTierGroup
LoyaltyTierId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description ID of the loyalty tier. This is a relationship field. Relationship Name LoyaltyTier

Field	Details
	Relationship Type Lookup Refers To LoyaltyTier
Name	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description The name of the loyalty member tier.
ReasonForChange	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The reason to change a member tier.
Status	Type picklist Properties Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description Specifies the status of the loyalty member tier. Possible values are: - Active - Inactive - Trial The default value is <i>Active</i> .
TierExpirationDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The date on which the tier expires.
TierExpirationTime	Type dateTime

Field	Details
	Properties Create, Filter, Nillable, Sort, Update Description The date and time on which the tier membership ends.
TierMemberIdentifier	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description Unique membership number of the member tier.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[LoyaltyMemberTierChangeEvent](#) on page 252 (API Version 62.0)

Change events are available for the object.

[LoyaltyMemberTierHistory](#) (API version 62.0)

Sharing is available for the object.

LoyaltyPartnerProduct

Represents products and product categories offered by loyalty partners. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
EndDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The date on which a partner ends the offering of a product for accrual or redemption.

Field	Details
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyPartnerId	Type reference Properties Create, Filter, Group, Sort Description ID of the loyalty program partner. This is a relationship field. Relationship Name LoyaltyPartner Relationship Type Master-detail Refers To LoyaltyProgramPartner
Name	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description The name of the loyalty partner product.
PartnerCategoryDerivedFieldId	Type reference Properties Filter, Group, Nillable, Sort Description A lookup field to show product category in Partner Product.

Field	Details
	This is a relationship field. Relationship Name PartnerCategoryDerivedField Relationship Type Lookup Refers To ProductCategory
PartnerProductCost	Type currency Properties Create, Filter, Nillable, Sort, Update Description The cost incurred by the partner when members redeem the product during the promotion. This field is available in API version 55.0 and later.
ProductDerivedFieldId	Type reference Properties Filter, Group, Nillable, Sort Description A lookup field to show product in Partner Product. This is a relationship field. Relationship Name ProductDerivedField Relationship Type Lookup Refers To Product2
ProductId	Type reference Properties Create, Filter, Group, Sort, Update Description Required. ID of the partner product. This is a polymorphic relationship field. Relationship Name Product Relationship Type Lookup

Field	Details
	Refers To Product2, ProductCategory
StartDate	Type date Properties Create, Filter, Group, Sort, Update Description Required. The date on which a partner starts the offering of a product for accrual or redemption.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[LoyaltyPartnerProductChangeEvent](#) on page 252 (API Version 52.0)

Change events are available for the object.

[LoyaltyPartnerProductHistory](#) (API version 52.0)

History is available for tracked fields of the object.

LoyaltyPgmCurrencySubtype

Represents a subtype of a fixed-model non-qualifying loyalty program currency. This object is available in API version 61.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Special Access Rules

This object is available only if the Loyalty Management - Advanced license is enabled in your org.

Fields

Field	Details
CostPerUnit	Type currency Properties Create, Filter, Nillable, Sort, Update

Field	Details
	Description Monetary value of each unit of the loyalty program currency subtype.
Description	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description Description of the loyalty program currency subtype.
IsActive	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Required. Indicates whether the loyalty program currency subtype is active (true) or not (false). The default value is <code>false</code> .
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description Most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description Most recent date on which a user viewed this record.
LoyaltyProgramCurrencyId	Type reference Properties Create, Filter, Group, Sort Description Required. The fixed-model non-qualifying loyalty program currency associated with the loyalty program currency subtype. This field is a relationship field.

Field	Details
	Relationship Name LoyaltyProgramCurrency Relationship Type Master-detail Refers To LoyaltyProgramCurrency (the master object)
Name	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description Required. The name of the loyalty program currency subtype.
UsagePriorityNumber	Type int Properties Create, Filter, Group, idLookup, Sort, Update Description Required. The usage priority number of the loyalty program currency subtype. The usage priority number decides the order in which the points are redeemed across currency subtypes when multiple currency subtypes expire on the same date.

LoyaltyPgmEngmtAttribute

Represents information about an engagement attribute for a loyalty program. This object is available in API version 55.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
DefaultValue	Type string Properties Create, Filter, Group, Nillable, Sort, Update

Field	Details
	Description The default value for the loyalty program engagement attribute.
Description	Type textarea Properties Create, Filter, Group, Nillable, Sort, Update Description The description for the loyalty program engagement attribute.
EndDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The date when the loyalty program engagement attribute ends.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyProgramId	Type reference Properties Create, Filter, Group, Sort Description The loyalty program that's related to the loyalty program engagement attribute. This is a relationship field. Relationship Name LoyaltyProgram

Field	Details
	Relationship Type Master-detail Refers To LoyaltyProgram
MaximumResetCountPerPeriod	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description The number of times the current value of an engagement attribute can be reset during a period.
Name	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description The name of the loyalty program engagement attribute.
ResetValueOn	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The day on which loyalty program members' current value for the engagement attribute must be reset. If the reset frequency is Weekly, specify the day of the week when the current value is reset. To reset the current value on a Sunday, enter 1. To reset the current value on a Saturday, enter 7. If the reset frequency is Monthly, enter a value between 1 and 31. To reset the current value on the fifteenth of every month, enter 15. If the reset frequency is Yearly, enter a value between 1 and 12. To reset the current value on the first day of April month, enter 4. If the reset frequency is custom, enter the number of days after which the current value is reset.
ShouldOverwriteAttrCurrVal	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether the associated member's current value is overwritten every period (<code>true</code>) or not (<code>false</code>). The default value is ' <code>false</code> '.

Field	Details
StartDate	Type date Properties Create, Filter, Group, Sort, Update Description Required. The date when the loyalty program engagement attribute ends.
Status	Type picklist Properties Create, Defaulted on create, Filter, Group, Restricted picklist, Sort, Update Description Required. The status of the loyalty program engagement attribute. Possible values are: • Active • Draft • Inactive The default value is 'Draft'.
TargetValue	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The value that the loyalty program member must reach to qualify for rewards of the engagement attribute.
ValueResetFrequency	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The frequency at which the loyalty program member's engagement attribute value is reset. Possible values are: • CustomPeriod—Custom Period • Daily • Monthly • Weekly • Yearly

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[LoyaltyPgmEngmtAttributeChangeEvent](#) on page 252 (API Version 62.0)

Change events are available for the object.

[LoyaltyPgmEngmtAttributeHistory](#)

History is available for tracked fields of the object.

LoyaltyPgmEngmtAttrProm

Represents a junction between Loyalty Program Engagement Attribute and Promotion. Associate an engagement attribute with a promotion if you want to track the engagement attribute during the promotion period. This object is available in API version 55.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyProgramEngmtAttrId	Type reference Properties Create, Filter, Group, Sort Description The loyalty program engagement attribute associated with the promotion.

Field	Details
	This is a relationship field. Relationship Name LoyaltyProgramEngmtAttr Relationship Type Master-detail Refers To LoyaltyPgmEngmtAttribute
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The name of the loyalty program engagement attribute promotion. The name is autopopulated.
PromotionId	Type reference Properties Create, Filter, Group, Sort Description The promotion associated with the loyalty program engagement attribute. This is a relationship field. Relationship Name Promotion Relationship Type Master-detail Refers To Promotion

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[LoyaltyPgmEngmtAttrPromChangeEvent](#) on page 252 (API Version 62.0)

Change events are available for the object.

[LoyaltyPgmEngmtAttrPromHistory](#)

History is available for tracked fields of the object.

LoyaltyPgmGroupMbrRlnsp

Represents a junction between a Corporate or Individual type loyalty program member who is part of a group and the Group type loyalty program member. This object is available in API version 53.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
AccountId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The ID of the account with which the loyalty program member is associated. This field is a relationship field. Relationship Name Account Refers To Account
CanMemberRedeemFromGroup	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Required. Indicates whether the member can redeem points from a group. This field isn't used. You can customize and use this field based on your org's requirements. The default value is 'false'.
ContactId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The ID of the contact with which the loyalty program member is associated. This field is a relationship field.

Field	Details
	Relationship Name Contact
	Refers To Contact
ContributionCalculationType	Type picklist
	Properties Create, Filter, Group, Restricted picklist, Sort, Update
	Description Required. Specifies the basis on which the member's point contribution to a group is calculated. Possible values are: Percentage
LastReferencedDate	Type dateTime
	Properties Filter, Nillable, Sort
	Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime
	Properties Filter, Nillable, Sort
	Description The most recent date on which a user viewed this record.
LoyaltyProgramGroupMemberId	Type reference
	Properties Create, Filter, Group, Sort
	Description ID of the Group type loyalty program member that the related Individual or Corporate type loyalty program member is a part of. This is a relationship field.
	Relationship Name LoyaltyProgramGroupMember
	Relationship Type Master-detail

Field	Details
	Refers To LoyaltyProgramMember
MemberName	Type string Properties Filter, Group, Nillable, Sort Description The name of the related loyalty program member record. For an Individual type loyalty program member, it's the first name and the last name of the contact record associated with loyalty program member record. For a Corporate type loyalty program member, it's the name of the account record associated with the loyalty program member record. This field is read-only.
MemberPointContributionPercent	Type percent Properties Create, Filter, Nillable, Sort, Update Description The percentage of a group member's total accrued points that's transferred to the group.
MemberRole	Type picklist Properties Create, Filter, Group, Nillable, Sort, Update Description Specifies the role of the loyalty program member in a group. You can add values to the picklist. This field isn't used. You can customize and use this field based on your org's requirements. Possible values are: - Primary - Secondary
Name	Type string Properties Filter, Group, idLookup, Sort Description The name of the loyalty program member group relationship record.
RelatedLoyaltyProgramMemberId	Type reference

Field	Details
	Properties Create, Filter, Group, Nillable, Sort
	Description The related Individual or Corporate type loyalty program member associated with the loyalty program group member relationship. This is a relationship field.
	Relationship Name RelatedLoyaltyProgramMember
	Relationship Type Lookup
	Refers To LoyaltyProgramMember

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[LoyaltyPgmGroupMbrInspChangeEvent](#) on page 252 (API Version 62.0)

Change events are available for the object.

[LoyaltyPgmGroupMbrInspHistory](#)

History is available for tracked fields of the object.

LoyaltyPgmMbrAttributeVal

Represents information about the value reached by a loyalty program member for an engagement attribute. This object is available in API version 55.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`

Fields

Field	Details
CumulativeValue	Type double
	Properties Create, Filter, Nillable, Sort

Field	Details
	Description The cumulative value achieved by a loyalty program member for the related loyalty program engagement attribute across all the periods.
CurrentValue	Type string Properties Create, Filter, Group, Nillable, Sort Description The value achieved by a loyalty program member for the related loyalty program member engagement attribute in during a specific period.
EndDate	Type date Properties Create, Filter, Group, Nillable, Sort Description The end date of the period for which the member's progress towards achieving the related engagement attribute target is tracked.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastResetDate	Type date Properties Create, Filter, Group, Nillable, Sort Description The date on which the related loyalty program member's current value was last reset.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.

Field	Details
LoyaltyPgmEngmtAttributeId	Type reference Properties Create, Filter, Group, Sort Description Required. The loyalty program engagement attribute that's related to the loyalty program member engagement attribute value. This is a relationship field. Relationship Name LoyaltyPgmEngmtAttribute Relationship Type Lookup Refers To LoyaltyPgmEngmtAttribute
LoyaltyProgramMemberId	Type reference Properties Create, Filter, Group, Sort Description The loyalty program member that's related to the loyalty program member engagement attribute value. This is a relationship field. Relationship Name LoyaltyProgramMember Relationship Type Master-detail Refers To LoyaltyProgramMember
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The name of the loyalty program engagements attribute related to the loyalty program member engagement attribute value.
ResetCount	Type int

Field	Details
	Properties Create, Filter, Group, Nillable, Sort Description The number of times the current value of the engagement attribute is reset.
StartDate	Type date Properties Create, Filter, Group, Sort Description Required. The start date of the period for which the member's progress towards achieving the related engagement attribute target is tracked.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[LoyaltyPgmMbrAttributeValChangeEvent](#) on page 252 (API Version 62.0)

Change events are available for the object.

LoyaltyPgmMbrLinkedPtnr

Represents the association between a loyalty program member and their membership with a partner. It acts as an association that facilitates points accrual and data exchange in a joint loyalty program. This object is available in API version 66.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
Comments	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The description for the unlink reason.

Field	Details
ExternalIdentifier	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The external identifier of the loyalty program member with partner.
IsActive	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether the linkage is active (<code>true</code>) or inactive (<code>false</code>).
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The timestamp when the current user last accessed this record indirectly, for example, through a list view or related record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The timestamp when the current user last viewed this record or list view. If this value is null, and <code>LastReferenceDate</code> is not null, the user accessed this record or list view indirectly.
LinkDate	Type dateTime Properties Create, Filter, Sort Description The date on which a loyalty program member is linked to a partner.
LoyaltyProgramMemberId	Type reference Properties Create, Filter, Group, Sort

Field	Details
	Description Required. The ID of the loyalty program member to be linked to a partner. This field is a relationship field. Relationship Name LoyaltyProgramMember Relationship Type Master-detail Refers To LoyaltyProgramMember (the master object)
LoyaltyProgramPartnerId	Type reference Properties Create, Filter, Group, Sort Description Required. The ID of the partner to be linked to the loyalty program member. This field is a relationship field. Relationship Name LoyaltyProgramPartner Refers To LoyaltyProgramPartner
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The name of the loyalty program member linked partner.
UnlinkDate	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The date on which a loyalty program member is unlinked from a partner.
UnlinkReasonType	Type picklist Properties Create, Filter, Group, Nillable, Sort, Update

Field	Details
	Description The reason for unlinking a loyalty program member from a partner.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[LoyaltyPgmMbrLinkedPtrnChangeEvent](#) on page 252

Change events are available for the object.

[LoyaltyPgmMbrLinkedPtrnHistory](#) on page 262

History is available for tracked fields of the object.

LoyaltyPgmMbrPromEligView

Represents a virtual object that provides a consolidated view of the loyalty program member's eligibility for a promotion and their enrollment status in eligible promotions. This object is available in API version 56.0 and later.

Supported Calls

`create()`, `describeSObjects()`, `query()`

LoyaltyPgmMbrPromEligView isn't available in the Object Manager and can't be customized.

Fields

Field	Details
CurrPromotionLiabilityAmt	Type currency Properties Create, Filter, Nillable, Sort Description The current liability amount of the related promotion.
CurrencyIsoCode	Type picklist Properties Create, Defaulted on create, Filter, Group, Restricted picklist, Sort Description The currency ISO code of the promotion. This field is only available in orgs where multiple currencies are enabled.

Field	Details
	Possible values are: • USD—U.S. Dollar The default value is <code>USD</code>.
CurrentPromotionPointsCount	Type int Properties Create, Filter, Group, Nillable, Sort Description The liability amount incurred for the related promotion till date.
DefaultVoucherDefinitionId	Type reference Properties Create, Filter, Group, Nillable, Sort Description The voucher definition that's used as the default in the related promotion. This field is a relationship field. Relationship Name DefaultVoucherDefinition Relationship Type Lookup Refers To VoucherDefinition
Description	Type textarea Properties Create, Filter, Nillable, Sort Description The description of the LoyaltyPgmMbrPromEligView record.
DoesPromHavePtWithFctr	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort Description Indicates whether the related promotion has a point with factor. The default value is <code>false</code>.

Field	Details
EndDate	Type date Properties Create, Filter, Group, Nillable, Sort Description The end date of the related promotion.
EndTime	Type dateTime Properties Create, Filter, Nillable, Sort Description End date and time of the promotion.
ExternalId	Type string Properties Create, Filter, Group, Nillable, Sort Description A combination of the related loyalty program member ID and the related promotion ID.
FulfillmentAction	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort Description Specifies the fulfillment action of the related promotion. Possible values are: • CreditPoints • IssueVoucher
IsMbrEligBsdCmpnOrSgmt	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort Description Indicates if the member who is eligible for a promotion is part of a campaign or segment. The default value is <code>false</code>.
IsPromotionEnrollmentRqr	Type boolean

Field	Details
	Properties Create, Defaulted on create, Filter, Group, Sort Description Indicates whether the related promotion requires member enrollment. The default value is <code>false</code>.
IssuedPromotionVoucherCount	Type int Properties Create, Filter, Group, Nillable, Sort Description The number of vouchers issued for the related promotion.
LoyaltyProgramCurrencyId	Type reference Properties Create, Filter, Group, Nillable, Sort Description The loyalty program currency that's used to credit points for the related promotion. This field is a relationship field. Relationship Name LoyaltyProgramCurrency Relationship Type Lookup Refers To LoyaltyProgramCurrency
LoyaltyProgramId	Type reference Properties Create, Filter, Group, Nillable, Sort Description The loyalty program that's associated with the related promotion. This field is a relationship field. Relationship Name LoyaltyProgram Relationship Type Lookup Refers To LoyaltyProgram

Field	Details
LoyaltyProgramMemberId	Type reference Properties Create, Filter, Group, Nillable, Sort Description The loyalty program member associated with the related promotion. This field is a relationship field. Relationship Name LoyaltyProgramMember Relationship Type Lookup Refers To LoyaltyProgramMember
LoyaltyPromotionType	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort Description Specifies the type of the related promotion. Possible values are: • Cumulative • Joint • Standard
MaxPromotionLiabilityAmount	Type currency Properties Create, Filter, Nillable, Sort Description The maximum liability amount that the related promotion can incur.
MaxPromotionVoucherCount	Type int Properties Create, Filter, Group, Nillable, Sort Description The maximum vouchers that can be issued for the related promotion.
MaximumPromotionRewardValue	Type int

Field	Details
	Properties Create, Filter, Group, Nillable, Sort Description The maximum reward value of the related promotion.
MemberEligibilityCategory	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort Description Specifies whether the loyalty program member is eligible for the promotion. Possible values are: • Eligible • EligibleButNotEnrolled • Ineligible
Objective	Type textarea Properties Create, Filter, Nillable, Sort Description The objective of the related promotion.
PromEnrollmentStartDate	Type date Properties Create, Filter, Group, Nillable, Sort Description The date when the enrollment in the related promotion starts.
PromotionCumulativeUsageTgt	Type int Properties Create, Filter, Group, Nillable, Sort Description The cumulative usage target of the related promotion.
PromotionEnrollmentEndDate	Type date Properties Create, Filter, Group, Nillable, Sort

Field	Details
	Description The date when the enrollment in the related promotion ends.
PromotionId	Type reference Properties Create, Filter, Group, Nillable, Sort Description The promotion related to the promotion related to the LoyaltyPgmMbrPromEligView record. This field is a relationship field. Relationship Name Promotion Relationship Type Lookup Refers To Promotion
PromotionImageUrl	Type url Properties Create, Filter, Group, Nillable, Sort Description The URL of the promotion's image location. This field is available in API version 58.0 and later.
PromotionName	Type string Properties Create, Filter, Group, Nillable, Sort Description The name of the related promotion.
PromotionPointFactor	Type double Properties Create, Filter, Nillable, Sort Description The points factor of the related promotion.
PromotionTrxnJournalType	Type picklist

Field	Details
	Properties Create, Filter, Group, Nillable, Restricted picklist, Sort Description The transaction journal type that's associated with the related promotion. Possible values are: • Accrual • Redemption
StartDate	Type date Properties Create, Filter, Group, Sort Description Required. The start date of the related promotion.
StartDateTime	Type dateTime Properties Create, Filter, Nillable, Sort Description The start date and time of the promotion.
TotalPromotionRewardPointsVal	Type int Properties Create, Filter, Group, Nillable, Sort Description The total points credited to the loyalty program member for the related promotion.

LoyaltyPgmPartnerCurrency

Represents a junction between a loyalty program partner and a loyalty program currency. This object is available in API version 55.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
AccrualConversionFactor	Type double Properties Create, Filter, Nillable, Sort, Update Description The factor by which points earned by a loyalty program member for an accrual-type transaction are converted to determine the number of points to be debited from the associated loyalty program partner's points balance or balance amount.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyProgramCurrencyId	Type reference Properties Create, Filter, Group, Sort Description Required. The loyalty program currency associated with the loyalty program partner currency record. This is a relationship field. Relationship Name LoyaltyProgramCurrency Relationship Type Lookup Refers To LoyaltyProgramCurrency

Field	Details
LoyaltyProgramPartnerId	Type reference Properties Create, Filter, Group, Sort Description The loyalty program partner associated with the loyalty program partner currency record. This is a relationship field. Relationship Name LoyaltyProgramPartner Relationship Type Master-detail Refers To LoyaltyProgramPartner
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The name of the loyalty program partner currency.
RedemptionConversionFactor	Type double Properties Create, Filter, Nillable, Sort, Update Description The factor by which points redeemed by a loyalty program member for a redemption-type transaction are converted to determine the number of points or amount to be credited to the associated loyalty program partner's points balance or balance amount.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[LoyaltyPgmPartnerCurrencyHistory](#)

History is available for tracked fields of the object.

LoyaltyPgmPartnerPromotion

Represents a junction between the Promotion and Loyalty Program Partner objects. This object is available in API version 55.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
ContributionPercentage	Type percent Properties Create, Filter, Sort, Update Description Required. The percentage of contribution given by a partner to the promotion.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyProgramPartnerId	Type reference Properties Create, Filter, Group, Nillable, Sort Description The loyalty program partner associated with the loyalty program partner prepaid pack or postpaid balance amount. This is a relationship field. Relationship Name LoyaltyProgramPartner Relationship Type Lookup

Field	Details
	Refers To LoyaltyProgramPartner
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The name of the loyalty program partner promotion.
PromotionId	Type reference Properties Create, Filter, Group, Sort Description The joint promotion in which the loyalty program partner is a part of. This is a relationship field. Relationship Name Promotion Relationship Type Master-detail Refers To Promotion

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[LoyaltyPgmPartnerPromotionEvent](#) on page 252 (API Version 62.0)

Change events are available for the object.

[LoyaltyPgmPartnerPromotionHistory](#)

History is available for tracked fields of the object.

LoyaltyPgmPtnrLdgrSummary

Represents the aggregated ledger used by a loyalty program to credit points to or debit points from a prepaid partner's points pack. This ledger can also be used to add or deduct the monetary value of points from the postpaid partner's bill. This object is available in API version 55.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
ActivityDate	Type date Properties Create, Filter, Group, Sort, Update Description Required. The date on which points that are part of the associated loyalty program partner prepaid pack are debited or credited. For postpaid billing, it is the date when the postpaid amount of the loyalty program partner is debited or credited.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyPgmPtnrPrepaidPackId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The loyalty program partner prepaid pack associated with the loyalty program partner ledger. This is a relationship field. Relationship Name LoyaltyPgmPtnrPrepaidPack Relationship Type Lookup

Field	Details
	Refers To LoyaltyPgmPtnrPrepaidPack
LoyaltyProgramPartnerId	Type reference Properties Create, Filter, Group, Sort Description The loyalty program partner associated with the loyalty program. This is a relationship field. Relationship Name LoyaltyProgramPartner Relationship Type Master-detail Refers To LoyaltyProgramPartner
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The name of the loyalty program partner ledger summary.
Status	Type picklist Properties Create, Defaulted on create, Filter, Group, Restricted picklist, Sort, Update Description Required. Specifies the status of the loyalty program partner ledger. Possible values are: • Pending Partner doesn't have the adequate balance and hence the ledger is pending to be processed. • Processed Partner's ledger is processed. The default value is Pending .
TotalPoints	Type double

Field	Details
	Properties Create, Filter, Sort, Update
	Description Required. The number of points debited from or credited to the associated loyalty program partner's prepaid pack or postpaid bill amount.
Type	Type picklist
	Properties Create, Defaulted on create, Filter, Group, Restricted picklist, Sort, Update
	Description Required. Specifies the type of the loyalty program partner ledger. Possible values are: • Credit • CreditCancellation—Credit Cancellation • Debit • DebitCancellation—Debit Cancellation The default value is Credit.

LoyaltyPgmPtnrPrepaidPack

Represents information about a prepaid set of points purchased by a loyalty program partner. This object is available in API version 55.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
EndDate	Type date
	Properties Create, Filter, Group, Nillable, Sort, Update
	Description The date until when the loyalty program partner prepaid pack is effective.

Field	Details
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyProgramPartnerId	Type reference Properties Create, Filter, Group, Sort Description The loyalty program partner associated with the loyalty program partner prepaid pack. This is a relationship field. Relationship Name LoyaltyProgramPartner Relationship Type Master-detail Refers To LoyaltyProgramPartner
Name	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description The name of the prepaid points pack.
OrderId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The order associated with the loyalty program partner prepaid pack.

Field	Details
	This is a relationship field. Relationship Name Order Relationship Type Lookup Refers To Order
PurchaseDate	Type date Properties Create, Filter, Group, Sort Description Required. The date when the prepaid pack was purchased by the loyalty program partner.
PurchasedPoints	Type double Properties Create, Filter, Sort Description Required. The number of points credited to the associated loyalty program partner when a prepaid pack of points is purchased.
RemainingPoints	Type double Properties Create, Filter, Nillable, Sort, Update Description The number of points remaining in the loyalty program partner's prepaid pack.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[LoyaltyPgmPtnrPrepaidPackHistory](#)

History is available for tracked fields of the object.

LoyaltyProgram

Represents the loyalty program within an organization. All the information related to members, tiers, tier groups, program currencies, vouchers, benefits, partners, promotions, transactions are associated with the program. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
Description	Type string Properties Create, Filter, Nillable, Sort, Update Description Description of the loyalty program.
EscrowPeriod	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description The number of days after which the escrow points are credited to the existing points balance. This field is available in API version 53.0 and later.
IsPrimary	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether a loyalty program is the primary program. The default value is 'false'. This field is available in API version 52.0 and later.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort

Field	Details
	Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
Name	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description Required. The name of the loyalty program. The name must contain at least one alphanumeric character and can't contain special characters such as % and +.
OwnerId	Type reference Properties Create, Defaulted on create, Filter, Group, Sort, Update Description ID of the owner of this record. This is a polymorphic relationship field. Relationship Name Owner Relationship Type Lookup Refers To Group, User
Status	Type picklist Properties Create, Defaulted on create, Filter, Group, Restricted picklist, Sort, Update Description Required. The status of the loyalty program. Possible values are: - Active - Inactive

Field	Details
Type	Type picklist Properties Create, Defaulted on create, Filter, Group, Restricted picklist, Sort, Update Description Specifies the type of loyalty program. Possible values are: • LOYALTY_PROGRAM • REFERRAL_PROGRAM The default value is LOYALTY_PROGRAM.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[LoyaltyProgramChangeEvent](#) on page 252 (API Version 62.0)

Change events are available for the object.

[LoyaltyProgramFeed](#)

Feed tracking is available for the object.

[LoyaltyProgramHistory](#)

History is available for tracked fields of the object.

[LoyaltyProgramOwnerSharingRule](#)

Sharing rules are available for the object.

[LoyaltyProgramShare](#)

Sharing is available for the object.

LoyaltyProgramBadge

Represents information about the badge that's associated with a loyalty program. This object is available in API version 56.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
Description	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The description of the loyalty program badge.
ImageUrl	Type url Properties Create, Filter, Nillable, Sort, Update Description The URL of the image location for the loyalty program badge.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyProgramId	Type reference Properties Create, Filter, Group, Sort Description The loyalty program that's associated with the badge. This field is a relationship field. Relationship Name LoyaltyProgram Relationship Type Master-detail

Field	Details
	Refers To LoyaltyProgram
Name	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description The name of the loyalty program badge.
StartDate	Type date Properties Create, Filter, Group, Sort, Update Description Required. The date from when the loyalty program badge can be assigned to members.
Status	Type picklist Properties Create, Filter, Group, Restricted picklist, Sort, Update Description Required. Specifies the status of the loyalty program badge. Possible values are: • Active • Draft • Inactive
ValidityDuration	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description Specifies the unit of measure for the validity period.
ValidityDurationUnit	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The number of days or months in the validity period.

Field	Details
	Possible values are: • Days • Months
ValidityEndDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The date after which the badge is no longer assigned to members. Select a value for this field if the badge's validity type is <code>FixedDate</code>.
ValidityType	Type picklist Properties Create, Filter, Group, Restricted picklist, Sort, Update Description Required. Specifies the model used to determine the duration for which the badge is valid for a member. Possible values are: • <code>FixedDate</code>—Fixed Date • <code>Period</code>

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[LoyaltyProgramBadgeChangeEvent](#) on page 252 (API Version 62.0)

Change events are available for the object.

[LoyaltyProgramBadgeHistory](#)

History is available for tracked fields of the object.

LoyaltyProgramCurrency

Represents qualifying and non-qualifying currencies associated with the program. Qualifying currency represents the engagement of a member with the program and it's associated with a tier group. Non-qualifying currency represent the points that the member earns for redemption. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
CostperUnit	Type currency Properties Create, Filter, Nillable, Sort, Update Description The cost for each currency associated with a loyalty program.
CurrencyType	Type picklist Properties Create, Defaulted on create, Filter, Group, Restricted picklist, Sort Description Required. Type of the loyalty program currency. Possible values are: • NonQualifying • Qualifying
DoesAllowLedgerTracing	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether points credited by accrual transactions and debited by redemption transactions for a fixed-model non-qualifying currency is traced (true) or not (false). This field is available in API version 61.0 and later. The default value is <code>false</code>.
DoesAllowNegativeBalance	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether a non-qualifying currency allows members to hold a negative point balance. This field is available in API version 61.0 and later.

Field	Details
	The default value is <code>false</code> .
ExpiryModel	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The model evaluates the type of expiration of points based on the fixed date or customer's activity. Possible values are: • Activity • Fixed
ExpiryPeriodFrequency	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description The frequency type to evaluate the expiration of currency.
ExpiryPeriodUnit	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description Unit of the period in which currency expiration is done. Possible values are: • Days • Months • Years
ExtendExpiration	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The field decides if the expiration date is changed on the exact day or the last day of the month. Possible values are: • None

Field	Details
	TillEndOfMonth
HasEscrow	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether points can be escrowed for the loyalty program currency. This field is available in API version 53.0 and later. The default value is <code>false</code>.
IsActive	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Required. Indicates whether the loyalty program currency is active.
IsPrimary	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether the loyalty program currency is primary.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyProgramId	Type reference

Field	Details
	Properties Create, Filter, Group, Sort Description The ID of the loyalty program. This is a relationship field. Relationship Name LoyaltyProgram Relationship Type Master-detail Refers To LoyaltyProgram
LoyaltyTierGroupId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The ID of the loyalty tier group. This is a relationship field. Relationship Name LoyaltyTierGroup Relationship Type Lookup Refers To LoyaltyTierGroup
Name	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description The name of the loyalty program currency.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[LoyaltyProgramCurrencyChangeEvent](#) on page 252 (API Version 62.0)

Change events are available for the object.

LoyaltyProgramCurrencyTier

Represents a junction between non-qualifying loyalty program currency and loyalty tier. This object is available in API version 61.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Special Access Rules

This object is available only if the Loyalty Management - Growth or the Loyalty Management - Advanced license is enabled in your org.

Fields

Field	Details
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description Most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description Most recent date on which a user viewed this record.
LoyaltyProgramCurrencyId	Type reference Properties Create, Filter, Group, Sort Description Non-qualifying loyalty program currency associated with the loyalty tier. This field is a relationship field. Relationship Name LoyaltyProgramCurrency Relationship Type Master-detail

Field	Details
	Refers To LoyaltyProgramCurrency (the master object)
LoyaltyTierId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description Loyalty tier associated with the loyalty program currency. This field is a relationship field. Relationship Name LoyaltyTier Relationship Type Lookup Refers To LoyaltyTier
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description Autogenerated number of the loyalty program currency tier.
NegativePointsBalanceLimit	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description Maximum negative point balance that members of the loyalty tier can hold for the associated loyalty program currency.

LoyaltyProgramMbrPromotion

Represents details about the promotions to be applied for a loyalty program member. Tracks the enrollment status, promotion status, and progress for a promotion of a member. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
CumulativeUsageCompletePercent	Type percent Properties Filter, Nillable, Sort Description The percentage of the promotion completed by the member.
CumulativeUsageCompleted	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description The amount of the frequency type promotion completed by the member.
CumulativeUsageTarget	Type int Properties Filter, Group, Nillable, Sort Description The target of the frequency type promotion that must be achieved by the member to complete the promotion.
IsAutoEnrolled	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether the loyalty program member is automatically enrolled in the promotion. The default value is <code>false</code>.
IsEnrollmentActive	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update

Field	Details
	Description Indicates whether the enrollment for the promotion is active.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyProgramMemberId	Type reference Properties Create, Filter, Group, Sort Description The loyalty program member associated with the promotion. This is a relationship field. Relationship Name LoyaltyProgramMember Relationship Type Lookup Refers To LoyaltyProgramMember
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description Unique identifier for the loyalty program member promotion.
PromotionId	Type reference

Field	Details
	Properties Create, Filter, Group, Nillable, Sort
	Description The ID of the loyalty program member promotion. This is a relationship field.
	Relationship Name Promotion
	Relationship Type Lookup
	Refers To Promotion

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[LoyaltyProgramMbrPromotionEvent](#)

Change events are available for the object.

[LoyaltyProgramMbrPromotionHistory](#)

History is available for tracked fields of the object.

LoyaltyProgramMember

Represents information about member who belongs to a loyalty program. The member can be individual or corporate type. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
AccountId	Type reference
	Properties Create, Filter, Group, Nillable, Sort, Update

Field	Details
	Description ID of the account with which the loyalty program member is associated. This field is optional for a Group type member. This is a relationship field. Relationship Name Account Relationship Type Lookup Refers To Account
CanReceivePartnerPromotions	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether the member can receive partner promotions for the program (<code>true</code>) or not (<code>false</code>).
CanReceivePromotions	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether the member can receive promotions for the program (<code>true</code>) or not (<code>false</code>).
ContactId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description ID of the contact with which the loyalty program member is associated. This field is optional for a Group type member. This is a relationship field. Relationship Name Contact Relationship Type Lookup Refers To Contact

Field	Details
EnrollmentChannel	Type picklist Properties Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description Channel through which membership is enrolled. Possible values are: • CallCenter • Email • Franchise • Mobile • Partner • POS • Print • Social • Store • Web
EnrollmentDate	Type date Properties Create, Filter, Group, Sort, Update Description Required. Date when the member enrolled to the loyalty program.
GroupCreatedByMemberId	Type reference Properties Create, Filter, Group, Nillable, Sort Description The Individual or Corporate type loyalty program member who created the group or requested for the group to be created. This field is available in API version 53.0 and later. This is a relationship field. Relationship Name GroupCreatedByMember Relationship Type Lookup Refers To LoyaltyProgramMember

Field	Details
GroupName	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The name of the group when the type of membership in the loyalty program is Group. This field is available in API version 53.0 and later.
IsEligibleForTierAssessment	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether the member is eligible for tier assessment. A member is eligible for tier assessment when there is a change in their qualifying points balance. This field is available in API version 58.0 and later. The default value is <code>false</code>.
LastActivityDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The date when the loyalty program member last performed a transaction.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
MemberStatus	Type picklist

Field	Details
	Properties Create, Filter, Group, Sort, Update Description Required. The status of the member in the loyalty program. Possible values are: • Active • Inactive • Merged—This value is available in API version 56.0 and later.
MemberType	Type picklist Properties Create, Defaulted on create, Filter, Group, Restricted picklist, Sort Description Required. The type of membership in the loyalty program. Possible values are: • Corporate • Group • Individual
MembershipEndDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The date on which the membership ends.
MembershipLastRenewalDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The most recent date when the membership was renewed.
MembershipNumber	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description The membership number of the loyalty program.

Field	Details
OwnerId	Type reference Properties Create, Defaulted on create, Filter, Group, Sort, Update Description The ID of the user who owns this record. This is a polymorphic relationship field. Relationship Name Owner Relationship Type Lookup Refers To Group, User
ProgramId	Type reference Properties Create, Filter, Group, Nillable, Sort Description ID of the loyalty program. This is a relationship field. Relationship Name Program Relationship Type Lookup Refers To LoyaltyProgram
ReferralCode	Type string Properties Create, Filter, Group, idLookup, Nillable, Sort, Update Description The code that the member can share with others who want to join the loyalty program. If no referral code is entered during enrollment, an automatic referral code is generated for the member. This field is available in API version 57.0 and later.
ReferredById	Type reference Properties Create, Filter, Group, Nillable, Sort, Update

Field	Details
	Description Reference to Member who referred to the program. This is a relationship field. Relationship Name ReferredBy Relationship Type Lookup Refers To LoyaltyProgramMember
RelatedCorporateMembershipId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The related corporate membership ID of the loyalty program. This is a relationship field. Relationship Name RelatedCorporateMembership Relationship Type Lookup Refers To LoyaltyProgramMember
StatementFrequency	Type picklist Properties Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The frequency of loyalty balance statements. Possible values are: • Monthly • Quarterly
StatementLastGeneratedDate	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The date on which the statement is last generated.

Field	Details
StatementMethod	Type picklist Properties Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The mode of loyalty balance statements. Possible values are: - Email - Mail

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[LoyaltyProgramMemberChangeEvent](#)

Change events are available for the object.

[LoyaltyProgramMemberHistory](#)

History is available for tracked fields of the object.

[LoyaltyProgramMemberOwnerSharingRule](#)

Sharing rules are available for the object.

[LoyaltyProgramMemberShare](#)

Sharing is available for the object.

LoyaltyProgramMemberBadge

Represents a badge that's assigned to a loyalty program member. This object is available in API version 56.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
EndDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update

Field	Details
	Description The date until when the loyalty program badge is assigned to the related loyalty program member.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyProgramBadgeId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The loyalty program badge that's associated with the loyalty program member. This field is a relationship field. Relationship Name LoyaltyProgramBadge Relationship Type Lookup Refers To LoyaltyProgramBadge
LoyaltyProgramMemberId	Type reference Properties Create, Filter, Group, Sort Description The loyalty program member that's associated with the badge. This field is a relationship field. Relationship Name LoyaltyProgramMember

Field	Details
	Relationship Type Lookup Refers To LoyaltyProgramMember
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The autogenerated unique identifier of loyalty program member badge.
Reason	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The reason why the member badge was assigned.
StartDate	Type date Properties Create, Filter, Group, Sort, Update Description Required. The date from when the loyalty program badge is assigned to the related loyalty program member.
Status	Type picklist Properties Create, Filter, Group, Sort, Update Description Required. Specifies the status of the member badge. Possible values are: • Active • Expired

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

LoyaltyProgramMemberBadgeChangeEvent on page 252 (API Version 62.0)

Change events are available for the object.

LoyaltyProgramMemberBadgeHistory

History is available for tracked fields of the object.

LoyaltyProgramMemberCase

Represents a junction between a loyalty program member and a case. This object is available in API version 52.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
CaseId	Type reference Properties Create, Filter, Group, Sort, Update Description Required. The Case record associated with the loyalty program member case. This is a relationship field. This field is unique within your organization. Relationship Name Case Relationship Type Lookup Refers To Case
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort

Field	Details
	Description The most recent date on which a user viewed this record.
LoyaltyProgramMemberId	Type reference Properties Create, Filter, Group, Sort, Update Description Required. The loyalty program member associated with the loyalty program member case. This is a relationship field. Relationship Name LoyaltyProgramMember Relationship Type Lookup Refers To LoyaltyProgramMember
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The name of the loyalty program member case.
OwnerId	Type reference Properties Create, Defaulted on create, Filter, Group, Sort, Update Description The owner of the loyalty program member case. This is a polymorphic relationship field. Relationship Name Owner Relationship Type Lookup Refers To Group, User

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

LoyaltyProgramMemberCaseChangeEvent on page 252 (API Version 62.0)

Change events are available for the object.

LoyaltyProgramMemberCaseHistory

History is available for tracked fields of the object.

LoyaltyProgramMemberCaseOwnerSharingRule

Sharing rules are available for the object.

LoyaltyProgramMemberCaseShare

Sharing is available for the object.

LoyaltyProgramMemberMerge

Represents information about merged loyalty program memberships. This object is available in API version 56.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
Name	Type string

Field	Details
	Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The autogenerated unique identifier of LoyaltyProgramMemberMerge records.
OwnerId	Type reference Properties Create, Defaulted on create, Filter, Group, Sort, Update Description ID of the owner of this record. This field is a polymorphic relationship field. Relationship Name Owner Relationship Type Lookup Refers To Group, User
Reason	Type string Properties Create, Filter, Group, Sort, Update Description Required. The reason for merging the memberships.
SourceMemberId	Type reference Properties Create, Filter, Group, Sort, Update Description Required. The loyalty program member whose membership is merged with the target loyalty program member. This field is a relationship field. Relationship Name SourceMember Relationship Type Lookup Refers To LoyaltyProgramMember

Field	Details
SrcMbrDataBody	Type base64 Properties Create, Nillable, Update Description The details of the source loyalty program member, such as transaction journals, ledgers, vouchers, tiers, and currencies.
SrcMbrDataContentType	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description Specifies the format in which the source member data is stored. Possible values are: • application/html • application/java-archive • application/javascript • application/json • application/msword • application/octet-stream • application/octet-stream;type=unknown • application/opx • application/pdf • application/postscript • application/rtf • application/vnd.box.notes • application/vnd.google-apps.document • application/vnd.google-apps.drawing • application/vnd.google-apps.form • application/vnd.google-apps.presentation • application/vnd.google-apps.script • application/vnd.google-apps.spreadsheet • application/vnd.ms-excel • application/vnd.ms-excel.sheet.macroEnabled.12 • application/vnd.ms-infopath • application/vnd.ms-powerpoint • application/vnd.ms-powerpoint.presentation.macroEnabled.12 • application/vnd.ms-word.document.macroEnabled.12

Field	Details
	• application/vnd.oasis.opendocument.presentation • application/vnd.oasis.opendocument.spreadsheet • application/vnd.oasis.opendocument.text • application/vnd.openxmlformats-officedocument.presentationml.presentation • application/vnd.openxmlformats-officedocument.presentationml.slideshow • application/vnd.openxmlformats-officedocument.presentationml.template • application/vnd.openxmlformats-officedocument.spreadsheetml.sheet • application/vnd.openxmlformats-officedocument.spreadsheetml.template • application/vnd.openxmlformats-officedocument.wordprocessingml.document • application/vnd.openxmlformats-officedocument.wordprocessingml.template • application/vnd.visio • application/x-gzip • application/x-java-source • application/x-javascript • application/x-shockwave-flash • application/x-sql • application/x-zip-compressed • application/xhtml+xml • application/xml • application/zip • audio/mp4 • audio/mpeg • audio/ogg • audio/x-aac • audio/x-ms-wma • audio/x-ms-wmv • audio/x-wav • image/bmp • image/gif • image/jpeg • image/jpeg;type=mobile • image/png • image/svg+xml • image/tiff • image/vnd.adobe.photoshop • image/vnd.dwg • image/x-photoshop • message/rfc822

Field	Details
	• model/vnd.usdz+zip • text/css • text/csv • text/html • text/plain • text/rtf • text/snote • text/stypi • text/vnd.salesforce.quip-chat • text/vnd.salesforce.quip-doc • text/vnd.salesforce.quip-sheet • text/vnd.salesforce.quip-slides • text/vnd.salesforce.quip-template • text/vtt • text/webviewhtml • text/x-c • text/x-c++ • text/xml • video/mp4 • video/mpeg • video/ogg • video/quicktime • video/x-m4v • video/x-ms-asf • video/x-msvideo
SrcMbrDataLength	Type int Properties Filter, Group, Nillable, Sort Description The number of characters in the SrcMbrDataBody field.
SrcMbrDataName	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The unique identifier of the data stored in the SrcMbrDataBody field.

Field	Details
Status	Type picklist Properties Create, Filter, Group, Restricted picklist, Sort, Update Description Required. Specifies the status of merge. Possible values are: • Merged • Unmerged
TargetMemberId	Type reference Properties Create, Filter, Group, Sort, Update Description Required. The loyalty program member that the source membership was merged into. This field is a relationship field. Relationship Name TargetMember Relationship Type Lookup Refers To LoyaltyProgramMember

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[LoyaltyProgramMemberMergeShare](#)

Sharing is available for the object.

LoyaltyProgramPartner

Represents partners that are associated with a given loyalty program. Partners can be an accrual partner, a redemption partner, or both. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
AccrualCostperUnit	Type currency Properties Create, Filter, Nillable, Sort, Update Description The cost per point associated with the transactions of accrual partner. This field is available in API version 52.0 and later.
AlertValue	Type double Properties Create, Filter, Nillable, Sort, Update Description The value that's used to send an alert to a partner to indicate the minimum points required when the billing type is prepaid or the maximum outstanding amount when the billing type is postpaid. This field is available in API version 55.0 and later.
BillingType	Type picklist Properties Create, Filter, Group, Restricted picklist, Sort, Update Description Specifies the method in which the loyalty program partner is billed. Possible values are: • Prepaid • Postpaid This field is available in API version 55.0 and later.
CreditedPoints	Type double Properties Create, Filter, Nillable, Sort, Update Description The number of points credited to partners (using prepaid billing) for member redemptions, accrual cancellations, or any custom processes. This field is available in API version 55.0 and later.
DebitedPoints	Type double

Field	Details
	Properties Create, Filter, Nillable, Sort, Update Description The total number of points debited from the partner (using prepaid billing) for member accruals, redemption cancellations, or any custom processes. This field is available in API version 55.0 and later.
IsMemberLinkingAllowed	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether a member is allowed to link with a partner (<code>true</code>) or not (<code>false</code>). When you allow linking, you must enter an effective start date. The default value is <code>false</code>.
LastProcessedDateTime	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The last date and time when the prepaid points balance or the outstanding postpaid bill amount for a partner was calculated. This field is available in API version 55.0 and later.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyPartnerId	Type reference Properties Create, Filter, Group, Sort, Update

Field	Details
	Description Required. Account ID of the loyalty program partner. This is a relationship field. Relationship Name LoyaltyPartner Relationship Type Lookup Refers To Account
LoyaltyProgramId	Type reference Properties Create, Filter, Group, Sort, Update Description Required. ID of the loyalty program. This is a relationship field. Relationship Name LoyaltyProgram Relationship Type Lookup Refers To LoyaltyProgram
MemberLinkingEffEndDate	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The date after which members can't be linked to a partner.
MemberLinkingEffStartDate	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The date from which members can be linked to a partner.
Name	Type string

Field	Details
	Properties Create, Filter, Group, idLookup, Sort, Update Description The name of the loyalty program partner.
OwnerId	Type reference Properties Create, Defaulted on create, Filter, Group, Sort, Update Description The ID of the user who owns this record. This is a polymorphic relationship field. Relationship Name Owner Relationship Type Lookup Refers To Group, User
PartnerCategory	Type picklist Properties Create, Filter, Group, Nillable, Sort, Update Description The category of the program partnership. Possible values are: • Alliance • Bilateral
PartnerIndustry	Type picklist Properties Create, Filter, Group, Sort, Update Description Required. The industry type of the program partner. Picklist values aren't provided for this field and must be added based on the requirements of the organization.
PartnerStatus	Type picklist Properties Create, Defaulted on create, Filter, Group, Restricted picklist, Sort, Update

Field	Details
	Description Required. The status of the partner in the loyalty program. The default value is <code>Active</code>. Possible values are: • <code>Active</code> • <code>Inactive</code> • <code>Pending</code>
PartnerType	Type picklist Properties Create, Filter, Group, Restricted picklist, Sort, Update Description Required. The type of the program partnership. Possible values are: • <code>Accrual</code> • <code>Both</code> • <code>Redemption</code>
PartnershipEndDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The date on which the partnership with the program ends.
PartnershipStartDate	Type date Properties Create, Filter, Group, Sort, Update Description Required. The date on which the partnership with the program starts.
PointsBalance	Type double Properties Filter, Nillable, Sort Description The number of points available with the loyalty program partner using prepaid billing. This field is available in API version 55.0 and later. This is a calculated field.

Field	Details
PostpaidBillAmount	Type currency Properties Create, Filter, Nillable, Sort, Update Description The total amount of all credits minus debits for partner transactions from the partnership start date. A positive value indicates that the partner is liable to the program. A negative value indicates the program is liable to the partner. The total amount that either the partner is liable to pay to the program or the program is liable to pay to the partner. This field is available in API version 55.0 and later.
PurchasedPoints	Type double Properties Create, Filter, Nillable, Sort, Update Description The total number of points that the partner (using prepaid billing) has purchased across all their packs. This field is available in API version 55.0 and later.
RedemptionCostperUnit	Type currency Properties Create, Filter, Nillable, Sort, Update Description The cost per point associated with the transactions of redemption partner. This field is available in API version 52.0 and later.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[LoyaltyProgramPartnerChangeEvent](#) on page 252 (API Version 62.0)

Change events are available for the object.

[LoyaltyProgramPartnerHistory](#) (API version 52.0)

History is available for tracked fields of the object.

[LoyaltyProgramPartnerOwnerSharingRule](#)

Sharing rules are available for the object.

[LoyaltyProgramPartnerShare](#) (API version 52.0)

Sharing is available for the object.

LoyaltyProgramPartnerLedger

Represents the ledger used by a loyalty program to credit points to or debit points from a prepaid partner's points pack or from a postpaid partner's point balance. This ledger can also be used to add or deduct the monetary value of points from the partner's postpaid bill. This object is available in API version 55.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
ActivityDate	Type dateTime Properties Create, Filter, Sort, Update Description Required. The date on which the loyalty program member transacts with the loyalty program partner.
LoyaltyLedgerId	Type reference Properties Create, Filter, Group, Sort, Update Description Required. Specifies the type of the loyalty program partner ledger. This is a relationship field. Relationship Name LoyaltyLedger Relationship Type Lookup Refers To LoyaltyLedger
LoyaltyProgramMemberId	Type reference Properties Create, Filter, Group, Sort, Update Description Required. The loyalty program member that's related to the loyalty program partner ledger.

Field	Details
	This is a relationship field. Relationship Name LoyaltyProgramMember Relationship Type Lookup Refers To LoyaltyProgramMember
LoyaltyProgramPartnerId	Type reference Properties Create, Filter, Group, Sort, Update Description Required. The loyalty program partner that's related to the loyalty program partner ledger. This is a relationship field. Relationship Name LoyaltyProgramPartner Relationship Type Lookup Refers To LoyaltyProgramPartner
Points	Type double Properties Create, Filter, Sort, Update Description Required. The number of points credited to or debited from the loyalty program partner's points balance or postpaid account balance.
PromotionId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The promotion that's related to the loyalty program partner ledger. This is a relationship field. Relationship Name Promotion Relationship Type Lookup

Field	Details
	Refers To Promotion
TransactionJournalId	Type reference Properties Create, Filter, Group, Sort, Update Description Required. The transaction journal that's related to the loyalty program partner ledger. This is a relationship field. Relationship Name TransactionJournal Relationship Type Lookup Refers To TransactionJournal
Type	Type picklist Properties Create, Defaulted on create, Filter, Group, Restricted picklist, Sort, Update Description Required. Specifies the type of the loyalty program partner ledger. Possible values are: • Credit • CreditCancellation—Credit Cancellation • Debit • DebitCancellation—Debit Cancellation The default value is Credit.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

LoyaltyProgramPartnerLedgerChangeEvent

Change events are available for the object.

LoyaltyProgramProcess

Represents information about loyalty program processes that determine how transaction journals are processed. This object is available in API version 54.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
Description	Type textarea Properties Create, Filter, Nillable, Sort, Update Description The description of the loyalty program process.
ExecutionType	Type picklist Properties Create, Defaulted on create, Filter, Group, Restricted picklist, Sort Description Specifies the method used for executing the loyalty program process. Determines how transaction journals are processed. If you select Real Time, incoming transactions are processed as soon as they're created. If you select Batch, then Run Program Process for Transaction Journal action in Flows is used to schedule when pending transaction journals are processed. Possible values are: • Batch • BatchAndRealTime—Both • RealTime The default value is RealTime.
JournalSubTypeId	Type reference Properties Create, Filter, Group, Nillable, Sort Description The ID of the journal subtype record associated with the loyalty program process record. This is a relationship field.

Field	Details
	Relationship Name JournalSubType Relationship Type Lookup Refers To JournalSubType
JournalTypeId	Type reference Properties Create, Filter, Group, Nillable, Sort Description The ID of the journal type record associated with the loyalty program process record. This is a relationship field. Relationship Name JournalType Relationship Type Lookup Refers To JournalType
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyProgramId	Type reference Properties Create, Filter, Group, Sort Description The loyalty program associated with the loyalty program process record.

Field	Details
	This is a relationship field. Relationship Name LoyaltyProgram Relationship Type Lookup Refers To LoyaltyProgram
LoyaltyTierGroup	Type reference Properties Create, Filter, Group, Sort Description The tier group of a program that's associated with the loyalty program process. This field is available in API version 56.0 and later. This is a relationship field. Relationship Name LoyaltyProgram Relationship Type Lookup Refers To LoyaltyProgram
Name	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description Required. The name of the loyalty program process.
OwnerId	Type reference Properties Create, Defaulted on create, Filter, Group, Sort, Update Description The ID of the user who owns this record. This is a polymorphic relationship field. Relationship Name Owner Relationship Type Lookup

Field	Details
	Refers To Group, User
ProcessType	Type picklist Properties Create, Filter, Group, Restricted picklist, Sort Description Required. Specifies the type of loyalty program process. Possible values are: • <code>CancelVoucher</code>—This value is available in API version 56.0 and later. • <code>CreditPointsToMembers</code>—This value is available in API version 56.0 and later. • <code>DebitPointsFromMembers</code>—This value is available in API version 56.0 and later. • <code>EnrollInPromotion</code>—This value is available in API version 56.0 and later. • <code>GetMemberPromotions</code>—This value is available in API version 56.0 and later. • <code>IssueVoucher</code>—This value is available in API version 56.0 and later. • <code>OptOutPromotion</code>—This value is available in API version 58.0 and later. • <code>TierProcessing</code>—This value is available in API version 56.0 and later. • <code>TransactionJournal</code> • <code>UnenrollMember</code>—This value is available in API version 56.0 and later. • <code>UpdateMemberDetails</code>—This value is available in API version 56.0 and later.
Status	Type picklist Properties Create, Defaulted on create, Filter, Group, Restricted picklist, Sort, Update Description Specifies the status of the loyalty program process. Possible values are: • <code>Active</code> • <code>Draft</code> • <code>Inactive</code> The default value is <code>Draft</code> .

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

LoyaltyProgramProcessHistory

History is available for tracked fields of the object.

LoyaltyProgramProcessOwnerSharingRule

Sharing rules are available for the object.

LoyaltyProgramProcessShare

Sharing is available for the object.

LoyaltyProgramWidget

Represents the configuration of a widget that a user can apply to the out-of-the-box widget templates and FlexCards provided by Omnistudio. The configuration helps implement custom UI changes, such as promotion type, promotion tag line, and so on for a user. This object is available in API version 57.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
CustomizedContent	Type textarea Properties Create, Nillable, Update Description The detail about the customization that's applied to a widget. For example, promotion_title: Great Times
Description	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The description of the widget.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.

Field	Details
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The name of the widget.
OwnerId	Type reference Properties Create, Defaulted on create, Filter, Group, Sort, Update Description The owner of the widget. This field is a polymorphic relationship field. Relationship Name Owner Relationship Type Lookup Refers To Group, User
ProgramId	Type reference Properties Create, Filter, Group, Sort Description Required. The identifier of the loyalty program that's associated with the widget. This field is a relationship field. Relationship Name Program Relationship Type Lookup Refers To LoyaltyProgram

Field	Details
PromotionId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The ID of the promotion that the widget is associated with. This field is a relationship field. Relationship Name Promotion Relationship Type Lookup Refers To Promotion
PromotionStageId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The ID of the promotion stage that the widget is associated with. This field is a relationship field. Relationship Name PromotionStage Relationship Type Lookup Refers To PromotionStage
TemplateName	Type string Properties Create, Filter, Group, Sort Description Required. The name of the widget template.
TemplateVersion	Type int Properties Create, Filter, Group, Sort Description Required. The version of the widget template.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[LoyaltyProgramWidgetShare](#)

Sharing is available for the object.

LoyaltyTier

Represents information about tiers of a loyalty program. A tier group has a set of tiers in a sequential order. This object is available in API version 51.0 and later.

You can set up Loyalty Tiers to group the members based on their qualifying points. Qualifying points are awarded to members based on their activities in a loyalty program. The hierarchies are created in the tiers using the sequence number. The qualifying points are used to assess a member's tier for upgrade or downgrade.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
Color	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The color that represents the tier. This field is available in API version 62.0 and later.
Description	Type textarea Properties Create, Filter, Nillable, Sort, Update Description Detailed description of the tier.
EligibleEnrollmentType	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The type of enrollment for which a member is eligible.

Field	Details
	Possible values are: • Free • Paid • SegmentBased • TierBased • Trial
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyProgramId	Type reference Properties Filter, Group, Nillable, Sort Description The ID of the loyalty program associated with the loyalty tier. This field is available in API version 64.0 and later. This field is a relationship field. Relationship Name LoyaltyProgram Refers To LoyaltyProgram
LoyaltyTierGroupId	Type reference Properties Create, Filter, Group, Sort Description ID of the loyalty tier group. This is a relationship field.

Field	Details
	Relationship Name LoyaltyTierGroup
	Relationship Type Lookup
	Refers To LoyaltyTierGroup
MaximumEligibleBalance	Type double
	Properties Create, Filter, Nillable, Sort, Update
	Description The maximum balance that's required for loyalty program members to be eligible for the tier. This field is available in API version 58.0 and later.
MinimumEligibleBalance	Type double
	Properties Create, Filter, Nillable, Sort, Update
	Description The minimum balance that's required for loyalty program members to be eligible for the tier. This field is available in API version 58.0 and later.
Name	Type string
	Properties Create, Filter, Group, idLookup, Sort, Update
	Description The name of the loyalty tier.
SequenceNumber	Type int
	Properties Create, Filter, Group, idLookup, Nillable, Sort, Update
	Description The sequence number indicates the ranking of the tier. The tier with the highest sequence number indicates the highest tier.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[LoyaltyTierChangeEvent](#) on page 252 (API Version 62.0)

Change events are available for the object.

[LoyaltyTierHistory](#)

History is available for tracked fields of the object.

LoyaltyTierBenefit

A junction object that associates the loyalty benefits with the loyalty program tiers. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
BenefitId	Type reference Properties Create, Filter, Group, Sort, Update Description Required. Benefit associated with the loyalty tier benefit. This is a relationship field. Relationship Name Benefit Relationship Type Lookup Refers To Benefit
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.

Field	Details
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyTierId	Type reference Properties Create, Filter, Group, Sort Description Tier associated with the loyalty benefit. This is a relationship field. Relationship Name LoyaltyTier Relationship Type Master-detail Refers To LoyaltyTier
Name	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description The name of the loyalty tier benefit.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[LoyaltyTierBenefitChangeEvent](#) on page 252 (API Version 62.0)

Change events are available for the object.

LoyaltyTierEligibilitySrc

Represents the customer groups that are eligible for club membership. This object is available in API version 64.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyTierId	Type reference Properties Create, Filter, Group, Sort Description The ID of the parent loyalty tier for which eligibility source is associated with. This field is a relationship field. Relationship Name LoyaltyTier Relationship Type Master-detail Refers To LoyaltyTier (the master object)
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description Name of the loyalty tier.

Field	Details
SourceReferenceId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The ID of the source for determining club enrollment eligibility. For example, if a customer is part of a specific campaign, they are eligible for club membership. This field is a polymorphic relationship field. Relationship Name SourceReference Refers To ActionableList, LoyaltyTier

LoyaltyTierGroup

Represents information about tier groups of a loyalty program. Tier group categorizes members based on their loyalty points. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
Category	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The category of the loyalty tier group. Possible values are: • Clubs • Tier_Group - Tier Group
Description	Type textarea

Field	Details
	Properties Create, Filter, Nillable, Sort, Update Description Detailed description of the tier group.
EndDate	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The end date of the loyalty tier group.
ExtendExpiration	Type picklist Properties Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description Required. The field decides if the tier is changed on the exact day or the last day of the month. Possible values are: • MemberEnrollmentAnniversary— Select this option only when TierModel is Anniversary. This change is available from API version 52.0 and later. • None • QualifyingPointsResetDate— This value can only be selected when TierModel is Fixed. This change is available from API version 52.0 and later. • TillEndOfMonth— This value isn't supported in API version 52.0 and later.
IsActive	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Required. Indicates whether the loyalty tier group is active (<code>true</code>) or not (<code>false</code>).
IsHiddenToMembers	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether the loyalty tier group is hidden (<code>true</code>) or not (<code>false</code>).

Field	Details
IsPrimary	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether the loyalty tier group is primary (<code>true</code>) or not (<code>false</code>).
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyProgramId	Type reference Properties Create, Filter, Group, Sort Description ID of the loyalty program. This is a relationship field. Relationship Name LoyaltyProgram Relationship Type Master-detail Refers To LoyaltyProgram
Name	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description The name of the loyalty tier group.

Field	Details
QpResetDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description Required. Start date of the QP reset process.
QpResetFrequency	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description Required. The frequency at which members' qualifying points must be reset.
QpResetPeriod	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description Required. Unit of the period for QP reset. Possible values are: • Days—This value isn't supported in API version 52.0 and later. • Months— This value isn't supported in API version 52.0 and later. • Years
StartDate	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The start date of the loyalty tier group.
TierAssessmentCurrencyId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The ID of the Loyalty Program Currency that's used to assess and assign tiers to members of the loyalty program. Members are assigned tiers based on their points balance for the currency. This field is available in API version 58.0 and later. This field is a relationship field.

Field	Details
	Relationship Name TierAssessmentCurrency Relationship Type Lookup Refers To LoyaltyProgramCurrency
TierModel	Type picklist Properties Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description Required. It's the model that decides the start date of the qualifying period. Possible values are: - Anniversary - Fixed
TierPeriodQuantity	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description Required. The period after which members' tier must be reset.
TierPeriodUnit	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description Required. The frequency type for the tier assessment. Possible values are: - Days—This value isn't supported in API version 52.0 and later. - Months - Years

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[LoyaltyTierGroupChangeEvent](#) on page 252 (API Version 62.0)

Change events are available for the object.

LoyaltyTierMshpFeeOption

Represents the options by which members can pay for tier or club memberships. This object is available in API version 63.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
EffectiveEndDateTime	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The end date and time of the loyalty tier membership.
EffectiveStartDateTime	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The start date and time of the loyalty tier membership.
FeeType	Type picklist Properties Create, Filter, Group, Sort, Update Description Specifies the type of payment for the membership fee.
IsActive	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether the membership is active (true) or not (false).

Field	Details
	The default value is <code>false</code> .
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyProgramCurrencyId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The ID of the non-qualifying loyalty program currency associated with the loyalty tier. This field is a relationship field. Relationship Name LoyaltyProgramCurrency Refers To LoyaltyProgramCurrency
LoyaltyTierId	Type reference Properties Create, Filter, Group, Sort Description The ID of the loyalty tier associated with the membership. This field is a relationship field. Relationship Name LoyaltyTier Relationship Type Master-detail Refers To LoyaltyTier (the master object)

Field	Details
MembershipDurationType	Type picklist Properties Create, Filter, Group, Sort, Update Description The duration of the loyalty tier membership.
MembershipFee	Type currency Properties Create, Filter, Nillable, Sort, Update Description The amount of the loyalty tier membership.
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The name of the tier membership fee option.
Points	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description The points used to pay for the membership fee.

LoyaltyTierPromotion

Represents a junction between a member tier and a loyalty promotion. This object is available in API version 63.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyTierId	Type reference Properties Create, Filter, Group, Sort Description The ID of the loyalty tier associated with the promotion. This field is a relationship field. Relationship Name LoyaltyTier Relationship Type Master-detail Refers To LoyaltyTier (the master object)
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The name of the tier promotion.
PromotionId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update

Field	Details
	Description The ID of the loyalty promotion associated with the tier. This field is a relationship field.
	Relationship Name Promotion
	Refers To Promotion

MemberBenefit

Represents details about a benefit that's assigned to a loyalty program member and has a benefit action that's triggered. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
AccountId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The ID of the account associated with the member benefit. This is a relationship field. Relationship Name Account Relationship Type Lookup Refers To Account
BenefitId	Type reference Properties Create, Filter, Group, Sort, Update

Field	Details
	Description Required. The ID of the benefit associated with the member benefit. This is a relationship field. Relationship Name Benefit Relationship Type Lookup Refers To Benefit
ContactId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The ID of the contact associated with the member benefit. This is a relationship field. Relationship Name Contact Relationship Type Lookup Refers To Contact
Description	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The description of the member benefit.
EndDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The end date of the member benefit.
LastReferencedDate	Type dateTime

Field	Details
	Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
MemberId	Type reference Properties Create, Filter, Group, Sort Description The ID of the member associated with the member benefit. This is a polymorphic relationship field. Relationship Name Member Relationship Type Lookup Refers To LoyaltyProgramMember
Name	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description The name of the member benefit.
PrimaryRefObjectRecord	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The ID of a record created by the member benefit.

Field	Details
PrimaryRefObjectRecordTitle	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The title of a record created by the member benefit.
SourceType	Type picklist Properties Create, Filter, Group, Nillable, Sort, Update Description Required. The source of the benefit. Possible values are: • <code>LoyaltyTierProcessing</code> • <code>Manual</code>
StartDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The start date of the member benefit.
Status	Type picklist Properties Create, Defaulted on create, Filter, Group, Restricted picklist, Sort, Update Description Required. The status of the member benefit. Possible values are: • <code>Expired</code> • <code>ExpiredPendingProcessing</code>—Expired - Awaiting Processing • <code>NewPendingProcessing</code>—New - Awaiting Processing • <code>Processed</code> • <code>UpdatedPendingProcessing</code>—Updated - Awaiting Processing The default value is <code>NewPendingProcessing</code>.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[MemberBenefitChangeEvent](#) on page 252 (API Version 62.0)

Change events are available for the object.

[MemberBenefitHistory](#)

History is available for tracked fields of the object.

NetworkReferencedObject

A junction object that associates the loyalty program with the Experience Cloud site. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description Unique identifier for the Network Referenced Object.

Field	Details
NetworkId	Type reference Properties Create, Filter, Group, Sort, Update Description Required. The Id of the Experience Cloud site associated with the loyalty program. This is a relationship field. Relationship Name Network Relationship Type Lookup Refers To Network
ReferencedObjectId	Type reference Properties Create, Filter, Group, Sort Description The object associated with the Experience Cloud site. This is a polymorphic relationship field. Relationship Name ReferencedObject Relationship Type Lookup Refers To LoyaltyProgram

Promotion

Represents a promotion associated with the loyalty program. It captures the information such as type of the promotion, issue of a voucher or credit points, promotion enrollment, and status of the promotion. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
BusinessUnitId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description ID of the internal organization unit that contains the details of the Marketing Cloud business unit associated with the promotion. This field is a relationship field. Relationship Name BusinessUnit Relationship Type Lookup Refers To InternalOrganizationUnit
CampaignId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The campaign associated with the promotion. This is a relationship field. Relationship Name Campaign Relationship Type Lookup Refers To Campaign
CumulativeUsageTarget	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description The number of times a member must participate or the amount of money spent to qualify for a reward.
CurrentLiabilityAmount	Type currency

Field	Details
	Properties Create, Filter, Nillable, Sort, Update Description The liability amount incurred for the promotion until now. This field is available in API version 52.0 and later.
CurrentPointCount	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description The liability amount incurred for the promotion until now. This field is available in API version 52.0 and later.
DefaultVoucherDefinitionId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The voucher associated with the promotion. This is a relationship field. Relationship Name DefaultVoucherDefinition Relationship Type Lookup Refers To VoucherDefinition
Description	Type textarea Properties Create, Filter, Nillable, Sort, Update Description Description of the promotion.
DisplayName	Type string Properties Create, Filter, Group, Nillable, Sort, Update

Field	Details
	Description The display name of the promotion. This field is available in API version 60.0 and later where Global Promotions Management is enabled.
EndDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description End date of the promotion.
EndTime	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The end date and time of the promotion. This field is available in API version 60.0 and later where Unified Promotions is enabled.
EnrollmentEndDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The last date to enroll in the promotion.
EnrollmentStartDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The first date to enroll in the promotion.
FulfillmentAction	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description Select how a loyalty member is rewarded for fulfilling promotion requirements. Possible values are: • <code>CreditPoints</code>—Credit Points

Field	Details
	• <code>IssueVoucher</code>—Issue Voucher
<code>GameDefinitionId</code>	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description ID of the game definition associated with the promotion. This field is a relationship field. Relationship Name GameDefinition Relationship Type Lookup Refers To GameDefinition
<code>ImageUrl</code>	Type url Properties Create, Filter, Nillable, Sort, Update Description The URL for the promotion image. This field is available in API version 57.0 and later.
<code>IsActive</code>	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether the promotion is active. This field is available in API version 52.0 and later.
<code>IsCloned</code>	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether the promotion has been created by cloning an existing promotion. This field is available in API version 61.0 and later. The default value is <code>false</code>.
<code>IsEnrollmentRequired</code>	Type boolean

Field	Details
	Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether enrollment is required to participate in the promotion.
IsPointsWithFactor	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether the promotion is applied by multiplying points with a factor or by using the fixed value in the TotalRewardPoints field. This field is available in API version 52.0 and later. The default value is <code>False</code>.
IsQuickPromotion	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether the records represent a quick promotion (<code>true</code>) or not (<code>false</code>). This field is available in API version 57.0 and later. The default value is <code>false</code>.
IssuedVoucherCount	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description The number of vouchers issued for the promotion until current date. This field is available in API version 52.0 and later.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime

Field	Details
	Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyProgramCurrencyId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The loyalty program currency associated with the promotion. This is a relationship field. Relationship Name LoyaltyProgramCurrency Relationship Type Lookup Refers To LoyaltyProgramCurrency
LoyaltyProgramId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The loyalty program associated with the promotion. This is a relationship field. Relationship Name LoyaltyProgram Relationship Type Lookup Refers To LoyaltyProgram
LoyaltyPromotionType	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The type of promotion. Possible values are:

Field	Details
	- Cumulative - Standard - Joint This field is available in API version 55.0 and later.
MaximumLiabilityAmount	Type currency Properties Create, Filter, Nillable, Sort, Update Description The maximum liability amount that can be incurred for the promotion.
MaximumRewardValue	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description The total value that can be accumulated for the promotion.
MaximumVoucherCount	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description The maximum number of vouchers that can be issued for the promotion. This field is available in API version 52.0 and later.
Name	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description Unique identifier for the promotion.
Objective	Type textarea Properties Create, Filter, Nillable, Sort, Update Description Business objective of the promotion. For example, launching a new product, keeping customers satisfied, or clearing stock.

Field	Details
OwnerId	Type reference Properties Create, Defaulted on create, Filter, Group, Sort, Update Description The ID of the user who owns this record. This is a polymorphic relationship field. Relationship Name Owner Relationship Type Lookup Refers To Group, User
PointFactor	Type double Properties Create, Filter, Nillable, Sort, Update Description The factor with which points are multiplied or divided when <code>IsPointsWithFactor</code> is set to true. This field is available in API version 52.0 and later.
PriorityNumber	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description The priority number of the promotion. This field is available in API version 60.0 and later, where Global Promotions Management is enabled.
ProductCatalogId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description ID of the product catalog associated with the promotion. This field is available in API version 60.0 and later, where Global Promotions Management is enabled. This field is a relationship field. Relationship Name ProductCatalog

Field	Details
	Relationship Type Lookup Refers To ProductCatalog
PromotionCode	Type string Properties Create, Filter, Group, idLookup, Nillable, Sort, Update Description The unique code of the promotion. This field is available in API version 60.0 and later, where Global Promotions Management is enabled.
PromotionConfiguration	Type textarea Properties Create, Nillable, Update Description The configuration of a quick promotion. This field is available in API version 57.0 and later.
RelatedPromotionId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description ID of the promotion that was cloned to create the promotion. This field is a relationship field. Relationship Name RelatedPromotion Relationship Type Lookup Refers To Promotion
RuleLibraryId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description ID of the promotion that was cloned to create the promotion. This field is available in API version 62.0 and later.

Field	Details
	This field is a relationship field. Relationship Name RelatedPromotion Relationship Type Lookup Refers To Promotion
StartDate	Type date Properties Create, Filter, Group, Sort, Update Description Required. The start date of the promotion.
StartDateTime	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The start date and time of the promotion. This field is available in API version 60.0 and later where Global Promotions Management is enabled.
Status	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The status of the promotion. Possible values are: - Cancelled - Complete - Draft - InProgress
TotalRewardPoints	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description The points awarded on availing the promotion.

Field	Details
TransactionJournalType	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The journal type that has been applied for the promotion. This field is available in API version 52.0 and later. Possible values are: • Accrual • Redemption

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

PromotionChangeEvent

Change events are available for the object.

PromotionFeed

Feed tracking is available for the object.

PromotionHistory

History is available for tracked fields of the object.

PromotionOwnerSharingRule

Sharing rules are available for the object.

PromotionShare

Sharing is available for the object.

PromotionActionableList

Represent of a junction between a promotion and an actionable list. This object is available in API version 62.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `undelete()`, `update()`, `upsert()`

Special Access Rules

The PromotionActionableList object is available in orgs that have the Actionable Segmentation setting enabled.

Fields

Field	Details
ActionableListId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The ID of the actionable list associated with the promotion actionable list. This field is a relationship field. Relationship Name ActionableList Refers To ActionableList
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The autogenerated number of the promotion actionable list.
PromotionId	Type reference Properties Create, Filter, Group, Sort Description The ID of the promotion associated with the promotion actionable list.

Field	Details
	This field is a relationship field.
	Relationship Name Promotion
	Relationship Type Master-detail
	Refers To Promotion (the master object)

PromotionChannel

Represents a promotion's eligible channel. This object is available in API version 60.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Special Access Rules

PromotionLimit is only available in orgs where Global Promotions Management is enabled.

Fields

Field	Details
EndDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The date from which the channel is eligible for the promotion.
EngagementChannelTypeId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description ID of the engagement channel type that the channel belongs to. This field is a relationship field.

Field	Details
	Relationship Name EngagementChannelType Relationship Type Lookup Refers To EngagementChannelType
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The autogenerated unique number of the record.
PromotionId	Type reference Properties Create, Filter, Group, Sort Description ID of the promotion for which the channel is eligible. This field is a relationship field. Relationship Name Promotion Relationship Type Lookup Refers To Promotion

Field	Details
RetailLocationGroupId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description ID of the retail location group that's eligible for the promotion. This field is a relationship field. Relationship Name RetailLocationGroup Relationship Type Lookup Refers To RetailLocationGroup
RetailStoreId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description ID of the retail store that's eligible for the promotion. This field is a relationship field. Relationship Name RetailStore Relationship Type Lookup Refers To RetailStore
StartDate	Type date Properties Create, Filter, Group, Sort, Update Description Required. The date till which the channel is eligible for the promotion.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

PromotionChannelFeed

Feed tracking is available for the object.

PromotionChannelHistory

History is available for tracked fields of the object.

PromotionExecutionEvalGrp

Represents a collection of promotions that are evaluated for execution as a group. This object is available in API version 65.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `undelete()`, `update()`, `upsert()`

Special Access Rules

The PromotionExecutionEvalGrp object is available in orgs that have the Promotion Evaluation and Execution setting enabled.

Fields

Field	Details
Description	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The description of the promotion execution evaluation group.
DiscountType	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The type of discount the promotions in the group offer. Possible values are: • CartDiscount • CrossProductDiscount • EligibleProductDiscount
EvaluationMethod	Type picklist

Field	Details
	Properties Create, Filter, Group, Restricted picklist, Sort, Update Description The method in which the promotions in the promotion group are evaluted for execution. Possible values are: • FirstEligiblePromotion • HighestDiscount • Stacked
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
Name	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description The name of the promotion execution evaluation group.
OwnerId	Type reference Properties Create, Defaulted on create, Filter, Group, Sort, Update Description The ID of the owner of this record. This field is a polymorphic relationship field. Relationship Name Owner Refers To Group, User

Field	Details
PricingProcedureId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The pricing procedure associated with the promotion group. This field is a relationship field. Relationship Name PricingProcedure Refers To ExpressionSet
RuleLibraryId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The rule library associated with the promotion. This field is a relationship field. Relationship Name RuleLibrary Refers To RuleLibrary
Status	Type picklist Properties Create, Filter, Group, Restricted picklist, Sort, Update Description The status of the promotion execution evaluation group. Possible values are: • Active • Deleted • Draft • Inactive

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

PromotionExecutionEvalGrpShare

Sharing is available for the object.

PromotionExecutionEvalGrpOwnerSharingRule

Sharing rules are available for the object.

PromotionExecEvalGrpItem

Represents either a promotion or a promotion group in a promotion execution evaluation group. This object is available in API version 65.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `undelete()`, `update()`, `upsert()`

Special Access Rules

The PromotionExecutionEvalGrp object is available in orgs that have the Promotion Evaluation and Execution setting enabled.

Fields

Field	Details
CurrentItemRecordId	Type reference Properties Create, Filter, Group, Sort, Update Description The ID of promotion or the promotion group that the item record belongs to. This field is a polymorphic relationship field. Relationship Name CurrentItemRecord Refers To Promotion, PromotionExecutionEvalGrp
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.

Field	Details
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The name of the promotion execution evaluation group item.
OwnerId	Type reference Properties Create, Defaulted on create, Filter, Group, Sort, Update Description The ID of the owner of this record. This field is a polymorphic relationship field. Relationship Name Owner Refers To Group, User
ParentPromEvaluationGroupId	Type reference Properties Create, Filter, Group, Sort, Update Description The ID of the parent execution evaluation group This field is a relationship field. Relationship Name ParentPromEvaluationGroup Refers To PromotionExecutionEvalGrp
PreviousItemRecordId	Type reference

Field	Details
	Properties Create, Filter, Group, Nillable, Sort, Update
	Description The ID of the promotion or promotion group that's sequenced before the current item. This field is a polymorphic relationship field.
	Relationship Name PreviousItemRecord
	Refers To Promotion, PromotionExecutionEvalGrp

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

PromotionExecEvalGrpItemShare

Sharing is available for the object.

PromotionExecEvalGrpItemOwnerSharingRule

Sharing rules are available for the object.

PromotionLimit

Represents the limit of a promotion. This object is available in API version 60.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Special Access Rules

PromotionLimit is only available in orgs where Global Promotions Management is enabled.

Fields

Field	Details
Category	Type picklist
	Properties Create, Filter, Group, Restricted picklist, Sort, Update

Field	Details
	Description Required. The category of promotion limit. Possible values are: • CustomerLevel • PromotionLevel
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyProgramCurrencyId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description ID of the loyalty program currency for which the promotion limit is set. Specify a loyalty program currency when the limit type is PointsCredited. This field is a relationship field. Relationship Name LoyaltyProgramCurrency Relationship Type Lookup Refers To LoyaltyProgramCurrency
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The autogenerated unique number of the record.

Field	Details
PromotionId	Type reference Properties Create, Filter, Group, Sort Description ID of the promotion that the limit is for. This field is a relationship field. Relationship Name Promotion Relationship Type Lookup Refers To Promotion
Type	Type picklist Properties Create, Filter, Group, Restricted picklist, Sort, Update Description Required. The type of promotion limit. Possible values are: • CartItemCount • FinancialLiability • OrderCount • PointsCredited • TotalUsage • UsagePerCustomer • VouchersIssued • Milestone
Value	Type double Properties Create, Filter, Sort, Update Description Required. The value of the promotion limit.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

PromotionLimitFeed

Feed tracking is available for the object.

PromotionLimitHistory

History is available for tracked fields of the object.

PromotionLimitUsage

Represents information about the total usage of a promotion limit. This object is available in API version 60.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Special Access Rules

PromotionLimitUsage is only available in orgs where Global Promotions Management is enabled.

Fields

Field	Details
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort

Field	Details
	Description The autogenerated unique number of the record.
PromotionLimitId	Type reference Properties Create, Filter, Group, Sort Description ID of the promotion limit for which the usage is tracked. This field is a relationship field. Relationship Name PromotionLimit Relationship Type Lookup Refers To PromotionLimit
Usage	Type double Properties Create, Filter, Sort, Update Description Required. The usage of the promotion limit till date.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

PromotionLimitUsageFeed

Feed tracking is available for the object.

PromotionLimitUsageHistory

History is available for tracked fields of the object.

PromotionLoyaltyPtnrProd

Represents a relationship between a promotion and a loyalty partner product. It indicates the products or product categories that are applied to a particular promotion. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyPartnerId	Type reference Properties Filter, Group, Nillable, Sort Description The loyalty partner associated with the promotion. This is a relationship field. Relationship Name LoyaltyPartner Relationship Type Lookup Refers To LoyaltyProgramPartner
LoyaltyPartnerProductId	Type reference Properties Create, Filter, Group, Sort, Update Description Required. The loyalty partner product associated with the promotion loyalty partner product.

Field	Details
	This is a relationship field. Relationship Name LoyaltyPartnerProduct Relationship Type Lookup Refers To LoyaltyPartnerProduct
LoyaltyProductId	Type reference Properties Filter, Group, Nillable, Sort Description The loyalty product associated with the promotion. This is a relationship field. Relationship Name LoyaltyProduct Relationship Type Lookup Refers To Product2
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description Unique identifier for the promotion loyalty partner product.
ProductCategoryId	Type reference Properties Filter, Group, Nillable, Sort Description The product category associated with the promotion. This is a relationship field. Relationship Name ProductCategory Relationship Type Lookup

Field	Details
	Refers To ProductCategory
PromotionId	Type reference Properties Create, Filter, Group, Sort Description The promotion associated with the promotion loyalty partner product. This is a relationship field. Relationship Name Promotion Relationship Type Lookup Refers To Promotion

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

PromotionLoyaltyPtnrProdtChangeEvent on page 252 (API Version 62.0)

Change events are available for the object.

PromotionLoyaltyPtnrProdtFeed

Feed tracking is available for the object.

PromotionLoyaltyPtnrProdtHistory

History is available for tracked fields of the object.

PromotionMarketSegment

Represents a relationship between a promotion and Data Cloud segment. It indicates the segments targeted for a particular promotion. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
CurrencyIsoCode	Type picklist Properties Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The currency ISO code of the promotion. This field is only available in orgs where multiple currencies are enabled. Possible values are: • USD—U.S. Dollar The default value is USD.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
MarketSegmentId	Type reference Properties Create, Filter, Group, Sort, Update Description The ID of the market segment associated with the promotion market segment.
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description Unique identifier for the promotion market segment.

Field	Details
PromotionId	Type reference Properties Create, Filter, Group, Sort Description The ID of the promotion associated with the promotion market segment. This is a relationship field. Relationship Name Promotion Relationship Type Lookup Refers To Promotion
PromotionSegmentId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The ID of the related promotion segment. This field is accessible in orgs with a Commerce Cloud license, where users have access to Promotions. The field associates a promotion with its segments. Promotion segments are different from Data Cloud segments. This field is a relationship field. Relationship Name PromotionSegment Relationship Type Lookup Refers To PromotionSegment

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

PromotionMarketSegmentChangeEvent on page 252 (API Version 62.0)

Change events are available for the object.

PromotionMarketSegmentFeed

Feed tracking is available for the object.

PromotionMarketSegmentHistory

History is available for tracked fields of the object.

PromotionPartyUsage

Represents the details of number of times a promotion has been used by a customer. This object is available in API version 60.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Special Access Rules

PromotionPartyUsage is only available in orgs where Global Promotions Management is enabled.

Fields

Field	Details
AccountId	Type reference Properties Filter, Group, Nillable, Sort Description ID of the account associated with the organization that has used the promotion. This field is a relationship field. Relationship Name Account Relationship Type Lookup Refers To Account
AdditionalAttributeInfo	Type textarea Properties Create, Nillable, Update Description Specifies the details of the additional promotional attribute for Contact, Account or a member in JSON format.

Field	Details
ContactId	Type reference Properties Filter, Group, Nillable, Sort Description ID of the contact associated with the person who has used the promotion. This field is a relationship field. Relationship Name Contact Relationship Type Lookup Refers To Contact
IssuedVoucherCount	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description The number of vouchers that have been issued to the customer as the associated promotion's reward.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyProgramMemberId	Type reference Properties Filter, Group, Nillable, Sort

Field	Details
	Description ID of the loyalty program member who has used the promotion. This field is a relationship field. Relationship Name LoyaltyProgramMember Relationship Type Lookup Refers To LoyaltyProgramMember
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The autogenerated unique number of the record.
OrderCount	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description The number of orders for which the customer has applied the promotion.
PartyId	Type reference Properties Create, Filter, Group, Sort Description ID of the person or the organization that has been provided the discount. This field is a polymorphic relationship field. Relationship Name Party Relationship Type Lookup Refers To Account, Contact, LoyaltyProgramMember
PointsCredited	Type double

Field	Details
	Properties Create, Filter, Nillable, Sort, Update Description The number of points that have been credited to a loyalty program member as the associated promotion's reward.
PromotionId	Type reference Properties Create, Filter, Group, Sort, Update Description Required. ID of the promotion for which the customer's usage is tracked. This field is a relationship field. Relationship Name Promotion Relationship Type Lookup Refers To Promotion
Type	Type picklist Properties Filter, Group, Nillable, Restricted picklist, Sort Description The type of customer for whom the promotion usage is tracked. Possible values are: - Account - Contact - LoyaltyProgramMember
UsageCount	Type double Properties Create, Filter, Nillable, Sort, Update Description The number of times a party has used a promotion.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

PromotionPartyUsageChangeEvent

Change events are available for the object.

PromotionPartyUsageFeed

Feed tracking is available for the object.

PromotionPartyUsageHistory

History is available for tracked fields of the object.

PromotionPartyUsageDetail

Represents the amount that's discounted by a promotion for a transaction journal. This object is available in API version 60.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `undelete()`, `update()`, `upsert()`

Special Access Rules

PromotionPartyUsageDetail is only available in orgs where Global Promotions Management is enabled.

Fields

Field	Details
AccountId	Type reference Properties Filter, Group, Nillable, Sort Description ID of the account associated with the organization that has used the promotion. This field is a relationship field. Relationship Name Account Relationship Type Lookup Refers To Account

Field	Details
ContactId	Type reference Properties Filter, Group, Nillable, Sort Description ID of the contact associated with the person who has used the promotion. This field is a relationship field. Relationship Name Contact Relationship Type Lookup Refers To Contact
CouponId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The coupon that's applied as a part of the promotion to provide discounts to the customer. This field is available in API version 66.0 and later. This field is a relationship field. Relationship Name Coupon Refers To Coupon
DiscountAmount	Type currency Properties Create, Filter, Nillable, Sort, Update Description The monetary discount provided to the person or organization who has availed the promotion.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.

Field	Details
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyProgramMemberId	Type reference Properties Filter, Group, Nillable, Sort Description ID of the loyalty program member who has used the promotion. This field is a relationship field. Relationship Name LoyaltyProgramMember Relationship Type Lookup Refers To LoyaltyProgramMember
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The autogenerated unique number of the record.
PartyId	Type reference Properties Create, Filter, Group, Sort Description ID of the person or the organization that has been provided the discount. This field is a polymorphic relationship field. Relationship Name Party Relationship Type Lookup Refers To Account, Contact, LoyaltyProgramMember

Field	Details
PartyType	Type picklist Properties Filter, Group, Nillable, Restricted picklist, Sort Description The type of customer that has been provided the discount. Possible values are: - Account - Contact - LoyaltyProgramMember
PromotionId	Type reference Properties Create, Filter, Group, Sort, Update Description Required. The promotion that has provided the discount to the customer. This field is a relationship field. Relationship Name Promotion Relationship Type Lookup Refers To Promotion
TransactionJournalId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description ID of the transaction journal that represents the customer activity for which the promotion has been applied. This field is a relationship field. Relationship Name TransactionJournal Relationship Type Lookup Refers To TransactionJournal

PromotionProduct

Represents a junction between a promotion and an product that's eligible for the promotion. This object is available in API version 60.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Special Access Rules

PromotionProduct is available in orgs where Global Promotions Management is enabled.

Fields

Field	Details
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The autogenerated unique number of the record.
ProductId	Type reference Properties Create, Filter, Group, Sort, Update

Field	Details
	Description Required. ID of the product that's eligible for the associated promotion. This field is a relationship field. Relationship Name Product Relationship Type Lookup Refers To Product2
PromotionId	Type reference Properties Create, Filter, Group, Sort Description The promotion for which eligible product is selected. This field is a relationship field. Relationship Name Promotion Relationship Type Lookup Refers To Promotion

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

PromotionProductFeed

Feed tracking is available for the object.

PromotionProductHistory

History is available for tracked fields of the object.

PromotionProductCategory

Represents a junction between a promotion and a product category that's eligible for the promotion. This object is available in API version 60.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Special Access Rules

PromotionProductCategory is only available in orgs where Global Promotions Management is enabled.

Fields

Field	Details
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The autogenerated unique number of the record.
ProductCategoryId	Type reference Properties Create, Filter, Group, Sort, Update Description Required. of the product category that's eligible for the associated promotion. This field is a relationship field. Relationship Name ProductCategory

Field	Details
	Relationship Type Lookup Refers To ProductCategory
PromotionId	Type reference Properties Create, Filter, Group, Sort Description The promotion for which eligible product category is selected. This field is a relationship field. Relationship Name Promotion Relationship Type Lookup Refers To Promotion

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

PromotionProductCategoryFeed

Feed tracking is available for the object.

PromotionProductCategoryHistory

History is available for tracked fields of the object.

PromotionRuleTemplate

Represents the configuration of a promotion rule template. This object is available in API version 60.0 and later.

Supported Calls

`describeSObjects()`, `query()`

Special Access Rules

PromotionRuleTemplate is only available in orgs where Global Promotions Management is enabled. The object isn't available in the Object Manager and can't be customized.

Fields

Field	Details
ApiName	Type string Properties Filter, Group, Nillable, Sort Description The API name of the promotion rule template.
Category	Type string Properties Filter, Group, Nillable, Sort Description The category of the promotion rule template.
Description	Type textarea Properties Filter, Nillable, Sort Description The description of the promotion rule template.
ExternalId	Type string Properties Filter, Group, Nillable, Sort Description Reserved for internal use.
IconUrl	Type url Properties Filter, Group, Nillable, Sort Description The URL of the promotion rule template's icon.
JournalTypeId	Type reference Properties Filter, Group, Nillable, Sort

Field	Details
	Description The ID of the journal type that represents the type of customer activity the promotion rule template is applicable for. This field is a relationship field. Relationship Name JournalType Relationship Type Lookup Refers To JournalType
Label	Type string Properties Filter, Group, Nillable, Sort Description The name of the promotion rule template.
RuleConfiguration	Type textarea Properties Nillable Description The promotion rule configuration in JSON format.
RuleRewardConfiguration	Type textarea Properties Nillable Description The rule reward configuration details in JSON format, which are applicable for the promotion rule template.

PromotionStage

Represents a stage when the promotion sends an email to the promotion's target audience. With Global Promotions Management, a promotion launch email is sent after the promotion is activated. This object is available in API version 60.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Special Access Rules

PromotionStage is only available in orgs where Global Promotions Management is enabled.

Fields

Field	Details
ApiName	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description Required. The unique name of the promotion stage.
Configuration	Type textarea Properties Create, Nillable, Update Description The configuration of the promotion stage.
Description	Type textarea Properties Create, Filter, Nillable, Sort, Update Description The description of the promotion stage.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime

Field	Details
	Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
Name	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description The name of the promotion stage.
PromotionId	Type reference Properties Create, Filter, Group, Sort Description The ID of the promotion that the stage belongs to. This field is a relationship field. Relationship Name Promotion Relationship Type Lookup Refers To Promotion
SequenceNumber	Type int Properties Create, Filter, Group, Sort, Update Description Required. The sequence number of the stage in a promotion. The sequence number of the Promotion Launched stage is 1.
Type	Type picklist Properties Create, Filter, Group, Restricted picklist, Sort, Update Description Required. The type of promotion stage. Possible values are:

Field	Details
	Promotion Launched

PromotionStageEmailTemplate

Represents a junction between a promotion stage and an email template. With Global Promotions Management, a promotion's launch email is associated with the promotion's Promotion Launched stage. You can customize the promotion launch email based on your company's branding. This object is available in API version 60.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Special Access Rules

PromotionStageEmailTemplate is only available in orgs where Global Promotions Management is enabled.

Fields

Field	Details
EmailCustomConfiguration	Type textarea Properties Create, Nillable, Update Description The customized content of an email template.
EmailTemplateId	Type reference Properties Create, Filter, Group, Sort, Update Description Required. The ID of the email template associated with the promotion stage. This field is a relationship field. Relationship Name EmailTemplate Relationship Type Lookup Refers To EmailTemplate

Field	Details
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The unique number of the promotion stage email template.
PromotionStageId	Type reference Properties Create, Filter, Group, Sort Description The ID of the promotion stage that the associated email template belongs to. This field is a relationship field. Relationship Name PromotionStage Relationship Type Lookup Refers To PromotionStage

TransactionJournal

Represents information about loyalty transactions that track activities, member behavior, and adjustments. This object is available in API version 51.0 and later.



Warning: You can add up to 50 custom fields in the TransactionJournal object.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
ActivityDate	Type dateTime Properties Create, Filter, Sort, Update Description Required. The date of the transaction.
BookedFareClass	Type picklist Properties Create, Filter, Group, Nillable, Sort, Update Description The class of ticket service that the member booked.
BookedRoomType	Type picklist Properties Create, Filter, Group, Nillable, Sort, Update Description The type of room the member booked.
BookingDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The date of the member booked a reservation or purchased goods or services.
Brand	Type string Properties Create, Filter, Group, Nillable, Sort, Update

Field	Details
	Description The hotel chain where the member stayed or the brand of the purchased or rented goods or services.
Channel	Type picklist Properties Create, Filter, Group, Nillable, Sort, Update Description The channel the member used for transactions.
CouponId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The coupon that's applied as a part of a promotion to provide discounts on the transaction. This field is available in API version 66.0 and later. This field is a relationship field. Relationship Name Coupon Refers To Coupon
CurrencyIsoCode	Type picklist Properties Create, Defaulted on create, Filter, Group, Restricted picklist, Sort, Update Description The three-letter ISO 4217 currency code of the monetary currency in which the transaction took place. Possible values are: • USD The default value is USD.
Destination	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The ending location for a member's trip or the place where an order shipped to.

Field	Details
Distance	Type double Properties Create, Filter, Nillable, Sort, Update Description The miles or kilometers traveled by a member or an order's shipping distance.
EndDate	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The end date associated with the transaction.
EngagementChannelTypeId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The ID of the engagement channel that the customer used for the purchase. This field is available in API version 61.0 and later. This field is a relationship field. Relationship Name EngagementChannelType Relationship Type Lookup Refers To EngagementChannelType
ErrorDescription	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The description of the error.
EscrowPointsCreditDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update

Field	Details
	Description The date on which the escrow points are credited to the associated loyalty program member's point balance. This field is available in API version 53.0 and later.
Establishment	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The unique identifier for the location where the member stayed or purchased goods or services.
ExternalTransactionNumber	Type string Properties Create, Filter, Group, idLookup, Nillable, Sort, Update Description The transaction number from the external systems.
FlightNumber	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The airline flight number the member booked.
Industry	Type picklist Properties Create, Filter, Group, Nillable, Sort, Update Description Specifies the type of the industry.
InvoiceDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The date the invoice was generated.
IsParentTransaction	Type boolean

Field	Details
	Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether the transaction journal represents the primary activity within a group of activities. This field is available in API version 60.0 and later. The default value is <code>false</code>.
JournalDate	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The date of the journal.
JournalReason	Type picklist Properties Create, Filter, Group, Nillable, Sort, Update Description Specifies the reason for the action that created a transaction journal. This field can store up to 999 values. After the field reaches 999 values, rename existing values that you no longer use instead of deleting values and adding new ones. Possible values are: • Accrual Cancellation Requested by Member • Order Canceled • Product Returned • Product or Service Unavailable • Redemption Cancellation Requested by Member • Resolve Disputed Points • Service Recovery
JournalSubTypeId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The subtype of journal transaction. This is a relationship field. Relationship Name JournalSubType

Field	Details
	Relationship Type Lookup Refers To JournalSubType
JournalTypeId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The type of journal transaction. This is a relationship field. Relationship Name JournalType Relationship Type Lookup Refers To JournalType
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyProgramId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The loyalty program associated with the transaction. This is a relationship field.

Field	Details
	Relationship Name LoyaltyProgram Relationship Type Lookup Refers To LoyaltyProgram
MemberId	Type reference Properties Create, Filter, Group, Sort, Update Description The member associated with the transaction. This field is a polymorphic relationship field. Relationship Name Member Relationship Type Lookup Refers To LoyaltyProgramMember
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The autogenerated number of the transaction journal.
OrderId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The order associated with the transaction. This is a relationship field. Relationship Name Order Relationship Type Lookup Refers To Order

Field	Details
OrderItemId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The order items associated with the transaction. This is a relationship field. Relationship Name OrderItem Relationship Type Lookup Refers To OrderItem
Origin	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The starting location for a member's trip or the place where an order shipped from.
PartnerId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description Participating partners where members can add or redeem points. This is a relationship field. Relationship Name Partner Relationship Type Lookup Refers To LoyaltyProgramPartner
PaymentMethod	Type picklist Properties Create, Filter, Group, Nillable, Sort, Update Description The member's payment method.

Field	Details
ProductCategoryId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The category of the product associated with the transaction. This is a relationship field. Relationship Name ProductCategory Relationship Type Lookup Refers To ProductCategory
ProductId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The product associated with the transaction. This is a relationship field. Relationship Name Product Relationship Type Lookup Refers To Product2
PromotionId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The promotion associated with the transaction journal. This field is available in API version 52.0 and later. This is a relationship field. Relationship Name Promotion Relationship Type Lookup

Field	Details
	Refers To Promotion
Quantity	Type double Properties Create, Filter, Nillable, Sort, Update Description The length of the booked stay or the order quantity.
QuantityUnitOfMeasureId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The unit of measure for the booked stay or the order quantity. This is a relationship field. Relationship Name QuantityUnitOfMeasure Relationship Type Lookup Refers To UnitOfMeasure
RedeemedPointsExpirationInformation	Type textarea Properties Create, Nillable, Update Description The details of the fixed type non-qualifying points that are redeemed in a transaction journal and their corresponding expiration dates. This field is available in API version 55.0 and later. RedeemedPointsExpirationInformation isn't able in the Object Manager and can't be customized.
ReferredMemberId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The name of the person the member referred. This is a relationship field.

Field	Details
	Relationship Name ReferredMember Relationship Type Lookup Refers To LoyaltyProgramMember
ReferredPartyId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The ID of the contact or account associated with the transaction journal. The contact ID belongs to the person whose activity the transaction journal represents. This field is available in API version 60.0 and later. Relationship Name ReferredParty Relationship Type Lookup Refers To Account, Contact
RelatedCorporateMembershipId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The corporate membership associated with the transaction. This is a relationship field. Relationship Name RelatedCorporateMembership Relationship Type Lookup Refers To LoyaltyProgramMember
RelatedJournalId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update

Field	Details
	Description The related journal entry associated with this transaction. This is a relationship field. Relationship Name RelatedJournal Relationship Type Lookup Refers To TransactionJournal
StartDate	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The start date associated with the transaction.
Status	Type picklist Properties Create, Defaulted on create, Filter, Group, Restricted picklist, Sort, Update Description Required. The status of the transaction. Possible values are: - Cancelled - Error - Pending - Processed
TotalPromotionDiscount	Type currency Properties Create, Filter, Nillable, Sort, Update Description The total discount provided by the associated promotion for the activity that the transaction journal represents. This field is available in API version 61.0 and later.
TransactionAmount	Type currency Properties Create, Filter, Nillable, Sort, Update

Field	Details
	Description For reservations, the total fees or prepaid balance. For goods and services, the total purchase price.
TransactionLocation	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The location where the loyalty member purchased goods or services.
TransactionSourceLocationId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The ID of the retail store where the activity that the transaction journal represents has taken place. This field is available in API version 60.0 and later. This field is a relationship field. Relationship Name TransactionSourceLocation Relationship Type Lookup Refers To RetailStore
TraveledFareClass	Type picklist Properties Create, Filter, Group, Nillable, Sort, Update Description The class of ticket service that the member traveled.
Unique Identifier	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The unique identifier of the transaction journal.

Field	Details
UsageType	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort Description The type of usage. Possible values are: • <code>Loyalty</code> • <code>Rebates</code> • <code>UnifiedPromotions</code>—This value is available in API version 60.0 and later.
VoucherCode	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The voucher code generated by an external system.

Usage

You can query Transaction Journal records in your org using a combination of these indexed fields:

- `ActivityDate`
- `ExternalTransactionNumber`
- `Id`
- `JournalTypeId`
- `LoyaltyProgramId`
- `MemberId`
- `OrderId`
- `RelatedJournalId`
- `UsageType`

The TransactionJournal object in most orgs can store millions of records. A limited set of TransactionJournal fields are indexed to help optimize operations on the TransactionJournal object.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[TransactionJournalChangeEvent](#) on page 252 (API Version 63.0)

Change events are available for the object.

Voucher

Represents details about a voucher that's issued to a loyalty program member. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
AccountId	Type reference Properties Create, Filter, Group, Nillable, Sort Description The ID of the account to which the voucher is issued. This field is available in API version 62.0 and later. If you want to issue vouchers to accounts, set up a custom process based on the requirements of your company. This field is a relationship field. Relationship Name Account Refers To Account
BenefitId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The benefit associated with the voucher. This is a relationship field. Relationship Name Benefit Relationship Type Lookup Refers To Benefit

Field	Details
ContactId	Type reference Properties Create, Filter, Group, Nillable, Sort Description The contact associated with the member. This is a relationship field. Relationship Name Contact Relationship Type Lookup Refers To Contact
DiscountPercent	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description The discount percentage associated with the voucher.
EffectiveDate	Type date Properties Create, Filter, Group, Sort, Update Description Required. The date the voucher goes into effect.
EffectiveDateTime	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The date and time from when the voucher is available for redemption. This field is available in API version 62.0 and later.
ExpirationDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update

Field	Details
	Description The date the voucher expires.
ExpirationDateTime	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The date and time when the voucher expires and is no longer available for redemption. This field is available in API version 62.0 and later.
FaceValue	Type currency Properties Create, Filter, Nillable, Sort, Update Description The value of the voucher.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyProgramMemberId	Type reference Properties Create, Filter, Group, Nillable, Sort Description The member who has been issued the voucher. This is a relationship field. Relationship Name LoyaltyProgramMember

Field	Details
	Relationship Type Lookup Refers To LoyaltyProgramMember
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The autogenerated number of the voucher.
OwnerId	Type reference Properties Create, Defaulted on create, Filter, Group, Sort, Update Description The ID of the user who owns this record. This is a polymorphic relationship field. Relationship Name Owner Relationship Type Lookup Refers To Group, User
PromotionId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The promotion associated with the voucher. This is a relationship field. Relationship Name Promotion Relationship Type Lookup Refers To Promotion

Field	Details
Reason	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The reason why the voucher was issued.
RedeemedValue	Type currency Properties Create, Filter, Nillable, Sort, Update Description The voucher balance that was redeemed. This field is available in API version 53.0 and later.
RemainingValue	Type currency Properties Filter, Nillable, Sort Description The remaining value of the voucher after partial redemption. This field is available in API version 53.0 and later.
ReservationKey	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The key that's used to redeem a reserved voucher. This field is available in API version 62.0 and later.
ReservedDateTime	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The date and time when the voucher is reserved for redemption. This field is available in API version 62.0 and later.
ReservedValue	Type currency Properties Create, Filter, Nillable, Sort, Update

Field	Details
	Description The value of the voucher reserved by the customer. This field is available in API version 62.0 and later.
Status	Type picklist Properties Create, Defaulted on create, Filter, Group, Restricted picklist, Sort, Update Description Required. The status of the voucher. Possible values are: - Cancelled - Expired - Issued - Redeemed
TransactionJournalId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The transaction journal associated with the member voucher when the voucher results from accrual of points. This is a relationship field. Relationship Name TransactionJournal Relationship Type Lookup Refers To TransactionJournal
UseDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The date the voucher was used.
VoucherCode	Type string

Field	Details
	Properties Create, Filter, Group, Nillable, Sort, Update Description The voucher identifier generated by an external system.
VoucherDefinitionId	Type reference Properties Create, Filter, Group, Nillable, Sort Description The loyalty voucher definition associated with the voucher. This is a relationship field. Relationship Name VoucherDefinition Relationship Type Lookup Refers To VoucherDefinition

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

VoucherChangeEvent on page 252 (API Version 63.0)

Change events are available for the object.

VoucherHistory (API version 52.0)

History is available for tracked fields of the object.

VoucherShare (API version 52.0)

Sharing is available for the object.

VoucherDefinition

Represents details about a voucher definition that's associated with a loyalty program. This object is available in API version 51.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
AutogeneratedCodeLength	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description The length of the auto-generated voucher code. If no value is specified, the default is eight digits. This field is available in API version 57.0 and later.
CodePrefix	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The prefix of the auto-generated voucher code. This field is available in API version 57.0 and later.
CodeSuffix	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The suffix of the auto-generated voucher code. This field is available in API version 57.0 and later.
CostPerVoucher	Type currency Properties Create, Filter, Nillable, Sort, Update Description The liability incurred for each voucher that's issued. This field is available in API version 52.0 and later.
Description	Type textarea Properties Create, Filter, Nillable, Sort, Update Description The description of the voucher.

Field	Details
DiscountPercent	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description The discount percentage associated with the voucher.
EffectiveDate	Type date Properties Create, Filter, Group, Sort, Update Description Required. The date on which the voucher becomes valid.
EffectiveDateTime	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The date and time from when the vouchers issued by the voucher definition can be redeemed by customers. This field is available in API version 62.0 and later.
ExpirationDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The date the voucher expires. The expiration date is only valid for the fixed date model.
ExpirationDateTime	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description The date and time on which the vouchers issued by the voucher definition are no longer available for customers to redeem. This field is available in API version 62.0 and later.
ExpirationPeriod	Type int Properties Create, Filter, Group, Nillable, Sort, Update

Field	Details
	Description The number of days or months in the expiration period.
ExpirationPeriodUnit	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The unit for the expiration period. Possible values are: - Days - Months
ExpirationType	Type picklist Properties Create, Filter, Group, Restricted picklist, Sort, Update Description Required. The type of expiration for the voucher. Possible values are: - FixedDate - Period
FaceValue	Type currency Properties Create, Filter, Nillable, Sort, Update Description The value of the voucher.
HasTimeBasedVoucherPeriod	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort Description Indicates whether vouchers issued using the definition are effective and expire at a specific date and time. This field is available in API version 62.0 and later. The default value is <code>false</code> .
ImageUrl	Type url

Field	Details
	Properties Create, Filter, Nillable, Sort, Update Description The URL for the voucher definition image. This field is available in API version 57.0 and later.
IsActive	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Required. Indicates whether the voucher is active (<code>true</code>) or not (<code>false</code>). The default value is <code>false</code> .
IsPartialRedeemable	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether the voucher can be partially redeemed (<code>true</code>) or not (<code>false</code>). This field is available in API version 53.0 and later. The default value is <code>false</code> .
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user referenced this record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The most recent date on which a user viewed this record.
LoyaltyProgramId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update

Field	Details
	Description The ID of the loyalty program. This is a relationship field. Relationship Name LoyaltyProgram Relationship Type Lookup Refers To LoyaltyProgram
Name	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description The name of the voucher.
OwnerId	Type reference Properties Create, Defaulted on create, Filter, Group, Sort, Update Description The ID of the user who owns this record. This is a polymorphic relationship field. Relationship Name Owner Relationship Type Lookup Refers To Group, User
PartnerAccountId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The partner associated with the voucher. This is a relationship field. Relationship Name PartnerAccount

Field	Details
	Relationship Type Lookup
	Refers To Account
ProductCategoryDerivedFieldId	Type reference
	Properties Filter, Group, Nillable, Sort
	Description A look up to show product category in voucher definition. This is a relationship field.
	Relationship Name ProductCategoryDerivedField
	Relationship Type Lookup
	Refers To ProductCategory
ProductDerivedFieldId	Type reference
	Properties Filter, Group, Nillable, Sort
	Description A look up to show product in voucher definition. This is a relationship field.
	Relationship Name ProductDerivedField
	Relationship Type Lookup
	Refers To Product2
ProductId	Type reference
	Properties Create, Filter, Group, Nillable, Sort, Update
	Description The product associated with the voucher. This is a polymorphic relationship field.

Field	Details
	Relationship Name Product
	Relationship Type Lookup
	Refers To Product2, ProductCategory
Type	Type picklist
	Properties Create, Filter, Group, Restricted picklist, Sort, Update
	Description Required. The type of voucher. Possible values are: • <code>DiscountPercentage</code>—Discount Percentage • <code>FixedValue</code>—Fixed Value • <code>ProductOrService</code>—Product or Service

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

[VoucherDefinitionChangeEvent](#) on page 252 (API Version 62.0)

Change events are available for the object.

[VoucherDefinitionHistory](#)

History is available for tracked fields of the object.

[VoucherDefinitionShare](#)

Sharing is available for the object.

CHAPTER 2 Loyalty Management Fields on Standard Objects

In this chapter ...

- [Loyalty Management Fields on EngagementChannelType](#)
- [Loyalty Management Fields on InternalOrganizationUnit](#)

Loyalty Management has added standard fields to some standard Salesforce objects to represent information specific to a promotion. These fields are available only in orgs where Global Promotions Management is enabled.

Loyalty Management Fields on EngagementChannelType

Standard fields extend the EngagementChannelType object for use in Loyalty Management to represent to information promotion eligibility criteria. The Loyalty Management Fields on EngagementChannelType are available in API version 60.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
Delivery	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The type of channel that's eligible for a promotion. This field is only available in orgs where Global Promotions Management is enabled. Possible values are: • Online • Social

Loyalty Management Fields on InternalOrganizationUnit

Standard fields extend the InternalOrganizationUnit object for use in Loyalty Management to represent to information about Global Promotion Management's connection with a Marketing Cloud business unit. The Loyalty Management fields on InternalOrganizationUnit are available in API version 60.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Special Access Rules

The Delivery field is only available in orgs where Global Promotions Management is enabled.

Fields

Field	Details
BusinessUsage	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The business purpose of the record. Possible values are: • UNIFIED_PROMOTIONS—Global Promotions Management
EmailConfiguration	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The send classification that's used by the connected Marketing Cloud business unit to send emails.
OrganizationIdentifier	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The name of the company's Marketing Cloud business unit that Global Promotions Management is connected to.

CHAPTER 3 Coupon Objects in Loyalty Management

Global Promotions Management in Loyalty Management provides access to Coupon objects that you can use to create coupons and manage coupon redemption for your company's promotions. When Global Promotions Management is enabled, you don't need any additional license or user permission for these coupon objects.

For more information about the objects you can use to manage coupons for promotions:

- [Coupon](#)
- [CouponCodeRedemption](#)

CHAPTER 4 Loyalty Management Associated Objects

In this chapter ...

- [StandardObjectNameChangeEvent](#)
- [StandardObjectNameFeed](#)
- [StandardObjectNameHistory](#)
- [StandardObjectNameOwnerSharingRule](#)
- [StandardObjectNameShare](#)
- [Event](#)
- [Task](#)

This section provides a list of objects associated to standard objects and their standard fields.

StandardObjectNameChangeEvent

A `ChangeEvent` object is available for each object that supports Change Data Capture. You can subscribe to a stream of change events using Change Data Capture to receive data tied to record changes in Salesforce. Changes include record creation, updates to an existing record, deletion of a record, and undeletion of a record. A change event isn't a Salesforce object—it doesn't support CRUD operations or queries. It's included in the object reference so you can discover which Salesforce objects support change events.

Supported Calls

`describeSObjects()`

Special Access Rules

- All objects may not be available in your org. Some objects require specific feature settings and permissions to be enabled.
- For more special access rules, if any, see the documentation for the standard object. For example, for `AccountChangeEvent`, see the special access rules for `Account`.

Change Event Support

Change events are available for all custom objects and a subset of standard objects. Change events that correspond to custom settings are partially supported. They aren't supported in Apex triggers but are supported in other types of subscribers. For more information about standard object support, see the [Objects That Support Change Events](#) section below.

Change Event Name

The name of a change event is based on the name of the corresponding object for which it captures the changes.

Standard Object Change Event Name

```
<Standard_Object_Name>ChangeEvent
```

Example: `AccountChangeEvent`

Custom Object Change Event Name

```
<Custom_Object_Name>__ChangeEvent
```

Example: `MyCustomObject__ChangeEvent`

Change Event Fields

The fields that a change event can include correspond to the fields on the associated parent Salesforce object, with a few exceptions. For example, `AccountChangeEvent` fields correspond to the fields on `Account`.

The fields that a change event doesn't include are:

- The `IsDeleted` system field.
- The `SystemModStamp` system field.
- Any field whose value isn't on the record and is derived from another record or from a formula, except roll-up summary fields, which are included. Examples are formula fields. Examples of fields with derived values include `LastActivityDate` and `PhotoUrl`.

Each change event also contains header fields. The header fields are included inside the `ChangeEventHeader` field. They contain information about the event, such as whether the change was an update or delete and the name of the object, like `Account`.

In addition to the event payload, the event schema ID is included in the `schema` field. Also included is the event-specific field, `replayId`, which is used for retrieving past events.

Event Message Example

This example is an event message in JSON format for a new account record creation.

```
{
  "schema": "IeRuaY6cbI_HsV8Rv1Mc5g",
  "payload": {
    "ChangeEventHeader": {
      "entityName": "Account",
      "recordIds": [
        "<record_ID>"
      ],
      "changeType": "CREATE",
      "changeOrigin": "com/salesforce/api/soap/51.0;client=SfdcInternalAPI/",
      "transactionKey": "0002343d-9d90-e395-ed20-cf416ba652ad",
      "sequenceNumber": 1,
      "commitTimestamp": 1612912679000,
      "commitNumber": 10716283339728,
      "commitUser": "<User_ID>"
    },
    "Name": "Acme",
    "Description": "Everyone is talking about the cloud. But what does it mean?",
    "OwnerId": "<Owner_ID>",
    "CreatedDate": "2021-02-09T23:17:59Z",
    "CreatedBy": "<User_ID>",
    "LastModifiedDate": "2021-02-09T23:17:59Z",
    "LastModifiedBy": "<User_ID>"
  },
  "event": {
    "replayId": 6
  }
}
```

API Version and Schema

When you subscribe to change events, the subscription uses the latest API version and the event messages received reflect the latest field definitions. For more information, see [API Version and Event Schema](#) in the *Change Data Capture Developer Guide*.

Usage

For more information about Change Data Capture, see [Change Data Capture Developer Guide](#).

Objects That Support Change Events

These objects have associated `ChangeEvent` objects.

- BenefitType
- DigitalPass
- DigitalPassTemplate
- DigitalPassTplParameterChangeEvent
- GameParticipant
- GameParticipantReward
- GameReward
- JournalSubType
- JournalType
- LoyaltyMemberCurrency
- LoyaltyMemberTier
- LoyaltyPartnerProduct
- LoyaltyPgmEngmtAttribute
- LoyaltyPgmEngmtAttrProm
- LoyaltyPgmGroupMbrRInsp
- LoyaltyPgmMbrAttributeVal
- LoyaltyPgmPartnerPromotion
- LoyaltyProgram
- LoyaltyProgramBadge
- LoyaltyProgramCurrency
- LoyaltyPgmMbrLinkedPtnr
- LoyaltyProgramMbrPromotion
- LoyaltyProgramMember
- LoyaltyProgramMemberBadge
- LoyaltyProgramMemberCase
- LoyaltyProgramPartner
- LoyaltyTier
- LoyaltyTierBenefit
- LoyaltyTierGroup
- MemberBenefit
- Promotion
- PromotionLoyaltyPtnrProdt
- PromotionMarketSegment
- PromotionPartyUsage
- TransactionJournal
- Voucher
- VoucherDefinition

StandardObjectNameFeed

StandardObjectNameFeed is the model for all feed objects associated with standard objects. These objects represent the posts and feed-tracked changes of a standard object.

The object name is variable and uses *StandardObjectNameFeed* syntax. For example, *AccountFeed* represents the posts and feed-tracked changes on an account record. We list the available associated feed objects at the end of this topic. For specific version information, see the documentation for the standard object.

Supported Calls

`delete()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`

Special Access Rules

In the internal org, users can delete all feed items they created. This rule varies in Experience Cloud sites where threaded discussions and delete-blocking are enabled. Site members can delete all feed items they created, provided the feed items don't have content nested under them—like a comment, answer, or reply. Where the feed item has nested content, only feed moderators and users with the Modify All Data permission can delete threads.

To delete feed items they didn't create, users must have one of these permissions:

- Modify All Data
- Modify All Records on the parent object, like Account for AccountFeed
- Moderate Chatter



Note: Users with the Moderate Chatter permission can delete only the feed items and comments they can see.

Only users with this permission can delete items in unlisted groups.

For more special access rules, if any, see the documentation for the standard object. For example, for AccountFeed, see the special access rules for Account.

Fields

Field	Details
BestCommentId	Type reference Properties Filter, Group, Nillable, Sort Description The ID of the comment marked as best answer on a question post. This field is available in API version 44.0 and later.
Body	Type textarea

Field	Details
	Properties Nillable, Sort Description The body of the post. Required when <code>Type</code> is <code>TextPost</code>. Optional when <code>Type</code> is <code>ContentPost</code> or <code>LinkPost</code>.
<code>CommentCount</code>	Type int Properties Filter, Group, Sort Description The number of comments associated with this feed item.  Tip: In a feed that supports pre-moderation, <code>CommentCount</code> isn't updated until a comment is published. For example, say that you comment on a post that already has one published comment and your comment triggers moderation. Now there are two comments on the post, but the count says there's only one. In a moderated feed, comments aren't counted until approved by an admin or someone with <code>Can Approve Feed Post and Comment</code> or <code>Modify All Data</code>. Feed moderation has implications on how you retrieve feed comments. In a moderated feed, rather than retrieving comments by looping through <code>CommentCount</code>, go through pagination until the end of comments is returned.
<code>ConnectionId</code>	Type reference Properties Filter, Group, Nillable, Sort Description When a <code>PartnerNetworkConnection</code> modifies a record that is tracked, the <code>CreatedBy</code> field contains the ID of the system administrator. The <code>ConnectionId</code> contains the ID of the <code>PartnerNetworkConnection</code>. Available if Salesforce to Salesforce is enabled for your organization.
<code>ContentData</code>	Type base64 Properties Nillable Description Available in API version 36.0 and earlier only. Required if <code>Type</code> is <code>ContentPost</code>. Encoded file data in any format, and can't be 0 bytes. Setting this field automatically sets <code>Type</code> to <code>ContentPost</code>. Only visible when <code>Type</code> is set to <code>ContentPost</code>.

Field	Details
ContentDescription	Type textarea Properties Nillable, Sort Description Available in API version 36.0 and earlier only. The description of the file specified in <code>ContentData</code>. Only visible when <code>Type</code> is set to <code>ContentPost</code>.
ContentFileName	Type string Properties Group, Nillable, Sort Description Available in API version 36.0 and earlier only. This field is required if <code>Type</code> is <code>ContentPost</code>. The name of the file uploaded to the feed. Setting <code>ContentFileName</code> automatically sets <code>Type</code> to <code>ContentPost</code>. Only visible when <code>Type</code> is set to <code>ContentPost</code>.
ContentSize	Type int Properties Group, Nillable, Sort Description Available in API version 36.0 and earlier. The size of the file (in bytes) uploaded to the feed. This field is read-only and is automatically determined during insert. Only visible when <code>Type</code> is set to <code>ContentPost</code>.
ContentType	Type string Properties Group, Nillable, Sort Description Available in API version 36.0 and earlier. The MIME type of the file uploaded to the feed. This field is read-only and is automatically determined during insert. Only visible when <code>Type</code> is set to <code>ContentPost</code>.
FeedPostId	Type reference Properties Filter, Group, Nillable, Sort

[illegible]

Field	Details
	 Note: In API version 35.0 and later, the system replaces special characters in rich text with escaped HTML. In API version 34.0 and prior, all rich text appears as a plain-text representation.
LikeCount	Type int Properties Filter, Group, Sort Description The number of likes associated with this feed item.
LinkUrl	Type url Properties Nillable, Sort Description The URL of a <code>LinkPost</code>.
NetworkScope	Type picklist Properties Group, Nillable, Restricted picklist, Sort Description Specifies whether this feed item is available in the default Experience Cloud site, a specific Experience Cloud site, or all sites. This field is available in API version 26.0 and later, if digital experiences is enabled for your org. <code>NetworkScope</code> can have the following values: <code>NetworkId</code>—The ID of the Experience Cloud site in which the <code>FeedItem</code> is available. If left empty, the feed item is only available in the default Experience Cloud site. <code>AllNetworks</code>—The feed item is available in all Experience Cloud sites. Note the following exceptions for <code>NetworkScope</code>: - Only feed items with a Group or User parent can set a <code>NetworkId</code> or a null value for <code>NetworkScope</code>. - For feed items with a record parent, users can set <code>NetworkScope</code> only to <code>AllNetworks</code>. - You can't filter a feed item on the <code>NetworkScope</code> field.
ParentId	Type reference Properties Filter, Group, Sort

Field	Details
	Description ID of the record that is tracked in the feed. The detail page for the record displays the feed.
RelatedRecordId	Type reference Properties Group, Nillable, Sort Description ID of the ContentVersion record associated with a ContentPost. This field is null for all posts except ContentPost.
Title	Type string Properties Group, Nillable, Sort Description The title of the feed item. When the Type is LinkPost, the LinkUrl is the URL and this field is the link name.
Type	Type picklist Properties Filter, Group, Nillable, Restricted picklist, Sort Description The type of feed item: • ActivityEvent—indirectly generated event when a user or the API adds a Task associated with a feed-enabled parent record (excluding email tasks on cases). Also occurs when a user or the API adds or updates a Task or Event associated with a case record (excluding email and call logging). For a recurring Task with CaseFeed disabled, one event is generated for the series only. For a recurring Task with CaseFeed enabled, events are generated for the series and each occurrence. • AdvancedTextPost—created when a user posts a group announcement and, in Lightning Experience as of API version 39.0 and later, when a user shares a post. • AnnouncementPost—Not used. • ApprovalPost—generated when a user submits an approval. • BasicTemplateFeedItem—Not used. • CanvasPost—a post made by a canvas app posted on a feed. • CollaborationGroupCreated—generated when a user creates a public group. • CollaborationGroupUnarchived—Not used. • ContentPost—a post with an attached file.

Field	Details
	• CreatedRecordEvent—generated when a user creates a record from the publisher. • DashboardComponentAlert—generated when a dashboard metric or gauge exceeds a user-defined threshold. • DashboardComponentSnapshot—created when a user posts a dashboard snapshot on a feed. • LinkPost—a post with an attached URL. • PollPost—a poll posted on a feed. • ProfileSkillPost—generated when a skill is added to a user's Chatter profile. • QuestionPost—generated when a user posts a question. • ReplyPost—generated when Chatter Answers posts a reply. • RypplePost—generated when a user creates a Thanks badge in WDC. • TextPost—a direct text entry on a feed. • TrackedChange—a change or group of changes to a tracked field. • UserStatus—automatically generated when a user adds a post. Deprecated. The following values appear in the Type picklist for all feed objects but apply only to CaseFeed: • CaseCommentPost—generated event when a user adds a case comment for a case object • EmailMessageEvent—generated event when an email related to a case object is sent or received • CallLogPost—generated event when a user logs a call for a case through the user interface. CTI calls also generate this event. • ChangeStatusPost—generated event when a user changes the status of a case • AttachArticleEvent—generated event when a user attaches an article to a case  Note: If you set Type to ContentPost, also specify ContentData and ContentFileName.
Visibility	Type picklist Properties Filter, Group, Nillable, Restricted picklist, Sort Description Specifies whether this feed item is available to all users or internal users only. This field is available in API version 26.0 and later, if digital experiences is enabled for your organization. Visibility can have the following values: • AllUsers—The feed item is available to all users who have permission to see the feed item. • InternalUsers—The feed item is available to internal users only. Note the following exceptions for Visibility:

Field	Details
	- For record posts, <code>Visibility</code> is set to <code>InternalUsers</code> for all internal users by default. - External users can set <code>Visibility</code> only to <code>AllUsers</code>. - On user and group posts, only internal users can set <code>Visibility</code> to <code>InternalUsers</code>.

Usage

A feed for an object is automatically created when a user enables feed tracking for the object. Use feeds to track changes to records. For example, `AccountFeed` tracks changes to an account record. Use feed objects to retrieve the content of feed fields, such as type of feed or feed ID.

- `NewsFeed` and `UserProfileFeed` are available in API version 18.0 through API version 26.0. In API version 27.0 and later, `NewsFeed` and `UserProfileFeed` are no longer available in SOAP API. Use Connect REST API to access `NewsFeed` and `UserProfileFeed`.

Use the `NewsFeed` object to query and retrieve lead feed items associated with a converted lead record.

- For `NewsFeed` and `UserProfileFeed`, users who don't have the View All Data permission have the following limitations when querying records: Must specify a `LIMIT` clause and the limit must be less than or equal to 1000. Can include a `WHERE` clause that references object fields, but can't include references to fields in related objects. For example, you can filter by `CreatedDate` or `ParentId`, but not by `Parent.Name`. Can include an `ORDER BY` clause that references object fields, but can't include references to fields in related objects. For example, `ORDER BY CreatedDate` or `ParentId`, but not by `Parent.Name`. To query for the most recent feed items, `ORDER BY CreatedDate DESC, Id DESC`.

Note the following SOQL restrictions. No SOQL limit if logged-in user has View All Data permission. If not, specify a `LIMIT` clause of 1,000 records or fewer. SOQL `ORDER BY` on fields using relationships isn't available. Use `ORDER BY` on fields on the root object in the SOQL query.

- The name `Article_Type__Feed` is variable, where `Article Type` is the object name for the article type associated with the article. For example, `Offer__Feed` represents a feed on an article of type `Offer`.
- Field Service must be enabled in your organization for `ServiceAppointmentFeed`, `ServiceCrewFeed`, `ServiceMemberFeed`, `ServiceResourceCapacityFeed`, `ServiceResourceFeed`, `ServiceResourceSkillFeed`, `ServiceTerritoryFeed`, `ServiceTerritoryMemberFeed`, and `SkillRequirementFeed`.
- For `WorkOrderFeed`, Work Orders or Field Service must be enabled in your organization.
- On `UserFeed`, if you use the `FeedComment` object to comment on a user record, the user can delete the comment. For example, if John Smith adds a comment to the feed on Sasha Jones' user record, Sasha can delete the comment.

StandardObjectNameHistory

`StandardObjectNameHistory` is the model for all history objects associated with standard objects. These objects represent the history of changes to the values in the fields of a standard object.

The object name is variable and uses `StandardObjectNameHistory` syntax. For example, `AccountHistory` represents the history of changes to the values of an account record's fields. We list the available associated history objects at the end of this topic. For specific version information, see the documentation for the standard object.

Supported Calls

`describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`

You can also enable `delete()` in API version 42.0 and later. See [Enable delete of Field History and Field History Archive](#).

Special Access Rules

For specific special access rules, if any, see the documentation for the standard object. For example, for AccountHistory, see the special access rules for Account.

Fields

Field Name	Details
<i>StandardObjectId</i>	Type reference Properties Filter, Group, Sort Description ID of the standard object.
<i>DataType</i>	Type picklist Properties Filter, Group, Nillable, Restricted picklist, Sort Description Data type of the field that was changed.
<i>Field</i>	Type picklist Properties Filter, Group, Restricted picklist, Sort Description Name of the field that was changed.
<i>NewValue</i>	Type anyType Properties Nillable, Sort Description New value of the field that was changed.
<i>OldValue</i>	Type anyType

Field Name	Details
	Properties Nillable, Sort
	Description Old value of the field that was changed.

StandardObjectName OwnerSharingRule OwnerSharingRule

StandardObjectNameOwnerSharingRule is the model for all owner sharing rule objects associated with standard objects. These objects represent a rule for sharing a standard object with users other than the owner.

The object name is variable and uses *StandardObjectNameOwnerSharingRule* syntax. For example, *ChannelProgramOwnerSharingRule* is a rule for sharing a channel program with users other than the channel program owner. The available associated owner sharing rule objects are listed at the end of this topic. For specific version information, see the standard object documentation.

 **Note:** To enable access to this object, contact Salesforce customer support. But we recommend that you use Metadata API to programmatically update owner sharing rules instead because it triggers automatic sharing rule recalculation. The [SharingRules](#) Metadata API type is enabled for all orgs.

Supported Calls

`create()`, `delete()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `update()`, `upsert()`

Special Access Rules

For specific special access rules, if any, see the documentation for the standard object. For example, for *ChannelProgramOwnerSharingRule*, see the special access rules for *ChannelProgram*.

Fields

Field Name	Details
<code>AccessLevel</code>	Type picklist
	Properties Create, Filter, Group, Restricted picklist, Sort, Update
	Description Determines the level of access users have to records. Values are: • <code>Read</code> (read only) • <code>Edit</code> (read/write)

Field Name	Details
Description	Type textarea Properties Create, Filter, Nillable, Sort, Update Description Description of the sharing rule. Maximum length is 1,000 characters.
DeveloperName	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The unique name of the object in the API. This name can contain only underscores and alphanumeric characters and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. In managed packages, this field prevents naming conflicts on package installations. With this field, a developer can change the object's name in a managed package, and the changes are reflected in a subscriber's organization.  Note: When creating large sets of data, always specify a unique DeveloperName for each record. If no DeveloperName is specified, performance can slow while Salesforce generates one for each record.
GroupId	Type reference Properties Create, Filter, Group, Sort Description ID of the source group. Records that are owned by users in the source group trigger the rule to give access.
Name	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description Label of the sharing rule as it appears in the UI. Maximum length is 80 characters.
UserOrGroupId	Type reference Properties Create, Filter, Group, Sort

Field Name	Details
	Description ID of the user or group that you're granting access to.

StandardObjectNameShare

StandardObjectNameShare is the model for all share objects associated with standard objects. These objects represent a sharing entry on the standard object.

The object name is variable and uses *StandardObjectNameShare* syntax. For example, *AccountBrandShare* is a sharing entry on an account brand. For specific version information, see the standard object documentation.

You can only create, edit, and delete sharing entries for standard objects whose *RowCause* field is set to *Manual*. Sharing entries for standard objects with different *RowCause* values are created as a result of your Salesforce org's sharing configuration and are read-only. For some sharing mechanisms, such as sharing sets, sharing entries aren't stored at all.

 **Note:** While Salesforce currently maintains read-only sharing entries for multiple sharing mechanisms, it's possible that we'll stop storing certain share records to improve performance. As a best practice, don't create customizations that rely on the availability of these sharing entries. Any changes to sharing behavior will be communicated before they occur.

Supported Calls

`create()`, `delete()`, `describeObjects()`, `query()`, `retrieve()`, `update()`, `upsert()`

Special Access Rules

For specific special access rules, if any, see the documentation for the standard object. For example, for *AccountBrandShare*, see the special access rules for *AccountBrand*.

Fields

Field Name	Details
<code>AccessLevel</code>	Type picklist
	Properties Create, Filter, Group, Restricted picklist, Sort, Update
	Description The level of access allowed. Values are: <code>All</code> (owner) <code>Edit</code> (read/write) <code>Read</code> (read only)

Field Name	Details
ParentId	Type reference Properties Create, Filter, Group, Sort Description ID of the parent record.
RowCause	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort Description Reason that the sharing entry exists.
UserOrGroupId	Type reference Properties Create, Filter, Group, Sort Description ID of the user or group that has been given access to the object.

Event

Represents an event in the calendar. In the user interface, event and task records are collectively referred to as activities.

 **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.

 **Note:**

- An EventRelation object can't be related to a child event, and child events don't include the invitee related list.
- `query()`, `delete()`, and `update()` aren't allowed with events related to more than one contact in API versions 25.0 and earlier.
- `create()` and `update()` aren't available for read-only fields on Lightning Experience event series.
- `upsert()` and `undelete()` aren't supported for syncing changes made to events through the API using the feature Lightning Sync.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
AcceptedEventInviteeIds	Type JunctionIdList Properties Create, Update Description A string array of contact or lead IDs who accepted this event. This <code>JunctionIdList</code> is linked to the <code>AcceptedEventRelation</code> child relationship.  Warning: Adding a <code>JunctionIdList</code> field name to the <code>fieldsToNull</code> property deletes all related junction records. This action can't be undone.
AccountId	Type reference Properties Filter, Group, Nillable, Sort Description Represents the ID of the related account. The <code>AccountId</code> is determined as follows. If the value of <code>whatId</code> is any of the following objects, then Salesforce uses that object's <code>AccountId</code>. • Account • Opportunity • Contract • Custom object that's a child of Account If the value of the <code>whatId</code> field is any other object, and the value of the <code>whoId</code> field is a contact object, then Salesforce uses that contact's <code>AccountId</code>. If your org uses Shared Activities, Salesforce uses the <code>AccountId</code> of the primary contact. Otherwise, Salesforce sets the value of the <code>AccountId</code> field to <code>null</code>. For information on IDs, see ID Field Type. This is a relationship field. Relationship Name Account Relationship Type Lookup Refers To Account
ActivityDate	Type date

Field	Details
	Properties Create, Filter, Group, Nillable, Sort, Update Description Contains the event's due date if the <code>IsAllDayEvent</code> flag is set to <code>true</code>. This field is a date field with a timestamp that's always set to midnight in the Coordinated Universal Time (UTC) time zone. Don't attempt to alter the timestamp to account for time zone differences. Label is Due Date Only. This field is required in API versions 12.0 and earlier if the <code>IsAllDayEvent</code> flag is set to <code>true</code>. The value for this field and <code>StartDateTime</code> must match, or one of them must be <code>null</code>.
ActivityDateTime	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description Contains the event's due date if the <code>IsAllDayEvent</code> flag is set to <code>false</code>. The time portion of this field is always transferred in the Coordinated Universal Time (UTC) time zone. Translate the time portion to or from a local time zone for the user or the application, as appropriate. Label is Due Date Time. This field is required in API versions 12.0 and earlier if the <code>IsAllDayEvent</code> flag is set to <code>false</code>. The value for this field and <code>StartDateTime</code> must match, or one of them must be <code>null</code>.
ClientGuid	Type string Properties Filter, Group, Nillable, Sort Description The client globally unique identifier identifies the external API client used to create the event. Label is Client GUID.
CurrencyIsoCode	Type picklist Properties Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description Available only for orgs with the multicurrency feature enabled. Contains the ISO code for any currency allowed by the organization.
DeclinedEventInviteeIds	Type JunctionIdList

Field	Details
	Properties Create, Update Description A string array of contact, lead, or user IDs who declined this event. This <code>JunctionIdList</code> is linked to the <code>DeclinedEventRelation</code> child relationship.  Warning: Adding a <code>JunctionIdList</code> field name to the <code>fieldsToNull</code> property deletes all related junction records. This action can't be undone.
Description	Type textarea Properties Create, Nillable, Update Description Contains a text description of the event. Limit: 32,000 characters.
Division	Type picklist Properties Defaulted on create, Filter, Group, Restricted picklist, Sort Description A logical segment of your organization's data. For example, if your company is organized into different business units, you could create a division for each business unit, such as "North America," "Healthcare," or "Consulting." Available only if the organization has the Division permission enabled.
DurationInMinutes	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description Contains the event length, in minutes. Even though this field represents a temporal value, it's an integer type—not a Date/Time type. Required in API versions 12.0 and earlier if <code>IsAllDayEvent</code> is false. In API versions 13.0 and later, this field is optional, depending on the following: • If <code>IsAllDayEvent</code> is true, you can supply a value for either <code>DurationInMinutes</code> or <code>EndDateTime</code>. Supplying values in both fields is allowed if the values add up to the same amount of time. If both fields are <code>null</code>, the duration defaults to one day. • If <code>IsAllDayEvent</code> is false, a value must be supplied for either <code>DurationInMinutes</code> or <code>EndDateTime</code>. Supplying values in both fields is allowed if the values add up to the same amount of time.

Field	Details
	If the multiday event feature is enabled, then API versions 13.0 and later support values greater than 1440 for the <code>DurationInMinutes</code> field. API versions 12.0 and earlier can't access event objects whose <code>DurationInMinutes</code> is greater than 1440. For more information, see Multiday Events. Depending on your API version, errors with the <code>DurationInMinutes</code> and <code>EndDateTime</code> fields may appear in different places. - Versions 38.0 and before—Errors always appear in the <code>DurationInMinutes</code> field. - Versions 39.0 and later—If there's no value for the <code>DurationInMinutes</code> field, errors appear in the <code>EndDateTime</code> field. Otherwise, they appear in the <code>DurationInMinutes</code> field.
EndDate	Type date Properties Filter, Group, Nillable, Sort Description Read-only. Available in API versions 46.0 and later. This field supplies the date value that appears in the <code>EndDateTime</code> field. This field is a date field with a timestamp that is always set to midnight in the Coordinated Universal Time (UTC) time zone.
EndDateTime	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description Available in API versions 13.0 and later. The time portion of this field is always transferred in the Coordinated Universal Time (UTC) time zone. Translate the time portion to or from a local time zone for the user or the application, as appropriate. This field is optional, depending on the following: - If <code>IsAllDayEvent</code> is true, you can supply a value for either <code>DurationInMinutes</code> or <code>EndDateTime</code>. Supplying values in both fields is allowed if the values add up to the same amount of time. If both fields are <code>null</code>, the duration defaults to one day. - If <code>IsAllDayEvent</code> is false, a value must be supplied for either <code>DurationInMinutes</code> or <code>EndDateTime</code>. Supplying values in both fields is allowed if the values add up to the same amount of time. Depending on your API version, errors with the <code>DurationInMinutes</code> and <code>EndDateTime</code> fields may appear in different places. - Versions 38.0 and before—Errors always appear in the <code>DurationInMinutes</code> field. - Versions 39.0 and later—If there's no value for the <code>DurationInMinutes</code> field, errors appear in the <code>EndDateTime</code> field. Otherwise, they appear in the <code>DurationInMinutes</code> field.

Field	Details
EventSubtype	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort Description Provides standard subtypes to facilitate creating and searching for events. This field isn't updateable.
EventWhoIds	Type JunctionIdList Properties Create, Update Description A string array of contact or lead IDs used to create many-to-many relationships with a shared event. <code>EventWhoIds</code> is available when the shared activities setting is enabled. The first contact or lead ID in the list becomes the primary <code>whoId</code> if you don't specify a primary <code>whoId</code>. If you set the <code>EventWhoIds</code> field to null, all entries in the list are deleted and the value of <code>whoId</code> is added as the first entry.  Warning: Adding a <code>JunctionIdList</code> field name to the <code>fieldsToNull</code> property deletes all related junction records. This action can't be undone.
GroupEventType	Type picklist Properties Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort Description Read-only. Available in API versions 19.0 and later. The possible values are: • 0 (Non-group event)—An event with no invitees. • 1 (Group event)—An event with invitees. • 2 (Proposed event)—An event created when a user requests a meeting with a contact, lead, or person account using the Salesforce user interface. When the user confirms the meeting, the proposed event becomes a group event. You can't create, edit, or delete proposed events in the API. This value is no longer used in API version 41.0 and later. • 3 (IsRecurrence2 Series Pattern)—An event representing an event series recurrence pattern in Lightning Experience.
IsAllDayEvent	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update

Field	Details
	Description Indicates whether the <code>ActivityDate</code> field (<code>true</code>) or the <code>ActivityDateTime</code> field (<code>false</code>) is used to define the date or time of the event. Label is All-Day Event. See also DurationInMinutes and EndDateTime.
IsArchived	Type boolean Properties Defaulted on create, Filter, Group, Sort Description Indicates whether the event has been archived.
IsChild	Type boolean Properties Defaulted on create, Filter, Group, Sort Description Indicates whether the event is a child of another event (<code>true</code>) or not (<code>false</code>). For a child event, you can update <code>IsReminderSet</code> and <code>ReminderDateTime</code> only. You can query and delete a child event. If the objects related to the child event are different from those objects related to the parent event (this difference is possible if you use API version 25.0 or earlier) and one of the objects related to the child event is deleted, the objects related to the parent event are updated to ensure data integrity.
IsClientManaged	Type boolean Properties Defaulted on create, Filter, Group, Sort Description Indicates whether the event is managed by an external client. If the value of this field is false, the event isn't owned or managed by an external client, and Salesforce can be used to update it. If the value is true, Salesforce can be used to change only noncritical fields on the event. Label is Is Client Managed.
IsGroupEvent	Type boolean Properties Defaulted on create, Filter, Group, Sort Description Indicates whether the event is a group event—that is, whether it has invitees (<code>true</code>) or not (<code>false</code>).

Field	Details
IsPrivate	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether users other than the creator of the event can (<code>false</code>) or can't (<code>true</code>) see the event details when viewing the event user's calendar. However, users with the View All Data or Modify All Data permission can see private events in reports and searches, or when viewing other users' calendars. Private events can't be associated with opportunities, accounts, cases, campaigns, contracts, leads, or contacts. Label is Private.
IsRecurrence	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort Description Indicates whether a Salesforce Classic event is scheduled to repeat itself (<code>true</code>) or only occurs one time (<code>false</code>). This field is read-only when updating records, but not when creating them. If this field value is <code>true</code>, then <code>RecurrenceEndDateOnly</code>, <code>RecurrenceStartDateTime</code>, <code>RecurrenceType</code>, and any recurrence fields associated with the given recurrence type must be populated. Label is Create recurring series of events.
IsRecurrence2	Type boolean Properties Defaulted on create, Filter, Group, Sort Description Read-only. This field is available in API version 44.0 and later. Indicates whether a Lightning Experience event is scheduled to repeat (<code>true</code>) or only occurs one time (<code>false</code>). If this field value is <code>true</code>, then <code>Recurrence2PatternText</code> and <code>Recurrence2PatternVersion</code> must be populated. Label is Repeat.
IsRecurrence2Exception	Type boolean Properties Defaulted on create, Filter, Group, Sort Description Read-only. This field is available in API version 44.0 and later. Indicates whether an individual event in a Lightning Experience event series has a recurrence pattern that's different from the rest of the series, making it an exception.

Field	Details
IsRecurrence2Exclusion	Type boolean Properties Defaulted on create, Filter, Group, Sort Description Read-only. This field is available in API version 44.0 and later. Indicates when updates to a Lightning Experience event series recurrence pattern have been made, but affect future event occurrences only. For past event occurrences, <code>IsRecurrence2Exclusion</code> is set to <code>true</code>, excluding past occurrences from the series recurrence pattern.
IsReminderSet	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether the activity is a reminder (<code>true</code>) or not (<code>false</code>).
IsVisibleInSelfService	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether an event associated with an object can be viewed in the Customer Portal (<code>true</code>) or not (<code>false</code>). If your org has enabled digital experiences, events marked <code>IsVisibleInSelfService</code> are visible to any external user in the Experience Cloud site, as long as the user has access to the record the event was created on. This field is available when - Customer Portal or partner portal is enabled OR - Digital experiences is enabled and you have Customer Portal or partner portal licenses
Location	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description Contains the location of the event.
OwnerId	Type reference Properties Create, Defaulted on create, Filter, Group, Sort, Update

Field	Details
	Description Contains the ID of the user or public calendar who owns the event. Label is Assigned to ID. This is a polymorphic relationship field. Relationship Name Owner Relationship Type Lookup Refers To Calendar, User
Recurrence2PatternStartDate	Type dateTime Properties Filter, Nillable, Sort Description Read-only. This field is available in API version 44.0 and later. Indicates the date and time when the Lightning Experience event series begins. The time portion of this field is always transferred in the Coordinated Universal Time (UTC) time zone. Translate the time portion to or from a local time zone for the user or the application, as appropriate.
Recurrence2PatternText	Type textarea Properties Create, Nillable Description The RRULE that describes the recurrence pattern for Lightning Experience event series. Supports a subset of the RFC 5545 standard for internet calendaring and scheduling. See the Event Series section in this topic for usage examples. This field has a maximum length of 512 characters. This field is available in API version 44.0 and later, and has the <code>Create</code> property in API version 52.0 and later.
Recurrence2PatternTimeZone	Type string Properties Filter, Group, Nillable, Sort Description This field is available in API version 44.0 and later. Indicates the time zone in which the Lightning Experience event series was created or updated. This field uses standard Java TimeZone IDs. For example, America/Los_Angeles.

Field	Details
Recurrence2PatternVersion	Type picklist Properties Filter, Group, Nillable, Restricted picklist, Sort, Description Read-only. This field is available in API version 44.0 and later. Indicates the standard specifications for Lightning Experience event series recurrence patterns. The only possible value is 1 (RFC 5545 v4 RRULE)—RFC 5545 is a standard set of specifications for internet calendaring and scheduling that <code>IsRecurrence2</code> adheres to for series recurrence patterns. RFC 5545 specifications for series recurrence patterns are called RRULES. For examples of RRULE usage, see the Lightning Experience Event Series and Recurring Events section in this topic.
RecurrenceActivityId	Type reference Properties Filter, Group, Nillable, Sort Description Read-only. Not required on create. Contains the ID of the main record of the Salesforce Classic recurring event. Subsequent occurrences have the same value in this field.
RecurrenceDayOfMonth	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description Indicates the day of the month on which the event repeats.
RecurrenceDayOfWeekMask	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description Indicates the day or days of the week on which the Salesforce Classic recurring event repeats. This field contains a bitmask. The values are as follows: • Sunday = 1 • Monday = 2 • Tuesday = 4 • Wednesday = 8 • Thursday = 16 • Friday = 32 • Saturday = 64

Field	Details
	Multiple days are represented as the sum of their numerical values. For example, Tuesday and Thursday = 4 + 16 = 20.
RecurrenceEndDateOnly	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description Indicates the last date on which the event repeats. For multiday Salesforce Classic recurring events, this date is the day on which the last occurrence starts. This field is a date field with a timestamp that is always set to midnight in the Coordinated Universal Time (UTC) time zone. Don't attempt to alter the timestamp to account for time zone differences.
RecurrenceInstance	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description Indicates the frequency of the Salesforce Classic event's recurrence. For example, 2nd or 3rd.
RecurrenceInterval	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description Indicates the interval between Salesforce Classic recurring events.
RecurrenceMonthOfYear	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description Indicates the month in which the Salesforce Classic recurring event repeats.
RecurrenceStartDateTime	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description Indicates the date and time when the Salesforce Classic recurring event begins. The value must precede the <code>RecurrenceEndDateOnly</code>. The time portion of this field is always

Field	Details
	transferred in the Coordinated Universal Time (UTC) time zone. Translate the time portion to or from a local time zone for the user or the application, as appropriate.
RecurrenceTimeZoneSidKey	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description Indicates the time zone associated with a Salesforce Classic recurring event. For example, "UTC-8:00" for Pacific Standard Time.
RecurrenceType	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description Indicates how often the Salesforce Classic event repeats. For example, daily, weekly, or every nth month (where "nth" is defined in <code>RecurrenceInstance</code>).
ReminderDateTime	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description Represents the time when the reminder is scheduled to fire, if <code>IsReminderSet</code> is set to <code>true</code>. If <code>IsReminderSet</code> is set to <code>false</code>, then the user may have deselected the reminder checkbox in the Salesforce user interface, or the reminder has already fired at the time indicated by the value.
ShowAs	Type picklist Properties Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description Indicates how this event appears when another user views the calendar: Busy, Out of Office, or Free. Label is Show Time As.
StartDateTime	Type dateTime Properties Create, Filter, Nillable, Sort, Update

Field	Details
	Description Indicates the start date and time of the event. Available in versions 13.0 and later. If the Event <code>IsAllDayEvent</code> flag is set to true (indicating that it's an all-day Event), then the event start date information is contained in the <code>StartDateTime</code> field. The time portion of this field is always transferred in the Coordinated Universal Time (UTC) time zone. Translate the time portion to or from a local time zone for the user or the application, as appropriate. If the Event <code>IsAllDayEvent</code> flag is set to false (indicating that it isn't an all-day event), then the event start date information is contained in the <code>StartDateTime</code> field. The time portion is always transferred in the Coordinated Universal Time (UTC) time zone. You need to translate the time portion to or from a local time zone for the user or the application, as appropriate. If this field has a value, then <code>ActivityDate</code> and <code>ActivityDateTime</code> must either be <code>null</code> or match the value of this field.
Subject	Type combobox Properties Create, Filter, Nillable, Sort, Update Description The subject line of the event, such as Call, Email, or Meeting. Limit: 255 characters.
Type	Type picklist Properties Create, Filter, Group, Nillable, Sort, Update Description Indicates the event type, such as Call, Email, or Meeting.
UndecidedEventInviteeIds	Type JunctionIdList Properties Create, Update Description A string array of contact, lead, or user IDs who are undecided about this event. This <code>JunctionIdList</code> is linked to the <code>UndecidedEventRelation</code> child relationship.  Warning: Adding a <code>JunctionIdList</code> field name to the <code>fieldsToNull</code> property deletes all related junction records. This action can't be undone.
WhatCount	Type int

Field	Details
	Properties Filter, Group, Nillable, Sort Description Available if your organization has enabled Shared Activities. Represents the count of related EventRelations pertaining to the <code>whatId</code>. The count of the <code>whatId</code> must be 1 or less.
<code>WhatId</code>	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The <code>whatId</code> represents nonhuman objects such as accounts, opportunities, campaigns, cases, or custom objects. <code>whatIds</code> are polymorphic. Polymorphic means a <code>whatId</code> is equivalent to the ID of a related object. The label is <code>Related To ID</code>. This is a polymorphic relationship field. Relationship Name What Relationship Type Lookup Refers To Account, Accreditation, AssessmentIndicatorDefinition, AssessmentTask, AssessmentTaskContentDocument, AssessmentTaskDefinition, AssessmentTaskOrder, Asset, AssetRelationship, AssignedResource, Award, BoardCertification, BusinessLicense, BusinessMilestone, BusinessProfile, Campaign, CareBarrier, CareBarrierDeterminant, CareBarrierType, CareDeterminant, CareDeterminantType, CareDiagnosis, CareInterventionType, CareMetricTarget, CareObservation, CareObservationComponent, CarePgmProvHealthcareProvider, CarePreauth, CarePreauthItem, CareProgram, CareProgramCampaign, CareProgramEligibilityRule, CareProgramEnrollee, CareProgramEnrolleeProduct, CareProgramEnrollmentCard, CareProgramGoal, CareProgramProduct, CareProgramProvider, CareProgramTeamMember, CareProviderAdverseAction, CareProviderFacilitySpecialty, CareProviderSearchableField, CareRegisteredDevice, CareRequest, CareRequestDrug, CareRequestExtension, CareRequestItem, CareSpecialty, CareSpecialtyTaxonomy, CareTaxonomy, Case, CommSubscriptionConsent, ContactEncounter, ContactEncounterParticipant, ContactRequest, Contract, CoverageBenefit, CoverageBenefitItem, CreditMemo, DelegatedAccount, DocumentChecklistItem, EnrollmentEligibilityCriteria, HealthcareFacility, HealthcareFacilityNetwork, HealthcarePayerNetwork, HealthcarePractitionerFacility, HealthcareProvider, HealthcareProviderNpi, HealthcareProviderSpecialty, HealthcareProviderTaxonomy, IdentityDocument, Image, IndividualApplication, Invoice, ListEmail, Location, MemberPlan, Opportunity, Order, OtherComponentTask, PartyConsent, PersonLifeEvent, PlanBenefit, PlanBenefitItem, ProcessException, Product2, ProductItem, ProductRequest, ProductRequestLineItem, ProductTransfer, PurchaserPlan, ReceivedDocument, ResourceAbsence, ReturnOrder, ReturnOrderLineItem,

Field	Details
	ServiceAppointment, ServiceResource, Shift, Shipment, ShipmentItem, Solution, Visit, VisitedParty, VolunteerProject, WorkOrder, WorkOrderLineItem
WhoCount	Type int Properties Filter, Group, Nillable, Sort Description Available to organizations that have Shared Activities enabled. Represents the count of related EventRelations pertaining to the <code>WhoId</code>.
WhoId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The Whold represents a human such as a lead or a contact. Wholds are polymorphic. Polymorphic means a Whold is equivalent to a contact's ID or a lead's ID. The label is <code>Name ID</code>. If Shared Activities is enabled, the value of this field is the ID of the related lead or primary contact. If you add, update, or remove the Whold field, you might encounter problems with triggers, workflows, and data validation rules that are associated with the record. The label is <code>Name ID</code>. If the <code>JunctionIdList</code> field is used, all <code>WhoIds</code> are included in the relationship list. Beginning in API version 37.0, if the contact or lead ID in the <code>WhoId</code> field isn't in the <code>EventWhoIds</code> list, no error occurs and the ID is added to the <code>EventWhoIds</code> as the primary <code>WhoId</code>. If <code>WhoId</code> is set to null, an arbitrary ID from the existing <code>EventWhoIds</code> list is promoted to the primary position. This is a polymorphic relationship field. Relationship Name Who Relationship Type Lookup Refers To Contact, Lead

Usage

Use Event to manage calendar appointments.

Querying and Filtering Events

Queries on events are denied before they time out if they involve amounts of data that are deemed too large. In such cases, the exception code `OPERATION_TOO_LARGE` is returned. If you receive `OPERATION_TOO_LARGE`, refactor your query to return or scan a smaller amount of data.

When querying for events with a specific due date, you must filter on both the `ActivityDateTime` and `ActivityDate` fields. For example to find all events with a due date of February 14, 2003, you need two filters:

- One filter with the `ActivityDate` field equal to the Coordinated Universal Time (UTC) time zone on February 14, 2003.
- One filter with the `ActivityDate` field greater than or equal to midnight on February 14, 2003 in the user's local time zone AND less than or equal to midnight on February 15, 2003 in the user's local time zone.

Alternatively, in API version 13.0 and later, you can find events with a specific due date by filtering on `StartDateTime`. For example, to find all events with a due date of February 14, 2003, filter with the `StartDateTime` greater than or equal to midnight on February 14, 2003 in the user's local time zone AND less than or equal to midnight on February 15, 2003 in the user's local time zone.

The `EventId` field of an `EventRelation` object always points to the master record. An invitee on a group event can query the `EventRelation` object to view the master record.

Multiday Events

- Multiday events are available in API version 13.0 and later. Also, in earlier versions SOQL queries don't return multiday events.
- Multiday events are enabled through the user interface from Setup by entering *Activity Settings* in the *Quick Find* box, then selecting **Activity Settings**.
- If the multiday event feature is enabled, then API versions 13.0 and later support values greater than 1440 for the `DurationInMinutes` field. API versions 12.0 and earlier can't access event objects whose `DurationInMinutes` is greater than 1440.
- Multiday events can't exceed 14 days.

Event Series and Recurring Events

In Lightning Experience, events with multiple occurrences are called event series, and are indicated when the `IsRecurrence2` field is set to `true`. In Salesforce Classic, events with multiple occurrences are called recurring events, and are indicated when the `IsRecurrence` field is set to `true`. Both fields can't be set to true for the same event.

- Lightning Experience event series are available in API version 44.0 and later as read-only fields. Recurrence patterns, specified by the `Recurrence2PatternText` field, are creatable in API version 52.0 and later. Salesforce Classic recurring events are available in API version 7.0 and later. In earlier versions, SOQL queries don't return any Lightning Experience event series.
- After an event is created, you can't change the values of `IsRecurrence2` or `IsRecurrence` from `true` to `false` or vice versa.
- You can't set fields associated with `IsRecurrence2` for events where `IsRecurrence` is set to `true`, or vice versa.
- For Lightning Experience event series where `IsRecurrence2` is `true`, if you'd like to delete a single or all remaining events, use the REST API call. For Salesforce Classic recurring events where `IsRecurrence` is `true`, all past and future events in the series are removed when you delete the recurring event series through the API. However, when you delete the recurring event series through the user interface, only future occurrences are removed.
- For Lightning Experience event series in API version 58.0 and later, when you change a future event, events in the entire series also change. When you change a past event, `IsRecurrence2Exception` is set to `true` and only that past event changes.
- When creating a Salesforce Classic recurring event series, the duration of the event must be 24 hours or less. When the Salesforce Classic recurring event series is created, you can extend the length of individual occurrences beyond 24 hours if Multiday events are enabled; see **Multiday Events**.
- For Salesforce Classic recurring events, `RecurrenceStartDateTime`, `RecurrenceEndDateOnly`, `RecurrenceType`, and any properties associated with the given recurrence type (see the Recurrence Field Usage for Salesforce Classic Recurring Events table) must be populated.

- When updating a Salesforce Classic recurring event series, it's not possible to update the `EventRelation` for the event series object and the `EventRelation` for the series object occurrences at the same time.
- Lightning Experience event series have no series ID, so it's not possible to locate other occurrences in the series. In Salesforce Classic recurring events, you can use `RecurrenceActivityId` to locate other occurrences.
- For both Lightning Experience event series and Salesforce Classic recurring events, when a series repeats every day, month, or year, you can only schedule occurrences one time per day, month, or year. The week option lets you schedule occurrences multiple days per week.

[Limits for Lightning Experience event series](#) and [limits for Salesforce Classic recurring events](#) also apply.

Lightning Experience Event Series and Recurring Events

Use the `Recurrence2PatternText` field to specify the recurrence pattern for Lightning Experience event series. These recurrence patterns, called reference rules or RRULES, support a subset of the RFC 5545 standards. This table includes common RRULE examples.

Recurrence Pattern	RRULE Example
Every day for five days	<code>RRULE:FREQ=DAILY;INTERVAL=1;COUNT=5</code>
Every Monday, Tuesday, Wednesday, Thursday, and Friday with no end date	<code>RRULE:FREQ=WEEKLY;INTERVAL=1;BYDAY=MO,TU,WE,TH,FR</code>
Every two weeks on Monday and Friday for 10 occurrences	<code>RRULE:FREQ=WEEKLY;INTERVAL=2;BYDAY=MO,FR;COUNT=10</code>
Monthly on the first day of the month until January 1, 2020	<code>RRULE:FREQ=MONTHLY;INTERVAL=1;BYMONTHDAY=1;UNTIL=20200101T000000Z</code>
Yearly on July 4th for three years (in this example, specify the date using <code>StartDateTime</code>)	<code>RRULE:FREQ=YEARLY;INTERVAL=1;BYMONTH=7;BYMONTHDAY=4;COUNT=3</code>
Daily until January 1, 2022 with no end date	<code>RRULE:FREQ=DAILY;UNTIL=20220101T000000Z</code>
Every third Friday of the month with no end date	<code>RRULE:FREQ=MONTHLY;BYSETPOS=3;BYDAY=FR</code>

The RRULE defined by `Recurrence2PatternText` supports a subset of the RFC 5545 standard for internet calendaring and scheduling. Supported RRULE parts include `FREQ`, `BYMONTH`, `BYMONTHDAY`, `BYDAY`, `WKST`, `BYSETPOS`, `INTERVAL`, `UNTIL`, and `COUNT`.

When the event record is saved, the RRULE might be modified to follow the required format:

- The RRULE parts are placed in the following order: `FREQ`, `BYMONTH`, `BYMONTHDAY`, `BYDAY`, `WKST`, `BYSETPOS`, `INTERVAL`, `UNTIL`, and `COUNT`.
- Any missing default values are inserted. For example, if the RRULE doesn't include `INTERVAL`, then `INTERVAL=1` is added.
- The RRULE is prefaced with `RRULE:` if that preface is missing.

RRULE Part	Supported RFC 5545 Implementation
<code>FREQ</code>	Required. Indicates the type of recurrence rule. Allowed values are: • <code>DAILY</code>—supported parts include <code>FREQ</code>, <code>INTERVAL</code>, <code>UNTIL</code>, and <code>COUNT</code>. • <code>WEEKLY</code>—supported parts include <code>INTERVAL</code>, <code>UNTIL</code>, <code>COUNT</code>, <code>BYDAY</code>, and <code>WKST</code>. <code>BYDAY</code> is required, but can't be preceded by a number. For example, to indicate weekly on Tuesday and Thursday until September 1, 2023, use <code>RRULE:FREQ=WEEKLY;UNTIL=20230901T000000Z;BYDAY=TU,TH</code> • <code>MONTHLY</code>—supported patterns include:

RRULE Part	Supported RFC 5545 Implementation
	- BYMONTHDAY For example, to indicate monthly on the third day of the month use: <code>RRULE:FREQ=MONTHLY;BYMONTHDAY=3</code> - BYDAY and BYSETPOS For example, to indicate the last weekday of the month, use <code>RRULE:FREQ=MONTHLY;BYDAY=MO,TU,WE,TH,FR;BYSETPOS=-1</code> - BYDAY, where the BYDAY values are specified with a numeric value For example, to indicate monthly on the first Friday for 10 occurrences, use <code>RRULE:FREQ=MONTHLY;COUNT=10;BYDAY=1FR</code> - YEARLY—supported patterns include: - BYMONTH, BYDAY, and BYSETPOS For example, to indicate every year on the second Friday of January, use <code>RRULE:FREQ=YEARLY;BYMONTH=1;BYDAY=FR;BYSETPOS=2</code> - BYMONTH and BYMONTHDAY For example, to indicate every year on October 31, use <code>RRULE:FREQ=YEARLY;BYMONTH=10;BYMONTHDAY=31</code> For example, to create a maintenance pattern such as twice in May, and September on 7th and 15th; and one time in June/July/August on the 1st, use two RRULEs: <code>RRULE:FREQ=MONTHLY;BYMONTH=5,9;BYMONTHDAY=7,15</code> <code>RRULE:FREQ=MONTHLY;BYMONTH=6,7,8;BYMONTHDAY=1</code>
BYMONTH	The month. Valid values are 1 to 12.
BYMONTHDAY	The day of the month. Valid values are 1 to 31. If BYMONTHDAY is 31 and the month has fewer than 31 days, the event is created on the last day of the month.
BYDAY	A comma-separated list of days of the week. Valid values are SU, MO, TU, WE, TH, FR, SA. For RRULES with yearly or monthly frequency, BYDAY must be one of: - a single day - weekend days - weekdays - every day of the week Each BYDAY value can be preceded by an integer that indicates the nth occurrence of a specific day within the monthly or yearly RRULE. Allowed values are -1, 1, 2, 3, and 4. You can't use different numbers in the BYDAY values. For example, this RRULE isn't supported: <code>RRULE:FREQ=MONTHLY;INTERVAL=2;COUNT=10;BYDAY=1SU,-1SU</code> If BYDAY values are prefaced with a number, the RRULE can't include BYSETPOS.
WKST	Specifies the day on which the workweek starts. Valid values are MO, TU, WE, TH, FR, SA, and SU. Default value is based on the user's locale.

RRULE Part	Supported RFC 5545 Implementation
BYSETPOS	A comma-separated list of values that correspond to the nth occurrence within the set of recurrence instances specified by the rule. Valid values are -1, 1, 2, 3, or 4. Default value is 1. For example, to indicate the last weekday of the month, use: RRULE:FREQ=MONTHLY;BYDAY=MO,TU,WE,TH,FR;BYSETPOS=-1
INTERVAL	The repetition interval. Valid values are: - an integer between 1 and 999 if FREQ=DAILY - an integer between 1 and 99 if FREQ=WEEKLY or FREQ=MONTHLY 1 if FREQ=YEARLY Default value is 1.
UNTIL	Specifies the datetime in UTC format when the recurrence rule stops. The supported format is yyyyMMddTHH:mm:ssZ, for example: 20210419T083000Z. An RRULE can't contain both UNTIL and COUNT. A recurring event without either UNTIL or COUNT repeats indefinitely.
COUNT	The number of occurrences. Allowed values are 1 to 999. An RRULE can't contain both UNTIL and COUNT. A recurring event without either UNTIL or COUNT repeats indefinitely.
BYWEEKNO	Specifies a comma-separated list of values that specify weeks of the year. Valid values are: 1 to 53 -53 to -1 For example, to indicate specific weeks in a year, use: RRULE:BYWEEKNO=20,-20. This rule part can't be used when the FREQ rule part is set to anything other than YEARLY. For example, 3 represents the third week of the year. Note: Assuming a Monday week start, week 53 can only occur when Thursday is January 1 or if it's a leap year and Wednesday is January 1.
BYYEARDAY	Specifies a comma-separated list of values that specify days of the year. Valid values are: 1 to 366 -366 to -1 For example, to indicate specific days in a year, use: RRULE:BYYEARDAY=1,100,200; or, RRULE:BYYEARDAY=1,-2.

Salesforce Classic Event Series and Recurring Events

This table describes the usage of recurrence fields for Salesforce Classic recurring events. Each recurrence type must have all of its properties set. All unused properties must be set to null.

RecurrenceType Value	Properties	Example Pattern
RecursDaily	RecurrenceInterval	Every second day

RecurrenceType Value	Properties	Example Pattern
RecursEveryWeekday	RecurrenceDayOfWeekMask	Every weekday - can't be Saturday or Sunday
RecursMonthly	RecurrenceDayOfMonth RecurrenceInterval	Every second month, on the third day of the month
RecursMonthlyNth	RecurrenceInterval RecurrenceInstance RecurrenceDayOfWeekMask	Every second month, on the last Friday of the month
RecursWeekly	RecurrenceInterval RecurrenceDayOfWeekMask	Every three weeks on Wednesday and Friday
RecursYearly	RecurrenceDayOfMonth RecurrenceMonthOfYear	Every March on the 26th day of the month
RecursYearlyNth	RecurrenceDayOfWeekMask RecurrenceInstanceRecurrenceMonthOfYear	The first Saturday in every October

Attendees, Invitees, and Resources

The field `GroupEventType` indicates that event participants are included on an event. You can add a resource to an event only when the resource is available. The only attendance status that can be assigned to resources is Accepted. Events can't be saved when resources you've added aren't available.

JunctionIdList

To create an event using `JunctionIdList`, IDs are pulled from the related contacts and both the event and the `EventRelation` records are created in one API call. If the `EventRelation` fails, the event is rolled back because it's all done in a single API call.

```
public void createEventNew(Contact[] contacts) {
    String[] contactIds = new String[contacts.size()];
    for (int i = 0; i < contacts.size(); i++) {
        contactIds[i] = contacts[i].getID();
    }
    Event event = new Event();
    event.setSubject("New Event");
    event.setEventWhoIds(contactIds);
    SaveResult[] results = null;
    try {
        results = connection.create(new Event[] {
            task
        });
    } catch (ConnectionException ce) {
        ce.printStackTrace();
    }
}
```

Syncing Events with Lightning Sync

Attendee statuses (Accepted or Maybe, Declined, or No Response) sync from Microsoft® Exchange or Google to Salesforce, but not from Salesforce to Exchange or Google. Be wary of creating API flows that update attendee status in Salesforce for users set up to sync both ways. Eventually the original Exchange or Google status overrides the update made in Salesforce.

Shared Field-Level Security for Event and Task Objects

Metadata deployments for the Event object must include the field-level security for the Task object. Shared field-level security prevents each object from changing the field-level security of the associated object.

Metadata deployments that include field-level security for only one of either the Event or Task objects can cause field-level security changes to the other object that aren't reflected in the metadata.

- If field-level security is enabled for one object, then field-level security is enabled for both objects.
- If field-level security is disabled for one object, then it's disabled for both objects.



Note: A missing entry in the metadata is treated as field-level security being disabled.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

EventChangeEvent (API version 44.0)

Change events are available for the object.

EventFeed (API version 20.0)

Feed tracking is available for the object.

Task

Represents a business activity such as making a phone call or other to-do items. In the user interface, Task and Event records are collectively referred to as activities.



Note: Task fields related to calls are exclusive to Salesforce CRM Call Center. Also, `query()`, `delete()`, and `update()` aren't allowed with tasks related to more than one contact in API versions 23.0 and earlier.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Field Type
AccountId	Type reference Properties Filter, Group, Nillable, Sort Description Represents the ID of the related Account. The <code>AccountId</code> is determined as follows. If the value of <code>whatId</code> is any of the following objects, then Salesforce uses that object's <code>AccountId</code>. • Account

Field	Field Type
	• Opportunity • Contract • Custom object that is a child of Account If the value of the <code>whatId</code> field is any other object, and the value of the <code>whoId</code> field is a Contact object, then Salesforce uses that contact's <code>AccountId</code>. (If your organization uses Shared Activities, then Salesforce uses the <code>AccountId</code> of the primary contact.) Otherwise, Salesforce sets the value of the <code>AccountId</code> field to <code>null</code>. For information on IDs, see ID Field Type. This is a relationship field. Relationship Name Account Relationship Type Lookup Refers To Account
ActivityDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description Represents the due date of the task. This field has a timestamp that is always set to midnight in the Coordinated Universal Time (UTC) time zone. The timestamp is not relevant; do not attempt to alter it to accommodate time zone differences. Label is Due Date.  Note: This field can't be set or updated for a recurring task (<code>IsRecurrence</code> is <code>true</code>).
CallDisposition	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description Represents the result of a given call, for example, "we'll call back," or "call unsuccessful." Limit is 255 characters. Not subject to field-level security, available for any user in an organization with Salesforce CRM Call Center.
CallDurationInSeconds	Type int Properties Create, Filter, Group, Nillable, Sort, Update

Field	Field Type
	Description Duration of the call in seconds. Not subject to field-level security, available for any user in an organization with Salesforce CRM Call Center.
CallObject	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description Name of a call center. Limit is 255 characters. Not subject to field-level security, available for any user in an organization with Salesforce CRM Call Center.
CallType	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The type of call being answered: Inbound, Internal, or Outbound.
CompletedDateTime	Type dateTime Properties Filter, Nillable, Sort Description The date and time the task was saved with a Closed status. - For insert, if the task is saved with a Closed status the field is set. If the task is saved with an Open status the field is set to NULL. - For update, if the task is saved with a new Closed status, the field is reset. If the task is saved with a new non-closed status, the field is reset to NULL. If the task is saved with the same closed status (that is, unchanged) there is no change to the field.  Note: The status is a dynamic enum. If the Closed mapping is changed it won't cause an update of existing tasks. Only new insert/update operations are affected.
ConnectionReceivedId	Type reference Properties Filter, Group, Nillable, Sort

Field	Field Type
	Description ID of the PartnerNetworkConnection that shared this record with your organization. This field is available if you enabled Salesforce to Salesforce.
ConnectionSentId	Type reference Properties Filter, Group, Nillable, Sort Description ID of the PartnerNetworkConnection that you shared this record with. This field is available if you enabled Salesforce to Salesforce. This field is supported using API versions earlier than 15.0. In all other API versions, this field's value is null. You can use the new PartnerNetworkRecordConnection object to forward records to connections.
Description	Type textarea Properties Create, Nillable, Update Description Contains a text description of the task.
IsArchived	Type boolean Properties Defaulted on create, Filter, Group, Sort Description Indicates whether the event has been archived. The default value of this field is <code>false</code> .
IsClosed	Type boolean Properties Defaulted on create, Filter, Group, Sort Description Indicates whether the task has been completed (<code>true</code>) or not (<code>false</code>). The default value of this field is <code>false</code> . Is only set indirectly via the <code>Status</code> picklist. Label is Closed .
IsHighPriority	Type boolean Properties Defaulted on create, Filter, Group, Sort

Field	Field Type
	Description Indicates a high-priority task. This field is derived from the <code>Priority</code> field. The default value of this field is <code>false</code> .
<code>IsRecurrence</code>	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort Description Indicates whether the task is scheduled to repeat itself (<code>true</code>) or only occurs once (<code>false</code>). The default value of this field is <code>false</code> . This field is read-only on update, but not on create. If this field value is <code>true</code> , then <code>RecurrenceStartDateOnly</code> , <code>RecurrenceEndDateOnly</code> , <code>RecurrenceType</code> , and any recurrence fields associated with the given recurrence type must be populated. See Recurring Tasks .
<code>IsReminderSet</code>	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether a popup reminder has been set for the task (<code>true</code>) or not (<code>false</code>). The default value of this field is <code>false</code> .
<code>IsVisibleInSelfService</code>	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether a task associated with an object can be viewed in the Customer Portal (<code>true</code>) or not (<code>false</code>). If your organization has digital experiences enabled, tasks marked <code>IsVisibleInSelfService</code> are visible to any external user in the Experience Cloud site, as long as the user has access to the record the task was created on.
<code>OwnerId</code>	Type reference Properties Create, Defaulted on create, Filter, Group, Sort, Update Description ID of the User or Group who owns the record. Label is Assigned To ID . This field accepts Groups of type Queue only.

Field	Field Type
	In the user interface, Group IDs correspond with the queue's list view names. To create or update tasks assigned to Group, use v48.0 or later. This is a polymorphic relationship field. Relationship Name Owner Relationship Type Lookup Refers To Group, User
Priority	Type picklist Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Required. Indicates the importance or urgency of a task, such as high or low. The default value of this field is Normal.
RecurrenceActivityId	Type reference Properties Filter, Group, Nillable, Sort Description Read-only. Not required on create. ID of the main record of the recurring task. Subsequent occurrences have the same value in this field.
RecurrenceDayOfMonth	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description The day of the month in which the task repeats.
RecurrenceDayOfWeekMask	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description The day or days of the week on which the task repeats. This field contains a bitmask. The values are as follows: • Sunday = 1

Field	Field Type
	- Monday = 2 - Tuesday = 4 - Wednesday = 8 - Thursday = 16 - Friday = 32 - Saturday = 64 Multiple days are represented as the sum of their numerical values. For example, Tuesday and Thursday = 4 + 16 = 20.
RecurrenceEndDateOnly	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The last date on which the task repeats. This field has a timestamp that is always set to midnight in the Coordinated Universal Time (UTC) time zone. The timestamp is not relevant; do not attempt to alter it to accommodate time zone differences.
RecurrenceInstance	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The frequency of the recurring task. Possible values are: - First—1st - Fourth—4th - Last—last - Second—2nd - Third—3rd
RecurrenceInterval	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description The interval between recurring tasks.
RecurrenceMonthOfYear	Type picklist

Field	Field Type
	Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The month of the year in which the task repeats.
RecurrenceRegeneratedType	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description Represents what triggers a repeating task to repeat. Add this field to a page layout together with the <code>RecurrenceInterval</code> field, which determines the number of days between the triggering date (due date or close date) and the due date of the next repeating task in the series. Label is Repeat This Task. This field has the following picklist values: • None: The task doesn't repeat. • After due date: The next repeating task will be due the specified number of days after the current task's due date. • After the task is closed: The next repeating task will be due the specified number of days after the current task is closed. • (Task closed): This task, now closed, was opened as part of a repeating series.  Note: When tasks in a series are set to repeat after their due date, Salesforce doesn't create recurrences that would have been due in the past. Instead, Salesforce keeps adding the interval until a repeated task has a due date in the future. For example, suppose that someone sets a task to repeat three days after it's due. But, that person doesn't complete the task (mark it Closed) until five days after it's due. Instead of creating a task that's already overdue, Salesforce gives the new task a due date of tomorrow. This due date is equivalent to 6 days after the due date; two intervals of three days each. If that person completes the repeating task (marks it Closed) before the due date, the next task is still due three days after the due date.
RecurrenceStartDateOnly	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The date when the recurring task begins. Must be a date and time before <code>RecurrenceEndDateOnly</code>.

Field	Field Type
RecurrenceTimeZoneSidKey	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The time zone associated with the recurring task. For example, "UTC-8:00" for Pacific Standard Time.
RecurrenceType	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description Indicates how often the task repeats. For example, daily, weekly, or every nth month (where "nth" is defined in <code>RecurrenceInstance</code>).
ReminderDateTime	Type dateTime Properties Create, Filter, Nillable, Sort, Update Description Represents the time when the reminder is scheduled to fire, if <code>IsReminderSet</code> is set to <code>true</code>. If <code>IsReminderSet</code> is set to <code>false</code>, then the user may have deselected the reminder checkbox in the Salesforce user interface, or the reminder has already fired at the time indicated by the value.
Status	Type picklist Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Required. Indicates the status of the task. The default value of this field is <code>Not Started</code>. Each predefined <code>Status</code> field implies a value for the <code>IsClosed</code> flag. To obtain picklist values, query the <code>TaskStatus</code> object. Possible values are: • Completed • Deferred • In Progress • Not Started • Waiting on someone else  Note: This field can't be updated for recurring tasks (<code>IsRecurrence</code> is <code>true</code>).

Field	Field Type
Subject	Type combobox Properties Create, Filter, Nillable, Sort, Update Description The subject line of the task, such as “Call” or “Send Quote.” Limit: 255 characters.
TaskSubtype	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort Description Provides standard subtypes to facilitate creating and searching for specific task subtypes. This field isn’t updateable. TaskSubtype values: • Task • Email • LinkedIn —Available in API version 56.0 and later. • List Email • Cadence • Call  Note: The Cadence subtype is an internal value used by Sales Engagement, and can’t be set manually.
TaskWhoIds	Type JunctionIdList Properties Create, Update Description A string array of contact or lead IDs related to this task. This <code>JunctionIdList</code> field is linked to the <code>TaskWhoRelations</code> child relationship. <code>TaskWhoIds</code> is only available when the shared activities setting is enabled. The first contact or lead ID in the list becomes the primary <code>whoId</code> if you don’t specify a primary <code>whoId</code>. If you set the <code>EventWhoIds</code> field to null, all entries in the list are deleted and the value of <code>whoId</code> is added as the first entry.  Warning: Adding a <code>JunctionIdList</code> field name to the <code>fieldsToNull</code> property deletes all related junction records. This action can’t be undone.
Type	Type picklist

Field	Field Type
	Properties Create, Filter, Group, Nillable, Sort, Update Description The type of task, such as Call or Meeting.
WhatCount	Type int Properties Filter, Group, Nillable, Sort Description Available to organizations that have Shared Activities enabled. Count of related TaskRelations pertaining to WhatId. Count of the WhatId must be 1 or less.
WhatId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The WhatId represents nonhuman objects such as accounts, opportunities, campaigns, cases, or custom objects. WhatIds are polymorphic. Polymorphic means a WhatId is equivalent to the ID of a related object. The label is Related To ID. This is a polymorphic relationship field. Relationship Name What Relationship Type Lookup Refers To Account, Accreditation, AssessmentIndicatorDefinition, AssessmentTask, AssessmentTaskContentDocument, AssessmentTaskDefinition, AssessmentTaskOrder, Asset, AssetRelationship, AssignedResource, Award, BoardCertification, BusinessLicense, BusinessMilestone, BusinessProfile, Campaign, CareBarrier, CareBarrierDeterminant, CareBarrierType, CareDeterminant, CareDeterminantType, CareDiagnosis, CareInterventionType, CareMetricTarget, CareObservation, CareObservationComponent, CarePgmProvHealthcareProvider, CarePreauth, CarePreauthItem, CareProgram, CareProgramCampaign, CareProgramEligibilityRule, CareProgramEnrollee, CareProgramEnrolleeProduct, CareProgramEnrollmentCard, CareProgramGoal, CareProgramProduct, CareProgramProvider, CareProgramTeamMember, CareProviderAdverseAction, CareProviderFacilitySpecialty, CareProviderSearchableField, CareRegisteredDevice, CareRequest, CareRequestDrug, CareRequestExtension, CareRequestItem, CareSpecialty, CareSpecialtyTaxonomy, CareTaxonomy, Case, CommSubscriptionConsent, ContactEncounter, ContactEncounterParticipant, ContactRequest, Contract, CoverageBenefit, CoverageBenefitItem, CreditMemo, DelegatedAccount, DocumentChecklistItem, EnrollmentEligibilityCriteria, HealthcareFacility,

Field	Field Type
	HealthcareFacilityNetwork, HealthcarePayerNetwork, HealthcarePractitionerFacility, HealthcareProvider, HealthcareProviderNpi, HealthcareProviderSpecialty, HealthcareProviderTaxonomy, IdentityDocument, Image, IndividualApplication, Invoice, ListEmail, Location, MemberPlan, Opportunity, Order, OtherComponentTask, PartyConsent, PersonLifeEvent, PlanBenefit, PlanBenefitItem, ProcessException, Product2, ProductItem, ProductRequest, ProductRequestLineItem, ProductTransfer, PurchaserPlan, ReceivedDocument, ResourceAbsence, ReturnOrder, ReturnOrderLineItem, ServiceAppointment, ServiceResource, Shift, Shipment, ShipmentItem, Solution, Visit, VisitedParty, VolunteerProject, WorkOrder, WorkOrderLineItem
WhoCount	Type int Properties Filter, Group, Nillable, Sort Description Available to organizations that have Shared Activities enabled. Count of related TaskRelations pertaining to WhoId.
WhoId	Type reference Properties Create, Filter, Group, Nillable, Sort, Update Description The Whold represents a human such as a lead or a contact. Wholds are polymorphic. Polymorphic means a Whold is equivalent to a contact's ID or a lead's ID. The label is Name ID. If Shared Activities is enabled, the value of this field is the ID of the related lead or primary contact. If you add, update, or remove the Whold field, you might encounter problems with triggers, workflows, and data validation rules that are associated with the record. The label is Name ID. Beginning in API version 37.0, if the contact or lead ID in the whoId field is not in the TaskWhoIds list, no error occurs and the ID is added to the TaskWhoIds as the primary whoId. If whoId is set to null, an arbitrary ID from the existing TaskWhoIds list is promoted to the primary position. This is a polymorphic relationship field.
	Relationship Name Who Relationship Type Lookup Refers To Contact, Lead

Usage

Recurring Tasks

- Recurring tasks are available in API version 16.0 and later.
- After a task is created, it can't be changed from recurring to nonrecurring or vice versa.
- When a user creates a series of recurring tasks, Salesforce creates a main record and subsequent occurrences. For the main record, `IsRecurrence` is set to `true` and other fields that define the recurrence pattern are populated. The ID of the main record of the recurring task is saved in the subsequent occurrences, in the `RecurrenceActivityId` field.
- When you delete a recurring task series through the API, all open and closed task occurrences in the series are removed. However, when you delete a recurring task series through the user interface, only open tasks occurrences (`IsClosed` is `false`) in the series are removed.
- If `IsRecurrence` is `true`, then `RecurrenceStartDateOnly`, `RecurrenceEndDateOnly`, `RecurrenceType`, and any properties associated with the given recurrence type (see the following table) must be populated.
- When you change the `RecurrenceStartDateOnly` field or the recurrence pattern, all open tasks occurrences in the series are deleted and new open task occurrences are created based on the new recurrence pattern. The following fields determine the recurrence pattern: `RecurrenceType`, `RecurrenceTimeZoneSidKey`, `RecurrenceInterval`, `RecurrenceDayOfWeekMask`, `RecurrenceDayOfMonth`, `RecurrenceInstance`, and `RecurrenceMonthOfYear`.
- When you change the value of `RecurrenceEndDateOnly` to an earlier date (for example, from January 20 to January 10), all open task occurrences in the series with the `ActivityDate` value greater than the new end date value are deleted. Other open and closed task occurrences in the series are not affected.
- When you change the value of `RecurrenceEndDateOnly` to a later date (for example, from January 10 to January 20), new task occurrences are created up to the new end date. Existing open and closed tasks in the series are not affected.

This table describes the usage of recurrence fields for Salesforce Classic recurring events. Each recurrence type must have all of its properties set. All unused properties must be set to null.

RecurrenceType Value	Properties	Example Pattern
RecursDaily	RecurrenceInterval	Every second day
RecursEveryWeekday	RecurrenceDayOfWeekMask	Every weekday - can't be Saturday or Sunday
RecursMonthly	RecurrenceDayOfMonth RecurrenceInterval	Every second month, on the third day of the month
RecursMonthlyNth	RecurrenceInterval RecurrenceInstance RecurrenceDayOfWeekMask	Every second month, on the last Friday of the month
RecursWeekly	RecurrenceInterval RecurrenceDayOfWeekMask	Every three weeks on Wednesday and Friday
RecursYearly	RecurrenceDayOfMonth RecurrenceMonthOfYear	Every March on the 26th day of the month
RecursYearlyNth	RecurrenceDayOfWeekMask RecurrenceInstance RecurrenceMonthOfYear	The first Saturday in every October

JunctionIdList

The `JunctionIdList` field is now implemented in the Event and Task objects. With a single API call, it's easy to create many-to-many relationships between the Event or Task object with contacts, leads, or users.

To create a Task with related Contacts without `JunctionIdList`, you first have to create the task, then use the returned task ID to create the `TaskRelation` records. If the `TaskRelation` save call fails, error handling is your responsibility because the task has already been committed to the database.

```
public void createTasksOld(Contact[] contacts) {
    Task task = new Task();
    task.setSubject("New Task");
    SaveResult[] results = null;
    try {
        results = connection.create(new Task[] {
            task
        });
        if (results[0].isSuccess()) {
            TaskRelation[] relations = new TaskRelation[contacts.size()];
            for (int i = 0; i < contacts.length; i++) {
                relations[i] = new TaskRelation();
                relations[i].setTaskId(results[0].getID());
                relations[i].setRelationId(contacts[i].getID());
            }
            results = connection.create(relations);
        }
    } catch (ConnectionException ce) {
        ce.printStackTrace();
    }
}
```

To create a task using `JunctionIdList`, IDs are pulled from the related contacts and both the task and the `TaskRelation` records are created in one API call. If the `TaskRelation` fails, the task is rolled back because it's all done in a single API call.

```
public void createTaskNew(Contact[] contacts) {
    String[] contactIds = new String[contacts.size()];
    for (int i = 0; i < contacts.size(); i++) {
        contactIds[i] = contacts[i].getID();
    }
    Task task = new Task();
    task.setSubject("New Task");
    task.setTaskWhoIds(contactIds);
    SaveResult[] results = null;
    try {
        results = connection.create(new Task[] {
            task
        });
    } catch (ConnectionException ce) {
        ce.printStackTrace();
    }
}
```

Shared Field-Level Security for Event and Task Objects

Metadata deployments for the Task object should always include the field-level security for the Event object. Shared field-level security prevents each object from changing the field-level security of the associated object.

Metadata deployments that include field-level security for only one of either the Event or Task objects can cause field-level security changes to the other object that aren't reflected in the metadata.

- If field-level security is enabled for one object, then field-level security is enabled for both objects.
- If field-level security is disabled for one object, then it's disabled for both objects.



Note: A missing entry in the metadata is treated as field-level security being disabled.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

TaskChangeEvent (API version 44.0)

Change events are available for the object.

TaskFeed (API version 20.0)

Feed tracking is available for the object.

CHAPTER 5 Apex Reference

In this chapter ...

- [ConnectApi Namespace](#)
- [LoyaltyManagement Namespace](#)

This Apex reference goes into detail about the built-in Apex `ConnectApi` classes and interfaces.

For Apex-related objects and SOAP API calls, see [Objects and SOAP API Calls and Headers for Apex](#) in the *SOAP API Developer Guide*.

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](#).

[LoyaltyConnect Methods](#)

This is a method for `LoyaltyConnect`. Method is static.

[LoyaltyManagementConnect Methods](#)

This is a method for `LoyaltyManagementConnect`. Method is static.

[LoyaltyVoucherConnect Methods](#)

This is a method for `LoyaltyVoucherConnect`. Method is static.

[Input Classes](#)

These are Loyalty Management Apex input classes.

[Output Classes](#)

These are Loyalty Management Apex output classes.

[ConnectApi Enums Enums](#)

Enums specific to the `ConnectApi` namespace.

LoyaltyConnect Methods

This is a method for `LoyaltyConnect`. Method is static.

[getMemberBenefits\(memberId\)](#)

Get benefits for a loyalty program member.

getMemberBenefits (memberId)

Get benefits for a loyalty program member.

API Version

51.0

Requires Chatter

No

Signature

```
public static ConnectApi.MemberBenefitsOutput getMemberBenefits(String memberId)
```


Parameters

memberId

Type: String

ID of the loyalty program member.

Return Value

Type: [ConnectApi.MemberBenefitsOutput](#)

LoyaltyManagementConnect Methods

This is a method for `LoyaltyManagementConnect`. Method is static.

[enrollCorporateMember\(loyaltyProgramName, memberDetails\)](#)

Enroll corporate members in a loyalty program.

[enrollIndividualMember\(loyaltyProgramName, memberDetails\)](#)

Enroll individual members in a loyalty program.

[getMemberProfileUsingLoyaltyProgramMemberId\(loyaltyProgramName, memberId, programCurrencyName\)](#)

Get the details of a loyalty program member using the program member ID.

[getMemberProfileUsingMembershipNumber\(loyaltyProgramName, membershipNumber, programCurrencyName\)](#)

Get the details of a loyalty program member using the program membership number.

enrollCorporateMember(loyaltyProgramName, memberDetails)

Enroll corporate members in a loyalty program.

API Version

55.0

Requires Chatter

No

Signature

```
public static ConnectApi.CorporateMemberOutput enrollCorporateMember(String
loyaltyProgramName, ConnectApi.CorporateMemberInput memberDetails)
```

Parameters

loyaltyProgramName

Type: String

The name of the loyalty program in which the member is enrolled.

memberDetails

Type: [ConnectApi.CorporateMemberInput](#)

Input representation of the corporate member create request.

Return Value

Type: `ConnectApi.CorporateMemberOutput`

Example

```
public class TestCorporateEnrollment {
    Map<String, String> additionalDetails = new Map<String, String>();
    ConnectApi.MemberAccountInput account = new ConnectApi.MemberAccountInput();
    additionalDetails.put('BillingCity', 'Delhi');
    account.additionalAccountFieldValues = additionalDetails;
    account.name = 'BBBB';
    ConnectApi.CorporateMemberInput member = new ConnectApi.CorporateMemberInput();
    member.associatedAccountDetails = account;
    DateTime myDateTime = DateTime.newInstance(2021, 1, 31, 7, 8, 16);
    member.enrollmentDate = myDateTime;
    member.membershipNumber = 'abcdefgh';
    member.memberStatus = 'Active';
    ConnectApi.CorporateMemberOutput output =
    ConnectApi.LoyaltyManagementConnect.enrollCorporateMember('Program-999',member);
}
```

enrollIndividualMember(loyaltyProgramName, memberDetails)

Enroll individual members in a loyalty program.

API Version

55.0

Requires Chatter

No

Signature

```
public static ConnectApi.IndividualMemberOutput enrollIndividualMember(String
loyaltyProgramName, ConnectApi.IndividualMemberInput memberDetails)
```

Parameters

loyaltyProgramName

Type: String

The name of the loyalty program in which the member is enrolled.

memberDetails

Type: `ConnectApi.IndividualMemberInput`

Input representation of the individual member create request.

Return Value

Type: `ConnectApi.IndividualMemberOutput`

Example

```
public class TestIndividualEnrollment {
    Map<String, String> additionalDetails = new Map<String, String>();
    ConnectApi.MemberContactInput contact = new ConnectApi.MemberContactInput();
    additionalDetails.put('PersonMailingCity', 'Delhi');
    contact.additionalContactFieldValues = additionalDetails;
    contact.lastName = 'BBBB';
    ConnectApi.IndividualMemberInput member = new ConnectApi.IndividualMemberInput();
    member.associatedContactDetails = contact;
    DateTime myDateTime = DateTime.newInstance(2021, 1, 31, 7, 8, 16);
    member.enrollmentDate = myDateTime;
    member.membershipNumber = 'abcdefg';
    member.memberStatus = 'Active';
    ConnectApi.IndividualMemberOutput output =
    ConnectApi.LoyaltyManagementConnect.enrollIndividualMember('Program1',member);
}
```

getMemberProfileUsingLoyaltyProgramMemberId(loyaltyProgramName, memberId, programCurrencyName)

Get the details of a loyalty program member using the program member ID.

API Version

55.0

Requires Chatter

No

Signature

```
public static ConnectApi.MemberDetailsOutput
getMemberProfileUsingLoyaltyProgramMemberId(String loyaltyProgramName, String memberId,
String programCurrencyName)
```

Parameters

loyaltyProgramName

Type: String

The name of the loyalty program.

memberId

Type: String

The ID of the loyalty program member.The ID of the loyalty program member.

programCurrencyName

Type: String

The name of the loyalty program currency associated with the member.

Use this parameter to get the details of the member's points-related information for a specific currency.

Return Value

Type: `ConnectApi.MemberDetailsOutput`

Example

```
public class TestMemberId {  
    ConnectApi.MemberDetailsOutput details =  
    ConnectApi.LoyaltyManagementConnect.getMemberProfileUsingLoyaltyProgramMemberId('program1',  
    '01Mxx00000007X1EAI', 'ruble');  
}
```

getMemberProfileUsingMembershipNumber(loyaltyProgramName, membershipNumber, programCurrencyName)

Get the details of a loyalty program member using the program membership number.

API Version

55.0

Requires Chatter

No

Signature

```
public static ConnectApi.MemberDetailsOutput getMemberProfileUsingMembershipNumber(String  
loyaltyProgramName, String membershipNumber, String programCurrencyName)
```

Parameters

loyaltyProgramName

Type: String

The name of the loyalty program.

membershipNumber

Type: String

The membership number of the loyalty program member.

programCurrencyName

Type: String

The name of the loyalty program currency associated with the member.

Use this parameter to get the details of the member's points-related information for a specific currency.

Return Value

Type: `ConnectApi.MemberDetailsOutput`

Example

```
public class TestMembershipNumber {  
    ConnectApi.MemberDetailsOutput details =  
    ConnectApi.LoyaltyManagementConnect.getMemberProfileUsingMembershipNumber('program1',  
    'Member15', 'ruble');  
}
```

LoyaltyVoucherConnect Methods

This is a method for `LoyaltyVoucherConnect`. Method is static.

`redeem(programName, membershipNumber, voucherCode)`

Redeem a voucher for a program member. The status of the voucher gets updated to redeemed and the use date gets updated to the current date.

`redeem(programName, membershipNumber, voucherCode, amount)`

Redeem partial value of a voucher for a program member. The status of the voucher gets updated to redeemed and the use date gets updated to the current date.

`redeem(programName, membershipNumber, voucherCode, voucherNumber)`

Redeem the single-use voucher for a loyalty program member. The status of the voucher gets updated to redeemed and the use date gets updated to the current date.

`redeem(programName, membershipNumber, voucherCode, voucherNumber, amount)`

Redeem partial value of a voucher for a program member. The status of the voucher gets updated to redeemed and the use date gets updated to the current date.

`redeem(programName, membershipNumber, voucherCode)`

Redeem a voucher for a program member. The status of the voucher gets updated to redeemed and the use date gets updated to the current date.

API Version

51.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.RedeemVoucherOutput redeem(String programName, String membershipNumber, String voucherCode)
```

Parameters

programName

Type: String

The loyalty program name.

membershipNumber

Type: String

The loyalty program membership number.

voucherCode

Type: String

The voucher code to redeem.



Note: The voucher definition associated with the voucher code must be active to redeem the voucher.

Status Code

Returns HTTP 201 on success.

redeem(programName, membershipNumber, voucherCode, amount)

Redeem partial value of a voucher for a program member. The status of the voucher gets updated to redeemed and the use date gets updated to the current date.

API Version

53.0

Requires Chatter

Yes

Signature

```
public static ConnectApi.RedeemVoucherOutput redeem(String programName, String membershipNumber, String voucherCode, ConnectApi.RedeemVoucherInput amount)
```

Parameters

programName

Type: String

The loyalty program name.

membershipNumber

Type: String

The loyalty program membership number.

voucherCode

Type: String

The voucher code to redeem.



Note: The voucher definition associated with the voucher code must be active to redeem the voucher.

amount

Type: [ConnectApi.RedemVoucherInput](#) on page 318

The value of the voucher to be redeemed.

Return Value

Type: [ConnectApi.RedemVoucherOutput](#) on page 319

Example

```
public class TestRedeemVoucher {
    ConnectApi.RedemVoucherInput redeemInput = new ConnectApi.RedemVoucherInput();
    redeemInput.redeemAmount = 10.2;
    ConnectApi.RedemVoucherOutput output =
ConnectApi.LoyaltyVoucherConnect.redeem('DemoProgram', 'M1234', 'VC001', redeemInput);
}
```

redeem(programName, membershipNumber, voucherCode, voucherNumber)

Redeem the single-use voucher for a loyalty program member. The status of the voucher gets updated to redeemed and the use date gets updated to the current date.

API Version

55.0

Requires Chatter

No

Signature

```
public static ConnectApi.RedemVoucherOutput redeem(String programName, String
membershipNumber, String voucherCode, String voucherNumber)
```

Parameters

programName

Type: String

Description

membershipNumber

Type: String

Description

voucherCode

Type: String

Description

voucherNumber

Type: String

Description

Return Value

Type: [ConnectApi.RedeemVoucherOutput](#)

Example

```
public class TestRedeemVoucher {  
    ConnectApi.RedeemVoucherOutput output =  
    ConnectApi.LoyaltyVoucherConnect.redeem('DemoProgram', 'M1234', 'VC001', '00000006');  
}
```

redeem(programName, membershipNumber, voucherCode, voucherNumber, amount)

Redeem partial value of a voucher for a program member. The status of the voucher gets updated to redeemed and the use date gets updated to the current date.

API Version

55.0

Requires Chatter

No

Signature

```
public static ConnectApi.RedeemVoucherOutput redeem(String programName, String  
membershipNumber, String voucherCode, String voucherNumber, ConnectApi.RedeemVoucherInput  
amount)
```

Parameters

programName

Type: String

The loyalty program name.

membershipNumber

Type: String

The loyalty program membership number.

voucherCode

Type: String

The voucher code to redeem.



Note: The voucher definition associated with the voucher code must be active to redeem the voucher.

voucherNumber

Type: String

The auto-generated voucher number.

amount

Type: [ConnectApi.RedeemVoucherInput](#)

The value of the voucher to be redeemed.

Return Value

Type: [ConnectApi.RedeemVoucherOutput](#)

Example

```
public class TestRedeemVoucher {
    ConnectApi.RedeemVoucherInput redeemInput = new ConnectApi.RedeemVoucherInput();
    redeemInput.redeemAmount = 10.2;
    ConnectApi.RedeemVoucherOutput output =
ConnectApi.LoyaltyVoucherConnect.redeem('DemoProgram', 'M1234', 'VC001', '00000006',
redeemInput);
}
```

Input Classes

These are Loyalty Management Apex input classes.

[ConnectApi.CorporateMemberInput](#)

Input representation of the corporate member create request.

[ConnectApi.IndividualMemberInput](#)

Input representation of the individual member create request.

[ConnectApi.MemberAccountInput](#)

Input representation of the loyalty member account.

[ConnectApi.MemberContactInput](#)

Input representation of the loyalty member contact.

[ConnectApi.RedeemVoucherInput](#)

Input representation to redeem amount from a voucher.

ConnectApi.CorporateMemberInput

Input representation of the corporate member create request.

Property	Type	Description	Required or Optional	Available Version
additionalMemberFieldValues	Map<String,String>	The custom attributes of the member in the loyalty program member record. • Keys • Values	Optional	55.0
associatedAccountDetails	ConnectApi.AccountDetails	The account details of the member.	Required	55.0
canReceivePartnerPromotions	Boolean	Indicates whether the member can receive partner promotions for the program (<code>true</code>) or not (<code>false</code>).	Optional	55.0
canReceivePromotions	Boolean	Indicates whether the member can receive promotions for the program (<code>true</code>) or not (<code>false</code>).	Optional	55.0
createTransactionJournals	Boolean	Indicates whether to create the transaction journal records (<code>true</code>) or not (<code>false</code>). The transaction journals records are created to: • Credit points to the member for enrolling in the loyalty program. • Credit points to the member who has referred the corporate member.	Optional	55.0
enrollmentChannel	ConnectApi.EnrollmentChannelResource	The channel used by members to enroll into the loyalty program. Possible values are: • CallCenter • Email • Franchise • Mobile • Partner • Pos • Print • Social • Store • Web	Optional	55.0

Property	Type	Description	Required or Optional	Available Version
enrollmentDate	Datetime	The date on which a member is enrolled. The format is: YYYY-MM-DDTHH:MM:SS.	Required	55.0
memberStatus	String	The status of the member. Possible values are: - Active - Inactive - Custom value	Required	55.0
membershipEndDate	Datetime	The date on which the loyalty program member's membership ends.	Optional	55.0
membershipNumber	String	The unique membership number of the loyalty program member.	Required	55.0
referredBy	String	The membership number of the loyalty program member who has referred the member.	Optional	55.0
transactionJournalStatementFrequency	ConnectApi.StatementFrequencyResource	The frequency at which transaction journal statement must be delivered to the member. Possible values are: - Monthly - Quarterly	Optional	55.0
transactionJournalStatementMethod	ConnectApi.StatementMethodResource	The method used to deliver transaction journal statements to the member. Possible values are: - Email - Mail	Optional	55.0

ConnectApi.IndividualMemberInput

Input representation of the individual member create request.

Property	Type	Description	Required or Optional	Available Version
additionalMemberFieldValues	Map<String,String>	The custom field values in the loyalty program member record. - Keys - Values	Required	55.0

Property	Type	Description	Required or Optional	Available Version
associatedContactDetails	ContactDetails ContactDetails	The contact record of the individual type member.	Required	55.0
canReceivePartnerPromotions	Boolean	Indicates whether the member can receive partner promotions for the program (<code>true</code>) or not (<code>false</code>).	Optional	55.0
canReceivePromotions	Boolean	Indicates whether the member can receive promotions for the program (<code>true</code>) or not (<code>false</code>).	Optional	55.0
createTransactionJournals	Boolean	Indicates whether to create the transaction journal records (<code>true</code>) or not (<code>false</code>). The transaction journal records are created to: • Credit points to the individual member for enrolling in the loyalty program. • Credit points to the member who has referred the individual member.	Optional	55.0
enrollmentChannel	ConnectApi.EnrollmentChannelResource	The channel used by members to enroll into the loyalty program. Possible values are: • <code>CallCenter</code> • <code>Email</code> • <code>Franchise</code> • <code>Mobile</code> • <code>Partner</code> • <code>Pos</code> • <code>Print</code> • <code>Social</code> • <code>Store</code> • <code>Web</code>	Optional	55.0
enrollmentDate	Datetime	The date on which a member is enrolled. The format is: <code>YYYY-MM-DDTHH:MM:SS</code> .	Required	55.0
memberStatus	String	The status of the member. Possible values are: • <code>Active</code> • <code>Inactive</code> • <code>Custom value</code>	Required	55.0

Property	Type	Description	Required or Optional	Available Version
membershipEndDate	Datetime	The date on which the loyalty program member's membership ends.	Optional	55.0
membershipNumber	String	The unique membership number of the loyalty program member.	Required	55.0
referredBy	String	The membership number of the loyalty program member who has referred the member.	Optional	55.0
relatedCorporateMembershipNumber	String	The external corporate membership of the member.	Optional	55.0
transactionJournalStatementFrequency	ConnectApi.StatementFrequencyResource	The frequency at which transaction journal statement must be delivered to the member. Possible values are: - Monthly - Quarterly	Optional	55.0
transactionJournalStatementMethod	ConnectApi.StatementMethodResource	The method used to deliver transaction journal statements to the member. Possible values are: - Email - Mail	Optional	55.0

ConnectApi.MemberAccountInput

Input representation of the loyalty member account.

Property	Type	Description	Required or Optional	Available Version
additionalAccountFieldValues	Map<String,String>	The additional attribute values that must be saved in the account record.	Optional	55.0
allowDuplicateRecords	Boolean	Indicates whether a duplicate member record can be created when a record with identical details already exists (<code>true</code>) or not (<code>false</code>).	Optional	55.0
name	String	The name of the account.	Required	55.0
phone	String	The phone number associated with the account.	Optional	55.0
website	String	The website associated with the account.	Optional	55.0

ConnectApi.MemberContactInput

Input representation of the loyalty member contact.

Property	Type	Description	Required or Optional	Available Version
<code>additionalContactFieldValues</code>	<code>Map<String,String></code>	The additional attributes that must be saved in the person account.	Optional	55.0
<code>allowDuplicateRecords</code>	<code>Boolean</code>	Indicates whether a duplicate record can be created for a member when an existing person account record already exists (<code>true</code>) or not (<code>false</code>).	Optional	55.0
<code>email</code>	<code>String</code>	The email ID of the contact.	Optional	55.0
<code>firstName</code>	<code>String</code>	The first name of the contact.	Optional	55.0
<code>lastName</code>	<code>String</code>	The last name of the contact.	Required	55.0

ConnectApi.RedeemVoucherInput

Input representation to redeem amount from a voucher.

Property	Type	Description	Required or Optional	Available Version
<code>currencyIsoCode</code>	<code>String</code>	Three-letter ISO 4217 currency code of the monetary currency in which the voucher is redeemed.	Optional	55.0
<code>redeemAmount</code>	<code>Double</code>	The value of the voucher to be redeemed. Specify this amount when the associated voucher definition is partially redeemable and is of the type Fixed Value.	Optional	53.0

Output Classes

These are Loyalty Management Apex output classes.

[ConnectApi.MemberBenefitsOutput](#)

Output representation of all benefits available for a member.

[ConnectApi.MemberBenefitOutput](#)

Output representation of an individual benefit available for a member.

[ConnectApi.RedeemVoucherOutput](#)

Output representation of a voucher redemption.

[ConnectApi.CorporateMemberOutput](#)

Output representation of the corporate member enrollment create request.

[ConnectApi.IndividualMemberOutput](#)

Output representation of the individual member create request.

[ConnectApi.MemberAccountOutput](#)

Output representation of the details of a loyalty program member's associated account record.

[ConnectApi.MemberContactOutput](#)

Output representation of the details of a loyalty program member's associated contact record.

[ConnectApi.MemberDetailsOutput](#)

Output representation of the details of a loyalty program member.

[ConnectApi.MemberCurrencyOutput](#)

Output representation of the details of the loyalty program member's associated currencies.

[ConnectApi.MemberTierOutput](#)

Output representation of the details of the loyalty program member's associated tiers.

[ConnectApi.TransactionJournalOutput](#)

Output representation of the transaction journal records.

ConnectApi.MemberBenefitsOutput

Output representation of all benefits available for a member.

Property Name	Type	Description	Available Version
memberBenefits	List< ConnectApi.MemberBenefitOutput >	List of all benefits available for a member.	51.0

ConnectApi.MemberBenefitOutput

Output representation of an individual benefit available for a member.

Property Name	Type	Description	Available Version
benefitId	String	ID of the benefit available for the member.	51.0
benefitName	String	The name of the benefit available for the member.	51.0
benefitTypeId	String	ID of the associated benefit type record.	51.0
benefitTypeName	String	The name of the associated benefit type.	51.0
createdRecordId	String	ID of the record created during benefit assignment.	51.0
createdRecordName	String	The display name of the record created during benefit assignment.	51.0
isActive	Boolean	Indicates whether the benefit is active.	51.0

ConnectApi.RedeemVoucherOutput

Output representation of a voucher redemption.

Property Name	Type	Description	Available Version
remainingAmount	Double	The voucher value after redemption.  Note: If the complete value of the voucher is redeemed, the remaining amount is zero.	53.0
currencyIsoCode	String	Three-letter ISO 4217 currency code of the remaining voucher value.  Note: If multicurrency isn't enabled, the currency code is set to the default currency code of an org.	56.0

ConnectApi.CorporateMemberOutput

Output representation of the corporate member enrollment create request.

Property Name	Type	Description	Available Version
accountId	String	The ID of the account record associated with the loyalty program member record.	55.0
loyaltyProgramMemberId	String	The ID of the loyalty program member record created for the corporate member.	55.0
loyaltyProgramName	String	The name of the loyalty program in which the corporate member is enrolled.	55.0
membershipNumber	String	The membership number created for the loyalty program member record.	55.0
transactionJournals	ConnectApi.TransactionJournalOutput	The list of the transaction journal records created after the request is successful.	55.0

ConnectApi.IndividualMemberOutput

Output representation of the individual member create request.

Property Name	Type	Description	Available Version
contactId	String	The ID of the loyalty program member record created for the individual member.	55.0
loyaltyProgramMemberId	String	The ID of the contact record associated with the loyalty program member record.	55.0
loyaltyProgramName	String	The name of the loyalty program in which the individual member is enrolled.	55.0
membershipNumber	String	The membership number created for the loyalty program member record.	55.0

Property Name	Type	Description	Available Version
transactionJournals	ConnectApi.TransactionJournalOutput	The list of the transaction journal records created after the request is successful.	55.0

ConnectApi.MemberAccountOutput

Output representation of the details of a loyalty program member's associated account record.

Property Name	Type	Description	Available Version
accountId	String	The ID of the associated account record.	55.0
name	String	The name of the associated account.	55.0

ConnectApi.MemberContactOutput

Output representation of the details of a loyalty program member's associated contact record.

Property Name	Type	Description	Available Version
contactId	String	The ID of the associated contact record.	55.0
email	String	The email of the associated contact.	55.0
firstName	String	The first name of the associated contact.	55.0
lastName	String	The last name of the associated contact.	55.0

ConnectApi.MemberDetailsOutput

Output representation of the details of a loyalty program member.

Property Name	Type	Description	Available Version
additionalLoyaltyProgramMemberFields	Map<String, Object>	The details of the custom fields added in the loyalty program member object.	55.0
associatedAccount	ConnectApi.MemberAccountOutput	The details of the account record associated with the loyalty program member.	55.0
associatedContact	ConnectApi.MemberContactOutput	The details of the contact record associated with the loyalty program member.	55.0
canReceivePartnerPromotions	Boolean	Indicates whether the loyalty program member can receive partner promotions (<code>true</code>) or not (<code>false</code>).	55.0
canReceivePromotions	Boolean	Indicates whether the loyalty program member can receive promotions <code>true</code>) or not (<code>false</code>).	55.0
enrollmentChannel	String	The channel used by the member to enroll in the loyalty program.	55.0

Property Name	Type	Description	Available Version
enrollmentDate	Datetime	The date on which the member enrolled in the loyalty program.	55.0
groupCreatedByMember	String	The membership number of the loyalty program member who created the group.	55.0
groupName	String	The name of the group to which the member belongs.	55.0
lastActivityDate	Datetime	The date on which the member performed their last activity.	55.0
loyaltyProgramMemberId	String	The ID of the loyalty program member.	55.0
loyaltyProgramName	String	The name of the loyalty program.	55.0
memberCurrencies	ContactApiMemberCurrenciesOutput	The currencies associated with the loyalty program member.	55.0
memberStatus	String	The status of the loyalty program member.	55.0
memberTiers	ContactApiMemberTiersOutput	The tiers associated with the loyalty program member.	55.0
memberType	String	The type of loyalty program member.	55.0
membershipEndDate	Datetime	The date on which the membership ends.	55.0
membershipLastRenewalDate	Datetime	The date on which the loyalty program member last renewed their membership.	55.0
membershipNumber	String	The membership number of the loyalty program member.	55.0
referredBy	String	The membership number of the loyalty program member who referred the member to the loyalty program.	55.0
relatedCorporateMembershipNumber	String	The membership number of the Corporate type loyalty program member associate with the member.	55.0
transactionJournalStatementFrequency	String	The frequency at which transaction journal statements must be delivered to the member.	55.0
transactionJournalStatementLastGeneratedDate	Datetime	The date on which the member's last transaction journal statement was generated.	55.0
transactionJournalStatementMethod	String	The method used to deliver transaction journal statements to the member.	55.0

ConnectApi.MemberCurrencyOutput

Output representation of the details of the loyalty program member's associated currencies.

Property Name	Type	Description	Available Version
additionalLoyaltyMemberCurrencyFields	Map<String, Object>	The details of the custom fields added in the Loyalty Member Currency object.	55.0
escrowPointsBalance	Double	The escrow points balance of the member for the currency.	55.0
expirablePoints	Double	The number of points that the member hasn't redeemed for the currency.	55.0
lastAccrualProcessedDate	Datetime	The date on which the member's accruals for the currency were last processed.	55.0
lastEscrowProcessedDate	Datetime	The date on which the member's escrow points were last rolled over to their points balance for the currency.	55.0
lastExpirationProcessRunDate	Datetime	The date on which the member's points were last expired for the currency.	55.0
lastPointsAggregationDate	Datetime	The date on which the member's points for the fixed type non-qualifying currency were last aggregated.	55.0
lastPointsResetDate	Datetime	The date on which the member's points for the currency were last reset.	55.0
loyaltyMemberCurrencyName	String	The name of the Loyalty Member Currency record.	55.0
loyaltyProgramCurrencyId	String	The ID of the loyalty program currency associated with the member.	55.0
loyaltyProgramCurrencyName	String	The name of the loyalty program currency associated with the member.	55.0
memberCurrencyId	String	The ID of the loyalty member currency record.	55.0
nextQualifyingPointsResetDate	Datetime	The date on which the member's points balance for the qualifying currency is reset next.	55.0
pointsBalance	Double	The points balance of the member for the currency.	55.0
qualifyingPointsBalanceBeforeReset	Double	The last point balance of the member before the member's points balance was reset to zero for the qualifying currency.	55.0
totalEscrowPointsAccrued	Double	The total number of points that the member has accrued in escrow for the currency.	55.0
totalEscrowRolloverPoints	Double	The total number of points that has rolled over from the member's escrow point balance to the member's points balance for the currency.	55.0

Property Name	Type	Description	Available Version
totalPointsAccrued	Double	The total number of points the member has accrued for the currency.	55.0
totalPointsExpired	Double	The total number of points that have expired for the member.	55.0
totalPointsRedeemed	Double	The total number of points the member has redeemed for the currency.	55.0

ConnectApi.MemberTierOutput

Output representation of the details of the loyalty program member's associated tiers.

Property Name	Type	Description	Available Version
additionalLoyaltyMemberTierFields	Map<String, Object>	The details of the custom fields added in the Loyalty Member Tier object.	55.0
areTierBenefitsAssigned	Boolean	Indicates whether the benefits associated with the tier have been assigned to the member (<code>true</code>) or not (<code>false</code>).	55.0
loyaltyMemberTierId	String	The tier ID associated with the loyalty program member.	55.0
loyaltyMemberTierName	String	The name of the loyalty member tier.	55.0
tierChangeReason	String	The reason why the member's was assigned the current tier.	55.0
tierChangeReasonType	String	The reason type why the member's was assigned the current tier.	55.0
tierEffectiveDate	Datetime	The date on which the member was assigned the tier.	55.0
tierExpirationDate	Datetime	The date on which the member's current tier expires.	55.0
tierGroupId	String	The ID of the tier group associated with the Loyalty Tier record.	55.0
tierGroupName	String	The name of the tier group.	55.0
tierId	String	The ID of the loyalty tier record.	55.0
tierSequenceNumber	Integer	The sequence number of the tier.	55.0

ConnectApi.TransactionJournalOutput

Output representation of the transaction journal records.

Property Name	Type	Description	Available Version
activityDate	Datetime	The date and time of when the transaction journal record is created.	55.0
journalSubType	String	The journal subtype of the transaction journal record.	55.0
journalType	String	The journal type of the transaction journal record.	55.0
loyaltyProgram	String	The loyalty program associated with the transaction journal record.	55.0
loyaltyProgramMember	String	The membership number of the loyalty program member associated with the transaction journal record.	55.0
referredMember	String	The membership number of the loyalty program member who has referred the member.	55.0
status	ConnectApi. JournalStatusResource	The status of the transaction journal record. Possible values are: - Cancelled - Error - Pending - Processed	55.0
transactionJournalId	String	The ID of the transaction journal record created after the request is successful.	55.0

ConnectApi Enums 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. EnrollmentChannelResource	Type of loyalty member enrollment channel. - CallCenter - Email - Franchise - Mobile - Partner - Pos - Print - Social - Store

Enum	Description
	• Web
ConnectApi. JournalStatusResource	Type of journal transaction status. • Cancelled • Error • Pending • Processed
ConnectApi. StatementFrequencyResource	Type of statement generation frequency. • Monthly • Quaterly
ConnectApi. StatementMethodResource	Type of statement delivery method. • Email • Mail

LoyaltyManagement Namespace

The `LoyaltyManagement` namespace provides Apex classes and methods to run the Loyalty Management actions.

Use the developer console to invoke these classes and methods. For more information, see [Open the Developer Console](#).



Tip: You can also choose to use the classes and methods of the `Invocable` namespace to call the Loyalty Management invocable actions from Apex.

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

[CdpBasedLoyaltyPromotionInput Class](#)

Use this class to retrieve promotions that are eligible for a member based on the member's market segment. This class receives the input parameter values for the Data Cloud based Loyalty Promotion action.

[CdpBasedLoyaltyPromotionInputBuilder Class](#)

Contains methods to build an instance of the `LoyaltyManagement.CdpBasedLoyaltyPromotionInput` class.

[CdpBasedLoyaltyPromotionOutput Class](#)

Represents the promotion records that the member is eligible for based on the member's Data Cloud market segment.

[ChangeTierInput Class](#)

Use this class to change the tier for a given loyalty program member. This class receives the input parameter values for the Change Tier action.

[ChangeTierInputBuilder Class](#)

Contains methods to build an instance of the `LoyaltyManagement.ChangeTierInput` class.

[ChangeTierOutput Class](#)

Represents the changed tier details for a loyalty program member.

[CreditPointsOutput Class](#)

Represents the details of points credited to a loyalty program member's balance.

[DebitPointsOutput Class](#)

Represents the details of points debited from a loyalty program member's balance.

[GetMemberActiveSegmentsInput Class](#)

Use this class to retrieve active Data Cloud market segments that a loyalty program member is part of. This class receives the input parameter values for the Get Member's Active Segments action.

[GetMemberActiveSegmentsInputBuilder Class](#)

Contains methods to build an instance of the `LoyaltyManagement.GetMemberActiveSegmentsInput` class.

[GetMemberActiveSegmentsOutput Class](#)

Represents the list of the active segment IDs that the specified loyalty program member is a part of.

[IssueVoucherInput Class](#)

Use this class to issue a voucher to a loyalty program member. This class receives the input parameter values for the Issue Voucher action.

[IssueVoucherInputBuilder Class](#)

Contains methods to build an instance of the `LoyaltyManagement.IssueVoucherInput` class.

[IssueVoucherOutput Class](#)

Represents the voucher ID issued to a loyalty program member.

[LoyaltyActionError Class](#)

A class that contains the errors related to the execution of a Loyalty Management action.

[LoyaltyActionResult Class](#)

A class that contains the result returned by a loyalty action.

[LoyaltyPromotionInput Class](#)

Use this class to retrieve active promotions that are eligible for a transaction journal. The action can process accrual or redemption transactions. This class receives the input parameter values for the Loyalty Promotion action.

[LoyaltyPromotionInputBuilder Class](#)

Contains methods to build an instance of the `LoyaltyManagement.LoyaltyPromotionInput` class.

[LoyaltyPromotionOutput Class](#)

Represents the active promotions based on the specified transaction journal.

[LoyaltyResources Class](#)

A class that executes a loyalty action and returns a result. Takes the input classes as input and returns the objects of output classes.

[MemberPointBalanceInput Class](#)

Use this class to retrieve loyalty points balance for a given loyalty program member. This class receives the input parameter values for the Get Member Points action.

[MemberPointBalanceInputBuilder Class](#)

Contains methods to build an instance of the `LoyaltyManagement.MemberPointsBalanceInput` class.

[MemberPointBalanceOutput Class](#)

Represents the loyalty points balance for a specified loyalty program member.

[MemberTierInput Class](#)

Use this class to retrieve the current tier for a specified loyalty program member. This class receives the input parameter values for the Get Member Tier action.

[MemberTierInputBuilder Class](#)

Contains methods to build an instance of the `LoyaltyManagement.MemberTierInput` class.

[MemberTierOutput Class](#)

Represents the tier details for a specified loyalty program member.

[PointsInput Class](#)

Use this class to credit loyalty points to a loyalty program member's balance or debit loyalty points from a loyalty program member's balance. This class receives the input parameter values for the Credit and Debit point actions.

[PointsInputBuilder Class](#)

Contains methods to build an instance of the `LoyaltyManagement.PointsInput` class.

[TransferMemberPointsToGroupsInput Class](#)

Use this class to transfer points from an individual member or a corporate member to the member's associated group. This class receives the input parameter values for the Transfer Member Points To Groups action.

[TransferMemberPointsToGroupsInputBuilder Class](#)

Contains methods to build an instance of the `LoyaltyManagement.TransferMemberPointsToGroupsInput` class.

[UpdateCumulativeUsageCompletedInput Class](#)

Use this class to update the progress made by a loyalty program member towards attaining a Cumulative type promotion. The action creates a Loyalty Program Member Promotion record if the promotion isn't associated with the member. This class receives the input parameter values for the Update Cumulative Usage Completed action.

[UpdateCumulativeUsageCompletedOutput Class](#)

Represents the details of the progress made by a loyalty program member towards attaining a Cumulative type promotion.

CdpBasedLoyaltyPromotionInput Class

Use this class to retrieve promotions that are eligible for a member based on the member's market segment. This class receives the input parameter values for the Data Cloud based Loyalty Promotion action.

Namespace

[LoyaltyManagement](#)

Example

```
public static void getCdpBasedLoyaltyPromotionsTest() {
    //...Create a list with LoyaltyManagement.CdpBasedLoyaltyPromotionInput as a datatype
    List<LoyaltyManagement.CdpBasedLoyaltyPromotionInput> loyaltyPromotionInputList = new
    List<LoyaltyManagement.CdpBasedLoyaltyPromotionInput>();

    //...Create an instance of the list object and set the input values
    loyaltyPromotionInputList.add(new LoyaltyManagement.CdpBasedLoyaltyPromotionInput input
    = new LoyaltyManagement.CdpBasedLoyaltyPromotionInputBuilder()
        .setMemberId('01MRM0000002BD12AM')
        .setPromotions(new List<SObject>{'0c8R000000002MqIAI', '0c8R000000002PqIAI'})
        .build());

    //...Pass the list that contains the input values to the
```



```

LoyaltyResources.getLoyaltyPromotionBasedOnSalesforceCDP() function.
    //...Store the output returned by the function in the CdpBasedLoyaltyPromotionOutput
    object
    List<LoyaltyManagement.CdpBasedLoyaltyPromotionOutput> results =
    LoyaltyManagement.LoyaltyResources.getLoyaltyPromotionBasedOnSalesforceCDP(loyaltyPromotionInputList);

    System.debug('Get Cdp Based Promotions:'+ results.get(0).isSuccess());
}

```

CdpBasedLoyaltyPromotionInputBuilder Class

Contains methods to build an instance of the `LoyaltyManagement.CdpBasedLoyaltyPromotionInput` class.

Namespace

[LoyaltyManagement](#)

[CdpBasedLoyaltyPromotionInputBuilder Methods](#)

CdpBasedLoyaltyPromotionInputBuilder Methods

The following are methods for `CdpBasedLoyaltyPromotionInputBuilder`.

[build\(\)](#)

Returns an instance of the `LoyaltyManagement.CdpBasedLoyaltyPromotionInput` object.

[setMemberId\(memberId\)](#)

Represents the ID of a loyalty program member for whom promotions must be filtered. This is a required field. Ensure that the member is part of a Data Cloud market segment.

[setPromotions\(promotions\)](#)

Represents the promotion records to filter the promotions that are applicable to the member's market segment.

build()

Returns an instance of the `LoyaltyManagement.CdpBasedLoyaltyPromotionInput` object.

Signature

```
public LoyaltyManagement.CdpBasedLoyaltyPromotionInput build()
```

Return Value

Type: [LoyaltyManagement.CdpBasedLoyaltyPromotionInput](#)

setMemberId(memberId)

Represents the ID of a loyalty program member for whom promotions must be filtered. This is a required field. Ensure that the member is part of a Data Cloud market segment.

Signature

```
public LoyaltyManagement.CdpBasedLoyaltyPromotionInputBuilder setMemberId(String  
memberId)
```

Parameters

memberId
Type: String

Return Value

Type: [LoyaltyManagement.CdpBasedLoyaltyPromotionInputBuilder](#)

setPromotions(promotions)

Represents the promotion records to filter the promotions that are applicable to the member's market segment.

Signature

```
public LoyaltyManagement.CdpBasedLoyaltyPromotionInputBuilder setPromotions(List<SObject>  
promotions)
```

Parameters

promotions
Type: List<SObject>

Return Value

Type: [LoyaltyManagement.CdpBasedLoyaltyPromotionInputBuilder](#)

CdpBasedLoyaltyPromotionOutput Class

Represents the promotion records that the member is eligible for based on the member's Data Cloud market segment.

Namespace

[LoyaltyManagement](#)

[CdpBasedLoyaltyPromotionOutput Methods](#)

CdpBasedLoyaltyPromotionOutput Methods

The following are methods for [CdpBasedLoyaltyPromotionOutput](#).

[getErrors\(\)](#)

If an error occurred, returns a list providing the error code and description.

getPromotions()

Returns a List of sObject data type that contains the promotion records.

isSuccess()

Returns a Boolean that is set to true if the Data Cloud based loyalty promotion operation was successful for this object, false otherwise.

getErrors()

If an error occurred, returns a list providing the error code and description.

Signature

```
public List<LoyaltyManagement.LoyaltyActionError> getErrors()
```

Return Value

Type: List<[LoyaltyManagement.LoyaltyActionError](#)>

getPromotions()

Returns a List of sObject data type that contains the promotion records.

Signature

```
public List<SObject> getPromotions()
```

Return Value

Type: List<SObject>

isSuccess()

Returns a Boolean that is set to true if the Data Cloud based loyalty promotion operation was successful for this object, false otherwise.

Signature

```
public Boolean isSuccess()
```

Return Value

Type: Boolean

ChangeTierInput Class

Use this class to change the tier for a given loyalty program member. This class receives the input parameter values for the Change Tier action.

Namespace

[LoyaltyManagement](#)

Example

```
public void changeTierTest(){
    //...Create a list with LoyaltyManagement.ChangeTierInput as a datatype
    List<LoyaltyManagement.ChangeTierInput> changeTierList = new
    List<LoyaltyManagement.ChangeTierInput>();

    //...Create an instance of the list object and set the input values
    changeTierList.add(new LoyaltyManagement.ChangeTierInputBuilder()
        .setLoyaltyProgramMemberId('0lcxx00006000001AAA')
        .setProgramName('baProgram')
        .setTierGroupName('QualifyingGroup')
        .setTargetTierName('Gold')
        .setChangeReason('Reason for change.')
        .build());

    //...Pass the list that contains the input values to the LoyaltyResources.changeTier()
    function.
    //...Store the output returned by the function in the ChangeTierOutput object
    List<LoyaltyManagement.ChangeTierOutput> changeTierActionResults =
    LoyaltyManagement.LoyaltyResources.changeTier(changeTierList);
    System.debug('Change Tier Successful Operation: '+
    changeTierActionResults.get(0).isSuccess());
}
```

ChangeTierInputBuilder Class

Contains methods to build an instance of the `LoyaltyManagement.ChangeTierInput` class.

Namespace

[LoyaltyManagement](#)

[ChangeTierInputBuilder Methods](#)

ChangeTierInputBuilder Methods

The following are methods for `ChangeTierInputBuilder`.

[build\(\)](#)

Returns an instance of the `LoyaltyManagement.ChangeTierInput` object.

[setChangeReason\(changeReason\)](#)

Represents the reason to change a member tier.

[setLoyaltyProgramMemberId\(loyaltyProgramMemberId\)](#)

Represents the ID of the loyalty program member. This is a required field.

[setProgramName\(programName\)](#)

Represents the loyalty program name associated with the tier. This is a required field.

[setTargetTierName\(targetTierName\)](#)

Represents the new tier to associate with the loyalty program member. This is a required field.

[setTierGroupName\(tierGroupName\)](#)

Represents the name of the loyalty tier group. This is a required field.

build()

Returns an instance of the `LoyaltyManagement.ChangeTierInput` object.

Signature

```
public LoyaltyManagement.ChangeTierInput build()
```

Return Value

Type: [LoyaltyManagement.ChangeTierInput](#)

setChangeReason(changeReason)

Represents the reason to change a member tier.

Signature

```
public LoyaltyManagement.ChangeTierInputBuilder setChangeReason(String changeReason)
```

Parameters

changeReason

Type: String

Return Value

Type: [LoyaltyManagement.ChangeTierInputBuilder](#)

setLoyaltyProgramMemberId(loyaltyProgramMemberId)

Represents the ID of the loyalty program member. This is a required field.

Signature

```
public LoyaltyManagement.ChangeTierInputBuilder setLoyaltyProgramMemberId(String  
loyaltyProgramMemberId)
```

Parameters

loyaltyProgramMemberId

Type: String

Return Value

Type: [LoyaltyManagement.ChangeTierInputBuilder](#)

setProgramName (programName)

Represents the loyalty program name associated with the tier. This is a required field.

Signature

```
public LoyaltyManagement.ChangeTierInputBuilder setProgramName(String programName)
```

Parameters

programName
Type: String

Return Value

Type: [LoyaltyManagement.ChangeTierInputBuilder](#)

setTargetTierName (targetTierName)

Represents the new tier to associate with the loyalty program member. This is a required field.

Signature

```
public LoyaltyManagement.ChangeTierInputBuilder setTargetTierName(String targetTierName)
```

Parameters

targetTierName
Type: String

Return Value

Type: [LoyaltyManagement.ChangeTierInputBuilder](#)

setTierGroupName (tierGroupName)

Represents the name of the loyalty tier group. This is a required field.

Signature

```
public LoyaltyManagement.ChangeTierInputBuilder setTierGroupName(String tierGroupName)
```

Parameters

tierGroupName
Type: String

Return Value

Type: [LoyaltyManagement.ChangeTierInputBuilder](#)

ChangeTierOutput Class

Represents the changed tier details for a loyalty program member.

Namespace

[LoyaltyManagement](#)

[ChangeTierOutput Methods](#)

ChangeTierOutput Methods

The following are methods for `ChangeTierOutput`.

[getErrors\(\)](#)

If an error occurred, returns a list providing the error code and description.

[getLoyaltyProgramMemberId\(\)](#)

Returns a String that contains the loyalty program member ID.

[getMemberTier\(\)](#)

Returns a String that contains the tier associated with the loyalty program member.

[getProgramName\(\)](#)

Returns a String that contains the loyalty program name.

[getTierGroupName\(\)](#)

Returns a String that contains the loyalty tier group name.

[isSuccess\(\)](#)

Returns a Boolean that is set to true if the change tier operation was successful for this object, false otherwise.

getErrors ()

If an error occurred, returns a list providing the error code and description.

Signature

```
public List<LoyaltyManagement.LoyaltyActionError> getErrors ()
```

Return Value

Type: List<[LoyaltyManagement.LoyaltyActionError](#)>

getLoyaltyProgramMemberId ()

Returns a String that contains the loyalty program member ID.

Signature

```
public String getLoyaltyProgramMemberId()
```

Return Value

Type: String

getMemberTier()

Returns a String that contains the tier associated with the loyalty program member.

Signature

```
public String getMemberTier()
```

Return Value

Type: String

getProgramName()

Returns a String that contains the loyalty program name.

Signature

```
public String getProgramName()
```

Return Value

Type: String

getTierGroupName()

Returns a String that contains the loyalty tier group name.

Signature

```
public String getTierGroupName()
```

Return Value

Type: String

isSuccess()

Returns a Boolean that is set to true if the change tier operation was successful for this object, false otherwise.

Signature

```
public Boolean isSuccess()
```


Return Value

Type: Boolean

CreditPointsOutput Class

Represents the details of points credited to a loyalty program member's balance.

Namespace

[LoyaltyManagement](#)

[CreditPointsOutput Methods](#)

CreditPointsOutput Methods

The following are methods for `CreditPointsOutput`.

[getCurrencyName\(\)](#)

Returns a String that contains the name of the program currency.

[getErrors\(\)](#)

If an error occurred, returns a list providing the error code and description.

[getJournalId\(\)](#)

Returns a String that contains the ID of the transaction journal.

[getLoyaltyProgramMemberId\(\)](#)

Returns a String that contains the ID of the loyalty program member.

[getProgramName\(\)](#)

Returns a String that contains the name of the loyalty program.

[isSuccess\(\)](#)

Returns a Boolean that is set to true if the credit points operation was successful for this object, false otherwise.

getCurrencyName ()

Returns a String that contains the name of the program currency.

Signature

```
public String getCurrencyName ()
```

Return Value

Type: String

getErrors ()

If an error occurred, returns a list providing the error code and description.

Signature

```
public List<LoyaltyManagement.LoyaltyActionError> getErrors()
```

Return Value

Type: List<LoyaltyManagement.LoyaltyActionError>

getJournalId()

Returns a String that contains the ID of the transaction journal.

Signature

```
public String getJournalId()
```

Return Value

Type: String

getLoyaltyProgramMemberId()

Returns a String that contains the ID of the loyalty program member.

Signature

```
public String getLoyaltyProgramMemberId()
```

Return Value

Type: String

getProgramName()

Returns a String that contains the name of the loyalty program.

Signature

```
public String getProgramName()
```

Return Value

Type: String

isSuccess()

Returns a Boolean that is set to true if the credit points operation was successful for this object, false otherwise.

Signature

```
public Boolean isSuccess()
```

Return Value

Type: Boolean

DebitPointsOutput Class

Represents the details of points debited from a loyalty program member's balance.

Namespace

[LoyaltyManagement](#)

[DebitPointsOutput Methods](#)

DebitPointsOutput Methods

The following are methods for `DebitPointsOutput`.

[getCurrencyName\(\)](#)

Returns a String that contains the name of the program currency.

[getErrors\(\)](#)

If an error occurred, returns a list providing the error code and description.

[getJournalId\(\)](#)

Returns a String that contains the ID of the transaction journal.

[getLoyaltyProgramMemberId\(\)](#)

Returns a String that contains the ID of the loyalty program member.

[getPoints\(\)](#)

Returns a Double that contains the total points available for the loyalty program member.

[getProgramName\(\)](#)

Returns a String that contains the name of the loyalty program.

[isSuccess\(\)](#)

Returns a Boolean that is set to true if the debit points operation was successful for this object, false otherwise.

getCurrencyName ()

Returns a String that contains the name of the program currency.

Signature

```
public String getCurrencyName ()
```

Return Value

Type: String

getErrors()

If an error occurred, returns a list providing the error code and description.

Signature

```
public List<LoyaltyManagement.LoyaltyActionError> getErrors()
```

Return Value

Type: List<[LoyaltyManagement.LoyaltyActionError](#)>

getJournalId()

Returns a String that contains the ID of the transaction journal.

Signature

```
public String getJournalId()
```

Return Value

Type: String

getLoyaltyProgramMemberId()

Returns a String that contains the ID of the loyalty program member.

Signature

```
public String getLoyaltyProgramMemberId()
```

Return Value

Type: String

getPoints()

Returns a Double that contains the total points available for the loyalty program member.

Signature

```
public Double getPoints()
```

Return Value

Type: Double

getProgramName()

Returns a String that contains the name of the loyalty program.

Signature

```
public String getProgramName()
```

Return Value

Type: String

isSuccess()

Returns a Boolean that is set to true if the debit points operation was successful for this object, false otherwise.

Signature

```
public Boolean isSuccess()
```

Return Value

Type: Boolean

GetMemberActiveSegmentsInput Class

Use this class to retrieve active Data Cloud market segments that a loyalty program member is part of. This class receives the input parameter values for the Get Member's Active Segments action.

Namespace

[LoyaltyManagement](#)

GetMemberActiveSegmentsInputBuilder Class

Contains methods to build an instance of the `LoyaltyManagement.GetMemberActiveSegmentsInput` class.

Namespace

[LoyaltyManagement](#)

[GetMemberActiveSegmentsInputBuilder Methods](#)

GetMemberActiveSegmentsInputBuilder Methods

The following are methods for `GetMemberActiveSegmentsInputBuilder`.

[build\(\)](#)

Returns an instance of the `LoyaltyManagement.GetMemberActiveSegmentsInput` object.

[setLoyaltyProgramMemberID\(loyaltyProgramMemberID\)](#)

Represents the ID of the loyalty program member to retrieve associated active segments for. This is a required field.

setSegmentIds(segmentIds)

Represents a comma-delimited list of segmentIds to retrieve for the specified member's associated segments.

build()

Returns an instance of the `LoyaltyManagement.GetMemberActiveSegmentsInput` object.

Signature

```
public LoyaltyManagement.GetMemberActiveSegmentsInput build()
```

Return Value

Type: [LoyaltyManagement.GetMemberActiveSegmentsInput](#)

setLoyaltyProgramMemberID(loyaltyProgramMemberID)

Represents the ID of the loyalty program member to retrieve associated active segments for. This is a required field.

Signature

```
public LoyaltyManagement.GetMemberActiveSegmentsInputBuilder  
setLoyaltyProgramMemberID(String loyaltyProgramMemberID)
```

Parameters

loyaltyProgramMemberID
Type: String

Return Value

Type: [LoyaltyManagement.GetMemberActiveSegmentsInputBuilder](#)

setSegmentIds(segmentIds)

Represents a comma-delimited list of segmentIds to retrieve for the specified member's associated segments.

Signature

```
public LoyaltyManagement.GetMemberActiveSegmentsInputBuilder setSegmentIds(List<String>  
segmentIds)
```

Parameters

segmentIds
Type: List<String>

Return Value

Type: [LoyaltyManagement.GetMemberActiveSegmentsInputBuilder](#)

GetMemberActiveSegmentsOutput Class

Represents the list of the active segment IDs that the specified loyalty program member is a part of.

Namespace

[LoyaltyManagement](#)

[GetMemberActiveSegmentsOutput Methods](#)

GetMemberActiveSegmentsOutput Methods

The following are methods for `GetMemberActiveSegmentsOutput`.

[getErrors\(\)](#)

If an error occurred, returns a list providing the error code and description.

[getMemberSegmentIDs\(\)](#)

Returns a comma-delimited list of the active segment IDs that the specified loyalty program member is a part of.

[isSuccess\(\)](#)

Returns a Boolean that is set to true if the get member active segments operation was successful for this object, false otherwise.

getErrors()

If an error occurred, returns a list providing the error code and description.

Signature

```
public List<LoyaltyManagement.LoyaltyActionError> getErrors()
```

Return Value

Type: [List<LoyaltyManagement.LoyaltyActionError>](#)

getMemberSegmentIDs()

Returns a comma-delimited list of the active segment IDs that the specified loyalty program member is a part of.



Note: If input segment IDs are passed, then the response is a subset of the specified segment IDs.

Signature

```
public List<String> getMemberSegmentIDs()
```

Return Value

Type: [List<String>](#)

isSuccess ()

Returns a Boolean that is set to true if the get member active segments operation was successful for this object, false otherwise.

Signature

```
public Boolean isSuccess()
```

Return Value

Type: Boolean

IssueVoucherInput Class

Use this class to issue a voucher to a loyalty program member. This class receives the input parameter values for the Issue Voucher action.

Namespace

[LoyaltyManagement](#)

Example

```
public void issueVoucherTest() {
    //...Create a list with LoyaltyManagement.IssueVoucherInput as a datatype
    List<LoyaltyManagement.IssueVoucherInput> issueVoucherList = new
    List<LoyaltyManagement.IssueVoucherInput>();

    //...Create an instance of the list object and set the input values
    issueVoucherList.add(new
    LoyaltyManagement.IssueVoucherInputBuilder().setLoyaltyProgramMemberId('01MRM0000002BD12AM')

        .setProgramName('LP01')
        .setVoucherDefinitionName('V1')
        .setVoucherCode('demoCode')
        .setJournalId('01VRM0000004DH22AM')
        .build());

    //...Pass the list that contains the input values to the LoyaltyResources.issueVoucher()
    function.
    //...Store the output returned by the function in the IssueVoucherOutput object
    List<LoyaltyManagement.IssueVoucherOutput> issueVoucherActionResults =
    LoyaltyManagement.LoyaltyResources.issueVoucher(issueVoucherList);
    System.debug('Issue Voucher Successful Operation:'+ issueVoucherActionResults.get(0));
}
```

IssueVoucherInputBuilder Class

Contains methods to build an instance of the `LoyaltyManagement.IssueVoucherInput` class.

Namespace

[LoyaltyManagement](#)

[IssueVoucherInputBuilder Methods](#)

IssueVoucherInputBuilder Methods

The following are methods for `IssueVoucherInputBuilder`.

[build\(\)](#)

Returns an instance of the `LoyaltyManagement.IssueVoucherInput` object.

[setAdditionalNotes\(additionalNotes\)](#)

Represents extra information about the voucher.

[setBenefitId\(benefitId\)](#)

Represents the benefit associated with the voucher.

[setContactId\(contactId\)](#)

Represents the contact associated with the loyalty program member.

[setDiscountPercent\(discountPercent\)](#)

Represents the discount percentage provided by the voucher.

[setEffectiveDate\(effectiveDate\)](#)

Represents the start date of the validity of the voucher.

[setExpiryDate\(expiryDate\)](#)

Represents the expiry date of the voucher.

[setFaceValue\(faceValue\)](#)

Represents the amount of discount provided by the voucher.

[setJournalId\(journalId\)](#)

Represents the transaction journal associated with the member voucher when the voucher results from accrual of points.

[setLoyaltyProgramMemberId\(loyaltyProgramMemberId\)](#)

Represents the ID of the loyalty program member to issue the voucher.

[setProgramName\(programName\)](#)

Represents the name of the loyalty program associated with the voucher.

[setPromotionId\(promotionId\)](#)

Represents the promotion associated with the voucher.

[setVoucherCode\(voucherCode\)](#)

Represents the voucher identifier generated by an external system. This is a required field.

[setVoucherDefinitionName\(voucherDefinitionName\)](#)

Represents the name of the voucher definition associated with the voucher code. This is a required field.

build()

Returns an instance of the `LoyaltyManagement.IssueVoucherInput` object.

Signature

```
public LoyaltyManagement.IssueVoucherInput build()
```

Return Value

Type: [LoyaltyManagement.IssueVoucherInput](#)

setAdditionalNotes (additionalNotes)

Represents extra information about the voucher.

Signature

```
public LoyaltyManagement.IssueVoucherInputBuilder setAdditionalNotes(String additionalNotes)
```

Parameters

additionalNotes
Type: String

Return Value

Type: [LoyaltyManagement.IssueVoucherInputBuilder](#)

setBenefitId (benefitId)

Represents the benefit associated with the voucher.

Signature

```
public LoyaltyManagement.IssueVoucherInputBuilder setBenefitId(String benefitId)
```

Parameters

benefitId
Type: String

Return Value

Type: [LoyaltyManagement.IssueVoucherInputBuilder](#)

setContactId (contactId)

Represents the contact associated with the loyalty program member.

Signature

```
public LoyaltyManagement.IssueVoucherInputBuilder setContactId(String contactId)
```

Parameters

contactId
Type: String

Return Value

Type: [LoyaltyManagement.IssueVoucherInputBuilder](#)

setDiscountPercent (discountPercent)

Represents the discount percentage provided by the voucher.

Signature

```
public LoyaltyManagement.IssueVoucherInputBuilder setDiscountPercent(Integer discountPercent)
```

Parameters

discountPercent
Type: Integer

The discount percentage provided by the voucher. Specify a value to override the discount percentage specified in the associated voucher definition. Specify this when the associated voucher definition is of the type Discount Percentage. This field is available in API version 53.0 and later.

Return Value

Type: [LoyaltyManagement.IssueVoucherInputBuilder](#)

setEffectiveDate (effectiveDate)

Represents the start date of the validity of the voucher.

Signature

```
public LoyaltyManagement.IssueVoucherInputBuilder setEffectiveDate(String effectiveDate)
```

Parameters

effectiveDate
Type: String
The supported date format is YYYY-MM-DD.

Return Value

Type: [LoyaltyManagement.IssueVoucherInputBuilder](#)

setExpiryDate(expiryDate)

Represents the expiry date of the voucher.

Signature

```
public LoyaltyManagement.IssueVoucherInputBuilder setExpiryDate(String expiryDate)
```

Parameters

expiryDate

Type: String

The supported date format is YYYY-MM-DD.

Return Value

Type: [LoyaltyManagement.IssueVoucherInputBuilder](#)

setFaceValue(faceValue)

Represents the amount of discount provided by the voucher.

Signature

```
public LoyaltyManagement.IssueVoucherInputBuilder setFaceValue(Double faceValue)
```

Parameters

faceValue

Type: Double

The amount of discount provided by the voucher. Specify a value to override the face value specified in the associated voucher definition. Specify this when the associated voucher definition is of the type Fixed Value. This field is available in API version 53.0 and later.

Return Value

Type: [LoyaltyManagement.IssueVoucherInputBuilder](#)

setJournalId(journalId)

Represents the transaction journal associated with the member voucher when the voucher results from accrual of points.

Signature

```
public LoyaltyManagement.IssueVoucherInputBuilder setJournalId(String journalId)
```

Parameters

journalId

Type: String

Return Value

Type: [LoyaltyManagement.IssueVoucherInputBuilder](#)

setLoyaltyProgramMemberId(loyaltyProgramMemberId)

Represents the ID of the loyalty program member to issue the voucher.

Signature

```
public LoyaltyManagement.IssueVoucherInputBuilder setLoyaltyProgramMemberId(String  
loyaltyProgramMemberId)
```

Parameters

loyaltyProgramMemberId
Type: String

Return Value

Type: [LoyaltyManagement.IssueVoucherInputBuilder](#)

setProgramName(programName)

Represents the name of the loyalty program associated with the voucher.

Signature

```
public LoyaltyManagement.IssueVoucherInputBuilder setProgramName(String programName)
```

Parameters

programName
Type: String

Return Value

Type: [LoyaltyManagement.IssueVoucherInputBuilder](#)

setPromotionId(promotionId)

Represents the promotion associated with the voucher.

Signature

```
public LoyaltyManagement.IssueVoucherInputBuilder setPromotionId(String promotionId)
```

Parameters

promotionId
Type: String

Return Value

Type: [LoyaltyManagement.IssueVoucherInputBuilder](#)

setVoucherCode (voucherCode)

Represents the voucher identifier generated by an external system. This is a required field.

Signature

```
public LoyaltyManagement.IssueVoucherInputBuilder setVoucherCode(String voucherCode)
```

Parameters

voucherCode
Type: String

Return Value

Type: [LoyaltyManagement.IssueVoucherInputBuilder](#)

setVoucherDefinitionName (voucherDefinitionName)

Represents the name of the voucher definition associated with the voucher code. This is a required field.

Signature

```
public LoyaltyManagement.IssueVoucherInputBuilder setVoucherDefinitionName(String voucherDefinitionName)
```

Parameters

voucherDefinitionName
Type: String

Return Value

Type: [LoyaltyManagement.IssueVoucherInputBuilder](#)

IssueVoucherOutput Class

Represents the voucher ID issued to a loyalty program member.

Namespace

[LoyaltyManagement](#)

[IssueVoucherOutput Methods](#)

IssueVoucherOutput Methods

The following are methods for `IssueVoucherOutput`.

`getErrors()`

If an error occurred, returns a list providing the error code and description.

`getVoucherId()`

Returns the ID of the voucher that's generated after the request is successful.

`isSuccess()`

Returns a Boolean that is set to true if the issue voucher operation was successful for this object, false otherwise.

`getErrors()`

If an error occurred, returns a list providing the error code and description.

Signature

```
public List<LoyaltyManagement.LoyaltyActionError> getErrors()
```

Return Value

Type: `List<LoyaltyManagement.LoyaltyActionError>`

`getVoucherId()`

Returns the ID of the voucher that's generated after the request is successful.

Signature

```
public String getVoucherId()
```

Return Value

Type: `String`

`isSuccess()`

Returns a Boolean that is set to true if the issue voucher operation was successful for this object, false otherwise.

Signature

```
public Boolean isSuccess()
```

Return Value

Type: `Boolean`

LoyaltyActionError Class

A class that contains the errors related to the execution of a Loyalty Management action.

Namespace

[LoyaltyManagement](#)

Example

```
public void creditPointsTestError() {
    List<LoyaltyManagement.PointsInput> creditPointsList = new
List<LoyaltyManagement.PointsInput>();
    creditPointsList.add(new LoyaltyManagement.PointsInputBuilder()
        .setLoyaltyProgramMemberId('01MRM0000002BD12AM')
        .setProgramName('LP01')
        .setProgramCurrencyName('Miles')
        .setPoints(1200)
        .setJournalId('01VT10000008Cn1MAE')
        .build());
    List<LoyaltyManagement.CreditPointsOutput> creditPointActionResults =
LoyaltyManagement.LoyaltyResources.creditPoints(creditPointsList);
    System.debug('Is Success:' + creditPointActionResults.get(0).isSuccess());
    System.debug('Errors:' + creditPointActionResults.get(0).getErrors());
}
```

[LoyaltyActionError Constructors](#)

[LoyaltyActionError Methods](#)

LoyaltyActionError Constructors

The following are constructors for `LoyaltyActionError`.

[LoyaltyActionError\(code, message\)](#)

Creates an instance of the `LoyaltyManagement.LoyaltyActionError` class.

LoyaltyActionError(code, message)

Creates an instance of the `LoyaltyManagement.LoyaltyActionError` class.

Signature

```
public LoyaltyActionError(String code, String message)
```

Parameters

code

Type: String

The error code.

message

Type: String

The error message.

LoyaltyActionError Methods

The following are methods for `LoyaltyActionError`.

[getCode\(\)](#)

Returns a String that contains an error code.

[getMessage\(\)](#)

Returns a String that contains an error description.

getCode ()

Returns a String that contains an error code.

Signature

```
public String getCode ()
```

Return Value

Type: String

getMessage ()

Returns a String that contains an error description.

Signature

```
public String getMessage ()
```

Return Value

Type: String

LoyaltyActionResult Class

A class that contains the result returned by a loyalty action.

Namespace

[LoyaltyManagement](#)

[LoyaltyActionResult Methods](#)

LoyaltyActionResult Methods

The following are methods for `LoyaltyActionResult`.

`getErrors()`

If an error occurred, returns a `LoyaltyManagement.LoyaltyActionError` object providing the error code and description.

`isSuccess()`

Returns a Boolean that is set to true if the Loyalty Management action operation was successful for this object, false otherwise.

`getErrors()`

If an error occurred, returns a `LoyaltyManagement.LoyaltyActionError` object providing the error code and description.

Signature

```
public List<LoyaltyManagement.LoyaltyActionError> getErrors()
```

Return Value

Type: `List<LoyaltyManagement.LoyaltyActionError>`

`isSuccess()`

Returns a Boolean that is set to true if the Loyalty Management action operation was successful for this object, false otherwise.

Signature

```
public Boolean isSuccess()
```

Return Value

Type: Boolean

LoyaltyPromotionInput Class

Use this class to retrieve active promotions that are eligible for a transaction journal. The action can process accrual or redemption transactions. This class receives the input parameter values for the Loyalty Promotion action.

Namespace

`LoyaltyManagement`

Example

```
public static void getLoyaltyPromotionsTest() {
    //...Create a list with 'LoyaltyManagement.LoyaltyPromotionInput' as a datatype
    List<LoyaltyManagement.LoyaltyPromotionInput> loyaltyPromotionInputList = new
    List<LoyaltyManagement.LoyaltyPromotionInput>();
}
```

```
//...Create an instance of the list object and set the input values
loyaltyPromotionInputList.add(new LoyaltyManagement.LoyaltyPromotionInput input = new
LoyaltyManagement.LoyaltyPromotionInputBuilder()
    .setJournalId('01VRM0000004DH22AM')
    .build());

//...Pass the list that contains the input values to the
'LoyaltyResources.getLoyaltyPromotions()' function.
//...Store the output returned by the function in the 'LoyaltyPromotionOutput' object
List<LoyaltyManagement.LoyaltyPromotionOutput> results =
LoyaltyManagement.LoyaltyResources.getLoyaltyPromotions(loyaltyPromotionInputList);
System.debug('Get Loyalty Promotions:'+ results.get(0).isSuccess());
}
```

LoyaltyPromotionInputBuilder Class

Contains methods to build an instance of the `LoyaltyManagement.LoyaltyPromotionInput` class.

Namespace

[LoyaltyManagement](#)

[LoyaltyPromotionInputBuilder Methods](#)

LoyaltyPromotionInputBuilder Methods

The following are methods for `LoyaltyPromotionInputBuilder`.

[build\(\)](#)

Returns an instance of the `LoyaltyManagement.LoyaltyPromotionInput` object.

[setJournalId\(journalId\)](#)

Represents the ID of an Accrual or Redemption type Transaction Journal record that retrieval of active loyalty promotions is based on. This is a required field.

build()

Returns an instance of the `LoyaltyManagement.LoyaltyPromotionInput` object.

Signature

```
public LoyaltyManagement.LoyaltyPromotionInput build()
```

Return Value

Type: [LoyaltyManagement.LoyaltyPromotionInput](#)

setJournalId(journalId)

Represents the ID of an Accrual or Redemption type Transaction Journal record that retrieval of active loyalty promotions is based on. This is a required field.

Signature

```
public LoyaltyManagement.LoyaltyPromotionInputBuilder setJournalId(String journalId)
```

Parameters

journalId
Type: String

Return Value

Type: [LoyaltyManagement.LoyaltyPromotionInputBuilder](#)

LoyaltyPromotionOutput Class

Represents the active promotions based on the specified transaction journal.

Namespace

[LoyaltyManagement](#)

[LoyaltyPromotionOutput Methods](#)

LoyaltyPromotionOutput Methods

The following are methods for `LoyaltyPromotionOutput`.

[getErrors\(\)](#)

If an error occurred, returns a list providing the error code and description.

[getPromotions\(\)](#)

Returns a List of sObject data type that contains the promotion records that are eligible for the transaction journal.

[isSuccess\(\)](#)

Returns a Boolean that is set to true if the loyalty promotion operation was successful for this object, false otherwise.

getErrors()

If an error occurred, returns a list providing the error code and description.

Signature

```
public List<LoyaltyManagement.LoyaltyActionError> getErrors()
```

Return Value

Type: List<[LoyaltyManagement.LoyaltyActionError](#)>

getPromotions()

Returns a List of sObject data type that contains the promotion records that are eligible for the transaction journal.

Signature

```
public List<SObject> getPromotions()
```

Return Value

Type: List<SObject>

isSuccess()

Returns a Boolean that is set to true if the loyalty promotion operation was successful for this object, false otherwise.

Signature

```
public Boolean isSuccess()
```

Return Value

Type: Boolean

LoyaltyResources Class

A class that executes a loyalty action and returns a result. Takes the input classes as input and returns the objects of output classes.

Namespace

[LoyaltyManagement](#)

[LoyaltyResources Methods](#)

LoyaltyResources Methods

The following are methods for `LoyaltyResources`.



Note: Methods in the `LoyaltyResources` class support processing up to 200 records per invocation. This class predates the `Invocable.Action` Apex wrapper, which supports up to 2000 records per invocation and is the recommended approach for invoking loyalty actions.

[changeTier\(changeTierList\)](#)

Use this method to change the tier for a loyalty program member.

[creditPoints\(creditPointsList\)](#)

Use this method to credit loyalty points to a loyalty program member's balance.

[debitPoints\(debitPointsList\)](#)

Use this method to debit loyalty points from a loyalty program member's balance.

[getLoyaltyPromotionBasedOnSalesforceCDP\(loyaltyPromotionInputList\)](#)

Use this method to retrieve promotions that are eligible for a member based on the member's market segment.

[getLoyaltyPromotions\(loyaltyPromotionInputList\)](#)

Use this method to retrieve active promotions that are eligible for a transaction journal. The action can process accrual or redemption transactions.

[getPointsBalance\(memberPointBalanceList\)](#)

Use this method to retrieve the loyalty points balance for a loyalty program member.

[getTier\(memberTierList\)](#)

Use this method to retrieve the current tier for a loyalty program member.

[issueVoucher\(issueVoucherList\)](#)

Use this method to issue a voucher for a loyalty program member.

[transferMemberPointsToGroups\(transferMemberPointsToGroupsList\)](#)

Use this method to transfer points from an individual member or a corporate member to the member's associated group.

[updateProgressForCumulativePromotionUsage\(updateCumulativeUsageCompletedInputList\)](#)

Use this method to update the progress made by a loyalty program member towards attaining a Cumulative type promotion.

changeTier (changeTierList)

Use this method to change the tier for a loyalty program member.

Signature

```
public static List<LoyaltyManagement.ChangeTierOutput>
changeTier (List<LoyaltyManagement.ChangeTierInput> changeTierList)
```

Parameters

changeTierList

Type: List<[LoyaltyManagement.ChangeTierInput](#)>

Set of inputs to change the tier for a loyalty program member.

Return Value

Type: List<[LoyaltyManagement.ChangeTierOutput](#)>

creditPoints (creditPointsList)

Use this method to credit loyalty points to a loyalty program member's balance.

Signature

```
public static List<LoyaltyManagement.CreditPointsOutput>  
creditPoints (List<LoyaltyManagement.PointsInput> creditPointsList)
```

Parameters

creditPointsList

Type: List<LoyaltyManagement.PointsInput>

Set of inputs to credit points to a loyalty program member's balance.

Return Value

Type: List<LoyaltyManagement.CreditPointsOutput>

debitPoints (debitPointsList)

Use this method to debit loyalty points from a loyalty program member's balance.

Signature

```
public static List<LoyaltyManagement.DebitPointsOutput>  
debitPoints (List<LoyaltyManagement.PointsInput> debitPointsList)
```

Parameters

debitPointsList

Type: List<LoyaltyManagement.PointsInput>

Set of inputs to debit points from a loyalty program member's balance.

Return Value

Type: List<LoyaltyManagement.DebitPointsOutput>

getLoyaltyPromotionBasedOnSalesforceCDP (loyaltyPromotionInputList)

Use this method to retrieve promotions that are eligible for a member based on the member's market segment.

Signature

```
public static List<LoyaltyManagement.CdpBasedLoyaltyPromotionOutput>  
getLoyaltyPromotionBasedOnSalesforceCDP (List<LoyaltyManagement.CdpBasedLoyaltyPromotionInput>  
loyaltyPromotionInputList)
```

Parameters

loyaltyPromotionInputList

Type: List<LoyaltyManagement.CdpBasedLoyaltyPromotionInput>

Set of inputs to retrieve promotions for a member based on the member's market segment.

Return Value

Type: List<LoyaltyManagement.CdpBasedLoyaltyPromotionOutput>

getLoyaltyPromotions (loyaltyPromotionInputList)

Use this method to retrieve active promotions that are eligible for a transaction journal. The action can process accrual or redemption transactions.

Signature

```
public static List<LoyaltyManagement.LoyaltyPromotionOutput>  
getLoyaltyPromotions (List<LoyaltyManagement.LoyaltyPromotionInput>  
loyaltyPromotionInputList)
```

Parameters

loyaltyPromotionInputList
Type: List<LoyaltyManagement.LoyaltyPromotionInput>
Set of inputs to retrieve active promotions for a transaction journal.

Return Value

Type: List<LoyaltyManagement.LoyaltyPromotionOutput>

getPointsBalance (memberPointBalanceList)

Use this method to retrieve the loyalty points balance for a loyalty program member.

Signature

```
public static List<LoyaltyManagement.MemberPointBalanceOutput>  
getPointsBalance (List<LoyaltyManagement.MemberPointBalanceInput> memberPointBalanceList)
```

Parameters

memberPointBalanceList
Type: List<LoyaltyManagement.MemberPointBalanceInput>
Set of inputs to get the loyalty points balance for a loyalty program member's balance.

Return Value

Type: List<LoyaltyManagement.MemberPointBalanceOutput>

getTier (memberTierList)

Use this method to retrieve the current tier for a loyalty program member.

Signature

```
public static List<LoyaltyManagement.MemberTierOutput>  
getTier (List<LoyaltyManagement.MemberTierInput> memberTierList)
```

Parameters

memberTierList

Type: List<loyalty_management.MemberTierInput>

Set of inputs to retrieve the current tier for a given loyalty program member.

Return Value

Type: List<loyalty_management.MemberTierOutput>

issueVoucher (issueVoucherList)

Use this method to issue a voucher for a loyalty program member.

Signature

```
public static List<LoyaltyManagement.IssueVoucherOutput>  
issueVoucher (List<LoyaltyManagement.IssueVoucherInput> issueVoucherList)
```

Parameters

issueVoucherList

Type: List<loyalty_management.IssueVoucherInput>

Set of inputs to issue a voucher.

Return Value

Type: List<loyalty_management.IssueVoucherOutput>

transferMemberPointsToGroups (transferMemberPointsToGroupsList)

Use this method to transfer points from an individual member or a corporate member to the member's associated group.

Signature

```
public static List<LoyaltyManagement.LoyaltyActionResult>  
transferMemberPointsToGroups (List<LoyaltyManagement.TransferMemberPointsToGroupsInput>  
transferMemberPointsToGroupsList)
```

Parameters

transferMemberPointsToGroupsList

Type: List<loyalty_management.TransferMemberPointsToGroupsInput>

Set of inputs to transfer points from an individual member or a corporate member.

Return Value

Type: List<[loyalty_management.LoyaltyActionResult](#)>

updateProgressForCumulativePromotionUsage (updateCumulativeUsageCompletedInputList)

Use this method to update the progress made by a loyalty program member towards attaining a Cumulative type promotion.

Signature

```
public static List<LoyaltyManagement.UpdateCumulativeUsageCompletedOutput>
updateProgressForCumulativePromotionUsage (List<LoyaltyManagement.UpdateCumulativeUsageCompletedInput>
updateCumulativeUsageCompletedInputList)
```

Parameters

updateCumulativeUsageCompletedInputList

Type: List<[LoyaltyManagement.UpdateCumulativeUsageCompletedInput](#)>

Set of inputs to update the progress for a Cumulative type promotion.

Return Value

Type: List<[LoyaltyManagement.UpdateCumulativeUsageCompletedOutput](#)>

MemberPointBalanceInput Class

Use this class to retrieve loyalty points balance for a given loyalty program member. This class receives the input parameter values for the Get Member Points action.

Namespace

[LoyaltyManagement](#)

Example

```
public void pointsBalanceTest() {
    //...Create a list with LoyaltyManagement.MemberPointsBalanceInput as a datatype
    List<LoyaltyManagement.MemberPointBalanceInput> memberPointsBalanceList = new
List<LoyaltyManagement.MemberPointBalanceInput>();

    //...Create an instance of the list object and set the input values
    memberPointsBalanceList.add(new LoyaltyManagement.MemberPointBalanceInputBuilder()
        .setLoyaltyProgramMemberId('01MRM000002BD12AM')
        .setProgramName('LP01')
        .setProgramCurrencyName('Miles')
        .build());

    //...Pass the list that contains the input values to the
    LoyaltyResources.getPointsBalance() function.
    //...Store the output returned by the function in the MemberPointsBalanceOutput object
```

```

    List<LoyaltyManagement.MemberPointBalanceOutput> pointBalanceActionResults =
    LoyaltyManagement.LoyaltyResources.getPointsBalance(memberPointsBalanceList);
    System.debug('Get Member Point Balance Successful Operation:' +
    pointBalanceActionResults.get(0));
    System.debug('Points balance : ' + pointBalanceActionResults.get(0).getPoints());
}

```

MemberPointBalanceInputBuilder Class

Contains methods to build an instance of the `LoyaltyManagement.MemberPointsBalanceInput` class.

Namespace

[LoyaltyManagement](#)

[MemberPointBalanceInputBuilder Methods](#)

MemberPointBalanceInputBuilder Methods

The following are methods for `MemberPointBalanceInputBuilder`.

[build\(\)](#)

Returns an instance of the `LoyaltyManagement.MemberPointsBalanceInput` object.

[setLoyaltyProgramMemberId\(loyaltyProgramMemberId\)](#)

Represents the ID of the loyalty program member whose points balance is to be fetched. This is a required field.

[setProgramCurrencyName\(programCurrencyName\)](#)

Represents the name of the program currency. This is a required field.

[setProgramName\(programName\)](#)

Represents the name of the loyalty program. This is a required field.

build()

Returns an instance of the `LoyaltyManagement.MemberPointsBalanceInput` object.

Signature

```
public LoyaltyManagement.MemberPointBalanceInput build()
```

Return Value

Type: [LoyaltyManagement.MemberPointBalanceInput](#)

setLoyaltyProgramMemberId(loyaltyProgramMemberId)

Represents the ID of the loyalty program member whose points balance is to be fetched. This is a required field.

Signature

```
public LoyaltyManagement.MemberPointBalanceInputBuilder setLoyaltyProgramMemberId(String  
loyaltyProgramMemberId)
```

Parameters

loyaltyProgramMemberId
Type: String

Return Value

Type: [LoyaltyManagement.MemberPointBalanceInputBuilder](#)

setProgramCurrencyName (programCurrencyName)

Represents the name of the program currency. This is a required field.

Signature

```
public LoyaltyManagement.MemberPointBalanceInputBuilder setProgramCurrencyName(String  
programCurrencyName)
```

Parameters

programCurrencyName
Type: String

Return Value

Type: [LoyaltyManagement.MemberPointBalanceInputBuilder](#)

setProgramName (programName)

Represents the name of the loyalty program. This is a required field.

Signature

```
public LoyaltyManagement.MemberPointBalanceInputBuilder setProgramName(String  
programName)
```

Parameters

programName
Type: String

Return Value

Type: [LoyaltyManagement.MemberPointBalanceInputBuilder](#)

MemberPointBalanceOutput Class

Represents the loyalty points balance for a specified loyalty program member.

Namespace

[LoyaltyManagement](#)

[MemberPointBalanceOutput Methods](#)

MemberPointBalanceOutput Methods

The following are methods for `MemberPointBalanceOutput`.

[getCurrencyName\(\)](#)

Returns a String that contains the name of the program currency.

[getErrors\(\)](#)

If an error occurred, returns a list providing the error code and description.

[getEscrowPointsBalance\(\)](#)

Returns a Double that contains the escrow points balance for a loyalty program member. These points are yet to be credited from Escrow to the loyalty member's points balance.

[getLoyaltyProgramMemberId\(\)](#)

Returns a String that contains the ID of the loyalty program member.

[getPoints\(\)](#)

Returns a Double that contains the points available for the loyalty program member.

[getProgramName\(\)](#)

Returns a String that contains the loyalty program name.

[isSuccess\(\)](#)

Returns a Boolean that is set to true if the member points operation was successful for this object, false otherwise.

getCurrencyName ()

Returns a String that contains the name of the program currency.

Signature

```
public String getCurrencyName ()
```

Return Value

Type: String

getErrors ()

If an error occurred, returns a list providing the error code and description.

Signature

```
public List<LoyaltyManagement.LoyaltyActionError> getErrors()
```

Return Value

Type: List<LoyaltyManagement.LoyaltyActionError>

getEscrowPointsBalance()

Returns a Double that contains the escrow points balance for a loyalty program member. These points are yet to be credited from Escrow to the loyalty member's points balance.

Signature

```
public Double getEscrowPointsBalance()
```

Return Value

Type: Double

getLoyaltyProgramMemberId()

Returns a String that contains the ID of the loyalty program member.

Signature

```
public String getLoyaltyProgramMemberId()
```

Return Value

Type: String

getPoints()

Returns a Double that contains the points available for the loyalty program member.

Signature

```
public Double getPoints()
```

Return Value

Type: Double

getProgramName()

Returns a String that contains the loyalty program name.

Signature

```
public String getProgramName()
```

Return Value

Type: String

isSuccess()

Returns a Boolean that is set to true if the member points operation was successful for this object, false otherwise.

Signature

```
public Boolean isSuccess()
```

Return Value

Type: Boolean

MemberTierInput Class

Use this class to retrieve the current tier for a specified loyalty program member. This class receives the input parameter values for the Get Member Tier action.

Namespace

[LoyaltyManagement](#)

Example

```
public void getTierTest() {
    //...Create a list with LoyaltyManagement.MemberTierInput as a datatype
    List<LoyaltyManagement.MemberTierInput> memberTiersList = new
    List<LoyaltyManagement.MemberTierInput>();

    //...Create an instance of the list object and set the input values
    memberTiersList.add(new
    LoyaltyManagement.MemberTierInputBuilder().setLoyaltyProgramMemberId('01MRM0000002BD12AM')

        .setProgramName('LP01')
        .setTierGroupName('TG1')
        .build());

    //...Pass the list that contains the input values to the LoyaltyResources.getTier()
    function.
    //...Store the output returned by the function in the MemberTierOutput object
    List<LoyaltyManagement.MemberTierOutput> memberTierActionResults =
    LoyaltyManagement.LoyaltyResources.getTier(memberTiersList);
    System.debug('Get Member Tier Successful Operation:' + memberTierActionResults.get(0));
}
```

```
System.debug('Member Tier : '+ memberTierActionResults.get(0).getMemberTier());
}
```

MemberTierInputBuilder Class

Contains methods to build an instance of the `LoyaltyManagement.MemberTierInput` class.

Namespace

[LoyaltyManagement](#)

[MemberTierInputBuilder Methods](#)

MemberTierInputBuilder Methods

The following are methods for `MemberTierInputBuilder`.

[build\(\)](#)

Returns an instance of the `LoyaltyManagement.MemberTierInput` object.

[setLoyaltyProgramMemberId\(loyaltyProgramMemberId\)](#)

Represents the ID of the loyalty program member whose tier details are to be fetched. This is a required field.

[setProgramName\(programName\)](#)

Represents the loyalty program name associated with the tier.

[setTierGroupName\(tierGroupName\)](#)

Represents the name of the tier group. This is a required field.

build()

Returns an instance of the `LoyaltyManagement.MemberTierInput` object.

Signature

```
public LoyaltyManagement.MemberTierInput build()
```

Return Value

Type: [LoyaltyManagement.MemberTierInput](#)

setLoyaltyProgramMemberId(loyaltyProgramMemberId)

Represents the ID of the loyalty program member whose tier details are to be fetched. This is a required field.

Signature

```
public LoyaltyManagement.MemberTierInputBuilder setLoyaltyProgramMemberId(String
loyaltyProgramMemberId)
```


Parameters

loyaltyProgramMemberId
Type: String

Return Value

Type: [LoyaltyManagement.MemberTierInputBuilder](#)

setProgramName (programName)

Represents the loyalty program name associated with the tier.

Signature

```
public LoyaltyManagement.MemberTierInputBuilder setProgramName(String programName)
```

Parameters

programName
Type: String

Return Value

Type: [LoyaltyManagement.MemberTierInputBuilder](#)

setTierGroupName (tierGroupName)

Represents the name of the tier group. This is a required field.

Signature

```
public LoyaltyManagement.MemberTierInputBuilder setTierGroupName(String tierGroupName)
```

Parameters

tierGroupName
Type: String

Return Value

Type: [LoyaltyManagement.MemberTierInputBuilder](#)

MemberTierOutput Class

Represents the tier details for a specified loyalty program member.

Namespace

[LoyaltyManagement](#)

[MemberTierOutput Methods](#)

MemberTierOutput Methods

The following are methods for `MemberTierOutput`.

[`getErrors\(\)`](#)

If an error occurred, returns a list providing the error code and description.

[`getLoyaltyProgramMemberId\(\)`](#)

Returns a String that contains the loyalty program member ID.

[`getMemberTier\(\)`](#)

Returns a String that contains the name of the tier associated with the program member.

[`getProgramName\(\)`](#)

Returns a String that contains the loyalty program name associated with the tier.

[`getSequenceNumber\(\)`](#)

Returns an Integer that contains the ranking of the tier.

[`getTierGroupName\(\)`](#)

Returns a String that contains the name of the tier group.

[`isSuccess\(\)`](#)

Returns a Boolean that is set to true if the member tier operation was successful for this object, false otherwise.

`getErrors()`

If an error occurred, returns a list providing the error code and description.

Signature

```
public List<LoyaltyManagement.LoyaltyActionError> getErrors()
```

Return Value

Type: List<[LoyaltyManagement.LoyaltyActionError](#)>

`getLoyaltyProgramMemberId()`

Returns a String that contains the loyalty program member ID.

Signature

```
public String getLoyaltyProgramMemberId()
```

Return Value

Type: String

getMemberTier()

Returns a String that contains the name of the tier associated with the program member.

Signature

```
public String getMemberTier()
```

Return Value

Type: String

getProgramName()

Returns a String that contains the loyalty program name associated with the tier.

Signature

```
public String getProgramName()
```

Return Value

Type: String

getSequenceNumber()

Returns an Integer that contains the ranking of the tier.

Signature

```
public Integer getSequenceNumber()
```

Return Value

Type: Integer

getTierGroupName()

Returns a String that contains the name of the tier group.

Signature

```
public String getTierGroupName()
```

Return Value

Type: String

isSuccess()

Returns a Boolean that is set to true if the member tier operation was successful for this object, false otherwise.

Signature

```
public Boolean isSuccess()
```

Return Value

Type: Boolean

PointsInput Class

Use this class to credit loyalty points to a loyalty program member's balance or debit loyalty points from a loyalty program member's balance. This class receives the input parameter values for the Credit and Debit point actions.

Namespace

[LoyaltyManagement](#)

Example

This example is to credit loyalty points to a member's balance:

```
public void creditPointsTest() {
    //...Create a list with LoyaltyManagement.PointsInput as a datatype
    List<LoyaltyManagement.PointsInput> creditPointsList = new
    List<LoyaltyManagement.PointsInput>();

    //...Create an instance of the list object and set the input values
    creditPointsList.add(new LoyaltyManagement.PointsInputBuilder()
        .setLoyaltyProgramMemberId('01MRM0000002BD12AM')
        .setProgramName('LP01')
        .setProgramCurrencyName('Miles')
        .setPoints(100)
        .setJournalId('01VRM0000004DH22AM')
        .build());

    //...Pass the list that contains the input values to the LoyaltyResources.creditPoints()
    function.
    //...Store the output returned by the function in the CreditPointsOutput object
    List<LoyaltyManagement.CreditPointsOutput> creditPointActionResults =
    LoyaltyManagement.LoyaltyResources.creditPoints(creditPointsList);
    System.debug('Credit Successful Operation:'+ creditPointActionResults.get(0));
}
```

This example is to debit loyalty points from a member's balance:

```
public void debitPointsTest() {
    //...Create a list with LoyaltyManagement.PointsInput as a datatype
    List<LoyaltyManagement.PointsInput> debitPointsList = new
    List<LoyaltyManagement.PointsInput>();

    //...Create an instance of the list object and set the input values
```

```

debitPointsList.add(new LoyaltyManagement.PointsInputBuilder()
    .setLoyaltyProgramMemberId('01MRM0000002BD12AM')
    .setProgramName('LP01')
    .setProgramCurrencyName('Miles')
    .setPoints(1000)
    .setJournalId('01VRM0000004COB2A2')
    .build());

//...Pass the list that contains the input values to the LoyaltyResources.debitPoints()
function.
//...Store the output returned by the function in the DebitPointsOutput object
List<LoyaltyManagement.DebitPointsOutput> debitPointActionResults =
LoyaltyManagement.LoyaltyResources.debitPoints(debitPointsList);
System.debug('Debit Successful Operation:'+ debitPointActionResults.get(0));
}

```

PointsInputBuilder Class

Contains methods to build an instance of the `LoyaltyManagement.PointsInput` class.

Namespace

[LoyaltyManagement](#)

[PointsInputBuilder Methods](#)

PointsInputBuilder Methods

The following are methods for `PointsInputBuilder`.

[build\(\)](#)

Returns an instance of the `LoyaltyManagement.PointsInput` object.

[setAdditionalNotes\(additionalNotes\)](#)

Represents extra information about the transactions performed by the loyalty program member.

[setCorrelationId\(correlationId\)](#)

Represents the ID of any Salesforce entity.

[setFlowId\(flowId\)](#)

Represents the ID of the flow associated with the credit or debit points action.

[setJournalId\(journalId\)](#)

Represents the ID of the transaction journal. This is a required field.

[setLoyaltyProgramMemberId\(loyaltyProgramMemberId\)](#)

Represents the ID of the loyalty program member to whom the points must be credited or debited. This is a required field.

[setPoints\(points\)](#)

Represents the points available with the loyalty program member. This is a required field.

[setPointsExpirationDate\(pointsExpirationDate\)](#)

Represents the date on which fixed-model non-qualifying points credited by the action expire.

[setProgramCurrencyName\(programCurrencyName\)](#)

Represents the name of the loyalty program currency.

[setLoyaltyProgramCurrencySubtypeName\(LoyaltyProgramCurrencySubtypeName\)](#)

This method sets specific `LoyaltyProgramCurrencySubtype` for the `LoyaltyProgramCurrency`.

[setProgramName\(programName\)](#)

Represents the name of the loyalty program.

build()

Returns an instance of the `LoyaltyManagement.PointsInput` object.

Signature

```
public LoyaltyManagement.PointsInput build()
```

Return Value

Type: [LoyaltyManagement.PointsInput](#)

setAdditionalNotes (additionalNotes)

Represents extra information about the transactions performed by the loyalty program member.

Signature

```
public LoyaltyManagement.PointsInputBuilder setAdditionalNotes(String additionalNotes)
```

Parameters

additionalNotes

Type: String

Return Value

Type: [LoyaltyManagement.PointsInputBuilder](#)

setCorrelationId (correlationId)

Represents the ID of any Salesforce entity.

Signature

```
public LoyaltyManagement.PointsInputBuilder setCorrelationId(String correlationId)
```

Parameters

correlationId
Type: String

Return Value

Type: [LoyaltyManagement.PointsInputBuilder](#)

setFlowId(flowId)

Represents the ID of the flow associated with the credit or debit points action.

Signature

```
public LoyaltyManagement.PointsInputBuilder setFlowId(String flowId)
```

Parameters

flowId
Type: String

Return Value

Type: [LoyaltyManagement.PointsInputBuilder](#)

setJournalId(journalId)

Represents the ID of the transaction journal. This is a required field.

Signature

```
public LoyaltyManagement.PointsInputBuilder setJournalId(String journalId)
```

Parameters

journalId
Type: String

Return Value

Type: [LoyaltyManagement.PointsInputBuilder](#)

setLoyaltyProgramMemberId(loyaltyProgramMemberId)

Represents the ID of the loyalty program member to whom the points must be credited or debited. This is a required field.

Signature

```
public LoyaltyManagement.PointsInputBuilder setLoyaltyProgramMemberId(String  
loyaltyProgramMemberId)
```

Parameters

loyaltyProgramMemberId
Type: String

Return Value

Type: [LoyaltyManagement.PointsInputBuilder](#)

setPoints (points)

Represents the points available with the loyalty program member. This is a required field.

Signature

```
public LoyaltyManagement.PointsInputBuilder setPoints(Double points)
```

Parameters

points
Type: Double

Return Value

Type: [LoyaltyManagement.PointsInputBuilder](#)

setPointsExpirationDate (pointsExpirationDate)

Represents the date on which fixed-model non-qualifying points credited by the action expire.

Signature

```
public LoyaltyManagement.PointsInputBuilder setPointsExpirationDate(String pointsExpirationDate)
```

Parameters

pointsExpirationDate
Type: String

The supported date format is YYYY-MM-DD. This input is available in API version 55.0 and later.

Return Value

Type: [LoyaltyManagement.PointsInputBuilder](#)

setProgramCurrencyName (programCurrencyName)

Represents the name of the loyalty program currency.

Signature

```
public LoyaltyManagement.PointsInputBuilder setProgramCurrencyName (String  
programCurrencyName)
```

Parameters

programCurrencyName
Type: String

Return Value

Type: [LoyaltyManagement.PointsInputBuilder](#)

setLoyaltyProgramCurrencySubtypeName (LoyaltyProgramCurrencySubtypeName)

This method sets specific `LoyaltyProgramCurrencySubtype` for the `LoyaltyProgramCurrency`.

Signature

```
public LoyaltyManagement.PointsInputBuilder setLoyaltyProgramCurrencySubtypeName (String  
loyaltyProgramCurrencySubtypeName)
```

Parameters

loyaltyProgramCurrencySubtypeName
Type: String

Return Value

Type: [LoyaltyManagement.PointsInputBuilder](#)

setProgramName (programName)

Represents the name of the loyalty program.

Signature

```
public LoyaltyManagement.PointsInputBuilder setProgramName (String programName)
```

Parameters

programName
Type: String

Return Value

Type: [LoyaltyManagement.PointsInputBuilder](#)

TransferMemberPointsToGroupsInput Class

Use this class to transfer points from an individual member or a corporate member to the member's associated group. This class receives the input parameter values for the Transfer Member Points To Groups action.

Namespace

[LoyaltyManagement](#)

Example

```
public static void transferPointsToGroupsTest() {

    //Create a list with LoyaltyManagement.TransferMemberPointsToGroupsInput as a datatype

    List<LoyaltyManagement.TransferMemberPointsToGroupsInput> transferPointsList =
        new List<LoyaltyManagement.TransferMemberPointsToGroupsInput>();

    //Create an instance of transferPointsList to set the transaction journal ID as input
    transferPointsList.add(new LoyaltyManagement.TransferMemberPointsToGroupsInputBuilder()

        .setJournalId('01VT10000008CnlMAE')
        .build());

    //Pass the list to the LoyaltyResources.transferMemberPointsToGroups() interface.
    //Store the output returned by the interface in an instance of the 'LoyaltyActionResult'
    object
    List<LoyaltyManagement.LoyaltyActionResult> transferResults =
        LoyaltyManagement.LoyaltyResources.transferMemberPointsToGroups(transferPointsList);

    //Return the status of the Apex call
    System.debug('Transfer points to groups is successful: '+
transferResults.get(0).isSuccess());
}
```

TransferMemberPointsToGroupsInputBuilder Class

Contains methods to build an instance of the `LoyaltyManagement.TransferMemberPointsToGroupsInput` class.

Namespace

[LoyaltyManagement](#)

[TransferMemberPointsToGroupsInputBuilder Methods](#)

TransferMemberPointsToGroupsInputBuilder Methods

The following are methods for `TransferMemberPointsToGroupsInputBuilder`.

build()

Returns an instance of the `LoyaltyManagement.TransferMemberPointsToGroupsInput` object.

setJournalId(journalId)

Represents the transaction journal that is used to transfer points to groups. This is a required field.

build()

Returns an instance of the `LoyaltyManagement.TransferMemberPointsToGroupsInput` object.

Signature

```
public LoyaltyManagement.TransferMemberPointsToGroupsInput build()
```

Return Value

Type: `LoyaltyManagement.TransferMemberPointsToGroupsInput`

setJournalId(journalId)

Represents the transaction journal that is used to transfer points to groups. This is a required field.

Signature

```
public LoyaltyManagement.TransferMemberPointsToGroupsInputBuilder setJournalId(String journalId)
```

Parameters

journalId

Type: String

The transaction journal must be associated with either an Individual or a Corporate type member. The points are transferred to the member's associated groups.

Return Value

Type: `LoyaltyManagement.TransferMemberPointsToGroupsInputBuilder`

UpdateCumulativeUsageCompletedInput Class

Use this class to update the progress made by a loyalty program member towards attaining a Cumulative type promotion. The action creates a Loyalty Program Member Promotion record if the promotion isn't associated with the member. This class receives the input parameter values for the Update Cumulative Usage Completed action.

Namespace

`LoyaltyManagement`

Example

```
public static void updateProgressForCumulativePromotionUsageTest() {

    //Create a list with LoyaltyManagement.UpdateCumulativeUsageCompletedInput as a datatype

    List<LoyaltyManagement.UpdateCumulativeUsageCompletedInput>
updateCumulativeUsageCompletedInputList =
    new List<LoyaltyManagement.UpdateCumulativeUsageCompletedInput>();

    //Create an instance of LoyaltyManagement.UpdateCumulativeUsageCompletedInput class and
    set the inputs
    LoyaltyManagement.UpdateCumulativeUsageCompletedInput input = new
    LoyaltyManagement.UpdateCumulativeUsageCompletedInput('01Mxx000000000EXEAY',
    '0c8xx000000000BJAAY', 200);
    LoyaltyManagement.UpdateCumulativeUsageCompletedInput input1 = new
    LoyaltyManagement.UpdateCumulativeUsageCompletedInput('01Mxx000000000G9EAI',
    '0c8xx000000000BJAAY', 200);
    updateCumulativeUsageCompletedInputList.add(input);
    updateCumulativeUsageCompletedInputList.add(input1);

    //Store the output returned by the interface in an instance of the
    UpdateCumulativeUsageCompletedOutput object
    List<LoyaltyManagement.UpdateCumulativeUsageCompletedOutput> results =

    LoyaltyManagement.LoyaltyResources.updateProgressForCumulativePromotionUsage(updateCumulativeUsageCompletedInputList);

    //Return the status of the Apex call
    System.debug('updateProgressForCumulativePromotionUsage Successful Operation:'+
    results.get(0).isSuccess());
}
```

[UpdateCumulativeUsageCompletedInput Constructors](#)

UpdateCumulativeUsageCompletedInput Constructors

The following are constructors for UpdateCumulativeUsageCompletedInput.

[UpdateCumulativeUsageCompletedInput\(loyaltyProgramMemberId, promotionId, usageIncrementForCumulativePromotion\)](#)

Creates an instance of the LoyaltyManagement.UpdateCumulativeUsageCompletedInput class.

UpdateCumulativeUsageCompletedInput(loyaltyProgramMemberId, promotionId, usageIncrementForCumulativePromotion)

Creates an instance of the LoyaltyManagement.UpdateCumulativeUsageCompletedInput class.

Signature

```
public UpdateCumulativeUsageCompletedInput (String loyaltyProgramMemberId, String promotionId, Integer usageIncrementForCumulativePromotion)
```

Parameters

loyaltyProgramMemberId

Type: String

The ID of the loyalty program member who progress towards attaining a cumulative promotion must be updated. This is a required field.

promotionId

Type: String

The ID of the Cumulative type promotion for which member's progress must be updated. This is a required field.

usageIncrementForCumulativePromotion

Type: Integer

The value or quantity that must be added to member's usage towards attaining the promotion. This is a required field.

UpdateCumulativeUsageCompletedOutput Class

Represents the details of the progress made by a loyalty program member towards attaining a Cumulative type promotion.

Namespace

[LoyaltyManagement](#)

[UpdateCumulativeUsageCompletedOutput Methods](#)

UpdateCumulativeUsageCompletedOutput Methods

The following are methods for `UpdateCumulativeUsageCompletedOutput`.

[getCumulativeUsageTarget\(\)](#)

Returns the target value or quantity which members must achieve to attain the promotion.

[getErrors\(\)](#)

If an error occurred, returns a list providing the error code and description.

[getUpdatedCumulativeUsageCompleted\(\)](#)

Returns the updated value or quantity by which the member has progressed towards attaining the promotion.

[isSuccess\(\)](#)

Returns a Boolean that is set to true if the update cumulative usage completed operation was successful for this object, false otherwise.

getCumulativeUsageTarget ()

Returns the target value or quantity which members must achieve to attain the promotion.

Signature

```
public Integer getCumulativeUsageTarget()
```

Return Value

Type: Integer

getErrors()

If an error occurred, returns a list providing the error code and description.

Signature

```
public List<LoyaltyManagement.LoyaltyActionError> getErrors()
```

Return Value

Type: List<LoyaltyManagement.LoyaltyActionError>

getUpdatedCumulativeUsageCompleted()

Returns the updated value or quantity by which the member has progressed towards attaining the promotion.

Signature

```
public Integer getUpdatedCumulativeUsageCompleted()
```

Return Value

Type: Integer

isSuccess()

Returns a Boolean that is set to true if the update cumulative usage completed operation was successful for this object, false otherwise.

Signature

```
public Boolean isSuccess()
```

Return Value

Type: Boolean

CHAPTER 6 Loyalty Management Business APIs

In this chapter ...

- [REST Reference](#)
- [Loyalty Management Connect API](#)
[Interchangeable Request Fields](#)

Loyalty Management Business APIs are RESTful APIs that are sometimes available as Apex classes and methods.

REST Reference

You can access Loyalty Management APIs using REST endpoints. These REST APIs follow similar conventions as Connect REST APIs.

To understand the architecture, authentication, rate limits, and how the requests and responses work, see Connect REST API Developer Guide.

[Resources](#)

Here's a list of Loyalty Management API resources.

[Request Bodies](#)

Here's a list of Loyalty Management API request bodies.

[Response Bodies](#)

Here's a list of Loyalty Management API response bodies.

Resources

Here's a list of Loyalty Management API resources.

[Club Member Enrollment](#)

Enroll members in a club and manage their details, including membership numbers, renewal periods, and end dates.

[Club Member Profile](#)

Get the profile details of a club member such as membership number, contact ID, and loyalty program name.

[Club Membership Renewal](#)

Renew the membership of club members and manage their membership details such as membership numbers, renewal periods, and end dates.

[Corporate Member Enrollments](#)

Enroll corporate members in a loyalty program. Create a business account and a Corporate type loyalty program member record for the new member.

[Individual Member Enrollments](#)

Enroll individual members in a loyalty program. Create a person account and an Individual type loyalty program member record for the new member.

[Link Program Member Partner](#)

Link a loyalty program member with a specified partner program. Enable integration between loyalty members and partner ecosystems, allowing members to earn, redeem, and manage benefits across connected programs. This API simplifies account linkage, enhances transparency, and supports real-time synchronization between loyalty and partner systems.

[Member Benefits](#)

Get benefits for a loyalty program member.

[Member Profile](#)

Get the details of a loyalty program member.

[Member Engagement Trail](#)

Fetches current and target values of the milestones defined in an engagement trail and the rewards received after achieving milestones.

Member Vouchers

Get vouchers that have been issued to a loyalty program member.

Redeem Voucher

Redeem a voucher for a loyalty program member. The status of the voucher gets updated to redeemed and the use date gets updated to the current date.

Transaction History

Get transaction journals of a loyalty member belonging to a loyalty program. Obtain either accrual or redemption transaction journal records.

Transaction Ledger Summary

Retrieve an aggregate view on the transaction journal and the loyalty ledger records.

Transaction Journals Execution

Use this resource to create and process a list of transaction journals against a corresponding loyalty program process.

Transaction Journals Simulation

Runs a simulation to determine how transaction journals are processed against their corresponding loyalty program processes. When transaction journals meet the criteria and conditions for a program process, actions that are set up in the process are triggered for the transaction journals.

Unlink Program Member Partner

Unlinks an existing association between a loyalty program member and a partner program. The API securely removes the linkage and updates related records to ensure data consistency across loyalty and partner systems.

Loyalty Program Process

Run a loyalty process created from a loyalty program process template.

Gamification

Who doesn't like to play a game? Engage your customers more by gamifying customer activities, such as placing an order, or events, such as birthdays and anniversaries.

Promotion

The promotion APIs allow you to set eligibility rules and limits for a promotion, choose a promotion template, and create a promotion. You can also get the details of a specific promotion, and view a list of eligible promotions for a customer's cart.

Club Member Enrollment

Enroll members in a club and manage their details, including membership numbers, renewal periods, and end dates.

Resource

```
loyalty-programs/{programName}/clubs/{clubName}/member-enrollment
```

Resource Example

```
loyalty-programs/SampleProgramName/clubs/SampleClubName/member-enrollment
```

Available version

64.0

HTTP methods

POST

URI Parameters

Name	Type	Description	Required or Optional	Available Version
programName	Integer	Name of the loyalty program.	Required	64.0
clubName	Integer	Name of the club.	Required	64.0

Request body for POST

```
Enroll an Existing Loyalty Program Member

{
  "enrollmentDate": "2025-03-10T00:00:00",
  "clubMembershipNumber": "NAME1",
  "loyaltyMembershipNumber": "ARC0001",
  "memberStatus": "Active",
  "membershipEndDate": "2025-03-25T00:00:00"
}

Enroll an Existing Contact (Person Account is Not needed)

{
  "enrollmentDate": "2025-03-10T00:00:00",
  "clubMembershipNumber": "NAME2",
  "contactId": "003xx000004WhFrAAK",
  "memberStatus": "Active",
  "membershipEndDate": "2025-03-25T00:00:00"
}

Enroll a New Person (Person Account is Not needed)

{
  "enrollmentDate": "2025-03-10T00:00:00",
  "clubMembershipNumber": "NAME3",
  "associatedContactDetails": {
    "firstName": "Name3",
    "lastName": "Name3",
    "email": "Name3@email.com",
    "allowDuplicateRecords": "false"
  },
  "memberStatus": "Active",
  "membershipEndDate": "2025-03-25T00:00:00"
}

Enroll an Existing Account

{
```

```

    "enrollmentDate": "2025-03-10T00:00:00",
    "clubMembershipNumber": "NAME4",
    "accountId": "001xx000003GYiBAAW",
    "memberStatus": "Active",
    "membershipEndDate": "2025-03-25T00:00:00"
  }

```

Enroll a New Account (Person Account is Not needed)

```

{
  "enrollmentDate": "2025-03-10T00:00:00",
  "clubMembershipNumber": "NAME5",
  "associatedAccountDetails": {
    "name": "NAME5",
    "phone": "0123456789",
    "website": "www.name5.com",
    "allowDuplicateRecords": "false"
  },
  "memberStatus": "Active",
  "membershipEndDate": "2025-03-25T00:00:00"
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
accountId	String	The ID of the associated account record.		64.0
associated Account Details	Loyalty Account Input on page 479	The business account details of the member.		64.0
associated Contact Details	Loyalty Contact Input on page 480	The contact associated with the club member.		64.0
club Membership Number	String	The membership number of the club member.		64.0
contactId	String	The contact details of the member.		64.0
enrollment Date	String	The date on which a member is enrolled into the club.		64.0
loyalty Membership Number	String	The membership number of the loyalty program member.		64.0
memberStatus	String	The status of the club member.		64.0
membershipBy	String	Member enrolled based on.		64.0

Name	Type	Description	Required or Optional	Available Version
membership EndDate	String	The end date of the club membership.		64.0

Response body for POST

[Club Member Enrollment Details](#) on page 521

Club Member Profile

Get the profile details of a club member such as membership number, contact ID, and loyalty program name.

Resource

```
/club/members
```

Resource Example

```
/club/members?clubName=NAME5&programName=SampleProgram
```

Available version

64.0

HTTP methods

GET

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
club Membership Number	String	The membership number of the club member.		64.0
clubName	String	The name of the club.		64.0
contactId	String	The contact ID of the member.		64.0
loyalty Membership Number	String	The membership number of the loyalty program member.		64.0
programName	String	The name of the loyalty program.		64.0

Response body for GET

[Club Member Details](#) on page 520

Club Membership Renewal

Renew the membership of club members and manage their membership details such as membership numbers, renewal periods, and end dates.

Resource

```
/loyalty-programs/${loyaltyProgramName}/clubs/${clubName}/membership-renewal
```

Resource Example

```
/loyalty-programs/SampleLoyaltyProgramName/clubs/SampleClubName/membership-renewal
```

Available version

64.0

HTTP methods

POST

URI Parameters

Name	Type	Description	Required or Optional	Available Version
programName	Integer	Name of the loyalty program. In case of an Independent Club, use the value Club Program.	Required	64.0
clubName	Integer	Name of the club.	Required	64.0

Request body for POST**JSON example**

```
{
  "clubMembershipNumber": "ARC0001",
  "renewalPeriodType": "months",
  "renewalPeriod": "3",
  "notes": "",
  "externalIdentifier": "",
  "OrderItemId" : ""
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
club Membership EndDate	String	The end date of the club membership.		64.0
club Membership Number	String	The membership number of the club member.		64.0
external Identifier	String	The external identifier of the club member.		64.0

Name	Type	Description	Required or Optional	Available Version
loyalty Membership Number	String	The membership number of the loyalty program member.		64.0
notes	String	Additional information about the club membership renewal.		64.0
orderId	String	The order associated with the club membership. This field is a relationship field.		64.0
renewal Period	Integer	The specific time period during which a club membership can be renewed.		64.0
renewal PeriodType	String	The unit of time used to measure the renewal period for a club membership. For example, Days , Months, or Years.		64.0

Response body for POST

[Club Membership Renewal](#) on page 522

Corporate Member Enrollments

Enroll corporate members in a loyalty program. Create a business account and a Corporate type loyalty program member record for the new member.

To use this resource, Person Accounts must be enabled in your org. The profile of the user who executes the resource must have Business Account selected as the default record type for business accounts. If the profile of the user has a different record type selected as the default record type for business accounts, an Account record of the selected record type is created for the member.

Resource

```
/loyalty-programs/${loyaltyProgramName}/corporate-member-enrollments
```

Example

```
https://yourInstance.salesforce.com/services/data/vXX.X
/loyalty-programs/program1/corporate-member-enrollments
```

Available version

55.0

Requires Chatter

No

HTTP methods

POST

Request body for POST**Root XML tag**

```
<corporateMemberEnrollment>
```

JSON example

```
{
  "enrollmentDate" : "2021-01-01T05:05:05",
  "membershipNumber" : "Member123",
  "associatedAccountDetails" : {
    "name" : "Name1",
    "phone" : "908989898",
    "website" : "www.ab.com",
    "allowDuplicateRecords" : "false",
    "additionalAccountFieldValues" : {
      "attributes" : {
        "CustomNumber__c" : "20.0"
      }
    }
  },
  "memberStatus" : "Active",
  "referredBy" : "Member12",
  "referredByMemberReferralCode" : "MPX12adf",
  "createTransactionJournals" : "true",
  "transactionJournalStatementFrequency" : "Monthly",
  "additionalMemberFieldValues" : {
    "attributes" : {
      "CustomBoolean__c" : "true"
    }
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalMemberFieldValues	Map<String, String>	The custom attributes of the member in the loyalty program member record. • Keys • Values	Optional	55.0
associatedAccountDetails	Loyalty Account Input	The business account details of the member.	Required	55.0
canReceivePartnerPromotions	Boolean	Indicates whether the member can receive partner promotions for the program (<code>true</code>) or not (<code>false</code>).	Optional	55.0
canReceivePromotions	Boolean	Indicates whether the member can receive promotions for the program.	Optional	55.0

Name	Type	Description	Required or Optional	Available Version
createTransaction Journals	Boolean	Indicates whether to create the transaction journal records (<code>true</code>) or not (<code>false</code>): The transaction journal records are created to: • Credit points to the member for enrolling. • Credit points to the member who has referred the new member.	Optional	55.0
enrollmentChannel	String	The channel used by members to enroll into a loyalty program. Possible values are: • CallCenter • Email • Franchise • Mobile • Partner • Pos • Print • Social • Store • Web	Optional	55.0
enrollmentDate	String	The date on which a member is enrolled.	Required	55.0
memberStatus	String	Status of the member. Possible values are: • Active • Inactive • Custom value	Required	55.0
membershipEndDate	String	The date on which the loyalty program member's membership ends.	Optional	55.0
membershipNumber	String	The unique membership number of the loyalty program member.	Required	55.0
referredBy	String	The ID of the loyalty program member who has referred the member.	Optional	55.0
referralCode	String	The referral code of the loyalty program member who has referred the member.	Optional	57.0

Name	Type	Description	Required or Optional	Available Version
transactionJournalStatementFrequency	String	The frequency at which transaction journal statements must be delivered to the member. Possible values are: - Daily - Monthly - Yearly	Optional	55.0
transactionJournalStatementMethod	String	The method used to deliver transaction journal statements to the member. Possible values are: - Email - Mail	Optional	55.0

Response body for POST[Corporate Member Enrollment Output](#)

Individual Member Enrollments

Enroll individual members in a loyalty program. Create a person account and an Individual type loyalty program member record for the new member.

To use this resource, Person Accounts must be enabled in your org. The profile of the user who executes the resource must have Person Account selected as the default record type for person accounts. If the profile of the user has a different record type selected as the default record type for person accounts, an Account record of the selected record type is created for the member.

Resource

```
/loyalty-programs/${loyaltyProgramName}/individual-member-enrollments
```

Example

```
https://yourInstance.salesforce.com/services/data/vXX.X
/loyalty-programs/program1/individual-member-enrollments
```

Available version

55.0

Requires Chatter

No

HTTP methods

POST

Request body for POST**Root XML tag**

```
<individualMemberEnrollment>
```

JSON example

```
{
  "enrollmentDate" : "2021-01-01T00:00:00",
  "membershipNumber" : "Member1",
  "associatedContactDetails" : {
    "firstName" : "Name1",
    "lastName" : "Name2",
    "email" : "ab@email.com",
    "allowDuplicateRecords" : "false",
    "additionalContactFieldValues" : {
      "attributes" : {
        "PersonMailingCity" : "Delhi",
        "CustomNumber__c" : "20.0"      }
    }
  },

  "memberStatus" : "Active",
  "referredBy" : "Member2",
  "referredByMemberReferralCode" : "MPX12adf",
  "createTransactionJournals" : "true",
  "transactionJournalStatementFrequency" : "Monthly",
  "transactionJournalStatementMethod" : "Mail",
  "enrollmentChannel" : "Email",
  "canReceivePromotions" : "true",
  "canReceivePartnerPromotions" : "true",
  "membershipEndDate" : "2022-01-01T00:00:00",
  "relatedCorporateMembershipNumber" : "Member3",
  "additionalMemberFieldValues" : {
    "attributes" : {
      "CustomBoolean__c" : "true"      }
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
<code>additionalMemberFieldValues</code>	Map<String, String>	The custom attributes of the member in the loyalty program member record. • Keys • Values	Required	55.0
<code>associatedContactDetails</code> on page 480	Loyalty Contact Input on page 480	The person account record details of the individual type member.	Required	55.0
<code>canReceivePartnerPromotions</code>	Boolean	Indicates whether the member can receive partner promotions for the program (true) or not (false).	Optional	55.0

Name	Type	Description	Required or Optional	Available Version
~~canReceivePromotions~~	Boolean	Indicates whether the loyalty program member can receive promotions.	Optional	55.0
~~createTransaction Journals~~	Boolean	Indicates whether to create the transaction journal records (<code>true</code>) or not (<code>false</code>): The transaction journal records are created to: • Credit points to the member for enrolling. • Credit points to the member who has referred the new member.	Optional	55.0
~~enrollmentChannel~~	String	The channel used to by loyalty program member to enroll in to the loyalty program. Possible values are: • Pos • Web • Email • CallCenter • Social • Mobile • Store • Franchise • Partner • Print	Optional	55.0
~~enrollmentDate~~	Date	The date on which the loyalty program member is enrolled in to the loyalty program. The format is: <code>YYYY-MM-DDTHH:mm:ss</code>. For example: <code>2020-09-08T00:00:00</code>. This field must be provided in the org's configured timezone, not in Central UTC. The system doesn't convert this value from UTC to org timezone internally. In other words, the provided datetime already represents the correct org-local enrollment date and time.	Required	55.0

Name	Type	Description	Required or Optional	Available Version
memberStatus	String	The status of the loyalty program member. Possible values are: - Active - Inactive - Custom	Required	55.0
membershipEndDate	String	The date on which the loyalty program member's membership ends.	Optional	55.0
membershipNumber	String	The unique membership number of the loyalty program member.	Required	55.0
referredBy	String	The membership number of the loyalty program member who has referred the member.	Optional	55.0
referralCode	String	The referral code of the loyalty program member who has referred the member.	Optional	57.0
relatedCorporateMembershipNumber	String	External corporate membership number of the member.	Optional	55.0
transactionJournalStatementFrequency	String	The frequency at which transaction journal statements must be delivered to the member. Possible values are: - Monthly - Quarterly	Optional	55.0
transactionJournalStatementMethod	String	The method used to deliver transaction journal statements to the member. Possible values are: - Mail - Email	Optional	55.0

Response body for POST[Individual Member Enrollment Output](#)

Link Program Member Partner

Link a loyalty program member with a specified partner program. Enable integration between loyalty members and partner ecosystems, allowing members to earn, redeem, and manage benefits across connected programs. This API simplifies account linkage, enhances transparency, and supports real-time synchronization between loyalty and partner systems.

Resource

```
/loyalty/programs/programName/members/membershipNumber/partners/partnerName/link
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v66.0/loyalty/programs/MarriottBonvoy/members/M0001/partners/AMEX/link
```

Available version

66.0

HTTP methods

POST

Request body for POST**JSON example**

```
{
  "externalId": "U10001",
  "additionalAttributes": {
    "Card_Type__c": "Green Card"
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalAttributes	Object	Attributes of the member that must be saved in the linkage record.	Optional	66.0
externalId	String	External ID of the loyalty program member associated with the partner.	Optional	66.0

Response body for POST[Link Program Member Partner](#)

Member Benefits

Get benefits for a loyalty program member.

Resource

```
/connect/loyalty/member/${memberId}/memberbenefits
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v66.0/connect/loyalty/member/01M0000A0th124242af/memberbenefits
```

Available version

51.0

Requires Chatter

No

HTTP methods

GET

Response body[Member Benefits Output](#)

Member Profile

Get the details of a loyalty program member.

 **Note:** For community users, the Get Member Profile API supports context-based resolution. When membershipNumber isn't provided, the API automatically returns the profile of the logged-in member.

Resource

```
/loyalty-programs/{loyaltyProgramName}/members
```

Examples

```
https://yourInstance.salesforce.com/services/data/v66.0/loyalty-programs/{baLoyalty}/members
https://yourInstance.salesforce.com/services/data/v66.0/loyalty-programs/{baLoyalty}/members?memberId=007
https://yourInstance.salesforce.com/services/data/v66.0/loyalty-programs/{baLoyalty}/members?membershipNumber=doc_member1
https://yourInstance.salesforce.com/services/data/v66.0/loyalty-programs/{baLoyalty}/members?programCurrencyName=Yen
```

Available version

55.0

Requires Chatter

No

HTTP methods

GET

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
memberId	String	The ID of the loyalty program member.	Required	55.0
			 Note: This field is optional if you provide the membership number.	
membershipNumber	String	The membership number of the loyalty program member. If this value is omitted or empty, and the request is made by a logged-in community user, the API returns the member profile associated with the current user context.	Required	55.0
			 Note: This field is optional if you provide the member ID.	

Parameter Name	Type	Description	Required or Optional	Available Version
programCurrencyName	String	The name of the loyalty program currency associated with the member. Use this parameter to get the details of the member's points-related information for a specific currency.	Optional	55.0

Response body for GET[Member Profile Output](#)

Member Engagement Trail

Fetches current and target values of the milestones defined in an engagement trail and the rewards received after achieving milestones.

Resource

```
/services/data/vXX.X/loyalty/programs/{programName}/members/{membershipNumber}/engagement-trail?$promotionId={promotionId}
```

Examples

```
https://yourInstance.salesforce.com/services/data/vXX.X
/loyalty/programs/{programName}/members/{membershipNumber}/engagement-trail?$promotionId={promotionId}
```

```
https://yourInstance.salesforce.com/services/data/vXX.X
/loyalty/programs/{programName}/members/{membershipNumber}/engagement-trail?$promotionId={promotionId}
```

Available version

63.0

Requires Chatter

No

HTTP methods

GET

Query Parameters

Parameter Name	Type	Description	Required or Optional	Available Version
promotionId	String	The ID of the promotion.	Required	63.0

Path Parameters

Parameter Name	Type	Description	Required or Optional	Available Version
programName	String	Name of the loyalty program.	Required	63.0
membershipNumber	String	The membership number of the loyalty program member.	Required	63.0

Response Body

[Member Engagement Trail Output](#) on page 553

Member Vouchers

Get vouchers that have been issued to a loyalty program member.

Resource

```
/loyalty/programs/{programName}/members/{membershipNumber}/vouchers
```

Resource Examples

```
https://yourInstance.salesforce.com/services/data/v58.0  
/loyalty/programs/NTO/members/M1/vouchers
```

```
https://yourInstance.salesforce.com/services/data/v58.0  
/loyalty/programs/NTO/members/M1/vouchers?voucherStatus=Cancelled,Expired
```

```
https://yourInstance.salesforce.com/services/data/v58.0  
/loyalty/programs/NTO/members/M1/vouchers?pageNumber=2&voucherStatus=Issued&productName=Linea,Creta
```

Available version

58.0

Requires Chatter

No

HTTP methods

GET

Query parameters

Parameter Name	Type	Description	Required or Optional	Available Version
pageNumber	Integer	The page number to be returned. If you don't specify a value, the first page is returned.  Note: Each page contains 200 vouchers, sorted by their date of creation.	Optional	58.0
productCategoryId	String	The ID of a product category associated with the member vouchers to be retrieved.  Note: You can specify up to 20 product category IDs.	Optional	58.0
productCategoryName	String	The product category name associated with the vouchers to be retrieved.  Note: You can specify up to 20 product categories.	Optional	58.0

Parameter Name	Type	Description	Required or Optional	Available Version
productId	String	The product ID associated with the vouchers to be retrieved.  Note: You can specify up to 20 product IDs.	Optional	58.0
productName	String	The product name associated with the vouchers to be retrieved.  Note: You can specify up to 20 product names.	Optional	58.0
voucherStatus	String	The status of the member vouchers to filter the API response. Possible values are: - Cancelled - Expired - Issued - Redeemed	Optional	58.0
sortBy	String	The voucher field for sorting the API response. The default value is <code>ExpirationDate</code> . Possible values are: - CreatedDate - EffectiveDate - ExpirationDate	Optional	58.0
sortOrder	String	The sort order of the API response. The default value is <code>Ascending</code> . Possible values are: - Ascending - Descending	Optional	58.0

Response body for GET[Vouchers List](#)

Redeem Voucher

Redeem a voucher for a loyalty program member. The status of the voucher gets updated to redeemed and the use date gets updated to the current date.

Resource

```
/services/data/vXX.X/loyalty/programs/${programName}/members/${membershipNumber}/vouchers/${voucherCode}/redeem
```

Examples

```
https://yourInstance.salesforce.com/services/data/vXX.X
/loyalty/programs/DemoProgram/members/M1234/vouchers/VC001/redeem
```

```
https://yourInstance.salesforce.com/services/data/vXX.X
/loyalty/programs/DemoProgram/members/M1234/vouchers/VC001/redeem?voucherNumber=00000006
```

Available version

51.0

Requires Chatter

No

HTTP methods

POST

Parameters for POST

Parameter Name	Type	Description	Required or Optional	Available Version
membershipNumber	String	The loyalty program membership number.	Required	51.0
programName	String	The loyalty program name.	Required	51.0
voucherCode	String	The voucher code to be redeemed.  Note: The voucher definition associated with the voucher code must be active to redeem the voucher.	Required	51.0
voucherNumber	String	The auto-generated voucher number.  Note: Use this query parameter if multiple vouchers exist with the same voucher code.	Optional	55.0

Request body for POST

The request body is available in version 53.0 and later.

JSON example

```
{
  "contactID": "003xx000004WjzAAAS",
  "action": "Reserve",
  "reservationKey": "key1",
  "reservedValue": 50.57,
  "redeemAmount": 50.57,
  "currencyIsoCode": "Yen"
}
```

JSON example to redeem vouchers using `contactId` or `accountId`.

```
{
  "redemptionAmount": "12",
  "contactId": "003xx000004WkhEAAS",
  "accountId": "001xx000003H3NnAAK",
  "memberId": "01Mxx00000002HxEAI"
}
```



Note: Provide `contactId` to redeem the voucher. If `accountId` is provided, then the associated `contactId` is fetched and the voucher is redeemed

Properties

Name	Type	Description	Required or Optional	Available Version
<code>contactId</code>	String	ID of the contact associated with the account.	Optional	62.0
<code>currencyIsoCode</code>	String	Three-letter ISO 4217 currency code of the monetary currency in which the voucher is redeemed.	Optional	55.0
<code>redeemAmount</code>	Double	The value of the voucher to be redeemed. Specify a value when a member hasn't redeemed the full value of the voucher. This value is only applicable for vouchers issued using Fixed Value voucher definitions. A value isn't required when the member has redeemed the complete value of the voucher in a single transaction.  Note: If you don't specify a value to be redeemed for the fixed-value voucher, the voucher status changes to <code>Redeemed</code>, and the remaining value becomes zero.	Optional	53.0
<code>action</code>	ReservationAction	The action to change the status of a voucher. Valid values are: • <code>Reserve</code> • <code>Reinstate</code> • <code>Redeem</code> If you don't specify an action, the default behavior changes voucher status from <code>Issued</code> to <code>Redeemed</code>.	Required when changing the voucher status from <code>Issued</code> to <code>Reserved</code> , or from <code>Reserved</code> to <code>Issued</code> or <code>Redeemed</code> . Otherwise it is optional.	62.0

Name	Type	Description	Required or Optional	Available Version
reservedValue	Currency	Specify the amount reserved for partially redeemable vouchers.	Optional	62.0
reservationKey	String	Specify the key used to redeem or reinstate a reserved voucher.	Optional	62.0

Response body for POST

[Redeem Voucher Output](#) on page 588 (available in version 53.0 and later)

This table lists HTTP response code descriptions that are unique to this resource.

HTTP Response Code	Error Code	Description
400	INVALID_API_INPUT	The voucher is expired. Specify a voucher that hasn't expired.
400	INVALID_API_INPUT	The voucher is canceled. Specify a voucher that hasn't canceled.
400	INVALID_API_INPUT	Specify a valid voucher code and try again.
400	INVALID_API_INPUT	Specify a valid loyalty program name and try again.
400	INVALID_API_INPUT	Specify a valid loyalty program membership number and try again.
400	INVALID_API_INPUT	The voucher definition is inactive. Specify a voucher that's associated with an active voucher definition.
409	ALREADY_REDEEMED_VOUCHER	The voucher is redeemed. Specify a voucher that hasn't redeemed.
500	MULTIPLE_VOUCHERS	Duplicate vouchers for this voucher code. Either specify a different voucher code or ask your Salesforce admin for help.
500	UNKNOWN_EXCEPTION	The voucher redemption failed. Try again or ask your Salesforce admin for help.

SEE ALSO:

[Connect REST API Developer Guide: Status Codes and Error Responses](#)

Transaction History

Get transaction journals of a loyalty member belonging to a loyalty program. Obtain either accrual or redemption transaction journal records.

Resource

```
/connect/loyalty/programs/${programName}/transaction-history
```

Example

```
https://yourInstance.salesforce.com/services/data/v66.0/connect/loyalty/programs/NIO/transaction-history?page=2
```

Available version

56.0

Requires Chatter

No

HTTP methods

POST

Request parameters for POST

Parameter Name	Type	Description	Required or Optional	Available Version
programName	String	Name of the loyalty program.	Required	56.0
page	Integer	Number of the page you want returned. If you don't specify a value, the first page is returned.	Optional	56.0

 **Note:** By default, each page contains 200 records.

Request body for POST**JSON example**

```
{
  "membershipNumber": "doc_member1",
  "journalType": "Accrual",
  "journalSubType": "Social",
  "periodStartDate": "2022-01-27T00:00:00Z",
  "periodEndDate": "2022-11-27T00:00:00Z"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
journalSubType	String	The subtype of the transaction journal record.	Optional	56.0
journalType	String	The type of the transaction journal. Possible values include: • Accrual • Redemption	Required	56.0
membershipNumber	String	The membership number of the loyalty program member.	Required	56.0
periodEndDate	String	Retrieve transaction journals until this date.	Optional	56.0

Name	Type	Description	Required or Optional	Available Version
periodStartDate	String	Retrieve transaction journals starting on this date.	Required	56.0

Response body for POST

[Loyalty View Transaction Journal History Output](#)

Transaction Ledger Summary

Retrieve an aggregate view on the transaction journal and the loyalty ledger records.

Resource

```
/loyalty/programs/{programName}/members/{membershipNumber}/transaction-ledger-summary
```

Resource Examples

```
https://yourInstance.salesforce.com/services/data/v66.0/loyalty/programs/NTO/members/M1/transaction-ledger-summary
```

```
https://yourInstance.salesforce.com/services/data/v66.0/loyalty/programs/NTO/members/M1/transaction-ledger-summary?journalSubType={provide_value}
```

```
https://yourInstance.salesforce.com/services/data/v66.0/loyalty/programs/NTO/members/M1/transaction-ledger-summary?journalType={provide_value}&journalSubType={provide_value}
```

Available version

58.0

Requires Chatter

No

HTTP methods

GET

Query parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
journalSubTypeName	String	Returns the transaction journals filtered by the specified journal subtype.	Optional	58.0
additionalFields	String	Returns the list of transaction journal fields.	Optional	58.0
journalTypeName	String	Returns the transaction journals filtered by the specified journal type.	Optional	58.0
pageNumber	Integer	Returns the transaction journals for the specified page number.	Optional	58.0

 **Note:** The API returns the first page by default. Each page contains 200 transaction journals, which are sorted according to the

Parameter Name	Type	Description	Required or Optional	Available Version
		date the transaction journal record was created.		
periodEndDate	String	Returns the transaction journals filtered by the specified end date.	Optional	58.0
periodStartDate	String	Returns the transaction journals filtered by the specified start date.	Optional	58.0

Response body for GET[Transaction Journal Summary Output](#)

Transaction Journals Execution

Use this resource to create and process a list of transaction journals against a corresponding loyalty program process.

The loyalty program processes determine how transaction journals are processed. When transaction journals meet the criteria and conditions for a program process, actions that are set up in the process are triggered for the transaction journals. For more information on how to set up program process, see [LoyaltyProgramSetup](#).

Special Access Rules

To use this resource, your org must have either B2C - Loyalty, B2C - Loyalty Plus, Loyalty Management - Growth, or Loyalty Management - Advanced license enabled, and you must be assigned the Loyalty Management permission set.

Resource

```
/connect/realtime/loyalty/programs/${programName}
```

Resource example

```
https://yourInstance.salesforce.com/services/data/vXX.X/connect/realtime/loyalty/programs/CloudKicksInnerCircle
```

Available version

54.0

Requires Chatter

No

HTTP methods

POST

Request body for POST**JSON example**

In the JSON, you can either provide the details of transaction journals that aren't available in the org, or a list of IDs that belong to transaction journals available in the org. If you provide the details of transaction journals that aren't available in the org, each transaction journal's applicable loyalty program process applies the eligible rules, executes the applicable actions, and then creates the transaction journal.

Using the transaction journal details:

 **Note:** Some of the API request fields contain field names that are interchangeable. For information on the list of interchangeable field names, see [Loyalty Management Connect API Interchangeable Request Fields](#).

Here's a JSON example that creates a transaction journal for a purchase-related transaction made by a loyalty program member of a retail company. It has all the transaction journal-related information along with the details of the promotion applied, including the different types of rewards the transaction is eligible for. For information on the TransactionJournal object fields, see https://developer.salesforce.com/docs/atlas.en-us.loyalty.meta/loyalty/sforce_api_objects_transactionjournal.htm.

```
{
  "transactionJournals": [
    {
      "ActivityDate": "2023-11-27T12:45:19Z",
      "JournalTypeId": "01ERM00000001Bb2AI",
      "LoyaltyProgramId": "01pRM00000002YTYAY",
      "MemberId": "01MRM00000002Fmu2AE",
      "Status": "Pending",
      "ProductId": "01txx00000006iTsAAI",
      "ProductCategoryId": "0ZSSB00000002d8X4AQ",
      "Quantity": "4",
      "TransactionAmount": "500",
      "appliedPromotions": [
        {
          "promotionId": "0c8RM00000004FiXYAU",
          "couponCode": "COUPON100",
          "rewards": [
            {
              "rewardType": "Discount",
              "discountAmount": 15
            },
            {
              "rewardType": "Points",
              "loyaltyProgramCurrencyName": "Coins",
              "points": 100,
              "relatedInformation": "Test",
              "notes": "points reward"
            },
            {
              "rewardType": "Voucher",
              "voucherDefinitionName": "10% Discount Voucher",
              "voucherExpirationDate": "2023-12-15",
              "voucherEffectiveDate": "2023-11-29",
              "notes": "discount voucher"
            },
            {
              "rewardType": "Badge",
              "loyaltyProgramBadgeName": "Gen AI Badge",
              "memberBadgeStatus": "Active",
              "badgeValidityEndDate": "2023-12-15",
              "reason": "purchase"
            },
            {
              "rewardType": "Game",
              "gameDefinitionName": "SpintheWheelGame"
            }
          ]
        }
      ]
    }
  ]
}
```



```

    }
  ]
}
}

```

Here's a JSON example to create a transaction journal for a loyalty program member of an airline company.

```

{
  "transactionJournals": [
    {
      "ActivityDate": "2022-01-04T12:45:19Z",
      "JournalDate": "2022-01-04T00:45:19Z",
      "Brand": "Tata",
      "Establishment": "Vistara",
      "ExternalTransactionNumber": "P1-981950",
      "JournalTypeId": "01ET10000004CQp",
      "LoyaltyProgramId": "0lpT10000004CdN",
      "MemberId": "0lMT10000004CfA",
      "TransactionAmount": "117",
      "TransactionLocation": "HitechCity",
      "VoucherCode": "",
      "Origin": "Bangalore",
      "Distance": "5",
      "Destination": "Hyderabad",
      "FlightNumber": "VS-Ran-001",
      "Status": "Pending"
    }
  ]
}

```

Using the transaction journal ID:

 **Note:** This is supported in API version 55.0 and later.

```

{
  "transactionJournals": [
    {
      "Id": "01Vxx00000000cj"
    }
  ]
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
processInput	Map<String, Object>	List of the Transaction Journal records to process and create.	Required	54.0

Response body for POST

[Loyalty Engine Realtime Output](#)

Transaction Journals Simulation

Runs a simulation to determine how transaction journals are processed against their corresponding loyalty program processes. When transaction journals meet the criteria and conditions for a program process, actions that are set up in the process are triggered for the transaction journals.

Special Access Rules

To use this resource, your org must have either B2C - Loyalty, B2C - Loyalty Plus, Loyalty Management - Growth, or Loyalty Management - Advanced license enabled, and you must be assigned the Loyalty Management permission set.

Resource

```
/connect/realtime/loyalty/programs/${programName}
```

Resource example

```
https://yourInstance.salesforce.com/services/data/vXX.X  
/connect/realtime/loyalty/programs/StarAir_Program
```

Available version

55.0

Requires Chatter

No

HTTP methods

POST

Request body for POST

Root XML tag

```
<loyaltyEngineInput>
```

JSON example

 **Note:** Some of the API request fields contain field names that are interchangeable. For information on the list of interchangeable field names, see [Loyalty Management Connect API Interchangeable Request Fields](#).

```
{  
  "shouldCheckCouponUsageLimit": true  
  "transactionJournals": [  
    {  
      "ActivityDate": "2022-01-04T12:45:19Z",  
      "JournalDate": "2022-01-04T00:45:19Z",  
      "Brand": "",  
      "Establishment": "Apple",  
      "ExternalTransactionNumber": "P1-981950",  
      "JournalTypeId": "01ET10000004CQp",  
      "LoyaltyProgramId": "01pT10000004CdN",  
      "MemberId": "01MT10000004CfA",  
      "TransactionAmount": "117",  
      "TransactionLocation": "HitechCity",  
      "VoucherCode": "",  
      "Origin": "asdf",  
      "Distance": "5",  
      "Destination": "Hyderabad",  
      "FlightNumber": "SF-Ran-001",  
      "Status": "Pending"  
    }  
  ]  
}
```

```

],
"runSetting": {
  "isSimulation": true
}
}

```

 **Note:** Provide `contactId` to retrieve rewards related to a Person Account.

Using the transaction journal ID:

```

{
  "transactionJournals": [
    {
      "Id": "01Vxx000000000cj"
    }
  ],
  "runSetting": {
    "isSimulation": true
  }
}

```

Sample payload to retrieve rewards assigned to accounts.

```

{
  "transactionJournals": [
    {
      "ActivityDate": "2024-11-27T12:45:19Z",
      "JournalTypeId": "01ERM00000001Bb2AI",
      "LoyaltyProgramId": "01pRM000000002YTYAY",
      "ReferredPartyId": "001xx000003H3NnAAK",
      "Status": "Pending",
      "ProductId": "01txx00000006i44AAA",
      "ProductCategoryId": "0ZGxx0000000001GAA",
      "Quantity": "1",
      "TransactionAmount": "42",
      "appliedPromotions": [
        {
          "promotionId": "0c8xx00000000mPAAQ",
          "rewards": [
            {
              "rewardType": "Discount",
              "discountAmount": 5
            },
            {
              "rewardType": "Discount",
              "discountAmount": 10
            },
            {
              "rewardType": "Voucher",
              "voucherDefinitionName": "Voucher2",
              "voucherExpirationDate": "2025-08-07",
              "voucherEffectiveDate": "2024-08-08",
            },
            {
              "rewardType": "Game",

```

```
        "gameDefinitionName": "TestPostFacto",
      },
    ]
  }
}
]
```

Properties

Name	Type	Description	Required or Optional	Available Version
enableCouponLimit	Boolean	Indicates whether the coupon usage limit check is enabled.	Required	62.0
transactionJournals	Loyalty Engine Input Map	Represents the transaction journals records to process.	Required	54.0
runSetting	Loyalty Engine Run Settings Input	Loyalty runtime setting to run the simulation on the transaction journals.	Required for simulation, otherwise optional	55.0

Response body for POST

[Loyalty Engine Realtime Output](#)

JSON example

```
{
  "message": null,
  "processResult": null,
  "programName": "StarAir_Program",
  "simulationResults": {
    "transactionJournals": [
      {
        "executionSummary": {
          "pointsSummary": [
            {
              "changeInEscrowPointsBalance": 0,
              "changeInPointsBalance": 250,
              "loyaltyProgramCurrencyName": "VroomKing",
              "loyaltyProgramCurrencyType": "NonQualifying"
            },
            {
              "changeInEscrowPointsBalance": 20,
              "changeInPointsBalance": 0,
              "loyaltyProgramCurrencyName": "EcoMiles",
              "loyaltyProgramCurrencyType": "Qualifying"
            }
          ]
        },
        "vouchersIssuedCount": 0
      },
      {
        "processId": "9MXxx0000004Khg",

```

```

"processName": "Redemption Process",
"processRules": [
  {
    "actionResultSummary": [
      {
        "actionType": "CreditPoints",
        "changeInEscrowPointsBalance": 20,
        "changeInPointsBalance": 0,
        "escrowPointsCreditDate": "2022-03-05T00:00:00.000Z",
        "loyaltyProgramCurrencyName": "EcoMiles",
        "promotionId": "",
        "stepName": "Credit Brownie Points"
      },
      {
        "actionType": "CreditPoints",
        "changeInEscrowPointsBalance": 0,
        "changeInPointsBalance": 250,
        "loyaltyProgramCurrencyName": "VroomKing",
        "promotionId": "",
        "stepName": "Credit Base Points For Transaction"
      }
    ],
    "executionStatus": "Success",
    "executionSummary": {
      "pointsSummary": [
        {
          "changeInEscrowPointsBalance": 0,
          "changeInPointsBalance": 250,
          "loyaltyProgramCurrencyName": "VroomKing",
          "loyaltyProgramCurrencyType": "NonQualifying"
        },
        {
          "changeInEscrowPointsBalance": 20,
          "changeInPointsBalance": 0,
          "loyaltyProgramCurrencyName": "EcoMiles",
          "loyaltyProgramCurrencyType": "Qualifying"
        }
      ],
      "vouchersIssuedCount": 0
    },
    "processRuleName": "Rule 1",
    "ruleSteps": [
      {
        "childSteps": [
          {
            "actionParameters": [
              {
                "name": "Notes",
                "operator": "Equals",
                "value": {
                  "expression": "Credit Brownie Points",
                  "expressionValue": {
                    "dataType": "Text",
                    "value": "Credit Brownie Points"
                  }
                }
              }
            ]
          }
        ]
      }
    ]
  }
]

```

```

        },
        "parameterValues": {}
    },
    {
        "name": "ProgramCurrencyName",
        "operator": "Equals",
        "value": {
            "expression": "EcoMiles",
            "expressionValue": {
                "dataType": "Text",
                "value": "EcoMiles"
            },
            "parameterValues": {}
        }
    },
    {
        "name": "PointsToCredit",
        "operator": "Equals",
        "value": {
            "expression": "20",
            "expressionValue": {
                "dataType": "Number",
                "value": 20
            },
            "parameterValues": {}
        }
    }
],
"actionType": "CreditPoints",
"sequenceNumber": 1,
"status": "Success",
"stepName": "Credit Brownie Points",
"stepType": "Action"
}
],
"filterCondition": "1",
"filterCriteria": [
    {
        "filterCriteriaResult": true,
        "operator": "Equals",
        "sequence": 1,
        "sourceFieldName": {
            "expression": "{!TransactionJournal.Product.Name}",
            "parameterValues": {
                "TransactionJournal.Product.Name": {
                    "dataType": "Text",
                    "value": "Brownie"
                }
            }
        }
    },
    {
        "value": {
            "expression": "Brownie",
            "expressionValue": {

```

```

        "dataType": "Text",
        "value": "Brownie"
      },
      "parameterValues": {}
    },
    "valueType": "FixedValue"
  }
],
"isConditionMet": true,
"sequenceNumber": 1,
"status": "Success",
"stepName": "Check Member Tier",
"stepType": "Condition"
},
{
  "actionParameters": [
    {
      "name": "ProgramCurrencyName",
      "operator": "Equals",
      "value": {
        "expression": "VroomKing",
        "expressionValue": {
          "dataType": "Text",
          "value": "VroomKing"
        },
        "parameterValues": {}
      }
    },
    {
      "name": "PointsToCredit",
      "operator": "Equals",
      "value": {
        "expression": "{!DoublePoints}",
        "expressionValue": {
          "dataType": "Number",
          "value": 250
        },
        "parameterValues": {
          "TransactionJournal.TransactionAmount": {
            "dataType": "Number",
            "value": 125
          }
        },
        "resolvedExpression": "{(!TransactionJournal.TransactionAmount}
* 2) "
      }
    }
  ],
  {
    "name": "Notes",
    "operator": "Equals",
    "value": {
      "expression": "Base Points For Transaction",
      "expressionValue": {
        "dataType": "Text",

```

```

        "value": "Base Points For Transaction"
      },
      "parameterValues": {}
    }
  ],
  "actionType": "CreditPoints",
  "sequenceNumber": 2,
  "status": "Success",
  "stepName": "Credit Base Points For Transaction",
  "stepType": "Action"
}
],
},
{
  "actionResultSummary": [],
  "errorMessage": "This rule didn't process the transaction journal because
the activity date of the transaction journal isn't between the start date and the end
date of the Holiday Season Promotion promotion.",
  "executionStatus": "Failed",
  "executionSummary": {
    "pointsSummary": [],
    "vouchersIssuedCount": 0
  },
  "processRuleName": "Rule 2",
  "promotionName": "Holiday Season Promotion",
  "ruleSteps": [
    {
      "actionParameters": [
        {
          "name": "VoucherExpirationDate",
          "operator": "Equals",
          "value": {
            "expression": "DATEVALUE("2023-01-31 00:00:00")",
            "expressionValue": {
              "dataType": "Date",
              "value": "DATEVALUE("2023-01-31 00:00:00")"
            },
            "parameterValues": {}
          },
        },
        {
          "name": "VoucherDiscountPercentage",
          "operator": "Equals",
          "value": {
            "expression": "10",
            "expressionValue": {
              "dataType": "Number",
              "value": "10"
            },
            "parameterValues": {}
          },
        },
      ],
    },
  ],
}

```



```

      "name": "VoucherCode",
      "operator": "Equals",
      "value": {
        "expression": "&quot;QWDBHYWIGFOIHW&quot;",
        "expressionValue": {
          "dataType": "Text",
          "value": "&quot;QWDBHYWIGFOIHW&quot;"
        },
        "parameterValues": {}
      },
    },
    {
      "name": "VoucherEffectiveDate",
      "operator": "Equals",
      "value": {
        "expression": "DATEVALUE(&quot;2022-12-24 00:00:00&quot;)",
        "expressionValue": {
          "dataType": "Date",
          "value": "DATEVALUE(&quot;2022-12-24 00:00:00&quot;)"
        },
        "parameterValues": {}
      },
    },
    {
      "name": "Notes",
      "operator": "Equals",
      "value": {
        "expression": "Holiday Season Voucher",
        "expressionValue": {
          "dataType": "Text",
          "value": "Holiday Season Voucher"
        },
        "parameterValues": {}
      },
    },
    {
      "name": "VoucherDefinitionName",
      "operator": "Equals",
      "value": {
        "expression": "TimeBasedVoucherDef",
        "expressionValue": {
          "dataType": "Text",
          "value": "TimeBasedVoucherDef"
        },
        "parameterValues": {}
      },
    },
  ],
  "actionType": "IssueVoucher",
  "errorMessage": "",
  "sequenceNumber": 1,
  "status": "CriteriaNotMet",
  "stepName": "Issue Holiday Season Voucher",
  "stepType": "Action"

```

```
    }
  ]
}
},
"status": true
}
```

Unlink Program Member Partner

Unlinks an existing association between a loyalty program member and a partner program. The API securely removes the linkage and updates related records to ensure data consistency across loyalty and partner systems.

Resource

```
/loyalty/programs/programName/members/membershipNumber/partners/partnerName/unlink
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v66.0/loyalty/programs/MarriottBonvoy/members/M0001/partners/AMEX/unlink
```

 **Note:** A record is uniquely identified by membershipNumber and partnerName. If either isn't provided, the API returns an error requesting additional filter criteria.

Available version

66.0

HTTP methods

POST

Request body for POST

JSON example

Sample request to unlink the program member and partner.

```
{
  "linkNumber": "00000029",
  "externalId": "U10001",
  "unlinkReasonType": "USER_REQUEST",
  "comments": "Member requested to unlink the partner account."
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
comments	String	Description for the unlink reason.	Optional	66.0
externalId	String	Partner membership number of the loyalty program member.	Optional	66.0
linkNumber	String	Unique identifier for the program member and partner linkage.	Optional	66.0

Name	Type	Description	Required or Optional	Available Version
unlinkReasonType	String	Reason to unlink a loyalty program member and a partner.	Optional	66.0

Response body for POST[Unlink Program Member Partner](#)

Loyalty Program Process

Run a loyalty process created from a loyalty program process template.

Out-of-the-box loyalty program process templates come with a default set of process parameters. You can configure new parameters or remove the default ones by simply cloning a template and running the configured parameters as a customized loyalty process using the loyalty program process API. For example, when using the *CreditPointsToMembers* process template to credit points to a loyalty member, you can configure the currency value so that the credited points can be redeemed in the currency you've specified.

The request and response parameters for the loyalty program process API vary depending on the input and output parameters configured for a loyalty program process in a Salesforce org.

Resource

```
/connect/loyalty/programs/${programName}/program-processes/${processName}
```

Resource examples

Loyalty Process API	Loyalty Process Template Name	Description
/connect/loyalty/programs/<i>DemoProgram</i>/program-processes/<i>CancelVoucher</i>	CancelVoucher	Cancel a voucher.
/connect/loyalty/programs/<i>DemoProgram</i>/program-processes/<i>CreditPointsToMembers</i>	CreditPointsToMembers	Credit pre-calculated points to a loyalty member on every transaction.
/connect/loyalty/programs/<i>DemoProgram</i>/program-processes/<i>DebitPointsFromMembers</i>	DebitPointsFromMembers	Redeem a loyalty member's pre-calculated points on a transaction.
/connect/loyalty/programs/<i>DemoProgram</i>/program-processes/<i>EnrollInPromotion</i>	EnrollInPromotion	Enroll a loyalty member to a promotion.
/connect/loyalty/programs/<i>DemoProgram</i>/program-processes/<i>GetMemberPromotions</i>	GetMemberPromotions	View eligible promotions of a loyalty member.
/connect/loyalty/programs/<i>DemoProgram</i>/program-processes/<i>IssueVoucher</i>	IssueVoucher	Issue a new voucher to a loyalty member.

Loyalty Process API	Loyalty Process Template Name	Description
/connect/loyalty/programs/ DemoProgram/program-processes/ UnenrollMember	UnenrollMember	Unenroll a member from a loyalty program.
/connect/loyalty/programs/ DemoProgram/program-processes/ UpdateMemberDetails	UpdateMemberDetails	Update a loyalty member profile.
/connect/loyalty/programs/ DemoProgram/program-processes/ UpdateTier	TierProcessing	Change the tier for a loyalty member.

Available version

56.0

Requires Chatter

No

HTTP methods

POST

Request parameters

Name	Type	Description	Required or Optional	Available Version
programName	String	Name of the loyalty program.	Required	56.0
processName	String	Name of the process created from a loyalty program process template.	Required	56.0

Request body for POST**Properties**

Name	Type	Description	Required or Optional	Available Version
processParameters	Loyalty Process Parameter Map	Input parameters for running the loyalty process.	Required	56.0

Response body for POST[Loyalty Process Output](#)[Cancel a Voucher](#)

Change the status of a voucher to canceled.

[Credit Points to Members](#)

Credit pre-calculated points to a loyalty member on every transaction.

[Debit Points from Members](#)

Redeem a loyalty member's pre-calculated points on a transaction.

[Enroll for Promotions](#)

Enroll a loyalty member to a promotion.

[Get Member Promotions](#)

View all eligible promotions of a loyalty member.

[Issue a Voucher](#)

Issue a new voucher to a loyalty member.

[Unenroll a Member](#)

Unenroll a member from a loyalty program.

[Opt Out from a Promotion](#)

Opt out a loyalty program member from an available promotion.

[Update Member Details](#)

Update personal and preference details of a member.

[Update Member Tier](#)

Upgrade or downgrade tier for a loyalty member.

Cancel a Voucher

Change the status of a voucher to canceled.

Example URL

```
https://yourInstance.salesforce.com/services/data/v66.0/connect/loyalty/programs/DemoProgram/program-processes/cancelVoucher
```

The request and response parameters vary depending on the input and output parameters configured for the *CancelVoucher* process type in a Salesforce org.

Sample JSON Requests



Note: Some of the API request fields contain field names that are interchangeable. For information on the list of interchangeable field names, see [Loyalty Management Connect API Interchangeable Request Fields](#).

Here's a sample request with a member ID and voucher ID

```
{
  "processParameters": [
    {
      "MemberId": "01Mxx00000000BJEAY",
      "VoucherId": "0kdxx000000000B245"
    }
  ]
}
```

Here's a sample request with a membership number and voucher code

 **Note:** If the voucher code isn't unique, provide either a voucher ID or a number.

```
{
  "processParameters": [
    {
      "MembershipNumber": "003",
      "VoucherCode": "15OFF"
    }
  ]
}
```

Sample JSON Response

```
{
  "message": null,
  "outputParameters": {
    "outputParameters": {
      "results": [
        {
          "VoucherId": "0kDRM0000004Ct22AE"
        }
      ]
    }
  },
  "simulationDetails": {
  },
  "status": true
}
```

SEE ALSO:

[Loyalty Management Help: Create a Process Using Cancel Voucher Process Template](#)

Credit Points to Members

Credit pre-calculated points to a loyalty member on every transaction.

Example URL

```
https://yourInstance.salesforce.com/services/data/v66.0/connect/loyalty/programs/DemoProgram/program-processes/creditPoints
```

The request and response parameters vary depending on the input and output parameters configured for the *CreditPointsToMembers* process type in a Salesforce org.

Sample JSON Requests

 **Note:** Some of the API request fields contain field names that are interchangeable. For information on the list of interchangeable field names, see [Loyalty Management Connect API Interchangeable Request Fields](#).

Here's an example with a member ID:

```
{
  "processParameters": [
    {
      "TransactionJournal": {
        "ActivityDate": "2023-08-27T00:00:00Z",
        "ExternalTransactionNumber": "1",
        "InvoiceDate": "2023-08-23",
        "JournalTypeName": "Accrual",
        "MemberId": "01MRM0000004G5h2AE",
        "TransactionAmount": "45.0",
        "CurrencyIsoCode": "USD",
        "Status": "Processed"
      },
      "PointsToCredit": 100.0
    }
  ]
}
```

Here's an example with a membership number:

```
{
  "processParameters": [
    {
      "TransactionJournal": {
        "ActivityDate": "2023-08-27T00:00:00Z",
        "ExternalTransactionNumber": "1",
        "InvoiceDate": "2023-08-23",
        "JournalTypeName": "Accrual",
        "MembershipNumber": "M1256",
        "TransactionAmount": "45.0",
        "CurrencyIsoCode": "USD",
        "Status": "Processed"
      },
      "PointsToCredit": 100.0
    }
  ]
}
```

Sample JSON Response

```
{
  "message": null,
  "outputParameters": {
    "outputParameters": {
      "results": [
        {
          "PointsCredited": 100.0
        }
      ]
    }
  },
  "simulationDetails": {
```

```
    },  
    "status":true  
  }  
}
```

SEE ALSO:

[Loyalty Management Help: Create a Process Using Credit Points to Member Process Template](#)

Debit Points from Members

Redeem a loyalty member's pre-calculated points on a transaction.

Example URL

```
https://yourInstance.salesforce.com/services/data/v66.0/connect/loyalty/programs/DemoProgram/program-processes/debit  
Points
```

The request and response parameters vary depending on the input and output parameters configured for the *DebitPointsFromMembers* process type in a Salesforce org.

Sample JSON Requests



Note: Some of the API request fields contain field names that are interchangeable. For information on the list of interchangeable field names, see [Loyalty Management Connect API Interchangeable Request Fields](#).

Here's an example with a member ID:

```
{  
  "processParameters": [  
    {  
      "TransactionJournal": {  
        "ActivityDate": "2023-08-27T00:00:00Z",  
        "ExternalTransactionNumber": "1",  
        "InvoiceDate": "2023-08-23",  
        "JournalTypeName": "Redemption",  
        "MemberId": "01MRM0000004G5h2AE",  
        "TransactionAmount": "45.0",  
        "CurrencyIsoCode": "USD",  
        "Status": "Processed"  
      },  
      "PointsToDebit": 200.0  
    }  
  ]  
}
```

Here's an example with a membership number:

```
{  
  "processParameters": [  
    {  
      "TransactionJournal": {  
        "ActivityDate": "2023-08-27T00:00:00Z",  
        "ExternalTransactionNumber": "1",  
        "InvoiceDate": "2023-08-23",  

```



```
        "JournalTypeName": "Redemption",
        "MembershipNumber": "M1256",
        "TransactionAmount": "45.0",
        "CurrencyIsoCode": "USD",
        "Status": "Processed"
    },
    "PointsToDebit": 200.0
}
]
```

Sample JSON Response

```
{
  "message": null,
  "outputParameters": {
    "outputParameters": {
      "results": [
        {
          "PointsDebited": 200.0
        }
      ]
    }
  },
  "simulationDetails": {
  },
  "status": true
}
```

SEE ALSO:

[Loyalty Management Help: Create a Process Using Debit Points from Member Process Template](#)

Enroll for Promotions

Enroll a loyalty member to a promotion.

Example URL

```
https://yourInstance.salesforce.com/services/data/v66.0/connect/loyalty/programs/DemoProgram/program-processes/EnrollPromotion
```

The request and response parameters vary depending on the input and output parameters configured for the *EnrollInPromotion* process type in a Salesforce org.

Sample JSON Requests



Note: Some of the API request fields contain field names that are interchangeable. For information on the list of interchangeable field names, see [Loyalty Management Connect API Interchangeable Request Fields](#).

Here's an example with a member ID and promotion ID:

```
{
  "processParameters":[
    {
      "MemberId":"01Mxx00000000BJEAY",
      "PromotionId":"0c8xx00000000BJEAY"
    }
  ]
}
```

Here's an example with a membership number and promotion name:

```
{
  "processParameters":[
    {
      "MembershipNumber":"M1234",
      "PromotionName":"Flat20"
    }
  ]
}
```

Sample JSON Response

```
{
  "message":null,
  "outputParameters":{
    "outputParameters":{
      "results":[
        {
          "MemberId":"01MRM00000004GBn2AM",
          "LoyaltyProgramMbrPromotionId":"1lpRM00000004CEmYAM"
        }
      ]
    }
  },
  "simulationDetails":{
  },
  "status":true
}
```

SEE ALSO:

[Loyalty Management Help: Create a Process Using Enroll in Promotion Process Template](#)

Get Member Promotions

View all eligible promotions of a loyalty member.

Example URL

```
https://yourInstance.salesforce.com/services/data/v66.0/connect/loyalty/programs/DemoProgram/program-processes/ViewPromotions
```

The request and response parameters vary depending on the input and output parameters configured for the *GetMemberPromotions* process type in a Salesforce org.

Sample JSON Requests



Note: Some of the API request fields contain field names that are interchangeable. For information on the list of interchangeable field names, see [Loyalty Management Connect API Interchangeable Request Fields](#).

Here's an example with a member ID:

```
{
  "processParameters": [
    {
      "MemberId": "01MRM0000004G5h2AE"
    }
  ]
}
```

Here's an example with a membership number:

```
{
  "processParameters": [
    {
      "MembershipNumber": "M1234"
    }
  ]
}
```

Sample JSON Response

```
{
  "message": null,
  "outputParameters": {
    "outputParameters": {
      "results": [
        {
          "promotionEnrollmentRqr": false,
          "loyaltyPromotionType": "STANDARD",
          "memberEligibilityCategory": "Eligible",
          "promotionName": "doc_promo_1",
          "promotionId": "0c8RM0000004DnXYAU",
          "startDate": "2022-09-01"
        }
      ]
    }
  },
  "simulationDetails": {
  },
}
```

```
{
  "status":true
}
```

SEE ALSO:

[Loyalty Management Help: Create a Process Using Get Member Promotions Process Template](#)

Issue a Voucher

Issue a new voucher to a loyalty member.

Example URL

```
https://yourInstance.salesforce.com/services/data/v66.0/connect/loyalty/programs/DemoProgram/program-processes/Issue
Voucher
```

The request and response parameters vary depending on the input and output parameters configured for the *IssueVoucher* process type in a Salesforce org.

Sample JSON Request



Note: Some of the API request fields contain field names that are interchangeable. For information on the list of interchangeable field names, see [Loyalty Management Connect API Interchangeable Request Fields](#).

Here's an example with a member ID:

```
{
  "processParameters": [
    {
      "MemberId": "01Mxx00000000BJEAY",
      "VoucherCode": "15OFF",
      "VoucherFaceValue": 15,
      "VoucherExpirationDate": "2022-07-13"
    }
  ]
}
```

Here's an example with a membership number:

```
{
  "processParameters": [
    {
      "MembershipNumber": "003",
      "VoucherCode": "15OFF",
      "VoucherDiscountPercentage": 20
    }
  ]
}
```

Sample JSON Response

```
{
  "message": null,
}
```

```

    "outputParameters":{
      "outputParameters":{
        "results":[
          {
          }
        ]
      }
    },
    "simulationDetails":{
    },
    "status":true
  }

```

SEE ALSO:

[Loyalty Management Help: Create a Process Using Issue Voucher Process Template](#)

Unenroll a Member

Unenroll a member from a loyalty program.

Example URL

```

https://yourInstance.salesforce.com/services/data/v66.0/connect/loyalty/programs/DemoProgram/program-processes/unenrollMember

```

The request and response parameters vary depending on the input and output parameters configured for the *UnenrollMember* process type in a Salesforce org.

Sample JSON Requests



Note: Some of the API request fields contain field names that are interchangeable. For information on the list of interchangeable field names, see [Loyalty Management Connect API Interchangeable Request Fields](#).

Here's an example with a member ID:

```

{
  "processParameters": [
    {
      "MemberId": "01MRM0000004G6f2AE"
    }
  ]
}

```

Here's an example with a membership number:

```

{
  "processParameters": [
    {
      "CurrentMembershipNumber": "007"
    }
  ]
}

```

```

    ]
  }

```

Sample JSON Response

```

{
  "message":null,
  "outputParameters":{
    "outputParameters":{
      "results":[
        {
          "MemberId":"01MRM0000004F152AE"
        }
      ]
    }
  },
  "simulationDetails":{

  },
  "status":true
}

```

SEE ALSO:

[Loyalty Management Help: Create a Process Using Unenroll Member Process Template](#)

Opt Out from a Promotion

Opt out a loyalty program member from an available promotion.

Example URL

```

https://yourInstance.salesforce.com/services/data/v66.0/connect/loyalty/programs/DemoProgram/program-processes/LeavePromotion

```

The request and response parameters vary depending on the input and output parameters configured for the *OptOutPromotion* process type in a Salesforce org.

Sample JSON Requests



Note: Some of the API request fields contain field names that are interchangeable. For information on the list of interchangeable field names, see [Loyalty Management Connect API Interchangeable Request Fields](#).

Here's an example with a membership number and promotion name:

```

{
  "processParameters":[
    {
      "MembershipNumber":"M1",
      "PromotionName":"Promo1"
    }
  ]
}

```

Here's an example with a member ID and promotion ID:

```
{
  "processParameters": [
    {
      "MemberId": "01MRM0000004F152AE",
      "PromotionId": "1lpxx00000000CvAAI"
    }
  ]
}
```

Sample JSON Response

```
{
  "message": null,
  "outputParameters": {
    "outputParameters": {
      "results": [
        {
          "LoyaltyProgramMbrPromotionId": "1lpxx00000000CvAAI"
        }
      ]
    }
  },
  "simulationDetails": {
  },
  "status": true
}
```

Update Member Details

Update personal and preference details of a member.

Example URL

```
https://yourInstance.salesforce.com/services/data/v66.0/connect/loyalty/programs/DemoProgram/program-processes/UpdateMemberDetails
```

The request and response parameters vary depending on the input and output parameters configured for the *UpdateMemberDetails* process type in a Salesforce org.

Sample JSON Request



Note: Some of the API request fields contain field names that are interchangeable. For information on the list of interchangeable field names, see [Loyalty Management Connect API Interchangeable Request Fields](#).

Here's an example with a member ID:

```
{
  "processParameters": [
    {
      "MemberId": "01MRM0000004G5h2AE",
    }
  ]
}
```

```
        "MembershipNumber": "M1234",
        "Email": "adam.smith@nto.com",
        "EnrollmentDate": "2023-05-28",
        "LastName": "Smith"
    }
}
```

Here's an example with a current membership number:

```
{
  "processParameters": [
    {
      "CurrentMembershipNumber": "M1234",
      "MembershipNumber": "M1234",
      "Email": "adam.smith@nto.com",
      "EnrollmentDate": "2023-05-28",
      "LastName": "Smith"
    }
  ]
}
```

Sample JSON Response

```
{
  "message": null,
  "outputParameters": {
    "outputParameters": {
      "results": [
        {
          "MemberId": "01MRM0000004F102AE",
          "ContactId": "003RM0000088CugYAE",
          "TransactionJournalId": null
        }
      ]
    }
  },
  "simulationDetails": {
  },
  "status": true
}
```

SEE ALSO:

[Loyalty Management Help: Create a Process Using the Update Member Details Process Template](#)

Update Member Tier

Upgrade or downgrade tier for a loyalty member.

Example URL

```
https://yourInstance.salesforce.com/services/data/v66.0/connect/loyalty/programs/DemoProgram/program-processes/updateTier
```

The request and response parameters vary depending on the input and output parameters configured for the *TierProcessing* process type in a Salesforce org.

Sample JSON Requests



Note: Some of the API request fields contain field names that are interchangeable. For information on the list of interchangeable field names, see [Loyalty Management Connect API Interchangeable Request Fields](#).

Here's an example with a member ID:

```
{
  "processParameters": [
    {
      "MemberId": "01MRM0000004GBs2AM",
      "NewTier": "Gold",
      "ReasonForChange": "update tier"
    }
  ]
}
```

Here's an example with a membership number:

```
{
  "processParameters": [
    {
      "MembershipNumber": "M2_Chloe_Sun",
      "NewTier": "Gold",
      "ReasonForChange": "update tier"
    }
  ]
}
```

Sample JSON Response

```
{
  "message": null,
  "outputParameters": {
    "results": [
      {
        "CurrentTier": "Gold",
        "TierExpirationDate": "2023-10-31",
        "PreviousTier": "Silver",
        "EffectiveDate": "2023-09-07"
      }
    ]
  },
  "simulationDetails": {
  },
}
```

```
"status":true
}
```

SEE ALSO:

[Loyalty Management Help: Create a Process Using the Change Member Tier Process Template](#)

Gamification

Who doesn't like to play a game? Engage your customers more by gamifying customer activities, such as placing an order, or events, such as birthdays and anniversaries.

Assign eligible games to a customer, let the customer play the game, and then win a reward on an external system using the gamification REST APIs.

Games

Get a list of eligible, expired, and played games, such as Spin the Wheel, or Scratch card, for a loyalty program member, an account, or a contact.

Game Reward

Get the details of the reward that's provided to a participant after the participant plays a Spin the Wheel, or Scratch Card game.

Games

Get a list of eligible, expired, and played games, such as Spin the Wheel, or Scratch card, for a loyalty program member, an account, or a contact.

Resource

```
/game/participant/{participantId}/games
```

Resource Example

```
https://yourInstance.salesforce.com/services/data/v66.0/game/participant/01MRM0000004FHX2A2/games
https://yourInstance.salesforce.com/services/data/v66.0/game/participant/01MRM0000004FHX2A2/games?definitionId=3GDX0000004CA6GM
https://yourInstance.salesforce.com/services/data/v66.0/game/participant/01MRM0000004FHX2A2/games?gameParticipantRewardId=3PSEB0000000njJ4PQ
```

Available version

60.0

HTTP methods

GET

Query parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
definitionId	String	ID of the game definition associated with the game whose details are to be fetched.	Optional	60.0

Parameter Name	Type	Description	Required or Optional	Available Version
gameParticipantRewardId	String	ID of the participant's game reward associated with the game whose details are to be fetched.	Optional	60.0

Response body for GET[Games List](#)

Game Reward

Get the details of the reward that's provided to a participant after the participant plays a Spin the Wheel, or Scratch Card game.

Resource

```
/game/gameparticipantreward/{gameParticipantRewardId}/game-reward
```

Resource Example

```
https://yourInstance.salesforce.com/services/data/v66.0/game/gameparticipantreward/3PRSB0000000hjJ4AQ/game-reward
```

Available version

60.0

HTTP methods

GET

Response body for GET[Game Result](#)

Promotion

The promotion APIs allow you to set eligibility rules and limits for a promotion, choose a promotion template, and create a promotion. You can also get the details of a specific promotion, and view a list of eligible promotions for a customer's cart.

[Eligible Promotions List \(POST\)](#)

Get eligible promotions and their associated rewards for a cart and its line items.

[Promotion Configuration \(GET\)](#)

View configuration of a promotion, including its definition, eligibility criteria, rules, limits, and related rule library version. The Promotion Configuration API enables developers and system integrators in your company to view the configuration of active promotions in your Salesforce org and export the configuration to external systems such as offline point-of-sale (POS) systems.

[Promotion Rule Configuration \(GET, POST\)](#)

Create the rule configuration or fetch the rule configuration details for a given promotion ID. This API defines how a promotion works by combining eligibility criteria (what customers must do), rewards logic (what customers receive), target audience (who receives the reward), and template driven behavior. Each promotion can contain multiple rules, ordered by priority.

[Clone Promotion \(POST\)](#)

Clone an existing promotion along with its associated records

[Coupon Usage Increase \(POST\)](#)

Increase the usage count of a list of coupons.

[Coupon Usage Decrease \(POST\)](#)

Decrease the usage count of a list of coupons.

[Promotions Creation \(POST, PUT\)](#)

Create a promotion based on the provided promotion limits and eligibility conditions.

[Promotion Details \(GET, PUT\)](#)

Retrieve the details of a promotion. Update a promotion record.

[Promotion Evaluation and Execution \(POST\)](#)

Evaluate cart details and return the adjusted price of cart and line items after applying eligible promotions based on the selected promotion evaluation and execution method.

[Promotion Recommendations \(POST\)](#)

Retrieve the list of promotions recommended for a customer or a loyalty program member based on the product and personalization point.

[Promotion Reward Application \(POST\)](#)

Create and process transaction journals for customer orders, and apply the relevant promotions for the orders.

Eligible Promotions List (POST)

Get eligible promotions and their associated rewards for a cart and its line items.

Special Access Rules

You must have the Context Definitions and Context Rule Library settings enabled in your Salesforce org.

Resource

```
/global-promotions-management/eligible-promotions
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v66.0/global-promotions-management/eligible-promotions
```

Available version

60.0

HTTP methods

POST

Request body for POST

Root XML tag

CartInputMapRepresentation

JSON examples

Here's an example of the details of a customer's cart whose eligible promotions are to be fetched:



Note: Provide `contactId` to retrieve campaigns related to a Person Account.

```
{
  "cart": {
    "cartDetails": [
      {
        "activityStartDate": "2023-06-14T12:45:19Z",
```

```

        "contactId/loyaltyProgramMemberId/accountId": "003xx000004WjzAAAS",
        "transactionAmount": 499,
        "currencyISOCODE": "USD",
        "engagementChannelId": "0eFxx0000000008EAA",
        "sourceLocationId/sourceLocation": "0YQxx0000000001GAA",
        "cartLineDetails": [
            {
                "cartLineProduct": "Burger",
                "cartLineProductCode": "testProductCode",
                "cartLineProductStockKeepingUnit": "testSKU",
                "cartLineProductCatalog": "Food",
                "cartLineItemQuantity": 1,
                "cartLineItemAmount": 100,
                "cartLineProductCategory": "test"
            }
        ]
    }
}

```

Here's an example of the details of a loyalty program member's cart whose eligible loyalty program promotions are to be fetched:

```

{
  "cart": {
    "cartDetails": [
      {
        "activityStartDate": "2023-06-14T12:45:19Z",
        "contactId/loyaltyProgramMemberId/accountId": "003xx000004WjzAAAS",
        "currencyISOCODE": "INR",
        "transactionAmount": 499,
        "membershipNumber": "Member-005",
        "cartLineDetails": [
          {
            "cartLineProductId": "01txx0000006iTsAAI",
            "cartLineProductCode": "AnyProductCode",
            "cartLineProductStockKeepingUnit": "AnySKU",
            "cartLineItemQuantity": 2,
            "cartLineItemAmount": 99,
            "cartLineProductCatalogId": "0ZSSB0000002d8X4AQ"
          },
          {
            "cartLineProduct": "Coke",
            "cartLineProductCode": "AnyProductCode",
            "cartLineProductStockKeepingUnit": "AnySKU",
            "cartLineItemQuantity": 10,
            "cartLineItemAmount": 400,
            "cartLineProductCatalog": "Beverages"
          }
        ]
      }
    ]
  }
}

```

Here’s an example to fetch eligible loyalty program promotions for accounts:

```
{
  "cart": {
    "cartDetails": [
      {
        "activityStartDate": "2024-10-16T22:45:19Z",
        "accountId": "001xx000003H3NnAAK",
        "transactionAmount": 40,
        "currencyISOCode": "USD",
        "cartLineDetails": [
          {
            "cartLineProductCatalog": "Beverages",
            "cartLineProductCategory": "Coffee Creations",
            "cartLineProduct": "Caramel Latte",
            "cartLineItemQuantity": 5
          }
        ]
      }
    ]
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
cart	Map<String, Object>	Fields configured in the Cart and CartLine nodes of the <i>CartDetailGetEligiblePromotion</i> context service definition in your org. These fields represent the details of a cart and its line items whose eligible promotions are to be fetched. To get eligible promotions associated with a loyalty program member’s cart, the required parameters are: - membershipNumber or loyaltyProgramMemberId - currencyISOCode For other eligible promotions, the required parameters are: - contactId, contactLastName, or contactEmail - currencyISOCode Along with the required parameters, provide specific details of a cart, such as the total transaction amount, and the cart	Required	60.0

Name	Type	Description	Required or Optional	Available Version
		line item details, such as the line item quantity and amount, the product ID or name, the product catalog ID or name, the product code, and the product stock keeping unit.		
contextInstanceId	String	ID of the context instance.	Optional	65.0

Request parameters for POST

Parameter Name	Type	Description	Required or Optional	Available Version
ruleLibraryApiName	String	API Name of an active Rule Library with the usage type as Global Promotion Management. This parameter allows the API to check the eligibility of the cart against the Promotions associated with the specified Rule Library. This parameter is required if there are multiple active rule libraries configured with the Usage Type as Global Promotion Management.	Optional	59.0

Response body for POST

[Eligible Promotions List](#)

Promotion Configuration (GET)

View configuration of a promotion, including its definition, eligibility criteria, rules, limits, and related rule library version. The Promotion Configuration API enables developers and system integrators in your company to view the configuration of active promotions in your Salesforce org and export the configuration to external systems such as offline point-of-sale (POS) systems.

Special Access Rules

The Promotion Configuration API is only available in orgs where Global Promotions Management is available.

Resource

```
/global-promotions-management/promotions/promotionId/configuration
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v66.0/global-promotions-management/promotions/promotionId/configuration
```

Available version

61.0

HTTP methods

GET

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
promotionId	String	ID of the active promotion you want to view. The ID must belong to a promotion whose rules are configured using a non-Loyalty promotion template.	Required	61.0
~~relatedObjectAndFieldApiNames~~ relatedObjectAndFieldApiNames	String[]	The list of related object fields whose details you want to view for the promotion. To view the details of a field, specify the API name of the object that the field belongs to and the API name of the field, separated by a period. relatedObjectAndFieldApiNames can pass only one field per object.	Optional	61.0

Response body for GET[Promotion Configuration](#) on page 563**Promotion Rule Configuration (GET, POST)**

Create the rule configuration or fetch the rule configuration details for a given promotion ID. This API defines how a promotion works by combining eligibility criteria (what customers must do), rewards logic (what customers receive), target audience (who receives the reward), and template driven behavior. Each promotion can contain multiple rules, ordered by priority.

Resource

```
/global-promotions-management/promotions/promotionId/rule-configuration
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v66.0/global-promotions-management/promotions/promotionId/rule-configuration
```

Available version

60.0

HTTP methods

GET POST

Response body for GET[Template List Configuration](#)

Request body for POST**Properties**

Name	Type	Description	Required or Optional	Available Version
event Configuration	Map<String, Object>[]	List of criteria that decides when customers are eligible for the rewards of the promotion rule.	Required	60.0
journalSubType	String	ID of the journal sub type.	Required when journalType is used	60.0
journalSub TypeName	String	Name of the journal sub type.	Required when journalType is used	64.0
journalType	String	Name of the journal sub type.	Required for loyalty accrual or redemption templates	60.0
priority	Integer	Priority number of the rule in the promotion.	Required	60.0
promotionLimit	Integer	Milestone limit of the promotion.	Optional	63.0
reward Configuration	Map<String, Object>[]	List of rewards that members get when their activity meets the rule's eligibility criteria.	Required	60.0
ruleName	String	Name of the promotion rule.	Required	60.0
templateName	String	Name of the promotion template that's used in a promotion rule.	Required	60.0

Response body for POST[Unified Promotion Templates](#)**Clone Promotion (POST)**

Clone an existing promotion along with its associated records

Resource`/promotions/clone`**Available version**

61.0

HTTP methods

POST

JSON example

```
{
  "sourcePromotionId": "0c8xx00000000WHAAY",
  "name": "ups clone 4",
  "startDateTime": "2024-01-08T00:00:00.000Z",
  "endDateTime": "2024-01-09T00:00:00.000Z",
  "promotionCode": "sdfgy543wsd",
  "doesReuseEngagementAttributes": "true",
  "additionalFieldValues": {
    "additionalFieldValuesMap": {
      "Description": "test ups clone"
    }
  },
  "relatedRecords": {
    "relatedRecordsList": [
      {
        "relatedObjectApiName": "PromotionChannel",
        "relatedObjectPromotionFieldName": "PromotionId"
      },
      {
        "relatedObjectApiName": "PromotionLoyaltyPtnrProdt",
        "relatedObjectPromotionFieldName": "PromotionId"
      },
      {
        "relatedObjectApiName": "PromotionMarketSegment",
        "relatedObjectPromotionFieldName": "PromotionId"
      },
      {
        "relatedObjectApiName": "PromotionProduct",
        "relatedObjectPromotionFieldName": "PromotionId"
      },
      {
        "relatedObjectApiName": "PromotionProductCategory",
        "relatedObjectPromotionFieldName": "PromotionId"
      },
      {
        "relatedObjectApiName": "PromotionLimit",
        "relatedObjectPromotionFieldName": "PromotionId"
      }
    ]
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalFieldValues	Map<String, String>	Details of the promotion's additional fields.	Optional	61.0
doesReuseEngagementAttributes	Boolean	Indicates whether the new promotion uses the engagement attributes created for the source promotion to track customers'	Optional	61.0

Name	Type	Description	Required or Optional	Available Version
		promotion usage (true) or (not). Engagement reuse can result in shared limits.		
endTime	String	End date and time of the new promotion.	Required	61.0
name	String	Name of the new promotion.	Required	61.0
promotionCode	String	Code of the new promotion.	Optional	61.0
relatedRecords	Related Promotion Records Input [] on page 492	Related record for the source promotion that have to be created for the new promotion.	Optional	61.0
sourcePromotionId	String	ID of the promotion that's cloned.	Required	61.0
startTime	String	Start date and time of the new promotion.	Required	61.0

Response body for POST
[Clone Promotion](#) on page 525

Coupon Usage Increase (POST)

Increase the usage count of a list of coupons.

Resource

```
/global-promotions-management/coupons/usage-increase
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v66.0/global-promotions-management/coupons/usage-increase
```

Available version
63.0

HTTP methods
POST

Request body for POST

JSON example

```
{
  "couponCodeList": [
    "QWERTY",
    "ABCD"
  ],
  "activityDateTime": "2024-10-05T12:45:19Z",
  "externalTransactionNumber": "01Vxx000000006T",
  "customerId": "01Mxx0000000001"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
couponCodeList	List<String>	The codes of the list of coupons for which the usage must be increased or decreased.	Required	62.0
activityDateTime	DateTime	The date and time at which the customer redeemed the coupon.	Required	62.0
~~externalTransactionId~~	String	The externally generated unique identifier of cart for which the customer redeemed the coupon. This ID must be the same in both the decrease and increase usage APIs so that the API can correctly track coupon usage for the originating transaction.	Required	62.0
customerId	String	The ID of either the contact, the account, or the loyalty program member who redeemed the coupon.	Required	62.0

Response body for POST

Coupon Usage

Coupon Usage Decrease (POST)

Decrease the usage count of a list of coupons.

Resource

```
/global-promotions-management/coupons/usage-decrease
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v66.0/global-promotions-management/coupons/usage-decrease
```

Available version

63.0

HTTP methods

POST

Request body for POST

JSON example

```
{
  "couponCodeList": [
    "QWERTY",
    "ABCD"
  ],
  "activityDateTime": "2024-10-05T12:45:19Z",
}
```

```
{
  "externalTransactionNumber": "01Vxx000000006T",
  "customerId": "01Mxx0000000001"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
couponCodeList	List<String>	The codes of the list of coupons for which the usage must be increased or decreased.	Required	62.0
activityDateTime	DateTime	The date and time at which the customer redeemed the coupon.	Required	62.0
externalTransactionNumber	String	The externally generated unique identifier of cart for which the customer redeemed the coupon. This ID must be the same in both the decrease and increase usage APIs so that the API can correctly track coupon usage for the originating transaction.	Required	62.0
customerId	String	The ID of either the contact, the account, or the loyalty program member who redeemed the coupon.	Required	62.0

Response body for POST

[Coupon Usage](#)

Promotions Creation (POST, PUT)

Create a promotion based on the provided promotion limits and eligibility conditions.

Resource

```
/global-promotions-management/promotions
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v66.0/global-promotions-management/promotions
```

Available version

64.0

HTTP methods

POST PUT

Response body for GET

[Promotion](#)

Request body for POST**Properties**

Name	Type	Description	Required or Optional	Available Version
promotionDetails	Unified Promotions Input[]	Configuration of the promotion	Required	64.0
rules	Unified Promotion Template Input[]	List of rules that are configured for a promotion.	Required	64.0

Response body for POST[Promotion](#)**Request body for PUT****Properties**

Name	Type	Description	Required or Optional	Available Version
promotionDetails	Unified Promotions Input[]	Configuration of the promotion	Required	64.0
rules	Unified Promotion Template Input[]	List of rules that are configured for a promotion.	Required	64.0

Response body for PUT[Promotion](#)**Promotion Details (GET, PUT)**

Retrieve the details of a promotion. Update a promotion record.



Note: To update an existing promotion, you must first deactivate the promotion in the org.

Resource

```
/global-promotions-management/promotions/promotionId
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v66.0/global-promotions-management/promotions/promotionId
```

Available version

60.0

HTTP methods

GET, PUT

Request body for PUT**Root XML tag**

unifiedPromotionsInputRepresentation

JSON example for POST and PUT

```
{
  "promotionDetails": {
    "name": "Sample Promotion 6",
    "displayName": "Sample Promotion",
    "priorityNumber": 1,
    "description": "Sample promotion description",
    "promotionCode": "XYZ",
    "startDateTime": "2023-06-30T00:00:00.000Z",
    "endDateTime": "2023-12-30T00:00:00.000Z",
    "objective": "To increase sales",
    "isAutomatic": true,
    "loyaltyProgram": {
      "name": "NTO"
    },
  },
  "additionalFieldValues": {
    "attributes": {
      "isUnifiedPromotion__c": true
    }
  },
  "promotionEligibility": {
    "productCatalog": {
      "id": "0ZSxx000000001dGAA"
    },
  },
  "eligibleProductCategories": [
    {
      "id": "0ZGxx0000000001GAA"
    }
  ],
  "eligibleProducts": [
    {
      "name": "someProduct"
    }
  ],
  "eligibleChannels": [
    {
      "channelType": "****",
      "channel": {
        "id": "****",
        "name": "****"
      },
      "store": {
        "id": "****",
        "name": "****"
      },
      "retailLocationGroup": {
        "name": "****"
      }
    }
  ]
}
```

```

    ],
    "eligibleCampaign": {
      "name": "****"
    },
    "eligibleSegments": [
      {
        "name": "someSegment"
      }
    ],
    "eligibleCustomerEvents": {
      "event": "MemberBirthday",
      "eventPeriodUnit": ""
    },
    "eligibleEnrollmentPeriod": {
      "isEnrollmentRequired": true,
      "enrollmentStartDate": "",
      "enrollmentEndDate": ""
    },
    "eligibleLoyaltyPartnerProducts": [
      {
        "name": "",
        "id": ""
      }
    ],
    "eligibleLoyaltyTiers": [
      {
        "tierGroup": {
          "name": "",
          "id": ""
        },
        "tier": {
          "name": "",
          "id": ""
        }
      }
    ],
    "eligibleCoupons": [
      {
        "couponCode": "",
        "status": "",
        "description": "",
        "startDateTime": "",
        "endDateTime": "",
        "redemptionLimitOverall": "",
        "redemptionLimitPerCustomer": ""
      }
    ]
  },
  "promotionLimits": {
    "perLineItemLimit": 10,
    "pointsPerCustomerLimit": 5,
    "vouchersPerCustomerLimit": 5,
    "perCartLimit": 5,
    "usePerCustomerLimit": 5,

```



```

        "totalUseLimit": 25,
        "pointsLimit": 1000,
        "vouchersLimit": 100,
        "liabilityLimit": 10000,
        "loyaltyProgramCurrency": {
            "id": "01cxx0000000001AAA"
        }
    }
}
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalFieldValues	Map<String, String>	Custom, and additional field values that provide more details about the promotion.	Optional	60.0
businessUnit	Name And Id Input []	ID or name of the Marketing Cloud business unit associated with the promotion. To specify a business unit, you must have the Marketing Cloud license.	Optional	60.0
currencyIsoCode	String	ISO code of the active currency associated with the promotion. Before specifying the currency ISO code, make sure that your org supports multiple currencies.	Optional	60.0
description	String	Description of the promotion.	Optional	60.0
displayName	String	Display name of the promotion.	Optional	60.0
endDateTime	String	Date and time until which the promotion can be applied for customers.	Optional	60.0
isAutomatic	Boolean	Indicates whether the promotion is automatically applied for eligible carts (true) or not (false).	Optional	63.0
isEmailActivated	Boolean	Indicates whether the promotion's email is activated (<code>true</code>) or not (<code>false</code>). If set to true, an email template is created for the promotion's email in the org. The default value is <code>false</code> .	Optional	Small, 60.0

Name	Type	Description	Required or Optional	Available Version
loyaltyProgram	Name And Id Input []	ID or name of the active loyalty program the promotion applies to. You can't change the loyalty program of an existing promotion.	Optional	60.0
name	String	Unique name of the promotion.	Required	60.0
objective	String	Objective of the promotion.	Optional	60.0
priority Number	Integer	Number that determines the priority of the promotion in a list of eligible promotions. Priority number must be greater than zero.	Optional	60.0
promotionCode	String	Unique code of the promotion.	Optional	60.0
promotion Eligibility	Promotion Eligibility Input	Eligibility criteria of the promotion.	Optional	60.0
promotionLimits	Promotion Limit Input	Limits of the promotion, including the number of times the promotion can be applied for customers. A promotion limit must be greater than zero.	Optional	60.0
ruleLibrary	String	Rule library associated with the promotion.	Optional	62.0
startDateTime	String	Date and time from which the promotion can be applied for customers.	Required	60.0
termsAndConditions	String	Terms and conditions which are applied to the promotion.	Optional	63.0

Promotion Evaluation and Execution (POST)

Evaluate cart details and return the adjusted price of cart and line items after applying eligible promotions based on the selected promotion evaluation and execution method.

Resource

```
/global-promotions-management/promotion-execution
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v66.0/global-promotions-management/promotion-execution
```

Available version

65.0

HTTP methods

POST

Request parameters for POST

Parameter Name	Type	Description	Required or Optional	Available Version
rule LibraryApiName	String	API Name of an active Rule Library with the usage type as Global Promotion Management. This parameter allows the API to check the eligibility of the cart against the promotions associated with the specified rule library. This parameter is required if there are multiple active rule libraries configured with the Usage Type as Global Promotion Management.	Optional	65.0

Request body for POST**JSON example**

```
{
  "cart": {
    "cartDetails": [
      {
        "cartHeaderId__std": "01ddf0000003kfMAAW",
        "activityStartDate": "2023-06-14T12:45:19Z",
        "contactId/contactEmail/contactLastName/loyaltyProgramMemberId/membershipNumber/accountId":
        "003xx000004WjzAAAS",
        "transactionAmount": 110,
        "currencyISOCODE": "USD",
        "engagementChannelId": "0eFxx00000000008EAA",
        "sourceLocationId/sourceLocation": "0YQxx00000000001GAA",
        "cartPromotionAdditionalInformation__std": [
          {
            "cartCouponCode__std": "CART5"
          }
        ],
        "cartLineDetails": [
          {
            "cartLineProductId": "01txx0000006kfMAAQ",
            "cartLineItemQuantity": 2,
            "cartLineItemAmount": 100,
            "cartLineItemId": "abcd1",
            "cartLineNetUnitPrice__std": 50,
            "cartLinePromotionAdditionalInformation__std": [
              {
                "cartLineCouponCode__std": "PIZZA5"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```
    ],
  },
  {
    "cartLineProductId": "01txx0000006kfNAAQ",
    "cartLineItemQuantity": 2,
    "cartLineItemAmount": 10,
    "cartLineItemId": "abcd2",
    "cartLineNetUnitPrice__std": 5
  }
]
}
]
```

Properties

Name	Type	Description	Required or Optional	Available Version
cart	Map<String, Object>	Details of a cart and its line items used to evaluate and execute promotions. The payload must include <i>cartHeaderId__std</i>, which uniquely identifies the cart and is required for promotion evaluation and execution. Specify the applicable <i>currencyISOCode</i> for the transaction. To evaluate or execute coupon-based promotions, include the <i>couponCode</i>, which must match an active coupon or promotion configured in the system. Along with the required parameters, provide cart-level details such as the total transaction amount, and cart line item details such as line item quantity and amount, <i>cartLineNetUnitPrice__std</i> (required for evaluation of line-level and cross-line promotions), and product ID.	Required	65.0

Response body for POST

[Cart](#)

Promotion Recommendations (POST)

Retrieve the list of promotions recommended for a customer or a loyalty program member based on the product and personalization point.

Resource

```
/promotions/recommendation/product-promotion
```

Resource Example

```
https://yourInstance.salesforce.com/services/data/v66.0/promotions/recommendation/product-promotion?count=3&promotionsPerProductLimit=4
```

Available version

64.0

HTTP methods

POST

Query Parameters

Name	Type	Description	Required or Optional	Available Version
count	Integer	The maximum number of promotion recommendations to retrieve.	Optional	64.0
promotionsPerProductLimit	Integer	The maximum number of promotions that can be recommended for a product. (Note: The count supersedes the limit, suggesting that in case we put a limit of 3 for each product but it doesn't fullfill the count, it will display more promotions which depends on the count parameter)	Optional	64.0

Request body for POST

```
{
  "personalizationPoint": "string",
  "dataspace": "string",
  "membershipNumber": "string",
  "loyaltyProgramId": "string",
  "loyaltyProgramMemberId": "string",
  "individualId": "string",
  "eligibleChannel": {
    "channel": {
      "id": "string",
      "name": "string"
    },
    "retailStore": {
      "id": "string",
      "name": "string"
    }
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
dataspace	String	The data space that defines the boundary of a personalization context. This API currently only works with your Data Cloud instance's default data space.	Required	64.0
eligible Channel	Channel Input on page 462	The details of a channel for which to retrieve eligible promotions.	Optional	64.0
individualId	String	The ID of the of the individual for whom to retrieve recommended promotions. Optional if membership number or loyalty program member ID is provided.	Optional if membership number or loyalty program member ID is provided.	64.0
loyalty ProgramId	String	The loyalty program that the loyalty program member belongs to. The loyalty program ID is required when membership number is specified.	Required when membership number is specified.	64.0
loyalty Program MemberId	String	The ID of the of the loyalty program member for whom to retrieve recommended promotions. Optional if membership number or contact ID is provided.	Optional if membership number or contact ID is provided.	64.0
membership Number	String	The membership number of the loyalty program member for whom to retrieve recommended promotions.	Optional	64.0
personalization Point	String	The API name of the personalization point. A personalization point is a connection between data, profile data, personalization type, and schema, used to deliver personalized content to customers.	Required	64.0

Response body for POST

[Promotion Recommendations](#) on page 580

Promotion Reward Application (POST)

Create and process transaction journals for customer orders, and apply the relevant promotions for the orders.

Resource

```
/global-promotions-management/promotion-reward
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v66.0/global-promotions-management/promotion-reward
```

Available version

65.0

HTTP methods

POST

Request body for POST**JSON example**

```
{
  "shouldCheckCouponUsageLimit": true,
  "journalCreationMode": "CreateTransactionJournal",
  "transactionJournalList": [
    {
      "ActivityDate": "2025-10-08T12:45:19Z",
      "JournalTypeId": "01ET10000004CQp",
      "ReferredPartyId": "003SB000000U84F0YAJ",
      "Quantity": "1",
      "TransactionAmount": "5000",
      "CurrencyIsoCode": "USD",
      "Status": "Processed",
      "appliedPromotions": [
        {
          "promotionId": "0c8SB000000HASnYAO",
          "promotionCode": "GET100OFF",
          "couponCode": "COUPON100",
          "rewards": [
            {
              "rewardType": "ProvideDiscount",
              "discountAmount": 15
            },
            {
              "rewardType": "IssueVoucher",
              "voucherDefinitionName": "$100 off",
              "voucherCode": "SALE-4F8G-RT2P",
              "voucherEffectiveDate": "2025-10-09",
              "voucherExpirationDate": "2025-10-10",
              "notes": "discount voucher"
            },
            {
              "rewardType": "AssignGame",
              "gameDefinitionName": "SpintheWheelGame",
              "gameExpirationDate": "2025-10-10"
            }
          ]
        }
      ]
    }
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
<code>journalCreationMode</code>	String	Specifies whether to create a transaction journal record and, if so, where to store the data.	Optional	66.0
<code>shouldCheckCouponUsageLimit</code>	Boolean	Indicates whether the API must execute the Coupon Usage Increase API to verify the coupon usage count and increase the redemption usage count of the specified coupon code (<code>true</code>) or not (<code>false</code>).	Optional	65.0
<code>transactionJournalList</code>	List<Map<String, Object>>	The list of transaction journals to create.	Required	65.0

Response body for POST[Promotion Reward Application](#)

Request Bodies

Here's a list of Loyalty Management API request bodies.

[Cart Input](#)

The details of a cart and its line items that's used to evaluate and execute promotions.

[Cart Input Map](#)

Input representation of the summary of the cart whose eligible promotions are to be fetched.

[Channel Input](#)

Input representation details of a channel for which to retrieve eligible promotions.

[Clone Promotion Input](#)

Input representation that decides which existing promotion is cloned and the attributes of the new promotion that's created.

[Club Member Enrollment Input](#)

The input representation to enroll a club member.

[Club Membership Renewal Input](#)

The input representation for club membership renewal.

[Corporate Member Enrollment Input](#)

Input representation for creating a business account and a Corporate type Loyalty Program Member record for the member.

[Coupon Usage Input](#)

Input representation of coupon usage.

[Eligible Clubs Input](#)

The club tier groups and tiers whose members are eligible for the promotion.

[Eligible Coupon Input](#)

The details of an eligible coupon for the promotion.

[Eligible Enrollment Period Input](#)

Input representation of the enrollment eligibility details for the promotion.

[Eligible Loyalty Tiers Input](#)

Input representation of the loyalty program tier whose members are eligible for the promotion.

[Events Detail Input](#)

Input representation details of the customer event that is eligible for the promotion.

[Individual Member Enrollment Input](#)

Input representation for creating a person account and an Individual type Loyalty Program Member record for the member.

[Link Program Member Partner Input](#)

Input details to link a program member and partner.

[Loyalty Engine Input](#)

Input representation of the Loyalty Engine Runtime API.

[Loyalty Engine Input Map](#)

Input representation of the Transaction Journal record execution request.

[Loyalty Engine Run Settings Input](#)

Input representation of the loyalty engine settings to run simulation on the transaction journals.

[Loyalty Account Input](#)

Input representation for creating a business account for the member.

[Loyalty Contact Input](#)

Input representation of the loyalty program member's person account.

[Loyalty Process Input](#)

Input representation of the details of the loyalty program process action.

[Loyalty Process Input Options](#)

Input representation of the execution options for the loyalty process.

[Loyalty Process Parameter Map Input](#)

Input representation of the input parameters for the loyalty process.

[Loyalty View Transaction Journal History Input](#)

Input representation for getting the transaction journal records of a loyalty member.

[Name and ID Input](#)

Input representation of an object's name and ID.

[Product Details](#)

An input representation that contains specific details related to the product.

[Promotion Eligible Channel Input](#)

Input representation details of the channel eligible for the promotion.

[Promotion Eligibility Input](#)

Input representation details of the promotion eligibility criteria.

[Promotion Input](#)

Input representation details of a promotion.

[Promotion Limit Input](#)

Input representation details of the promotion limits.

[Promotion Recommendation Input](#)

The input representation to retrieve the recommended promotion.

[Redeem Voucher Input](#)

Input representation for voucher redemption.

[Related Promotion Records Input](#)

Input representation that decides which related records and fields of the existing promotion are cloned and associated with the new promotion.

[Unified Promotions Input](#)

Input representation of the promotion configuration to be created, or updated.

[Unified Promotion Template Input](#)

Input representation of the unified promotion template.

[Unlink Program Member Partner Input](#)

Input details to unlink a program member and partner.

[Promotion Reward Application Input](#)

The details required to create transaction journals. The request includes a list of transaction journals, which are to be created. It includes the list of promotions applied along with the rewards. If any of the rewards aren't applied, then everything is rolled back. No partial rewards are applied.

Cart Input

The details of a cart and its line items that's used to evaluate and execute promotions.

JSON example

```
{
  "cart": {
    "cartDetails": [
      {
        "cartHeaderId__std": "01ddf0000003kfMAAW",
        "activityStartDate": "2023-06-14T12:45:19Z",
        "contactId/contactEmail/contactLastName/loyaltyProgramMemberId/membershipNumber/accountId":
        "003xx000004WjzAAAS",
        "transactionAmount": 110,
        "currencyISOCODE": "USD",
        "engagementChannelId": "0eFxx00000000008EAA",
        "sourceLocationId/sourceLocation": "0YQxx00000000001GAA",
        "cartPromotionAdditionalInformation__std": [
          {
            "cartCouponCode__std": "CART5"
          }
        ],
        "cartLineDetails": [
          {
            "cartLineProductId": "01txx0000006kfMAAQ",
            "cartLineItemQuantity": 2,
            "cartLineItemAmount": 100,

```

```
    "cartLineItemId": "abcd1",
    "cartLineNetUnitPrice__std": 50,
    "cartLinePromotionAdditionalInformation__std": [
      {
        "cartLineCouponCode__std": "PIZZA5"
      }
    ],
  },
  {
    "cartLineProductId": "01txx0000006kfNAAQ",
    "cartLineItemQuantity": 2,
    "cartLineItemAmount": 10,
    "cartLineItemId": "abcd2",
    "cartLineNetUnitPrice__std": 5
  }
]
}
]
```

Properties

Name	Type	Description	Required or Optional	Available Version
cart	Map<String, Object>	Details of a cart and its line items used to evaluate and execute promotions. The payload must include <i>cartHeaderId__std</i>, which uniquely identifies the cart and is required for promotion evaluation and execution. Specify the applicable <i>currencyISOCode</i> for the transaction. To evaluate or execute coupon-based promotions, include the <i>couponCode</i>, which must match an active coupon or promotion configured in the system. Along with the required parameters, provide cart-level details such as the total transaction amount, and cart line item details such as line item quantity and amount, <i>cartLineNetUnitPrice__std</i> (required for evaluation of line-level and cross-line promotions), and product ID.	Required	65.0

Cart Input Map

Input representation of the summary of the cart whose eligible promotions are to be fetched.

Root XML tag

CartInputMapRepresentation

JSON examples

Here's an example of the details of a customer's cart whose eligible promotions are to be fetched:

 **Note:** Provide `contactId` to retrieve campaigns related to a Person Account.

```
{
  "cart": {
    "cartDetails": [
      {
        "activityStartDate": "2023-06-14T12:45:19Z",
        "contactId/loyaltyProgramMemberId/accountId": "003xx000004WjzAAAS",
        "transactionAmount": 499,
        "currencyISOCODE": "USD",
        "engagementChannelId": "0eFxx0000000008EAA",
        "sourceLocationId/sourceLocation": "0YQxx0000000001GAA",
        "cartLineDetails": [
          {
            "cartLineProduct": "Burger",
            "cartLineProductCode": "testProductCode",
            "cartLineProductStockKeepingUnit": "testSKU",
            "cartLineProductCatalog": "Food",
            "cartLineItemQuantity": 1,
            "cartLineItemAmount": 100,
            "cartLineProductCategory": "test"
          }
        ]
      }
    ]
  }
}
```

Here's an example of the details of a loyalty program member's cart whose eligible loyalty program promotions are to be fetched:

```
{
  "cart": {
    "cartDetails": [
      {
        "activityStartDate": "2023-06-14T12:45:19Z",
        "contactId/loyaltyProgramMemberId/accountId": "003xx000004WjzAAAS",
        "currencyISOCODE": "INR",
        "transactionAmount": 499,
        "membershipNumber": "Member-005",
        "cartLineDetails": [
          {
            "cartLineProductId": "01txx0000006iTsAAI",
            "cartLineProductCode": "AnyProductCode",
            "cartLineProductStockKeepingUnit": "AnySKU",
            "cartLineItemQuantity": 2,
            "cartLineItemAmount": 99,
          }
        ]
      }
    ]
  }
}
```

```
        "cartLineProductCatalogId":"0ZSSB0000002d8X4AQ"
      },
      {
        "cartLineProduct":"Coke",
        "cartLineProductCode": "AnyProductCode",
        "cartLineProductStockKeepingUnit": "AnySKU",
        "cartLineItemQuantity":10,
        "cartLineItemAmount":400,
        "cartLineProductCatalog":"Beverages"
      }
    ]
  }
}
```

Here’s an example to fetch eligible loyalty program promotions for accounts:

```
{
  "cart": {
    "cartDetails": [
      {
        "activityStartDate": "2024-10-16T22:45:19Z",
        "accountId": "001xx000003H3NnAAK",
        "transactionAmount": 40,
        "currencyISOCode": "USD",
        "cartLineDetails": [
          {
            "cartLineProductCatalog": "Beverages",
            "cartLineProductCategory": "Coffee Creations",
            "cartLineProduct": "Caramel Latte",
            "cartLineItemQuantity": 5
          }
        ]
      }
    ]
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
cart	Map<String, Object>	Fields configured in the Cart and CartLine nodes of the <i>CartDetailGetEligiblePromotion</i> context service definition in your org. These fields represent the details of a cart and its line items whose eligible promotions are to be fetched. To get eligible promotions associated with a loyalty program member’s cart, the required parameters are:	Required	60.0

Name	Type	Description	Required or Optional	Available Version
		- membershipNumber or loyaltyProgramMemberId - currencyISOCode For other eligible promotions, the required parameters are: - contactId, contactLastName, or contactEmail - currencyISOCode Along with the required parameters, provide specific details of a cart, such as the total transaction amount, and the cart line item details, such as the line item quantity and amount, the product ID or name, the product catalog ID or name, the product code, and the product stock keeping unit.		
contextInstanceId	String	ID of the context instance.	Optional	65.0

Channel Input

Input representation details of a channel for which to retrieve eligible promotions.

Properties

Name	Type	Description	Required or Optional	Available Version
channel	Name and ID Input on page 482	ID and/or name of the input records.		64.0
retailStore	Name and ID Input on page 482	ID and/or name of the input records.		64.0

Clone Promotion Input

Input representation that decides which existing promotion is cloned and the attributes of the new promotion that’s created.

JSON example

```
{
  "sourcePromotionId": "0c8xx00000000WHAAY",
  "name": "ups clone 4",
  "startDateTime": "2024-01-08T00:00:00.000Z",
  "endDateTime": "2024-01-09T00:00:00.000Z",
}
```

```

"promotionCode": "sdfgy543wsd",
"doesReuseEngagementAttributes": "true",
"additionalFieldValues": {
  "additionalFieldValuesMap": {
    "Description": "test ups clone"
  }
},
"relatedRecords": {
  "relatedRecordsList": [
    {
      "relatedObjectApiName": "PromotionChannel",
      "relatedObjectPromotionFieldName": "PromotionId"
    },
    {
      "relatedObjectApiName": "PromotionLoyaltyPtnrProdt",
      "relatedObjectPromotionFieldName": "PromotionId"
    },
    {
      "relatedObjectApiName": "PromotionMarketSegment",
      "relatedObjectPromotionFieldName": "PromotionId"
    },
    {
      "relatedObjectApiName": "PromotionProduct",
      "relatedObjectPromotionFieldName": "PromotionId"
    },
    {
      "relatedObjectApiName": "PromotionProductCategory",
      "relatedObjectPromotionFieldName": "PromotionId"
    },
    {
      "relatedObjectApiName": "PromotionLimit",
      "relatedObjectPromotionFieldName": "PromotionId"
    }
  ]
}
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalFieldValues	Map<String, String>	Details of the promotion's additional fields.	Optional	61.0
doesReuseEngagementAttributes	Boolean	Indicates whether the new promotion uses the engagement attributes created for the source promotion to track customers' promotion usage (true) or (not). Engagement reuse can result in shared limits.	Optional	61.0
endDateTime	String	End date and time of the new promotion.	Required	61.0
name	String	Name of the new promotion.	Required	61.0

Name	Type	Description	Required or Optional	Available Version
promotionCode	String	Code of the new promotion.	Optional	61.0
relatedRecords	Related Promotion Records Input [] on page 492	Related record for the source promotion that have to be created for the new promotion.	Optional	61.0
source PromotionId	String	ID of the promotion that's cloned.	Required	61.0
startDateTime	String	Start date and time of the new promotion.	Required	61.0

Club Member Enrollment Input

The input representation to enroll a club member.

Properties

Name	Type	Description	Required or Optional	Available Version
accountId	String	The ID of the associated account record.		64.0
associated Account Details	Loyalty Account Input on page 479	The business account details of the member.		64.0
associated Contact Details	Loyalty Contact Input on page 480	The contact associated with the club member.		64.0
club Membership Number	String	The membership number of the club member.		64.0
contactId	String	The contact details of the member.		64.0
enrollment Date	String	The date on which a member is enrolled into the club.		64.0
loyalty Membership Number	String	The membership number of the loyalty program member.		64.0
memberStatus	String	The status of the club member.		64.0
membershipBy	String	Member enrolled based on.		64.0
membership EndDate	String	The end date of the club membership.		64.0

Club Membership Renewal Input

The input representation for club membership renewal.

JSON example

```
{
  "clubMembershipNumber": "ARC0001",
  "renewalPeriodType": "months",
  "renewalPeriod": "3",
  "notes": "",
  "externalIdentifier": "",
  "OrderItemId" : ""
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
club Membership EndDate	String	The end date of the club membership.		64.0
club Membership Number	String	The membership number of the club member.		64.0
external Identifier	String	The external identifier of the club member.		64.0
loyalty Membership Number	String	The membership number of the loyalty program member.		64.0
notes	String	Additional information about the club membership renewal.		64.0
orderItemId	String	The order associated with the club membership. This field is a relationship field.		64.0
renewalPeriod	Integer	The specific time period during which a club membership can be renewed.		64.0
renewalPeriod Type	String	The unit of time used to measure the renewal period for a club membership. For example, Days , Months, or Years.		64.0

Corporate Member Enrollment Input

Input representation for creating a business account and a Corporate type Loyalty Program Member record for the member.

Root XML tag

```
<corporateMemberEnrollment>
```

JSON example

```
{
  "enrollmentDate" : "2021-01-01T05:05:05",
  "membershipNumber" : "Member123",
  "associatedAccountDetails" : {
    "name" : "Name1",
    "phone" : "908989898",
    "website" : "www.ab.com",
    "allowDuplicateRecords" : "false",
    "additionalAccountFieldValues" : {
      "attributes" : {
        "CustomNumber__c" : "20.0"
      }
    }
  },
  "memberStatus" : "Active",
  "referredBy" : "Member12",
  "referredByMemberReferralCode" : "MPX12adf",
  "createTransactionJournals" : "true",
  "transactionJournalStatementFrequency" : "Monthly",
  "additionalMemberFieldValues" : {
    "attributes" : {
      "CustomBoolean__c" : "true"
    }
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalMemberFieldValues	Map<String, String>	The custom attributes of the member in the loyalty program member record. - Keys - Values	Optional	55.0
associatedAccountDetails	Loyalty Account Input	The business account details of the member.	Required	55.0
canReceivePartnerPromotions	Boolean	Indicates whether the member can receive partner promotions for the program (<code>true</code>) or not (<code>false</code>).	Optional	55.0
canReceivePromotions	Boolean	Indicates whether the member can receive promotions for the program.	Optional	55.0

Name	Type	Description	Required or Optional	Available Version
createTransaction Journals	Boolean	Indicates whether to create the transaction journal records (<code>true</code>) or not (<code>false</code>): The transaction journal records are created to: • Credit points to the member for enrolling. • Credit points to the member who has referred the new member.	Optional	55.0
enrollmentChannel	String	The channel used by members to enroll into a loyalty program. Possible values are: • CallCenter • Email • Franchise • Mobile • Partner • Pos • Print • Social • Store • Web	Optional	55.0
enrollmentDate	String	The date on which a member is enrolled.	Required	55.0
memberStatus	String	Status of the member. Possible values are: • Active • Inactive • Custom value	Required	55.0
membershipEndDate	String	The date on which the loyalty program member's membership ends.	Optional	55.0
membershipNumber	String	The unique membership number of the loyalty program member.	Required	55.0
referredBy	String	The ID of the loyalty program member who has referred the member.	Optional	55.0
referredByMemberReferralCode	String	The referral code of the loyalty program member who has referred the member.	Optional	57.0

Name	Type	Description	Required or Optional	Available Version
transactionJournalStatementFrequency	String	The frequency at which transaction journal statements must be delivered to the member. Possible values are: - Daily - Monthly - Yearly	Optional	55.0
transactionJournalStatementMethod	String	The method used to deliver transaction journal statements to the member. Possible values are: - Email - Mail	Optional	55.0

Coupon Usage Input

Input representation of coupon usage.

JSON example

```
{
  "couponCodeList": [
    "QWERTY",
    "ABCD"
  ],
  "activityDateTime": "2024-10-05T12:45:19Z",
  "externalTransactionNumber": "01Vxx000000006T",
  "customerId": "01Mxx0000000001"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
couponCodeList	List<String>	The codes of the list of coupons for which the usage must be increased or decreased.	Required	62.0
activityDateTime	DateTime	The date and time at which the customer redeemed the coupon.	Required	62.0
externalTransactionNumber	String	The externally generated unique identifier of cart for which the customer redeemed the coupon. This ID must be the same in both the decrease and increase usage APIs so that the API can correctly track coupon usage for the originating transaction.	Required	62.0

Name	Type	Description	Required or Optional	Available Version
customerId	String	The ID of either the contact, the account, or the loyalty program member who redeemed the coupon.	Required	62.0

Eligible Clubs Input

The club tier groups and tiers whose members are eligible for the promotion.

Properties

Name	Type	Description	Required or Optional	Available Version
clubTier	String	The club's tier whose members are eligible for the promotion.		64.0
clubTierGroup	String	The club's tier group whose members are eligible for the promotion.		64.0

Eligible Coupon Input

The details of an eligible coupon for the promotion.

Properties

Name	Type	Description	Required or Optional	Available Version
couponCode	String	Specifies the coupon code that is eligible for the promotion.		63.0
description	String	Specifies the description of the coupon that is eligible for the promotion.		63.0
endDateTime	String	Specifies the end date and time of the coupon eligible for the promotion.		63.0
~~redemptionLimitOverall~~	Integer	Specifies the overall redemption limit of the coupon that is eligible for the promotion.		63.0
~~redemptionLimitPerCustomer~~	Integer	Specifies the redemption limit per customer of the coupon that is eligible for the promotion.		63.0
startDateTime	String	Specifies the start date and time of the coupon eligible for the promotion.		63.0
status	String	Specifies the status of the coupon that is eligible for the promotion.		63.0

Eligible Enrollment Period Input

Input representation of the enrollment eligibility details for the promotion.

Properties

Name	Type	Description	Required or Optional	Available Version
enrollment EndDate	String	Date until which customers can enroll for the promotion.	Optional	60.0
enrollment StartDate	String	Date from which customers can enroll for the promotion.	Optional	60.0
isEnrollment Required	Boolean	Indicates whether customers must enroll for the promotion (<code>true</code>) or not (<code>false</code>). The default value is <code>false</code> .	Optional	60.0

Eligible Loyalty Tiers Input

Input representation of the loyalty program tier whose members are eligible for the promotion.

Properties

Name	Type	Description	Required or Optional	Available Version
tier	Name And Id Input []	ID or name of the loyalty program tier whose members are eligible for the promotion.	Optional	60.0
tierGroup	Name And Id Input []	ID or name of the loyalty program tier group whose members are eligible for the promotion.	Optional	60.0

Events Detail Input

Input representation details of the customer event that is eligible for the promotion.

Properties

Name	Type	Description	Required or Optional	Available Version
event	String	Customer event eligible for the promotion. Possible values are: - CustomerBirthday - MemberAnniversary - MemberBirthday	Optional	60.0

Name	Type	Description	Required or Optional	Available Version
		MemberAnniversary and MemberBirthday are valid only for the loyalty program-related promotions.		
eventPeriod Unit	String	Period for which the promotion is eligible. Possible values are: • Date • Week • Month	Optional	60.0

Individual Member Enrollment Input

Input representation for creating a person account and an Individual type Loyalty Program Member record for the member.

Root XML tag

```
<individualMemberEnrollment>
```

JSON example

```
{
  "enrollmentDate" : "2021-01-01T00:00:00",
  "membershipNumber" : "Member1",
  "associatedContactDetails" : {
    "firstName" : "Name1",
    "lastName" : "Name2",
    "email" : "ab@email.com",
    "allowDuplicateRecords" : "false",
    "additionalContactFieldValues" : {
      "attributes" : {
        "PersonMailingCity" : "Delhi",
        "CustomNumber__c" : "20.0"
      }
    }
  },
  "memberStatus" : "Active",
  "referredBy" : "Member2",
  "referredByMemberReferralCode" : "MPX12adf",
  "createTransactionJournals" : "true",
  "transactionJournalStatementFrequency" : "Monthly",
  "transactionJournalStatementMethod" : "Mail",
  "enrollmentChannel" : "Email",
  "canReceivePromotions" : "true",
  "canReceivePartnerPromotions" : "true",
  "membershipEndDate" : "2022-01-01T00:00:00",
  "relatedCorporateMembershipNumber" : "Member3",
  "additionalMemberFieldValues" : {
    "attributes" : {
      "CustomBoolean__c" : "true"
    }
  }
}
```

```

    }
  }

```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalMemberFieldValues	Map<String, String>	The custom attributes of the member in the loyalty program member record. • Keys • Values	Required	55.0
associatedContact on page 480Details	Loyalty Contact Input on page 480	The person account record details of the individual type member.	Required	55.0
canReceivePartnerPromotions	Boolean	Indicates whether the member can receive partner promotions for the program (<code>true</code>) or not (<code>false</code>).	Optional	55.0
canReceivePromotions	Boolean	Indicates whether the loyalty program member can receive promotions.	Optional	55.0
createTransaction Journals	Boolean	Indicates whether to create the transaction journal records (<code>true</code>) or not (<code>false</code>): The transaction journal records are created to: • Credit points to the member for enrolling. • Credit points to the member who has referred the new member.	Optional	55.0
enrollmentChannel	String	The channel used to by loyalty program member to enroll in to the loyalty program. Possible values are: • Pos • Web • Email • CallCenter • Social • Mobile • Store • Franchise • Partner • Print	Optional	55.0

Name	Type	Description	Required or Optional	Available Version
enrollmentDate	Date	The date on which the loyalty program member is enrolled in to the loyalty program. The format is: YYYY-MM-DDTHH:mm:ss. For example: 2020-09-08T00:00:00. This field must be provided in the org's configured timezone, not in Central UTC. The system doesn't convert this value from UTC to org timezone internally. In other words, the provided datetime already represents the correct org-local enrollment date and time.	Required	55.0
memberStatus	String	The status of the loyalty program member. Possible values are: - Active - Inactive - Custom	Required	55.0
membershipEndDate	String	The date on which the loyalty program member's membership ends.	Optional	55.0
membershipNumber	String	The unique membership number of the loyalty program member.	Required	55.0
referredBy	String	The membership number of the loyalty program member who has referred the member.	Optional	55.0
referredByMemberReferralCode	String	The referral code of the loyalty program member who has referred the member.	Optional	57.0
relatedCorporateMembershipNumber	String	External corporate membership number of the member.	Optional	55.0
transactionJournalStatementFrequency	String	The frequency at which transaction journal statements must be delivered to the member. Possible values are: - Monthly - Quarterly	Optional	55.0
transactionJournalStatementMethod	String	The method used to deliver transaction journal statements to the member. Possible values are: Mail	Optional	55.0

Name	Type	Description	Required or Optional	Available Version
		Email		

Link Program Member Partner Input

Input details to link a program member and partner.

JSON example

```
{
  "externalId": "U10001",
  "additionalAttributes": {
    "Card_Type__c": "Green Card"
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalAttributes	Object	Attributes of the member that must be saved in the linkage record.	Optional	66.0
externalId	String	External ID of the loyalty program member associated with the partner.	Optional	66.0

Loyalty Engine Input

Input representation of the Loyalty Engine Runtime API.

Root XML tag

```
<loyaltyEngineInput>
```

JSON example



Note: Some of the API request fields contain field names that are interchangeable. For information on the list of interchangeable field names, see [Loyalty Management Connect API Interchangeable Request Fields](#).

```
{
  "shouldCheckCouponUsageLimit": true
  "transactionJournals": [
    {
      "ActivityDate": "2022-01-04T12:45:19Z",
      "JournalDate": "2022-01-04T00:45:19Z",
      "Brand": "",
      "Establishment": "Apple",
      "ExternalTransactionNumber": "P1-981950",
      "JournalTypeId": "01ET10000004CQp",
      "LoyaltyProgramId": "01pT10000004CdN",
      "MemberId": "01MT10000004CfA",
      "TransactionAmount": "117",
      "TransactionLocation": "HitechCity",
    }
  ]
}
```

```

        "VoucherCode": "",
        "Origin": "asdf",
        "Distance": "5",
        "Destination": "Hyderabad",
        "FlightNumber": "SF-Ran-001",
        "Status": "Pending"
    }
],
"runSetting": {
    "isSimulation": true
}
}

```



Note: Provide `contactId` to retrieve rewards related to a Person Account.

Using the transaction journal ID:

```

{
  "transactionJournals": [
    {
      "Id": "01Vxx00000000cj"
    }
  ],
  "runSetting": {
    "isSimulation": true
  }
}

```

Sample payload to retrieve rewards assigned to accounts.

```

{
  "transactionJournals": [
    {
      "ActivityDate": "2024-11-27T12:45:19Z",
      "JournalTypeId": "01ERM00000001Bb2AI",
      "LoyaltyProgramId": "01pRM00000002YTYAY",
      "ReferredPartyId": "001xx000003H3NnAAK",
      "Status": "Pending",
      "ProductId": "01txx00000006i44AAA",
      "ProductCategoryId": "02Gxx0000000001GAA",
      "Quantity": "1",
      "TransactionAmount": "42",
      "appliedPromotions": [
        {
          "promotionId": "0c8xx00000000mPAAQ",
          "rewards": [
            {
              "rewardType": "Discount",
              "discountAmount": 5
            },
            {
              "rewardType": "Discount",
              "discountAmount": 10
            },
            {

```

```
      "rewardType": "Voucher",
      "voucherDefinitionName": "Voucher2",
      "voucherExpirationDate": "2025-08-07",
      "voucherEffectiveDate": "2024-08-08",
    },
    {
      "rewardType": "Game",
      "gameDefinitionName": "TestPostFacto",
    },
  ],
}
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
~~includeCouponLimit~~	Boolean	Indicates whether the coupon usage limit check is enabled.	Required	62.0
transactionJournals	Loyalty Engine Input Map	Represents the transaction journals records to process.	Required	54.0
runSetting	Loyalty Engine Run Settings Input	Loyalty runtime setting to run the simulation on the transaction journals.	Required for simulation, otherwise optional	55.0

Loyalty Engine Input Map

Input representation of the Transaction Journal record execution request.

JSON example

In the JSON, you can either provide the details of transaction journals that aren't available in the org, or a list of IDs that belong to transaction journals available in the org. If you provide the details of transaction journals that aren't available in the org, each transaction journal's applicable loyalty program process applies the eligible rules, executes the applicable actions, and then creates the transaction journal.

Using the transaction journal details:

 **Note:** Some of the API request fields contain field names that are interchangeable. For information on the list of interchangeable field names, see [Loyalty Management Connect API Interchangeable Request Fields](#).

Here's a JSON example that creates a transaction journal for a purchase-related transaction made by a loyalty program member of a retail company. It has all the transaction journal-related information along with the details of the promotion applied, including the different types of rewards the transaction is eligible for. For information on the TransactionJournal object fields, see https://developer.salesforce.com/docs/atlas.en-us.loyalty.meta/loyalty/sforce_api_objects_transactionjournal.htm.

```
{
  "transactionJournals": [
    {
```

```

    "ActivityDate": "2023-11-27T12:45:19Z",
    "JournalTypeId": "01ERM00000001Bb2AI",
    "LoyaltyProgramId": "01pRM000000002YTYAY",
    "MemberId": "01MRM00000002Fmu2AE",
    "Status": "Pending",
    "ProductId": "01txx00000006iTsaAI",
    "ProductCategoryId": "0ZSSB00000002d8X4AQ",
    "Quantity": "4",
    "TransactionAmount": "500",
    "appliedPromotions": [
      {
        "promotionId": "0c8RM00000004FixYAU",
        "couponCode": "COUPON100",
        "rewards": [
          {
            "rewardType": "Discount",
            "discountAmount": 15
          },
          {
            "rewardType": "Points",
            "loyaltyProgramCurrencyName": "Coins",
            "points": 100,
            "relatedInformation": "Test",
            "notes": "points reward"
          },
          {
            "rewardType": "Voucher",
            "voucherDefinitionName": "10% Discount Voucher",
            "voucherExpirationDate": "2023-12-15",
            "voucherEffectiveDate": "2023-11-29",
            "notes": "discount voucher"
          },
          {
            "rewardType": "Badge",
            "loyaltyProgramBadgeName": "Gen AI Badge",
            "memberBadgeStatus": "Active",
            "badgeValidityEndDate": "2023-12-15",
            "reason": "purchase"
          },
          {
            "rewardType": "Game",
            "gameDefinitionName": "SpintheWheelGame"
          }
        ]
      }
    ]
  }
}

```

Here's a JSON example to create a transaction journal for a loyalty program member of an airline company.

```

{
  "transactionJournals": [
    {

```

```

    "ActivityDate": "2022-01-04T12:45:19Z",
    "JournalDate": "2022-01-04T00:45:19Z",
    "Brand": "Tata",
    "Establishment": "Vistara",
    "ExternalTransactionNumber": "P1-981950",
    "JournalTypeId": "01ET10000004CQp",
    "LoyaltyProgramId": "01pT10000004CdN",
    "MemberId": "01MT10000004CfA",
    "TransactionAmount": "117",
    "TransactionLocation": "HitechCity",
    "VoucherCode": "",
    "Origin": "Bangalore",
    "Distance": "5",
    "Destination": "Hyderabad",
    "FlightNumber": "VS-Ran-001",
    "Status": "Pending"
  }
]
}

```

Using the transaction journal ID:

 **Note:** This is supported in API version 55.0 and later.

```

{
  "transactionJournals": [
    {
      "Id": "01Vxx00000000cj"
    }
  ]
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
processInput	Map<String, Object>	List of the Transaction Journal records to process and create.	Required	54.0

Loyalty Engine Run Settings Input

Input representation of the loyalty engine settings to run simulation on the transaction journals.

Root XML tag

```
<loyaltyEngineRunSettingsRepresentation>
```

JSON example

```

{
  "runSetting": {
    "isSimulation": true
  }
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
isSimulation	Boolean	Indicates whether to run a simulation on the transaction journals (<code>true</code>) or not (<code>false</code>). Default value is <code>false</code> .	Required for simulation otherwise optional.	55.0

Loyalty Account Input

Input representation for creating a business account for the member.

Root XML tag

```
<loyaltyAccountInput>
```

JSON example

```
{
  "associatedAccountDetails": {
    "name": "Name1",
    "phone": "908989898",
    "website": "www.ab.com",
    "allowDuplicateRecords": "false",
    "additionalAccountFieldValues": {
      "attributes": {
        "CustomNumber__c": "20.0"
      }
    }
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalAccountFieldValues	Map<String, String>	The additional attribute values that must be saved in the account record.	Optional	55.0
allowDuplicateRecords	Boolean	Indicates whether a duplicate member record can be created, when an existing record with identical details exists in the org. This parameter works in conjunction with your Duplicate Rules. A Duplicate Rule's Action on Create setting determines if Salesforce allows or blocks a potential duplicate. For instructions on configuring these rules, see Manage Duplicate Records . The default value is <code>True</code> .	Optional	55.0
name	String	Name of the account.	Required	55.0

Name	Type	Description	Required or Optional	Available Version
phone	String	The phone number associated with the account.	Optional	55.0
website	String	The website associated with the account.	Optional	55.0

Loyalty Contact Input

Input representation of the loyalty program member's person account.

Root XML tag

```
<loyaltyContactInput>
```

JSON example

```
{
  "enrollmentDate" : "2021-01-01T00:00:00",
  "membershipNumber" : "Member1",
  "associatedContactDetails" : {
    "firstName" : "Name1",
    "lastName" : "Name2",
    "email" : "ab@gmail.com",
    "allowDuplicateRecords" : "false",
    "additionalContactFieldValues" : {
      "attributes" : {
        "PersonMailingCity" : "Delhi",
        "CustomNumber__c" : "20.0"
      }
    }
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalContactFieldValues	Map<String, String>	The additional attributes that must be saved in the person account.	Optional	55.0
allowDuplicateRecords	Boolean	Indicates if a duplicate person account record can be created for a member when an existing person account record exists. This parameter works in conjunction with your Duplicate Rules. A Duplicate Rule's Action on Create setting determines if Salesforce allows or blocks a potential duplicate. For instructions on configuring these rules, see Manage Duplicate Records . The default value is <code>True</code> .	Optional	55.0
email	String	Email ID of the person account.	Optional	55.0
firstName	String	First name of the person account.	Optional	55.0

Name	Type	Description	Required or Optional	Available Version
lastName	String	Last name of the person account.	Required	55.0

Loyalty Process Input

Input representation of the details of the loyalty program process action.

Properties

Name	Type	Description	Required or Optional	Available Version
processParameters	Loyalty Process Parameter Map	Input parameters for running the loyalty process.	Required	56.0

Loyalty Process Input Options

Input representation of the execution options for the loyalty process.

Properties

Name	Type	Description	Required or Optional	Available Version
executionType	Enum	Specifies the Rules Engine execution type. Possible values are: • Explainability—If the execution type is Explainability, the process is executed and the output parameters are returned in the response. • Simulation— If the execution type is Simulation, only the simulation details are returned in the response.	Required	56.0

Loyalty Process Parameter Map Input

Input representation of the input parameters for the loyalty process.

Properties

Name	Type	Description	Required or Optional	Available Version
parameterInput	Map<String, Object>	Input parameters for the loyalty process.	Required	56.0

Loyalty View Transaction Journal History Input

Input representation for getting the transaction journal records of a loyalty member.

JSON example

```
{
  "membershipNumber": "doc_member1",
  "journalType": "Accrual",
  "journalSubType": "Social",
  "periodStartDate": "2022-01-27T00:00:00Z",
  "periodEndDate": "2022-11-27T00:00:00Z"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
journalSubType	String	The subtype of the transaction journal record.	Optional	56.0
journalType	String	The type of the transaction journal. Possible values include: • Accrual • Redemption	Required	56.0
membershipNumber	String	The membership number of the loyalty program member.	Required	56.0
periodEndDate	String	Retrieve transaction journals until this date.	Optional	56.0
periodStartDate	String	Retrieve transaction journals starting on this date.	Required	56.0

Name and ID Input

Input representation of an object's name and ID.

Properties

Name	Type	Description	Required or Optional	Available Version
id	String	ID of the object.	Optional if name is specified.	60.0
name	String	Name of the object.	Optional if ID is specified.	60.0

Product Details

An input representation that contains specific details related to the product.

Properties

Name	Type	Description	Required or Optional	Available Version
id	String	The ID of the record.		64.0
name	String	The name of the record.		64.0
productCode	String	The product code of the record.		64.0
productType	ProductType	Specifies the product type that is eligible for the promotion.		64.0
stockKeepingUnit	String	The stock keeping unit that is eligible for the promotion.		64.0

Promotion Eligible Channel Input

Input representation details of the channel eligible for the promotion.

Properties

Name	Type	Description	Required or Optional	Available Version
channel	Name And Id Input []	ID or name of the channel eligible for the promotion. This field takes the IDs or names of <code>EngagementChannelType</code> configured in your org. Possible names for the online channel type are: • App • Mobile • Web Possible names for the social channel type are: • Facebook • Instagram • Twitter • WhatsApp	Optional	60.0
channelType	String	Type of the channel eligible for the promotion. Possible values are: • Offline • Online • Social	Optional	60.0

Name	Type	Description	Required or Optional	Available Version
retail LocationGroup	Name And Id Input []	ID or name of the retail location group eligible for the promotion. This field is valid only for the offline channel type.	Optional	60.0
store	Name And Id Input []	ID or name of the retail store eligible for the promotion. This field is valid only for the offline channel type.	Optional	60.0

Promotion Eligibility Input

Input representation details of the promotion eligibility criteria.

Properties

Name	Type	Description	Required or Optional	Available Version
areAllLoyalty MembersEligible	Boolean	Indicates whether all loyalty members are eligible for the promotion (<code>true</code>) or not (<code>false</code>). If this field is set to true, you can't specify loyalty tiers whose members are eligible for the promotion.	Optional	60.0
eligibleCampaign	Name And Id Input []	ID or name of the marketing campaign whose members are eligible for the promotion. Specify either an eligible campaign, a list of eligible segments, or loyalty tiers in a request, but not all. To specify an eligible marketing campaign, you must have the Marketing license.	Optional	60.0
eligibleChannels	Promotion Eligible Channel Input []	Channels that are eligible for the promotion.	Optional	60.0
eligibleClubs	ConnectApi.Eligible Club Input Representation	List of clubs whose members are eligible for the promotion.	Optional	64.0
eligibleCoupons	ConnectApi.Eligible Coupon Input Representation	Coupons that are eligible for the promotion	Optional	63.0
eligible CustomerEvents	Events Detail Input	Details of the customer events that the promotion is eligible for.	Optional	60.0

Name	Type	Description	Required or Optional	Available Version
eligibleEnrollmentPeriod	Eligible Enrollment Period Input	Period during which customers can enroll in the promotion.	Optional	60.0
eligibleLoyaltyPartnerProducts	Name And Id Input []	IDs or names of the loyalty partner products that are eligible for the promotion. You can specify only products that belong to loyalty partners associated with the specified loyalty program. Specify either a list of products and categories, or a list of loyalty partner products in a request, but not both.	Optional	60.0
eligibleLoyaltyTiers	Eligible Loyalty Tiers Input []	List of loyalty program tiers whose members are eligible for the promotion. Specify either an eligible campaign, a list of eligible segments, or loyalty tiers in a request, but not all.	Optional	60.0
eligibleProductCategories	Name And Id Input []	IDs or names of the product categories that are eligible for the promotion. Ensure that the number of eligible product categories you want to specify doesn't exceed the limit supported in your org.	Optional	60.0
eligibleProducts	Name And Id Input []	IDs or names of the products that are eligible for the promotion. Ensure that the number of eligible products you want to specify doesn't exceed the number supported in your org.	Optional	60.0
eligibleSegments	Name And Id Input []	IDs or names of the Data Cloud segments whose members are eligible for the promotion. You must have the Marketing user permission to specify market segments. Specify either an eligible campaign, a list of eligible segments, or loyalty tiers in a request, but not all.	Optional	60.0
productCatalog	Name And Id Input []	ID or name of the product catalog that's eligible for the promotion. This field can't be updated.	Optional	60.0

Promotion Input

Input representation details of a promotion.

JSON example for POST and PUT

```
{
  "promotionDetails": {
    "name": "Sample Promotion 6",
    "displayName": "Sample Promotion",
    "priorityNumber": 1,
    "description": "Sample promotion description",
    "promotionCode": "XYZ",
    "startDateTime": "2023-06-30T00:00:00.000Z",
    "endDateTime": "2023-12-30T00:00:00.000Z",
    "objective": "To increase sales",
    "isAutomatic": true,
    "loyaltyProgram": {
      "name": "NTO"
    },
    "additionalFieldValues": {
      "attributes": {
        "isUnifiedPromotion__c": true
      }
    },
    "promotionEligibility": {
      "productCatalog": {
        "id": "0ZSxx000000001dGAA"
      },
      "eligibleProductCategories": [
        {
          "id": "0ZGxx0000000001dGAA"
        }
      ],
      "eligibleProducts": [
        {
          "name": "someProduct"
        }
      ],
      "eligibleChannels": [
        {
          "channelType": "****",
          "channel": {
            "id": "****",
            "name": "****"
          },
          "store": {
            "id": "****",
            "name": "****"
          },
          "retailLocationGroup": {
            "name": "****"
          }
        }
      ]
    }
  }
}
```

```

    ],
    "eligibleCampaign": {
      "name": "****"
    },
    "eligibleSegments": [
      {
        "name": "someSegment"
      }
    ],
    "eligibleCustomerEvents": {
      "event": "MemberBirthday",
      "eventPeriodUnit": ""
    },
    "eligibleEnrollmentPeriod": {
      "isEnrollmentRequired": true,
      "enrollmentStartDate": "",
      "enrollmentEndDate": ""
    },
    "eligibleLoyaltyPartnerProducts": [
      {
        "name": "",
        "id": ""
      }
    ],
    "eligibleLoyaltyTiers": [
      {
        "tierGroup": {
          "name": "",
          "id": ""
        },
        "tier": {
          "name": "",
          "id": ""
        }
      }
    ],
    "eligibleCoupons": [
      {
        "couponCode": "",
        "status": "",
        "description": "",
        "startDateTime": "",
        "endDateTime": "",
        "redemptionLimitOverall": "",
        "redemptionLimitPerCustomer": ""
      }
    ]
  },
  "promotionLimits": {
    "perLineItemLimit": 10,
    "pointsPerCustomerLimit": 5,
    "vouchersPerCustomerLimit": 5,
    "perCartLimit": 5,
    "usePerCustomerLimit": 5,

```

```

        "totalUseLimit" : 25,
        "pointsLimit": 1000,
        "vouchersLimit": 100,
        "liabilityLimit": 10000,
        "loyaltyProgramCurrency": {
            "id": "01cxx0000000001AAA"
        }
    },
    },
    "rules": [
        {
            "templateName": "BuyXToGetRewards",
            "ruleName": "123",
            "priority": 1,
            "eventConfiguration": [
                {
                    "allOf": [
                        {
                            "anyOf": {
                                "purchaseType": "Product",
                                "applicablePurchaseList": [
                                    {
                                        "name": "Veg Burger"
                                    }
                                ],
                                "purchaseCriteria": {
                                    "type": "Quantity",
                                    "value": 1
                                }
                            }
                        }
                    ]
                }
            ],
            "rewardConfiguration": [
                {
                    "type": "ProvideDiscount",
                    "rewardDetails": {
                        "discountValue": "100",
                        "discountType": "PercentageOff"
                    },
                    "targetAudience": {
                        "audienceType": "LoyaltyTierMembers",
                        "audienceDetails": {
                            "tierGroup": {
                                "name": "Default"
                            },
                            "tier": {
                                "name": "Silver"
                            }
                        }
                    }
                }
            ],
        }
    ],
    {

```



```

        "type": "IssueVoucher",
        "rewardDetails": {
          "voucherDefinition": {
            "name": "GET 10% OFF"
          },
          "voucherExpirationDate": "2020-10-10"
        },
        "targetAudience": {
          "audienceType": "AllLoyaltyMembers"
        }
      }
    ]
  }
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
promotionDetails	Unified Promotions Input[]	Configuration of the promotion	Required	64.0
rules	Unified Promotion Template Input[]	List of rules that are configured for a promotion.	Required	64.0

Promotion Limit Input

Input representation details of the promotion limits.

Properties

Name	Type	Description	Required or Optional	Available Version
liabilityLimit	Double	Maximum financial liability that the promotion can incur.	Optional	60.0
liabilityPerCustomerLimit	Double	Maximum financial liability per customer that the promotion can incur.	Optional	60.0
loyaltyProgramCurrency	Name And Id Input []	ID or name of the active loyalty program currency associated with loyalty points.	Optional	60.0
perCartLimit	Integer	Maximum number of carts the promotion applies to.	Optional	60.0
perLineItemLimit	Integer	Maximum number of cart items the promotion applies to.	Optional	60.0
pointsLimit	Double	Maximum number of points that the promotion can credit.	Optional	60.0

Name	Type	Description	Required or Optional	Available Version
pointsPerCustomerLimit	Double	Maximum number of points that the promotion can credit to a customer.	Optional	60.0
rewardLimit	Integer	Maximum number of rewards that can be issued by the promotion.	Optional	63.0
totalUsageLimit	Integer	Maximum number of times the promotion can be applied in total.	Optional	64.0
usePerCustomerLimit	Integer	Maximum number of times the promotion can be applied for a customer.	Optional	60.0
vouchersLimit	Integer	Maximum number of vouchers that the promotion can issue.	Optional	60.0
vouchersPerCustomerLimit	Integer	Maximum number of vouchers that the promotion can issue to a customer.	Optional	60.0

Promotion Recommendation Input

The input representation to retrieve the recommended promotion.

Properties

Name	Type	Description	Required or Optional	Available Version
dataspace	String	The data space that defines the boundary of a personalization context. This API currently only works with your Data Cloud instance's default data space.	Required	64.0
eligibleChannel	Channel Input on page 462	The details of a channel for which to retrieve eligible promotions.	Optional	64.0
individualId	String	The ID of the of the individual for whom to retrieve recommended promotions. Optional if membership number or loyalty program member ID is provided.	Optional if membership number or loyalty program member ID is provided.	64.0
loyaltyProgramId	String	The loyalty program that the loyalty program member belongs to. The loyalty program ID is required when membership number is specified.	Required when membership number is specified.	64.0
loyaltyProgramMemberId	String	The ID of the of the loyalty program member for whom to retrieve recommended promotions. Optional if membership number or contact ID is provided.	Optional if membership number or contact ID is provided.	64.0

Name	Type	Description	Required or Optional	Available Version
membership Number	String	The membership number of the loyalty program member for whom to retrieve recommended promotions.	Optional	64.0
personalization Point	String	The API name of the personalization point. A personalization point is a connection between data, profile data, personalization type, and schema, used to deliver personalized content to customers.	Required	64.0

Redeem Voucher Input

Input representation for voucher redemption.

JSON example

```
{
  "contactID": "003xx000004WjzAAAS",
  "action": "Reserve",
  "reservationKey": "key1",
  "reservedValue": 50.57,
  "redeemAmount": 50.57,
  "currencyIsoCode": "Yen"
}
```

JSON example to redeem vouchers using `contactId` or `accountID`.

```
{
  "redemptionAmount": "12",
  "contactId": "003xx000004WkhEAAS",
  "accountId": "001xx000003H3NnAAK",
  "memberId": "01Mxx00000002HxEAI"
}
```



Note: Provide `contactId` to redeem the voucher. If `accountID` is provided, then the associated `contactId` is fetched and the voucher is redeemed

Properties

Name	Type	Description	Required or Optional	Available Version
contactID	String	ID of the contact associated with the account.	Optional	62.0
currencyIsoCode	String	Three-letter ISO 4217 currency code of the monetary currency in which the voucher is redeemed.	Optional	55.0
redeemAmount	Double	The value of the voucher to be redeemed. Specify a value when a member hasn't	Optional	53.0

Name	Type	Description	Required or Optional	Available Version
		redeemed the full value of the voucher. This value is only applicable for vouchers issued using Fixed Value voucher definitions. A value isn't required when the member has redeemed the complete value of the voucher in a single transaction.  Note: If you don't specify a value to be redeemed for the fixed-value voucher, the voucher status changes to <code>Redeemed</code>, and the remaining value becomes zero.		
action	ReservationAction	The action to change the status of a voucher. Valid values are: • Reserve • Reinstate • Redeem If you don't specify an action, the default behavior changes voucher status from <code>Issued</code> to <code>Redeemed</code>.	Required when changing the voucher status from <code>Issued</code> to <code>Reserved</code> , or from <code>Reserved</code> to <code>Issued</code> or <code>Redeemed</code> . Otherwise it is optional.	62.0
reservedValue	Currency	Specify the amount reserved for partially redeemable vouchers.	Optional	62.0
reservationKey	String	Specify the key used to redeem or reinstate a reserved voucher.	Optional	62.0

Related Promotion Records Input

Input representation that decides which related records and fields of the existing promotion are cloned and associated with the new promotion.

Properties

Name	Type	Description	Required or Optional	Available Version
relatedObject ApiName	String	API name of the promotion's related object whose records associated with the existing promotion must be cloned for the new promotion.	Small, 61.0	61.0
relatedObject PromotionFieldName	String	The name of the Promotion field in the related object whose record has to be	Small, 61.0	61.0

Name	Type	Description	Required or Optional	Available Version
		cloned and associated with the new promotion.		

Unified Promotions Input

Input representation of the promotion configuration to be created, or updated.



Note: Before updating an existing promotion, get all the existing details of the promotion using the [GET](#) on page 446 operation. To update an existing promotion using the PUT method, you must include all the existing field values of the promotion in the payload along with the new fields, or the fields to be updated.

Root XML tag

unifiedPromotionsInputRepresentation

JSON example for POST and PUT

```
{
  "promotionDetails": {
    "name": "Sample Promotion 6",
    "displayName": "Sample Promotion",
    "priorityNumber": 1,
    "description": "Sample promotion description",
    "promotionCode": "XYZ",
    "startDateTime": "2023-06-30T00:00:00.000Z",
    "endDateTime": "2023-12-30T00:00:00.000Z",
    "objective": "To increase sales",
    "isAutomatic": true,
    "loyaltyProgram": {
      "name": "NTO"
    },
  },
  "additionalFieldValues": {
    "attributes": {
      "isUnifiedPromotion__c": true
    }
  },
  "promotionEligibility": {
    "productCatalog": {
      "id": "0ZSxx000000001dGAA"
    },
    "eligibleProductCategories": [
      {
        "id": "0ZGxx000000001GAA"
      }
    ],
    "eligibleProducts": [
      {
        "name": "someProduct"
      }
    ],
    "eligibleChannels": [
      {
```

```

        "channelType": "****",
        "channel": {
            "id": "****",
            "name": "****"
        },
        "store": {
            "id": "****",
            "name": "****"
        },
        "retailLocationGroup": {
            "name": "****"
        }
    },
    ],
    "eligibleCampaign": {
        "name": "****"
    },
    ],
    "eligibleSegments": [
        {
            "name": "someSegment"
        }
    ],
    ],
    "eligibleCustomerEvents": {
        "event": "MemberBirthday",
        "eventPeriodUnit": ""
    },
    ],
    "eligibleEnrollmentPeriod": {
        "isEnrollmentRequired": true,
        "enrollmentStartDate": "",
        "enrollmentEndDate": ""
    },
    ],
    "eligibleLoyaltyPartnerProducts": [
        {
            "name": "",
            "id": ""
        }
    ],
    ],
    "eligibleLoyaltyTiers": [
        {
            "tierGroup": {
                "name": "",
                "id": ""
            },
            "tier": {
                "name": "",
                "id": ""
            }
        }
    ],
    ],
    "eligibleCoupons": [
        {
            "couponCode": "",
            "status": "",
            "description": "",

```

```

        "startDateTime": "",
        "endDateTime": "",
        "redemptionLimitOverall": "",
        "redemptionLimitPerCustomer": ""
    }
]
},
"promotionLimits": {
    "perLineItemLimit": 10,
    "pointsPerCustomerLimit": 5,
    "vouchersPerCustomerLimit": 5,
    "perCartLimit": 5,
    "usePerCustomerLimit": 5,
    "totalUseLimit": 25,
    "pointsLimit": 1000,
    "vouchersLimit": 100,
    "liabilityLimit": 10000,
    "loyaltyProgramCurrency": {
        "id": "01cxx0000000001AAA"
    }
}
}
}
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalFieldValues	Map<String, String>	Custom, and additional field values that provide more details about the promotion.	Optional	60.0
businessUnit	Name And Id Input []	ID or name of the Marketing Cloud business unit associated with the promotion. To specify a business unit, you must have the Marketing Cloud license.	Optional	60.0
currencyIsoCode	String	ISO code of the active currency associated with the promotion. Before specifying the currency ISO code, make sure that your org supports multiple currencies.	Optional	60.0
description	String	Description of the promotion.	Optional	60.0
displayName	String	Display name of the promotion.	Optional	60.0
endDateTime	String	Date and time until which the promotion can be applied for customers.	Optional	60.0

Name	Type	Description	Required or Optional	Available Version
isAutomatic	Boolean	Indicates whether the promotion is automatically applied for eligible carts (true) or not (false).	Optional	63.0
isEmail Activated	Boolean	Indicates whether the promotion's email is activated (<code>true</code>) or not (<code>false</code>). If set to true, an email template is created for the promotion's email in the org. The default value is <code>false</code> .	Optional	Small, 60.0
loyaltyProgram	Name And Id Input []	ID or name of the active loyalty program the promotion applies to. You can't change the loyalty program of an existing promotion.	Optional	60.0
name	String	Unique name of the promotion.	Required	60.0
objective	String	Objective of the promotion.	Optional	60.0
priority Number	Integer	Number that determines the priority of the promotion in a list of eligible promotions. Priority number must be greater than zero.	Optional	60.0
promotionCode	String	Unique code of the promotion.	Optional	60.0
promotion Eligibility	Promotion Eligibility Input	Eligibility criteria of the promotion.	Optional	60.0
promotionLimits	Promotion Limit Input	Limits of the promotion, including the number of times the promotion can be applied for customers. A promotion limit must be greater than zero.	Optional	60.0
ruleLibrary	String	Rule library associated with the promotion.	Optional	62.0
startDateTime	String	Date and time from which the promotion can be applied for customers.	Required	60.0
termsAndConditions	String	Terms and conditions which are applied to the promotion.	Optional	63.0

Unified Promotion Template Input

Input representation of the unified promotion template.

JSON example for POST and PUT

```

{
  "rules": [
    {
      "templateName": "BuyXToGetRewards",
      "ruleName": "123",
      "priority": 1,
      "eventConfiguration": [
        {
          "allOf": [
            {
              "anyOf": {
                "purchaseType": "Product",
                "applicablePurchaseList": [
                  {
                    "name": "Veg Burger"
                  }
                ],
                "purchaseCriteria": {
                  "type": "Quantity",
                  "value": 1
                }
              }
            }
          ]
        }
      ],
      "rewardConfiguration": [
        {
          "type": "ProvideDiscount",
          "rewardDetails": {
            "discountValue": "100",
            "discountType": "PercentageOff"
          },
          "targetAudience": {
            "audienceType": "LoyaltyTierMembers",
            "audienceDetails": {
              "tierGroup": {
                "name": "Default"
              },
              "tier": {
                "name": "Silver"
              }
            }
          }
        },
        {
          "type": "IssueVoucher",
          "rewardDetails": {
            "voucherDefinition": {
              "name": "GET 10% OFF"
            },
            "voucherExpirationDate": "2020-10-10"
          }
        }
      ]
    }
  ]
}

```

```

    },
    "targetAudience": {
      "audienceType": "AllLoyaltyMembers"
    }
  }
]
}
]
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
event Configuration	Map<String, Object>[]	List of criteria that decides when customers are eligible for the rewards of the promotion rule.	Required	60.0
journalSubType	String	ID of the journal sub type.	Required when journalType is used	60.0
journalSub TypeName	String	Name of the journal sub type.	Required when journalType is used	64.0
journalType	String	Name of the journal sub type.	Required for loyalty accrual or redemption templates	60.0
priority	Integer	Priority number of the rule in the promotion.	Required	60.0
promotionLimit	Integer	Milestone limit of the promotion.	Optional	63.0
reward Configuration	Map<String, Object>[]	List of rewards that members get when their activity meets the rule's eligibility criteria.	Required	60.0
ruleName	String	Name of the promotion rule.	Required	60.0
templateName	String	Name of the promotion template that's used in a promotion rule.	Required	60.0

Unlink Program Member Partner Input

Input details to unlink a program member and partner.

JSON example

Sample request to unlink the program member and partner.

```

{
  "linkNumber": "00000029",
  "externalId": "U10001",

```

```

    "unlinkReasonType": "USER_REQUEST",
    "comments": "Member requested to unlink the partner account."
  }

```

Properties

Name	Type	Description	Required or Optional	Available Version
comments	String	Description for the unlink reason.	Optional	66.0
externalId	String	Partner membership number of the loyalty program member.	Optional	66.0
linkNumber	String	Unique identifier for the program member and partner linkage.	Optional	66.0
unlinkReasonType	String	Reason to unlink a loyalty program member and a partner.	Optional	66.0

Promotion Reward Application Input

The details required to create transaction journals. The request includes a list of transaction journals, which are to be created. It includes the list of promotions applied along with the rewards. If any of the rewards aren't applied, then everything is rolled back. No partial rewards are applied.

JSON example

```

{
  "shouldCheckCouponUsageLimit": true,
  "journalCreationMode": "CreateTransactionJournal",
  "transactionJournalList": [
    {
      "ActivityDate": "2025-10-08T12:45:19Z",
      "JournalTypeId": "01ET10000004CQp",
      "ReferredPartyId": "003SB000000U84F0YAJ",
      "Quantity": "1",
      "TransactionAmount": "5000",
      "CurrencyIsoCode": "USD",
      "Status": "Processed",
      "appliedPromotions": [
        {
          "promotionId": "0c8SB000000HASnYAO",
          "promotionCode": "GET100FF",
          "couponCode": "COUPON100",
          "rewards": [
            {
              "rewardType": "ProvideDiscount",
              "discountAmount": 15
            },
            {
              "rewardType": "IssueVoucher",
              "voucherDefinitionName": "$100 off",
              "voucherCode": "SALE-4F8G-RT2P",
            }
          ]
        }
      ]
    }
  ]
}

```

```
        "voucherEffectiveDate": "2025-10-09",
        "voucherExpirationDate": "2025-10-10",
        "notes": "discount voucher"
      },
      {
        "rewardType": "AssignGame",
        "gameDefinitionName": "SpintheWheelGame",
        "gameExpirationDate": "2025-10-10"
      }
    ]
  }
]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
journalCreationMode	String	Specifies whether to create a transaction journal record and, if so, where to store the data.	Optional	66.0
shouldCheck CouponUsageLimit	Boolean	Indicates whether the API must execute the Coupon Usage Increase API to verify the coupon usage count and increase the redemption usage count of the specified coupon code (<code>true</code>) or not (<code>false</code>).	Optional	65.0
transaction JournalList	List<Map<String, Object>>	The list of transaction journals to create.	Required	65.0

Response Bodies

Here’s a list of Loyalty Management API response bodies.

[Additional Attributes](#)

Output representation of the additional details of a promotion.

[Additional Transaction Journal Attributes Output](#)

Output representation of the additional transaction journal fields.

[Applied Cart Line Promotion Details](#)

Specifies the list of cart line reward details.

[Applied Cart Promotion Details](#)

The details of the cart-level promotions applied to the cart.

[Badge Output](#)

Output representation of the badge assigned to a member.

[Cart](#)

The details of a cart and its line items for which to evaluate and execute promotions.

[Cart Details](#)

The details of the cart and its line items, including initial amount, adjusted price, and details of all applied discounts and promotions.

[Cart Line Details](#)

The details of the initial amount, adjusted price, and details of all applied discounts and promotions to cart line items within the cart.

[Club Member Benefits](#)

Represents the output for Club Member Benefits.

[Club Member Details](#)

Represents the output for Club Member Profile API.

[Club Member Enrollment Details](#)

Represents the output for Club Member Enrollment API.

[Club Member Promotions](#)

Represents the output for Club Member Promotions.

[Club Membership Renewal](#)

Represents the output for Club Membership Renewal API.

[Corporate Member Enrollment Output](#)

Output representation of the corporate member enrollment create request.

[Coupon Usage](#)

The details of the coupons for which usage has been increased or decreased.

[Coupon Usage Result](#)

Represents the response for coupon usage increase and decrease POST method.

[Clone Promotion](#)

Output representation of clone promotion request.

[Credit Points Action Output](#)

Output representation of the credit points action type.

[Debit Points Action Output](#)

Output representation of the debit points action type.

[Discount Details](#)

The details of the discount applied to a customer's cart.

[Eligible Promotions List](#)

Output representation of a list of eligible promotions.

[Eligible Loyalty Tiers Output](#)

Output representation of the eligible loyalty tiers and tier groups of a promotion.

[Events Detail Output](#)

Output representation of the customer event eligible for a promotion.

[Expression](#)

Output representation of the expression.

[Expression Value](#)

Output representation of the expression value.

[Game Assignment Output](#)

Output representation of the details of a game assigned to a member.

[Game Details](#)

Output representation of the details of a game definition.

[Game Details \(Promotion\)](#)

The details of the game that's assigned to the customer.

[Game Result](#)

Output representation of the game reward that a participant has won.

[Game Reward Details](#)

Output representation of the details of the game reward that a participant has won.

[Game Rewards List](#)

Output representation of the details of a possible game reward.

[Games List](#)

Output representation of the list of games assigned to a member.

[Issue Voucher Action Output](#)

Output representation of the issue voucher action type.

[Individual Member Enrollment Output](#)

Output representation of the individual member enrollment create request.

[Link Program Member Partner](#)

Output representation of the program member and partner linkage request.

[Loyalty Account Output](#)

Represents the details of the loyalty program member's associated account record.

[Loyalty Engine Realtime Output](#)

Output representation of the loyalty program process result.

[Loyalty Contact Output](#)

Represents the details of the loyalty program member's associated contact record.

[Loyalty Process Output](#)

Output representation of the loyalty process response.

[Loyalty Program Process Rule](#)

Output representation of the loyalty program process rules.

[Loyalty Program Process Rule Step](#)

Output representation of the loyalty program process rule step.

[Loyalty Program Process Rule Action](#)

Output representation of the loyalty program process rule action.

[Loyalty Program Process Rule Action Parameter](#)

Output representation of the loyalty program process rule action parameter.

[Loyalty Program Process Rule Condition](#)

Output representation of the loyalty program process rule condition.

[Loyalty Program Process Rule Condition Filter Criteria](#)

Output representation of the loyalty program process rule condition filter criteria.

[Loyalty Program Process Rule Condition Group](#)

Output representation of the loyalty program process rule condition group.

[Loyalty View Transaction Journal History Output](#)

Output representation of the view transaction history request.

[Member Benefits Output](#)

Output representation of all benefits available for a member.

[Member Benefit Output](#)

Output representation of an individual benefit available for a member.

[Member Currency Output](#)

Represents the details of the loyalty program member's associated currencies.

[Member Currency Change](#)

Output representation of the member currency change.

[Member Engagement Trail Output](#)

Output representation of member engagement trail API.

[Member Engagement Attribute](#)

Output representation of Member Engagement Attribute.

[Member Points Balance](#)

Output representation of the member currency result.

[Member Profile Output](#)

Output representation of the details of a loyalty program member.

[Member Currency](#)

Output representation of the member currency details.

[Member Tier Output](#)

Represents the details of the loyalty program member's associated tiers.

[Member Points Balance Change](#)

Output representation of the points balance change for a member.

[Name And Id Output](#)

Output representation of an object's name and ID.

[Participant Game Rewards List](#)

Output representation of the participant game reward object.

[Process Result](#)

Output representation of results of the processed Transaction Journals records.

[Points Change Output](#)

Output representation of the points debited from or credited to the member's points balance by the transaction journal.

[Points Summary](#)

Output representation of the point summary.

[Product Details](#)

The details of a product associated with the recommended promotion.

[Promotion Configuration Output](#)

Output representation of the promotion's details and its configuration.

[Promotion Coupon Availability](#)

Output representation of the promotion coupon availability.

[Promotion Coupon](#)

Output representation of the promotion coupon.

[Promotion Details](#)

Output representation of the details of an eligible promotion.

[Promotion Eligibility Output](#)

Output representation of the promotion eligibility criteria.

[Promotion Eligible Channel Output](#)

Output representation of the eligible channel for a promotion.

[Promotion Eligible Enrollment Period Output](#)

Output representation of the enrollment eligibility details of a promotion.

[Promotion Details](#)

Output representation of the details of an eligible promotion.

[Promotion Limit](#)

Output representation of the promotion limit of an eligible promotion.

[Promotion Limit Output](#)

Output representation of promotion limits.

[Promotion](#)

Output representation of the promotion with rule configuration.

[Promotion Recommendations](#)

The count and list of promotions recommended for the member or customer.

[Promotion Reward Application](#)

Output representation details of the transaction journals that are created and processed.

[Promotion Reward Details](#)

Output representation of the details of the reward offered by an eligible promotion rule.

[Promotion Rule](#)

Output representation of the details of a promotion rule.

[Promotion Rules List](#)

Output representation of an eligible promotion rule.

[Rule Action](#)

Output representation of the details of a promotion rule action.

[Rule Action Parameter](#)

Output representation of the details of a promotion rule action parameter.

[Rule Condition](#)

Output representation of the details of a promotion rule condition.

[Rule Filter Criteria](#)

Output representation of the details of a promotion rule filter criteria.

[Rule Library](#)

The details of the rule library associated with the promotion.

[Rule Parameter](#)

Output representation of the details of a promotion rule parameter.

[Recommendation Details](#)

The rank and details of the recommended promotions.

[Redeem Voucher Output](#)

Represents the response for the redeem voucher POST method.

[Summary](#)

Output representation of the execution summary of the transaction journals.

[Template Configuration](#)

Output representation of the template configuration.

[Template List Configuration](#)

The output representation of the list template configuration

[Transaction Journal Details](#)

The details of the transaction journal that's created.

[Transaction Details Output](#)

Output representation of the transaction journal ledger records.

[Transaction Journal Output](#)

Output representation of the transaction journals.

[Transaction Journal Map Output](#)

Output representation of the loyalty program process simulation result.

[Transaction Journal Result](#)

Output representation of the transaction journals result.

[Transaction Journal Process Result](#)

Output representation of the Transaction Journal result.

[Transaction Ledgers Output](#)

Output representation of the transaction journals and the ledgers associated with the specified loyalty program member.

[Update Cumulative Usage Action Output](#)

Output representation of the update cumulative usage action type.

[Unified Promotions](#)

Output representation of the promotion details.

[Unified Promotion Templates](#)

The output representation of the request to create a promotion.

[Unlink Program Member Partner](#)

Output representation of the program member and partner unlink request.

[Voucher](#)

Output representation of the details of a voucher.

[Voucher Details](#)

The details of the voucher that's issued to the customer.

[Vouchers List](#)

Output representation of the list of vouchers associated with the loyalty program member.

[Voucher Result](#)

Output representation of the voucher result.

[Voucher Output](#)

Output representation of the details about a voucher.

Additional Attributes

Output representation of the additional details of a promotion.

Property Name	Type	Description	Filter Group and Version	Available Version
attributes	Map<String,String>	Additional fields that provide more details of the promotion.	Small, 60.0	60.0

Additional Transaction Journal Attributes Output

Output representation of the additional transaction journal fields.

Property Name	Type	Description	Filter Group and Version	Available Version
fieldName	String	The name of the transaction journal field.	Small, 58.0	58.0
value	String	The value that's stored in the transaction journal field.	Small, 58.0	58.0

Applied Cart Line Promotion Details

Specifies the list of cart line reward details.

JSON example

This example shows a sample response.

```
"appliedCartLinePromotionDetails": [
  {
    "cartLineItemAmountAfterDiscount": 95,
    "cartLineItemAmountBeforeDiscount": 100,
    "cartLineItemDiscountAmount": 5,
    "couponCode": "PIZZA5",
    "currencyIsoCode": "USD",
    "displayName": "5 % off on Pizza",
    "isAutomatic": false,
    "promotionId": "0c8xx00000005NtAAI"
  }
],
"cartLineItemAmountAfterDiscount": 66.36,
"cartLineItemAmountBeforeDiscount": 100,
"cartLineItemId": "abcd1",
"cartLineItemQuantity": 2,
```

```

    "cartLineProduct": "Pizza",
    "totalCartLineItemDiscountAmount": 33.64,
    "totalCartLinePromotionsDiscountAmount": 5,
    "totalCartPromotionsDistributedDiscountAmount": 28.64
  },
  {
    "appliedCartLinePromotionDetails": [
      {
        "cartLineItemAmountAfterDiscount": 9,
        "cartLineItemAmountBeforeDiscount": 10,
        "cartLineItemDiscountAmount": 1,
        "currencyIsoCode": "USD",
        "displayName": "Get 10% off on Burger + Voucher",
        "isAutomatic": true,
        "promotionId": "0c8xx00000004y5AAA"
      },
      {
        "cartLineItemAmountAfterDiscount": 4.5,
        "cartLineItemAmountBeforeDiscount": 9,
        "cartLineItemDiscountAmount": 4.5,
        "currencyIsoCode": "USD",
        "displayName": "Buy 1 Pizza, Get 1 Burger Free",
        "isAutomatic": true,
        "promotionId": "0c8xx00000006TdAAI"
      }
    ]
  }
]

```

Property Name	Type	Description	Filter Group and Version	Available Version
~~cartLineItemAmountAfterDiscount~~	Double	The updated cart amount after the current line or cross line promotion discount.	Big, 65.0	65.0
~~cartLineItemAmountBeforeDiscount~~	Double	The cart line item amount before the current line or cross line promotion is applied.	Big, 65.0	65.0
~~cartLineItemDiscountAmount~~	Double	Amount of line or cross line promotion discounts applied to the line items.	Big, 65.0	65.0
couponCode	String	Coupon code of the applied line or cross line promotion.	Big, 65.0	65.0
currencyIsoCode	String	Three-letter ISO 4217 code of the monetary currency associated with the applied line or cross line promotion.	Big, 65.0	65.0
displayName	String	Display name of the applied line or cross line promotion.	Big, 65.0	65.0
isAutomatic	Boolean	Indicates whether the promotion is automatically applied for eligible carts (true) or not (false).	Big, 65.0	65.0
promotionCode	String	Code of the applied line or cross line promotion.	Big, 65.0	65.0

Property Name	Type	Description	Filter Group and Version	Available Version
promotionId	String	The ID of the applied line or cross line promotion.	Big, 65.0	65.0

Applied Cart Promotion Details

The details of the cart-level promotions applied to the cart.

JSON example

This example shows a sample response.

```
"appliedCartPromotionDetails": [  
  {  
    "cartAmountAfterDiscount": 99.5,  
    "cartAmountBeforeDiscount": 99.5,  
    "cartLevelDiscountAmount": 0,  
    "displayName": "Get 10% off on Burger + Voucher",  
    "isAutomatic": true,  
    "nonPricingRewards": {  
      "voucherRewards": [  
        {  
          "voucherDefinitionId": "2Xaxx00000000XtCAI",  
          "voucherDefinitionName": "Voucher1"  
        }  
      ]  
    },  
    "promotionId": "0c8xx00000004y5AAA"  
  },  
  {  
    "cartAmountAfterDiscount": 89.5,  
    "cartAmountBeforeDiscount": 99.5,  
    "cartLevelDiscountAmount": 10,  
    "displayName": "10% off on Cart",  
    "isAutomatic": true,  
    "nonPricingRewards": {},  
    "promotionId": "0c8xx00000005R7AAI"  
  },  
  {  
    "cartAmountAfterDiscount": 99.5,  
    "cartAmountBeforeDiscount": 99.5,  
    "cartLevelDiscountAmount": 0,  
    "displayName": "Cart Voucher",  
    "isAutomatic": true,  
    "nonPricingRewards": {  
      "voucherRewards": [  
        {  
          "voucherDefinitionId": "2Xaxx00000000XtCAI",  
          "voucherDefinitionName": "Voucher1"  
        }  
      ]  
    }  
  }  
],
```

```

    "promotionId": "0c8xx000000060bAAA"
  },
  {
    "cartAmountAfterDiscount": 79.5,
    "cartAmountBeforeDiscount": 99.5,
    "cartLevelDiscountAmount": 20,
    "isAutomatic": true,
    "nonPricingRewards": {},
    "promotionId": "0c8xx00000006wfAAA"
  },
  {
    "cartAmountAfterDiscount": 99.5,
    "cartAmountBeforeDiscount": 99.5,
    "cartLevelDiscountAmount": 0,
    "displayName": "LOY promotion",
    "isAutomatic": true,
    "nonPricingRewards": {
      "badgeRewards": [
        {
          "loyaltyProgramBadgeId": "0w8xx0000000001AAA",
          "badgeValidityEndDate": "2025-08-31T00:00:00.000Z",
          "loyaltyProgramBadgeName": "badge1"
        }
      ],
      "fixedPointsRewards": [
        {
          "loyaltyProgramCurrencyName": "NQP",
          "loyaltyProgramCurrencyId": "0lcxx00000001FRAAY",
          "points": 100
        }
      ]
    }
  },
  "promotionId": "0c8xx0000000EpNAAU"
},
{
  "cartAmountAfterDiscount": 99.5,
  "cartAmountBeforeDiscount": 99.5,
  "cartLevelDiscountAmount": 0,
  "isAutomatic": true,
  "nonPricingRewards": {
    "multiplierPointsRewards": [
      {
        "loyaltyProgramCurrencyName": "NQP",
        "pointsExpirationDate": "2025-09-07T00:00:00.000Z",
        "transactionJournalField": "TransactionJournal.TransactionAmount",
        "loyaltyProgramCurrencyId": "0lcxx00000001FRAAY",
        "multiplierValue": 44
      }
    ],
    "gameRewards": [
      {
        "gameDefinitionId": "3GDxx0000004C92GAE",
        "gameType": "Scratchcard",
        "gameDefinitionName": "GD1"
      }
    ]
  }
}

```

```

    }
  ]
},
"promotionId": "0c8xx0000000Fv7AAE"
}
]

```

Property Name	Type	Description	Filter Group and Version	Available Version
cartAmountAfterDiscount	Double	The updated cart amount after the current promotion is applied.	Big, 65.0	65.0
cartAmountBeforeDiscount	Double	Cart amount before the current promotion is applied.	Big, 65.0	65.0
cartLevelDiscountAmount	Double	Total amount of discounts from promotions.	Big, 65.0	65.0
couponCode	String	Coupon code of the applied promotion.	Big, 65.0	65.0
currencyIsoCode	String	Three-letter ISO 4217 code of the monetary currency associated with the applied promotion.	Big, 65.0	65.0
displayName	String	Display name of the applied promotion.	Big, 65.0	65.0
isAutomatic	Boolean	Indicates whether the promotion is automatically applied for eligible carts (<code>true</code>) or not (<code>false</code>).	Big, 65.0	65.0
nonPricingRewards	Map<String, Object>	The non-pricing rewards issued to customers as a part of the applied promotions. Possible rewards are: • <code>badgeRewards</code>: List of rewards for members in the form of badges. • <code>fixedPointsRewards</code>: List of rewards for members in the form of fixed number of loyalty points. • <code>gameRewards</code>: List of rewards in the form of games. • <code>multiplierPointsRewards</code>: List of rewards for members in the form of loyalty points that are calculated based on a multiplier and a transaction journal field. • <code>voucherRewards</code>: List of rewards in the form of vouchers.	Big, 65.0	65.0
promotionCode	String	Code of the applied promotion.	Big, 65.0	65.0
promotionId	String	The ID of the applied promotion.	Big, 65.0	65.0

Badge Output

Output representation of the badge assigned to a member.

Property Name	Type	Description	Filter Group and Version	Available Version
badgeValidityEndDate	String	Date from which the badge is no longer assigned to the member.	Small, 60.0	60.0
loyaltyProgramBadgeName	String	Name of the assigned badge.	Small, 60.0	60.0
memberBadgeStatus	String	Status of the assigned badge.	Small, 60.0	60.0
reason	String	Reason the badge was assigned to the member.	Small, 60.0	60.0

Cart

The details of a cart and its line items for which to evaluate and execute promotions.

JSON example

This example shows a sample response.

```
{
  "cart": {
    "cartDetails": [
      {
        "adjustedCartAmount": 69.5,
        "appliedCartPromotionDetails": [
          {
            "cartAmountAfterDiscount": 99.5,
            "cartAmountBeforeDiscount": 99.5,
            "cartLevelDiscountAmount": 0,
            "displayName": "Get 10% off on Burger + Voucher",
            "isAutomatic": true,
            "nonPricingRewards": {
              "voucherRewards": [
                {
                  "voucherDefinitionId": "2Xaxx00000000XtCAI",
                  "voucherDefinitionName": "Voucher1"
                }
              ]
            },
            "promotionId": "0c8xx00000004y5AAA"
          },
          {
            "cartAmountAfterDiscount": 89.5,
            "cartAmountBeforeDiscount": 99.5,
            "cartLevelDiscountAmount": 10,
            "displayName": "10% off on Cart",
            "isAutomatic": true,
            "nonPricingRewards": {},
            "promotionId": "0c8xx00000005R7AAI"
          }
        ]
      }
    ]
  }
}
```

```

    },
    {
      "cartAmountAfterDiscount": 99.5,
      "cartAmountBeforeDiscount": 99.5,
      "cartLevelDiscountAmount": 0,
      "displayName": "Cart Voucher",
      "isAutomatic": true,
      "nonPricingRewards": {
        "voucherRewards": [
          {
            "voucherDefinitionId": "2Xaxx00000000XtCAI",
            "voucherDefinitionName": "Voucher1"
          }
        ]
      },
      "promotionId": "0c8xx000000060bAAA"
    },
    {
      "cartAmountAfterDiscount": 79.5,
      "cartAmountBeforeDiscount": 99.5,
      "cartLevelDiscountAmount": 20,
      "isAutomatic": true,
      "nonPricingRewards": {},
      "promotionId": "0c8xx00000006wfAAA"
    },
    {
      "cartAmountAfterDiscount": 99.5,
      "cartAmountBeforeDiscount": 99.5,
      "cartLevelDiscountAmount": 0,
      "displayName": "LOY promotion",
      "isAutomatic": true,
      "nonPricingRewards": {
        "badgeRewards": [
          {
            "loyaltyProgramBadgeId": "0w8xx0000000001AAA",
            "badgeValidityEndDate": "2025-08-31T00:00:00.000Z",
            "loyaltyProgramBadgeName": "badge1"
          }
        ],
        "fixedPointsRewards": [
          {
            "loyaltyProgramCurrencyName": "NQP",
            "loyaltyProgramCurrencyId": "0lcxx00000001FRAAY",
            "points": 100
          }
        ]
      },
      "promotionId": "0c8xx0000000EpNAAU"
    },
    {
      "cartAmountAfterDiscount": 99.5,
      "cartAmountBeforeDiscount": 99.5,
      "cartLevelDiscountAmount": 0,
      "isAutomatic": true,

```



```

    "nonPricingRewards": {
      "multiplierPointsRewards": [
        {
          "loyaltyProgramCurrencyName": "NQP",
          "pointsExpirationDate": "2025-09-07T00:00:00.000Z",
          "transactionJournalField": "TransactionJournal.TransactionAmount",
          "loyaltyProgramCurrencyId": "01cxx00000001FRAAY",
          "multiplierValue": 44
        }
      ],
      "gameRewards": [
        {
          "gameDefinitionId": "3GDxx0000004C92GAE",
          "gameType": "Scratchcard",
          "gameDefinitionName": "GD1"
        }
      ]
    },
    "promotionId": "0c8xx0000000Fv7AAE"
  }
],
"cartHeaderId": "01ddf0000003kfMAAW",
"cartLineDetails": [
  {
    "appliedCartLinePromotionDetails": [
      {
        "cartLineItemAmountAfterDiscount": 95,
        "cartLineItemAmountBeforeDiscount": 100,
        "cartLineItemDiscountAmount": 5,
        "couponCode": "PIZZA5",
        "currencyIsoCode": "USD",
        "displayName": "5 % off on Pizza",
        "isAutomatic": false,
        "promotionId": "0c8xx00000005NtAAI"
      }
    ],
    "cartLineItemAmountAfterDiscount": 66.36,
    "cartLineItemAmountBeforeDiscount": 100,
    "cartLineItemId": "abcd1",
    "cartLineItemQuantity": 2,
    "cartLineProduct": "Pizza",
    "totalCartLineItemDiscountAmount": 33.64,
    "totalCartLinePromotionsDiscountAmount": 5,
    "totalCartPromotionsDistributedDiscountAmount": 28.64
  },
  {
    "appliedCartLinePromotionDetails": [
      {
        "cartLineItemAmountAfterDiscount": 9,
        "cartLineItemAmountBeforeDiscount": 10,
        "cartLineItemDiscountAmount": 1,
        "currencyIsoCode": "USD",
        "displayName": "Get 10% off on Burger + Voucher",
        "isAutomatic": true,

```

```
        "promotionId": "0c8xx00000004y5AAA"
      },
      {
        "cartLineItemAmountAfterDiscount": 4.5,
        "cartLineItemAmountBeforeDiscount": 9,
        "cartLineItemDiscountAmount": 4.5,
        "currencyIsoCode": "USD",
        "displayName": "Buy 1 Pizza, Get 1 Burger Free",
        "isAutomatic": true,
        "promotionId": "0c8xx00000006TdAAI"
      }
    ],
    "cartLineItemAmountAfterDiscount": 3.14,
    "cartLineItemAmountBeforeDiscount": 10,
    "cartLineItemId": "abcd2",
    "cartLineItemQuantity": 2,
    "cartLineProduct": "Burger",
    "totalCartLineItemDiscountAmount": 6.86,
    "totalCartLinePromotionsDiscountAmount": 5.5,
    "totalCartPromotionsDistributedDiscountAmount": 1.36
  }
],
"initialCartAmount": 110,
"totalCartLevelDiscountAmount": 30,
"totalDiscountAmount": 40.5,
"totalLineLevelDiscountAmount": 10.5
}
]
}
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
cartDetails	Cart Details[]	Details of the cart and its line items, including initial amount, adjusted price, and details of all applied discounts and promotions.	Big, 65.0	65.0

Cart Details

The details of the cart and its line items, including initial amount, adjusted price, and details of all applied discounts and promotions.

JSON example

This example shows a sample response.

```
{
  "cartDetails": [
    {
      "adjustedCartAmount": 69.5,
      "appliedCartPromotionDetails": [
        {
          "cartAmountAfterDiscount": 99.5,
```

```

    "cartAmountBeforeDiscount": 99.5,
    "cartLevelDiscountAmount": 0,
    "displayName": "Get 10% off on Burger + Voucher",
    "isAutomatic": true,
    "nonPricingRewards": {
      "voucherRewards": [
        {
          "voucherDefinitionId": "2Xaxx00000000XtCAI",
          "voucherDefinitionName": "Voucher1"
        }
      ]
    },
    "promotionId": "0c8xx00000004y5AAA"
  },
  {
    "cartAmountAfterDiscount": 89.5,
    "cartAmountBeforeDiscount": 99.5,
    "cartLevelDiscountAmount": 10,
    "displayName": "10% off on Cart",
    "isAutomatic": true,
    "nonPricingRewards": {},
    "promotionId": "0c8xx00000005R7AAI"
  },
  {
    "cartAmountAfterDiscount": 99.5,
    "cartAmountBeforeDiscount": 99.5,
    "cartLevelDiscountAmount": 0,
    "displayName": "Cart Voucher",
    "isAutomatic": true,
    "nonPricingRewards": {
      "voucherRewards": [
        {
          "voucherDefinitionId": "2Xaxx00000000XtCAI",
          "voucherDefinitionName": "Voucher1"
        }
      ]
    },
    "promotionId": "0c8xx000000060bAAA"
  },
  {
    "cartAmountAfterDiscount": 79.5,
    "cartAmountBeforeDiscount": 99.5,
    "cartLevelDiscountAmount": 20,
    "isAutomatic": true,
    "nonPricingRewards": {},
    "promotionId": "0c8xx00000006wfAAA"
  },
  {
    "cartAmountAfterDiscount": 99.5,
    "cartAmountBeforeDiscount": 99.5,
    "cartLevelDiscountAmount": 0,
    "displayName": "LOY promotion",
    "isAutomatic": true,
    "nonPricingRewards": {

```

```

        "badgeRewards": [
          {
            "loyaltyProgramBadgeId": "0w8xx0000000001AAA",
            "badgeValidityEndDate": "2025-08-31T00:00:00.000Z",
            "loyaltyProgramBadgeName": "badge1"
          }
        ],
        "fixedPointsRewards": [
          {
            "loyaltyProgramCurrencyName": "NQP",
            "loyaltyProgramCurrencyId": "0lcxx00000001FRAAY",
            "points": 100
          }
        ]
      },
      "promotionId": "0c8xx0000000EpNAAU"
    },
    {
      "cartAmountAfterDiscount": 99.5,
      "cartAmountBeforeDiscount": 99.5,
      "cartLevelDiscountAmount": 0,
      "isAutomatic": true,
      "nonPricingRewards": {
        "multiplierPointsRewards": [
          {
            "loyaltyProgramCurrencyName": "NQP",
            "pointsExpirationDate": "2025-09-07T00:00:00.000Z",
            "transactionJournalField": "TransactionJournal.TransactionAmount",
            "loyaltyProgramCurrencyId": "0lcxx00000001FRAAY",
            "multiplierValue": 44
          }
        ],
        "gameRewards": [
          {
            "gameDefinitionId": "3GDxx00000004C92GAE",
            "gameType": "Scratchcard",
            "gameDefinitionName": "GD1"
          }
        ]
      },
      "promotionId": "0c8xx0000000Fv7AAE"
    }
  ],
  "cartHeaderId": "01ddf00000003kfMAAW",
  "cartLineDetails": [
    {
      "appliedCartLinePromotionDetails": [
        {
          "cartLineItemAmountAfterDiscount": 95,
          "cartLineItemAmountBeforeDiscount": 100,
          "cartLineItemDiscountAmount": 5,
          "couponCode": "PIZZA5",
          "currencyIsoCode": "USD",
          "displayName": "5 % off on Pizza",

```

```

        "isAutomatic": false,
        "promotionId": "0c8xx00000005NtAAI"
    }
],
"cartLineItemAmountAfterDiscount": 66.36,
"cartLineItemAmountBeforeDiscount": 100,
"cartLineItemId": "abcd1",
"cartLineItemQuantity": 2,
"cartLineProduct": "Pizza",
"totalCartLineItemDiscountAmount": 33.64,
"totalCartLinePromotionsDiscountAmount": 5,
"totalCartPromotionsDistributedDiscountAmount": 28.64
},
{
    "appliedCartLinePromotionDetails": [
        {
            "cartLineItemAmountAfterDiscount": 9,
            "cartLineItemAmountBeforeDiscount": 10,
            "cartLineItemDiscountAmount": 1,
            "currencyIsoCode": "USD",
            "displayName": "Get 10% off on Burger + Voucher",
            "isAutomatic": true,
            "promotionId": "0c8xx00000004y5AAA"
        },
        {
            "cartLineItemAmountAfterDiscount": 4.5,
            "cartLineItemAmountBeforeDiscount": 9,
            "cartLineItemDiscountAmount": 4.5,
            "currencyIsoCode": "USD",
            "displayName": "Buy 1 Pizza, Get 1 Burger Free",
            "isAutomatic": true,
            "promotionId": "0c8xx00000006TdAAI"
        }
    ],
    "cartLineItemAmountAfterDiscount": 3.14,
    "cartLineItemAmountBeforeDiscount": 10,
    "cartLineItemId": "abcd2",
    "cartLineItemQuantity": 2,
    "cartLineProduct": "Burger",
    "totalCartLineItemDiscountAmount": 6.86,
    "totalCartLinePromotionsDiscountAmount": 5.5,
    "totalCartPromotionsDistributedDiscountAmount": 1.36
}
],
"initialCartAmount": 110,
"totalCartLevelDiscountAmount": 30,
"totalDiscountAmount": 40.5,
"totalLineLevelDiscountAmount": 10.5
}
]
}
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
adjustedCartAmount	Double	The final total amount of the cart after all promotions are applied.	Big, 65.0	65.0
appliedCartPromotionDetails	Applied Cart Promotion Details[]	Details of the cart-level promotions and non-pricing rewards of the line and cross line level promotions applied to the cart.	Big, 65.0	65.0
cartHeaderId	String	The ID of the cart on which promotions are applied.	Big, 65.0	65.0
cartLineDetails	Cart Line Details[]	Details of the initial amount, adjusted price, and details of all applied discounts and promotions to cart line items within the cart.	Big, 65.0	65.0
initialCartAmount	Double	Sum of the prices of all items in the cart before any promotions are applied.	Big, 65.0	65.0
totalCartLevelDiscountAmount	Double	Total amount of discounts applied to the the cart from cart-level promotions.	Big, 65.0	65.0
totalDiscountAmount	Double	Sum of all discounts applied to the the cart, including line-level, crossline-level and cart-level discounts.	Big, 65.0	65.0
totalLineLevelDiscountAmount	Double	Total amount of discounts applied to line items from line-level and crossline-level promotions.	Big, 65.0	65.0

Cart Line Details

The details of the initial amount, adjusted price, and details of all applied discounts and promotions to cart line items within the cart.

JSON example

This example shows a sample response.

```

"cartLineDetails": [
  {
    "appliedCartLinePromotionDetails": [
      {
        "cartLineItemAmountAfterDiscount": 95,
        "cartLineItemAmountBeforeDiscount": 100,
        "cartLineItemDiscountAmount": 5,
        "couponCode": "PIZZA5",
        "currencyIsoCode": "USD",
        "displayName": "5 % off on Pizza",
        "isAutomatic": false,
        "promotionId": "0c8xx00000005NtAAI"
      }
    ],
    "cartLineItemAmountAfterDiscount": 66.36,

```

```

    "cartLineItemAmountBeforeDiscount": 100,
    "cartLineItemId": "abcd1",
    "cartLineItemQuantity": 2,
    "cartLineProduct": "Pizza",
    "totalCartLineItemDiscountAmount": 33.64,
    "totalCartLinePromotionsDiscountAmount": 5,
    "totalCartPromotionsDistributedDiscountAmount": 28.64
  },
  {
    "appliedCartLinePromotionDetails": [
      {
        "cartLineItemAmountAfterDiscount": 9,
        "cartLineItemAmountBeforeDiscount": 10,
        "cartLineItemDiscountAmount": 1,
        "currencyIsoCode": "USD",
        "displayName": "Get 10% off on Burger + Voucher",
        "isAutomatic": true,
        "promotionId": "0c8xx00000004y5AAA"
      },
      {
        "cartLineItemAmountAfterDiscount": 4.5,
        "cartLineItemAmountBeforeDiscount": 9,
        "cartLineItemDiscountAmount": 4.5,
        "currencyIsoCode": "USD",
        "displayName": "Buy 1 Pizza, Get 1 Burger Free",
        "isAutomatic": true,
        "promotionId": "0c8xx00000006TdAAI"
      }
    ]
  }
]

```

Property Name	Type	Description	Filter Group and Version	Available Version
appliedCartLinePromotionDetails	Applied Cart Line Promotion Details[]	Details of the line or cross line promotions applied.	Big, 65.0	65.0
cartLineItemAmountAfterDiscount	Double	Updated price of the line items after the current promotions are applied.	Big, 65.0	65.0
cartLineItemAmountBeforeDiscount	Double	Price of the line item before the current promotions are applied.	Big, 65.0	65.0
cartLineItemId	String	The ID of cart line item on which promotions are applied.	Big, 65.0	65.0
cartLineItemQuantity	Integer	Quantity of the cart line item on which promotions are applied.	Big, 65.0	65.0
cartLineProduct	String	Product on which promotions are applied.	Big, 65.0	65.0
totalCartLineDiscountAmount	Double	Total amount of discounts applied to the line item, including discount from applied line or cross line promotions. and cart line.	Big, 65.0	65.0

Property Name	Type	Description	Filter Group and Version	Available Version
~~totalPromotionalDiscount~~	Double	The distributed amount is only from cart level promotions.		
~~totalLineItemDiscount~~	Double	Total amount of discount from all line or cross line promotions.	Big, 65.0	65.0
~~totalProratedDiscount~~	Double	The prorated amount of the total monetary discount from cart promotions that a discount distribution service applies to an individual line item.	Big, 65.0	65.0

Club Member Benefits

Represents the output for Club Member Benefits.

Property Name	Type	Description	Filter Group and Version	Available Version
benefitId	String	The ID of the benefit associated with the club.	Small, 63.0	64.0
benefitName	String	The name of the benefit associated with the club.	Small, 63.0	64.0
benefitTypeId	String	The ID of the benefit type associated with the club.	Small, 63.0	64.0
benefitTypeName	String	The name of the benefit type associated with the club.	Small, 63.0	64.0
description	String	The description of the benefit associated with the club.	Small, 63.0	64.0
isActive	Boolean	Indicates whether the benefit is active (true) or not (false).	Small, 63.0	64.0

Club Member Details

Represents the output for Club Member Profile API.

Property Name	Type	Description	Filter Group and Version	Available Version
benefits	Club Member Benefits on page 520	The list of benefits associated with the club member.	Small, 63.0	63.0
clubId	String	The ID of the club.	Small, 63.0	63.0
clubMembershipId	String	The membership ID of the club member.	Small, 63.0	63.0

Property Name	Type	Description	Filter Group and Version	Available Version
clubMembershipNumber	String	The membership number of the club member.	Small, 63.0	63.0
clubName	String	The name of the club.	Small, 63.0	63.0
membershipEndDate	String	The end date of club membership.	Small, 63.0	63.0
membershipStartDate	String	The start date of club membership.	Small, 63.0	63.0
promotions	Club Member Promotions on page 521	The list of promotions associated with the club member.	Small, 63.0	63.0

Club Member Enrollment Details

Represents the output for Club Member Enrollment API.

JSON Example

```
{
  "clubMembershipNumber": "NAME5",
  "message": null,
  "status": true
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
clubMembershipNumber	String	The membership number of the club member.	Small, 63.0	63.0
message	String	The message explaining why the request failed.	Small, 63.0	63.0
status	Boolean	Indicates whether the request succeeded (true) or failed (false).	Small, 63.0	63.0

Club Member Promotions

Represents the output for Club Member Promotions.

Property Name	Type	Description	Filter Group and Version	Available Version
endDate	String	The end date of the promotion.	Small, 63.0	64.0

Property Name	Type	Description	Filter Group and Version	Available Version
imageUrl	String	The URL of the image associated with the promotion.	Small, 63.0	64.0
loyalty PromotionType	String	The type of loyalty promotion.	Small, 63.0	64.0
member Eligibility Category	String	The eligibility category of the club member for the promotion.	Small, 63.0	64.0
promotion EnrollmentRqr	Boolean	Indicates whether the club members must enroll for the promotion (true) or not (false).	Small, 63.0	64.0
promotionId	String	The ID of the promotion.	Small, 63.0	64.0
promotionName	String	The name of the promotion.	Small, 63.0	64.0
startDate	String	The start date of the promotion.	Small, 63.0	64.0

Club Membership Renewal

Represents the output for Club Membership Renewal API.

JSON example

```
{
  "clubMembershipNumber": "ARC0001",
  "clubMembershipEndDate": "2026-01-01T00:00:00"
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
clubMembership EndDate	String	The end date of the club membership.	Small, 63.0	63.0
club Membership Number	String	The membership number of the club member.	Small, 63.0	63.0
message	String	The message explaining why the request failed.	Small, 63.0	63.0
status	Boolean	Indicates whether the request succeeded (true) or failed (false).	Small, 63.0	63.0

Corporate Member Enrollment Output

Output representation of the corporate member enrollment create request.

JSON example

```

* accountId: *001xx000003GgHX*
* loyaltyProgramMemberId: *01Mxx00000008K9*
* loyaltyProgramName: *program1*
* membershipNumber: *Member6*
* transactionJournals
  * [Item 1]
    * activityDate: *2021-01-01T08:00:00.000Z*
    * journalSubType: *Member Enrollment*
    * journalType: *Accrual*
    * loyaltyProgram: *program1*
    * loyaltyProgramMember: *Member6*
    * referredMember: *null*
    * status: *Pending*
    * transactionJournalId: *01Vxx00000002Ej*
  * [Item 2]
    * activityDate: *2021-01-01T08:00:00.000Z*
    * journalSubType: *Member Referral*
    * journalType: *Accrual*
    * loyaltyProgram: *program1*
    * loyaltyProgramMember: *Member1*
    * referredMember: *Member6*
    * status: *Pending*
    * transactionJournalId: *01Vxx00000002Ek*

```

Property Name	Type	Description	Filter Group and Version	Available Version
accountId	String	The ID of the loyalty program member's business account.	Small, 55.0	55.0
loyaltyProgramMemberId	String	The ID of the loyalty program member record created for the member.	Small, 55.0	55.0
loyaltyProgramName	String	The name of the loyalty program into which the member enrolled.	Small, 55.0	55.0
membershipNumber	String	The membership number of the enrolled member.	Small, 55.0	55.0
transactionJournals	Transaction Journal Output[]	The list of transaction journals that are created.	Small, 55.0	55.0

Coupon Usage

The details of the coupons for which usage has been increased or decreased.

Property Name	Type	Description	Filter Group and Version	Available Version
couponUsageResultList	Coupon Usage Result[]	The details of the coupons for which usage is increased or decreased.	Small, 63.0	63.0

Coupon Usage Result

Represents the response for coupon usage increase and decrease POST method.

Sample Response

```
[
  {
    "couponCode": "QWERTY",
    "isSuccess": true,
    "redemptionLimit": 10,
    "redemptionLimitPerCustomer": 3,
    "remainingRedemptionCount": 6,
    "remainingRedemptionCountForCustomer": 1
  },
  {
    "couponCode": "ABCD",
    "errorMessage": "Coupon code doesn't exist.",
    "isSuccess": false
  }
]
```

Property Name	Type	Description	Filter Group and Version	Available Version
couponCode	String	The code of the coupon for which usage has been increased or decreased.	Small, 62.0	62.0
redemptionLimit	Integer	The total number of times customers can redeem the coupon.	Small, 62.0	62.0
remainingRedemptionCount	Integer	The remaining number of times the coupon can be redeemed.	Small, 62.0	62.0
redemptionLimitPerCustomer	Integer	The total number of times each customer can redeem the coupon.	Small, 62.0	62.0
remainingRedemptionCountForCustomer	Integer	The remaining number of times the specified customer can redeem the coupon.	Small, 62.0	62.0
isSuccess	Boolean	Indicates whether the usage of the coupon was successfully increased or decreased.	Small, 62.0	62.0
errorMessage	String	The error message that contains the details of the why the coupon usage wasn't updated.	Small, 62.0	62.0

Clone Promotion

Output representation of clone promotion request.

JSON Example

```
{
  "promotionId": "01pxx00000000EXAAQ"
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
promotionId	String	The ID of the new promotion.	Small, 61.0	61.0

Credit Points Action Output

Output representation of the credit points action type.

Property Name	Type	Description	Filter Group and Version	Available Version
actionType	String	The type of action in the loyalty program process rule.	Small, 55.0	55.0
changeInEscrowPointsBalance	Double	The change in the escrow points balance of the loyalty program member associated with the transaction journal.	Small, 55.0	55.0
~~changeInPointsBalance~~	Double	The change in the points balance of the loyalty program member associated with the transaction journal.	Small, 55.0	55.0
escrowPointsCreditDate	String	The date on which points that are added to the escrow points balance of the loyalty program member associated with the transaction journal must be rolled over to the points balance of the member.	Small, 55.0	55.0
loyaltyProgramCurrencyName	String	The name of the loyalty program currency for which points have been credited to the loyalty program member associated with the transaction journal.	Small, 55.0	55.0
promotionId	String	The ID of the promotion associated with the rule.	Small, 55.0	55.0
stepName	String	The name of the step in the loyalty program process rule.	Small, 55.0	55.0

Debit Points Action Output

Output representation of the debit points action type.

Property Name	Type	Description	Filter Group and Version	Available Version
actionType	String	The type of action in the loyalty program process rule.	Small, 55.0	55.0
changeInPoints Balance	Double	The change in the points balance of the loyalty program member associated with the transaction journal.	Small, 55.0	55.0
loyaltyProgram CurrencyName	String	The name of the loyalty program currency for which points have been debited from the loyalty program member associated with the transaction journal.	Small, 55.0	55.0
promotionId	String	The ID of the promotion associated with the rule.	Small, 55.0	55.0
stepName	String	The name of the step in the loyalty program process rule.	Small, 55.0	55.0

Discount Details

The details of the discount applied to a customer's cart.

JSON example

This example shows a sample response.

```
"discountsAppliedList": [  
  {  
    "discountType": "ProvideDiscount",  
    "discountAmount": 15  
  }  
]
```

Property Name	Type	Description	Filter Group and Version	Available Version
discountAmount	Double	The discount amount provided to a customer.	Big, 65.0	65.0
discountType	String	The type of discount provided to a customer.	Big, 65.0	65.0

Eligible Promotions List

Output representation of a list of eligible promotions.

JSON example

```
{  
  "promotions": [  
    {
```

```

    "id": "0c85i0000009c6D",
    "promotionCode": "10OFF",
    "displayName": "10% off on all electronics",
    "priority": 1,
    "currencyCode": "USD",
    "coupon": {
      "couponDetails": {
        "couponCode": "COUPON-123",
        "status": "Active",
        "startDateTime": "2024-01-01",
        "endDateTime": null
      },
      "message": "SingleActive"
    },
    "isAutomatic": false,
    "additionalFields": {
      "startDate": "2023-09-10",
      "endDate": "2023-09-20",
      "promotionLocation__c": "HYD"
    },
    "limits": [
      {
        "type": "PerLineItem",
        "limit": 1
      }
    ],
    "rules": [
      {
        "ruleName": "10PercentElectronics",
        "benefits": [
          {
            "type": "Discount",
            "discountLevel": "Cart",
            "discountType": "Amount",
            "discountValue": 10
          },
          {
            "type": "Discount",
            "discountLevel": "CartLineItem",
            "lines": [
              {
                "id": "6c139faa-b749-11ed-afa1-0242ac120002",
                "quantity": 3
              },
              {
                "id": "346bded6-b74a-11ed-afa1-0242ac120002",
                "quantity": 2
              }
            ],
            "discountType": "Amount",
            "discountValue": 10
          }
        ],
        {
          "type": "Voucher",

```

```

        "voucherDefinition": "ELEC10",
        "voucherExpiryDate": "2023-12-31"
    }
  ],
  },
  {
    "ruleName": "5PercentAppliances",
    "benefits": [
      {
        "type": "Discount",
        "discountLevel": "Cart",
        "discountType": "Percent",
        "discountValue": 5
      },
      {
        "type": "FreeProduct",
        "lineItemid": "b749-11ed-afaf",
        "lines": [
          {
            "id": "6c139faa-b749-11ed-afaf-0242ac120002",
            "quantity": 1
          }
        ]
      }
    ]
  }
]
},
{
  "id": "0c85i0000009c6D",
  "promotionCode": "20OFF",
  "displayName": "Get 100 points on all clothing",
  "priority": 2,
  "currencyCode": "USD",
  "additionalFields": {
    "startDate": "2023-09-10"
  },
  "rules": [
    {
      "ruleName": "100PointsClothing",
      "benefits": [
        {
          "type": "Points",
          "points": "10",
          "loyaltyProgramCurrencyName": "NQP"
        },
        {
          "type": "Badge",
          "loyaltyProgramBadgeName": "Ranger"
        }
      ]
    }
  ]
}
}

```



```

    ]
  }

```

Property Name	Type	Description	Filter Group and Version	Available Version
Promotions	Promotion Details []	List of promotions that are eligible for a customer's cart. The list also contains the details of the rewards that each promotion offers to the customer.	Small, 60.0	60.0

Eligible Loyalty Tiers Output

Output representation of the eligible loyalty tiers and tier groups of a promotion.

Property Name	Type	Description	Filter Group and Version	Available Version
tier	Name And Id Output []	Loyalty tier eligible for the promotion.	Small, 60.0	60.0
tierGroup	Name And Id Output []	Loyalty tier group eligible for the promotion.	Small, 60.0	60.0

Events Detail Output

Output representation of the customer event eligible for a promotion.

Property Name	Type	Description	Filter Group and Version	Available Version
event	String []	Customer event eligible for the promotion. Possible values are: - CustomerBirthday - MemberAnniversary - MemberBirthday	Small, 60.0	60.0
eventPeriodUnit	String []	Period the promotion is eligible for. Possible values: - Date - Week - Month	Small, 60.0	60.0

Expression

Output representation of the expression.

Property Name	Type	Description	Filter Group and Version	Available Version
expression	String	The name of an expression.	Small, 55.0	55.0
expressionValue	Expression Value	The values of an expression.	Small, 55.0	55.0
parameterValues	Map<String, Expression Value >	The list of parameter values provided for an expression.	Small, 55.0	55.0
resolvedExpression	String	The resolved fields and values of an expression.	Small, 55.0	55.0

Expression Value

Output representation of the expression value.

Property Name	Type	Description	Filter Group and Version	Available Version
dataType	ExpressionDataType	The data type of an expression. Possible values are: • Boolean • Date • DateTime • Number • Subject • Text	Small, 55.0	55.0
value	Object	The value of an expression.	Small, 55.0	55.0

Game Assignment Output

Output representation of the details of a game assigned to a member.

Property Name	Type	Description	Filter Group and Version	Available Version
gameParticipantId	String	ID of the associated game participant.	Small, 60.0	60.0
gameParticipantReward	String	The ID of the game participant reward record that stores the details of the reward that the member receives after they play the game. The record is associated with the source transaction journal and the record that issues the reward to the participant.	Small, 60.0	60.0
gameDefinitionName	String	Name of the assigned game definition.	Small, 60.0	60.0

Property Name	Type	Description	Filter Group and Version	Available Version
gameExpirationDate	String	Date until which the assigned game is valid.	Small, 60.0	60.0
reason	String	Reason the game was assigned to the member.	Small, 60.0	60.0

Game Details

Output representation of the details of a game definition.

Property Name	Type	Description	Filter Group and Version	Available Version
description	String	Description of the game.	Small, 60.0	60.0
endDate	String	Date until which participants are eligible to play the game.	Small, 60.0	60.0
gameRewards	Game Rewards List []	List of possible rewards for a game.	Small, 60.0	60.0
name	String	Name of the game definition.	Small, 60.0	60.0
participantGameRewards	Participant Game Rewards List []	List of participant game reward records that contain the details of the game reward the participant has won for each game play, or can win after playing the game.	Small, 60.0	60.0
startDate	String	Date from which participants are eligible to play the game.	Small, 60.0	60.0
timeoutDuration	Integer	Duration of the game in seconds before the game session times out.	Small, 60.0	60.0
type	String	Type of the game. Possible values are: - Scratchcard - SpintheWheel	Small, 60.0	60.0

Game Details (Promotion)

The details of the game that's assigned to the customer.

JSON example

This example shows a sample response.

```
"gamesAssignedList": [
  {
    "gameDefinitionName": "SpintheWheelGame",
    "gameExpirationDate": "2025-10-10T00:00:00.000Z",
    "gameParticipantRewardId": "3PRSB0000004LFZ"
```

```
}
]
```

Property Name	Type	Description	Filter Group and Version	Available Version
gameDefinitionName	String	Name of the game that's assigned to a customer.	Big, 65.0	65.0
gameExpirationDate	String	Date on which the game expires for the customer.	Big, 65.0	65.0
gameParticipantRewardId	String	The ID of the game participant reward record that stores the details of the reward that the customer receives after they play the game.	Big, 65.0	65.0

Game Result

Output representation of the game reward that a participant has won.

JSON example

Here's an example of a reward won by the participant for the Spin the Wheel game:

```
{
  "errorMessage":null,
  "gameReward":[
    {
      "color":"01CD6C",
      "description":"Loyalty Points",
      "gameRewardId":"3GRSB0000000Caf4AE",
      "imageUrl":null,
      "issuedRewardReference":"01VSB000000032j2AA",
      "name":"1000 Reward Points",
      "rewardType":"LoyaltyPoints",
      "rewardValue":"100.0"
    }
  ],
  "status":true
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
errorMessage	String	Error message that provides the reason why the details of the participant reward weren't found.	Small, 60.0	60.0
gameReward	Game Reward Details []	Details of the game reward won by the participant.	Small, 60.0	60.0
status	Boolean	Indicates whether the request succeeded (<code>true</code>) or failed (<code>false</code>).	Small, 60.0	60.0

Game Reward Details

Output representation of the details of the game reward that a participant has won.

Property Name	Type	Description	Filter Group and Version	Available Version
color	String	Color of the segment in the Spin the Wheel game that represents the game reward.	Small, 60.0	60.0
description	String	Description of the game reward.	Small, 60.0	60.0
gameRewardId	String	ID of the game reward won by the participant.	Small, 60.0	60.0
imageUrl	String	Image URL of the segment in the Spin the Wheel game that represents the game reward.	Small, 60.0	60.0
issuedRewardReference	String	Voucher ID, or the transaction journal ID associated with the voucher, or the loyalty points rewarded to the participant.	Small, 60.0	60.0
name	String	Name of the game reward.	Small, 60.0	60.0
rewardType	String	Type of the game reward. Possible values are: • CustomReward • LoyaltyPoints • NoReward • Raffle • Voucher	Small, 60.0	60.0
rewardValue	String	Value of the voucher, or the number of points to credit as a game reward to participants.	Small, 60.0	60.0

Game Rewards List

Output representation of the details of a possible game reward.

Property Name	Type	Description	Filter Group and Version	Available Version
color	String	Color of the segment in the Spin the Wheel game that represents the game reward.	Small, 60.0	60.0
description	String	Description of the game reward.	Small, 60.0	60.0
imageUrl	String	Image URL of the segment in the Spin the Wheel game that represents the game reward.	Small, 60.0	60.0

Property Name	Type	Description	Filter Group and Version	Available Version
name	String	Name of the game reward.	Small, 60.0	60.0
rewardId	String	ID of the game reward.	Small, 60.0	60.0
rewardType	String	Type of the game reward. Possible values are: - CustomReward - LoyaltyPoints - NoReward - Raffle - Voucher	Small, 60.0	60.0
rewardValue	String	Value of the voucher, or the number of points to credit as a game reward to participants.	Small, 60.0	60.0

Games List

Output representation of the list of games assigned to a member.

JSON example

Here's an example of the details of an eligible Spin the Wheel game assigned to a member, and an already played Scratch Card game.

```
{
  "errorMessage":null,
  "gameDefinitions":[
    {
      "description":"Play Spin The Wheel to get amazing rewards",
      "endDate":"2024-07-31T19:00:00.000Z",
      "gameRewards":[
        {
          "color":"FF5733",
          "description":"Get 10% Off Voucher",
          "imageUrl":null,
          "name":"10 % Off",
          "rewardType":"Voucher",
          "rewardValue":null
        },
        {
          "color":"000000",
          "description":"Loyalty Points",
          "imageUrl":null,
          "name":"1000 Reward Points",
          "rewardType":"LoyaltyPoints",
          "rewardValue":"100.0"
        },
        {
          "color":"762E2E",
```

```

        "description": "Better Luck Next Time",
        "imageUrl": null,
        "name": "No Reward",
        "rewardType": "NoReward",
        "rewardValue": null
    }
],
"participantGameRewards": [
    {
        "expirationDate": "2024-01-04T20:00:00.000Z",
        "gameParticipantRewardId": "3PRSB0000000hjJ4AQ",
        "gameRewardId": null,
        "issuedRewardReference": null,
        "sourceActivity": null,
        "status": "YetToReward"
    }
],
"name": "Spin The Wheel",
"startDate": "2023-10-01T19:00:00.000Z",
"timeoutDuration": 10,
"type": "SpintheWheel"
},
{
    "description": "Play Scratch card to get amazing rewards",
    "endDate": "2024-07-31T19:00:00.000Z",
    "gameRewards": [
        {
            "color": "",
            "description": "Get 10% Off Voucher",
            "imageUrl": null,
            "name": "10 % Off",
            "rewardType": "Voucher",
            "rewardValue": null
        },
        {
            "color": "",
            "description": "Better Luck Next Time",
            "imageUrl": null,
            "name": "No Reward",
            "rewardType": "NoReward",
            "rewardValue": null
        }
    ],
    "participantGameRewards": [
        {
            "expirationDate": "2024-01-04T20:00:00.000Z",
            "gameParticipantRewardId": "2ABSB0000000hjJ3AM",
            "gameRewardId": "2GRSB0000000Caf4AB",
            "issuedRewardReference": "01VSB000000032j2AA",
            "sourceActivity": "0c85i0000009c6D",
            "status": "Rewarded"
        }
    ],
    "name": "Scratch Card",

```

```

        "startDate":"2023-10-01T19:00:00.000Z",
        "timeoutDuration":20,
        "type":"Scratchcard"
    }
],
"status":true
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
errorMessage	String	Message that provides the reason why a list of games weren't found for the participant.	Small, 60.0	60.0
gameDefinitions	Game Details []	List of game definitions associated with the games assigned to the member. A game definition has the details of a game, including: • The list of possible rewards for the game. • The participant game reward records that contain details of the game reward the participant has won for each game play, or can win after playing the game.	Small, 60.0	60.0
status	Boolean	Indicates whether the request succeeded (<code>true</code>) or failed (<code>false</code>).	Small, 60.0	60.0

Issue Voucher Action Output

Output representation of the issue voucher action type.

Property Name	Type	Description	Filter Group and Version	Available Version
actionType	String	The type of action in the loyalty program process rule.	Small, 55.0	55.0
discountPercent	Integer	The discount percentage provided by the voucher.	Small, 55.0	55.0
effectiveDate	String	The date the voucher goes into effect.	Small, 55.0	55.0
expiryDate	String	The date the voucher expires.	Small, 55.0	55.0
faceValue	Double	The value of the voucher.	Small, 55.0	55.0
productId	String	The ID of the product associated with the voucher.	Small, 55.0	55.0
promotionId	String	The ID of the promotion associated with the rule.	Small, 55.0	55.0

Property Name	Type	Description	Filter Group and Version	Available Version
stepName	String	The name of the step in the loyalty program process rule.	Small, 55.0	55.0
voucherCode	String	The code of the voucher.	Small, 55.0	55.0
voucherDefinition Name	String	The name of the voucher definition associated with the voucher code.	Small, 55.0	55.0
voucherId	String	The ID of the voucher.	Small, 55.0	55.0

Individual Member Enrollment Output

Output representation of the individual member enrollment create request.

JSON example

```
{
  "contactId": "003xx000004Ws08",
  "loyaltyProgramMemberId": "01Mxx00000008FJ",
  "loyaltyProgramName": "program1",
  "membershipNumber": "Member3",
  "transactionJournals": [
    {
      "activityDate": "2021-01-01T08:00:00.000Z",
      "journalSubType": "Member Enrollment",
      "journalType": "Accrual",
      "loyaltyProgram": "program1",
      "loyaltyProgramMember": "Member3",
      "referredMember": null,
      "status": "Pending",
      "transactionJournalId": "01Vxx000000029t"
    },
    {
      "activityDate": "2021-01-01T08:00:00.000Z",
      "journalSubType": "Member Referral",
      "journalType": "Accrual",
      "loyaltyProgram": "program1",
      "loyaltyProgramMember": "Member1",
      "referredMember": "Member3",
      "status": "Pending",
      "transactionJournalId": "01Vxx000000029u"
    }
  ]
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
contactId	String	The ID of the loyalty program member's person account.	Small, 55.0	55.0

Property Name	Type	Description	Filter Group and Version	Available Version
loyaltyProgramMemberId	String	The ID of the loyalty program member record created for the member.	Small, 55.0	55.0
loyaltyProgramName	String	The name of the loyalty program into which the member is enrolled.	Small, 55.0	55.0
membershipNumber	String	The membership number of the enrolled member.	Small, 55.0	55.0
transactionJournals	Transaction Journal Output[]	The list of transaction journals that are created.	Small, 55.0	55.0

Link Program Member Partner

Output representation of the program member and partner linkage request.

JSON example

```
{
  "linkNumber": "00000028",
  "message": null,
  "status": true
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
linkNumber	String	Unique identifier for the program member and partner linkage.	Big, 66.0	66.0
message	String	Message displayed when a loyalty program member is linked to a partner.	Big, 66.0	66.0
status	Boolean	Indicates whether the API request was successful (<code>true</code>) or not (<code>false</code>).	Big, 66.0	66.0

Loyalty Account Output

Represents the details of the loyalty program member's associated account record.

Property Name	Type	Description	Filter Group and Version	Available Version
accountId	String	The ID of the associated account record.	Small, 55.0	55.0
name	String	The name of the associated account.	Small, 55.0	55.0

Loyalty Engine Realtime Output

Output representation of the loyalty program process result.

JSON example

```
{
  "message":null,
  "processResult":{
    "memberPointsBalance":[
      {
        "memberCurrencies":[
          {
            "currencyName":"Tier Coins",
            "currencyType":"Qualifying",
            "escrowPointsBalance":0.0,
            "pointsBalance":350.0
          },
          {
            "currencyName":"Coins",
            "currencyType":"NonQualifying",
            "escrowPointsBalance":0.0,
            "pointsBalance":300.0
          }
        ],
        "memberShipNumber":"M5"
      }
    ],
    "transactionJournalResult":[
      {
        "badgesAssigned":[
          {
            "badgeValidityEndDate":"2023-12-15",
            "loyaltyProgramBadgeName":"Gen AI Badge",
            "memberBadgeStatus":"1",
            "reason":"Test"
          }
        ],
        "changeInMemberPoints":[
          {
            "changeInMemberCurrencies":[
              {
                "changeInPointsBalance":100.0,
                "currencyName":"Coins",
                "currencyType":"NonQualifying",
                "promotionId":"0c8RM0000004Fix"
              }
            ],
            "memberShipNumber":"M5"
          }
        ],
        "gamesAssigned":[
          {
            "gameDefinitionName":"3PRRM0000004C98",
            "gameExpirationDate":"2023-12-28T00:00:00.000Z",

```

```

        "reason": "Test"
      }
    ],
    "id": "01VRM0000002qnB2AQ",
    "voucherResult": {
      "vouchersIssued": [
        {
          "discountPercent": 10,
          "effectiveDate": "2023-11-29",
          "expirationDate": "2023-12-15",
          "promotionId": "0c8RM0000004Fix",
          "voucherDefinitionName": "10% Discount Voucher"
        }
      ]
    }
  }
}
],
},
"programName": "NTO",
"simulationResults": null,
"status": true
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
message	String	Error message.	Small, 54.0	54.0
processResult	Process Result	Details of each transaction journal in a request, and also the member points balance that is the currency-wise balances after all the transaction journals are processed. The process result collection is empty when the simulation is run.	Small, 54.0	54.0
programName	String	Name of the loyalty program.	Small, 55.0	55.0
simulationResults	Transaction Journal Map Output	Simulation result that details how the loyalty program process rules can process a transaction journal. The simulation result collection is empty when the loyalty program process is run.	Small, 55.0	55.0
status	Boolean	Indicates whether the request succeeded (<code>true</code>) or failed (<code>false</code>).	Small, 54.0	54.0



Important: The response contains only the details of the points, badges, games, and vouchers that are provided by the loyalty program process rules executed for the member's order. The response doesn't contain the details of the rewards provided by the flows or the subprocesses that the loyalty program process rules execute.

Loyalty Contact Output

Represents the details of the loyalty program member's associated contact record.

Property Name	Type	Description	Filter Group and Version	Available Version
contactId	String	The ID of the associated contact record.	Small, 55.0	55.0
email	String	The email of the associated contact.	Small, 55.0	55.0
firstName	String	The first name of the associated contact.	Small, 55.0	55.0
lastName	String	The last name of the associated contact.	Small, 55.0	55.0

Loyalty Process Output

Output representation of the loyalty process response.

Property Name	Type	Description	Filter Group and Version	Available Version
message	String	The message that provides details of the error in executing the loyalty process request.	Small, 56.0	56.0
outputParameters	Map<String, Object>	The parameters returned after the execution of a loyalty process.	Small, 56.0	56.0
simulationDetails	Map<String, Object>	Specifies the simulation execution type details including detailed steps on how the process is executed.	Small, 56.0	56.0
		 Note: This property is for future use.		
status	Boolean	Indicates whether the request succeeded (<code>true</code>) or failed (<code>false</code>).	Small, 56.0	56.0

Loyalty Program Process Rule

Output representation of the loyalty program process rules.

Property Name	Type	Description	Filter Group and Version	Available Version
actionResultSummary	Various	The result provided by each action that processed the transaction journals. The type of action must be configured in the loyalty program process rule. • CreditPoints—Credit loyalty points to a specified loyalty program	Small, 55.0	55.0

Property Name	Type	Description	Filter Group and Version	Available Version
		member's balance. See Credit Points Action Output. • DebitPoints—Debit loyalty points from a specified loyalty program member's balance. See Debit Points Action Output. • IssueVoucher—Issue a voucher for a loyalty program member. See Issue Voucher Action Output. • UpdateUsageForCumulativePromotion—Update the progress made by a loyalty program member towards attaining a cumulative type promotion. See Update Cumulative Action Output.		
errorMessage	String	The message stating the reason for error, if any.	Small, 55.0	55.0
executionStatus	String	The execution status of the loyalty program process rule. Valid values are: • CriteriaNotMet • Failed • Success	Small, 55.0	55.0
executionSummary	Summary	The summary of how the loyalty program process rules processed the transaction journals.	Small, 55.0	55.0
previousRuleName	String	The name of the loyalty program process rule that processed the transaction journals before the current loyalty program process rule.	Small, 55.0	55.0
processRuleName	String	The name of the loyalty program process rule that processed the transaction journals.	Small, 55.0	55.0
promotionName	String	The name of the loyalty program promotion associated with the rule.	Small, 55.0	55.0
ruleSteps	Various	A list of steps that are part of the loyalty process process rules. • Action—Loyalty program process rule action. See Loyalty Program Process Rule Action.	Small, 55.0	55.0

Property Name	Type	Description	Filter Group and Version	Available Version
		- Condition—Loyalty program process rule condition. See Loyalty Program Process Rule Condition. - ConditionGroup—Loyalty program process rule condition group. See Loyalty Program Process Rule Condition Group. - DefaultPath		

Loyalty Program Process Rule Step

Output representation of the loyalty program process rule step.

Property Name	Type	Description	Filter Group and Version	Available Version
errorMessage	String	The message stating the reason for error, if any.	Small, 55.0	55.0
sequenceNumber	Integer	The sequence number of a rule step.	Small, 55.0	55.0
status	String	The status of the loyalty program process rule step execution. Possible values are: - CriteriaNotMet - Failed - Success	Small, 55.0	55.0
stepName	String	The name of the step in the loyalty program process rule.	Small, 55.0	55.0
stepType	String	Specifies the type of step in the loyalty program process rule. Possible values are: - Action - Condition - ConditionGroup - DefaultPath	Small, 55.0	55.0

Loyalty Program Process Rule Action

Output representation of the loyalty program process rule action.

Property Name	Type	Description	Filter Group and Version	Available Version
actionParameters	Loyalty Program Process Rule Action Parameter[]	The list of parameters of the action.	Small, 55.0	55.0
actionType	String	The type of action in the loyalty program process rule. Possible values are: - AssignParameterValues - CreditPoints - DeditPoints - GetMemberAttributesValues - GetMemberPointBalance - GetMemberTier - GetOutputsFromDecisionTable - IncreaseUsageForCumulativePromotion - IssueVoucher - RunFlow - UpdateCurrentValueForMemberAttribute - UpdateUsageForCumulativePromotion	Small, 55.0	55.0
decisionTableName	String	The name of the decision table being executed with the action.	Small, 55.0	55.0
errorMessage	String	The message stating the reason for error, if any.	Small, 55.0	55.0
flowDefinitionName	String	The name of the flow definition that is run with the action.	Small, 55.0	55.0
sequenceNumber	Integer	The sequence number of a rule step in the process rule.	Small, 55.0	55.0
status	String	The status of the loyalty program process rule step execution. Possible values are: - CriteriaNotMet - Failed - Success	Small, 55.0	55.0
stepName	String	The name of the step in the loyalty program process rule.	Small, 55.0	55.0

Property Name	Type	Description	Filter Group and Version	Available Version
stepType	String	Specifies the type of step in the loyalty program process rule. Possible values are: - Action - Condition - ConditionGroup - DefaultPath	Small, 55.0	55.0

Loyalty Program Process Rule Action Parameter

Output representation of the loyalty program process rule action parameter.

Property Name	Type	Description	Filter Group and Version	Available Version
name	String	The name of the parameter.	Small, 55.0	55.0
operator	String	The type of operator used in the action. Possible value is: Equals	Small, 55.0	55.0
sequenceNumber	Integer	The sequence number of the parameter in the action.	Small, 55.0	55.0
value	Expression	The value of the parameter.	Small, 55.0	55.0

Loyalty Program Process Rule Condition

Output representation of the loyalty program process rule condition.

Property Name	Type	Description	Filter Group and Version	Available Version
childSteps	Loyalty Program Process Rule Step []	The child steps of the condition.	Small, 55.0	55.0
errorMessage	String	The message stating the reason for error, if any.	Small, 55.0	55.0
filterCondition	String	The logic that is specified to apply the filter conditions.	Small, 55.0	55.0
filterCriteria	Loyalty Program Process Rule Condition Filter Criteria []	The filter criteria of a condition.	Small, 55.0	55.0

Property Name	Type	Description	Filter Group and Version	Available Version
isConditionMet	Boolean	Indicates whether the transaction journal meets this condition (<code>true</code>) or not (<code>false</code>).	Small, 55.0	55.0
sequenceNumber	Integer	Sequence number of a rule step in a rule.	Small, 55.0	55.0
status	String	The status of the loyalty program process rule step execution. Possible values are: - CriteriaNotMet - Failed - Success	Small, 55.0	55.0
stepName	String	The name of the step in the loyalty program process rule.	Small, 55.0	55.0
stepType	String	Specifies the type of step in the loyalty program process rule. Possible values are: - Action - Condition - ConditionGroup - DefaultPath	Small, 55.0	55.0

Loyalty Program Process Rule Condition Filter Criteria

Output representation of the loyalty program process rule condition filter criteria.

Property Name	Type	Description	Filter Group and Version	Available Version
filterCriteria Result	Boolean	Indicates whether the transaction journal meets this filter criteria (<code>true</code>) or not (<code>false</code>).	Small, 55.0	55.0
operator	String	The operator of a condition filter criteria. Possible values are: - Contains - DoesNotContain - Equals - GreaterThan - GreaterThanOrEquals - IsNull	Small, 55.0	55.0

Property Name	Type	Description	Filter Group and Version	Available Version
		- IsNull - LessThan - LessThanOrEquals - NotEquals		
sequence	Integer	The sequence number of the criteria in a rule condition.	Small, 55.0	55.0
sourceFieldName	Expression	The source field of a condition or action.	Small, 55.0	55.0
value	Expression	The value of an expression.	Small, 55.0	55.0
valueType	String	The type of value that is specified for a condition or an action parameter. Possible values are: - FixedValue - Formula - Lookup - Parameter	Small, 55.0	55.0

Loyalty Program Process Rule Condition Group

Output representation of the loyalty program process rule condition group.

Property Name	Type	Description	Filter Group and Version	Available Version
childSteps	Loyalty Program Process Rule Step[]	The child steps of the condition group.	Small, 55.0	55.0
errorMessage	String	The message stating the reason for error, if any.	Small, 55.0	55.0
sequenceNumber	Integer	The sequence number of a rule step in a rule.	Small, 55.0	55.0
status	String	The status of the loyalty program process rule step execution. Possible values are: - CriteriaNotMet - Failed - Success	Small, 55.0	55.0
stepName	String	The name of the step in the loyalty program process rule.	Small, 55.0	55.0

Property Name	Type	Description	Filter Group and Version	Available Version
stepType	String	Specifies the type of step in the loyalty program process rule. Possible values are: • Action • Condition • ConditionGroup • DefaultPath	Small, 55.0	55.0

Loyalty View Transaction Journal History Output

Output representation of the view transaction history request.

JSON example

```
{
  "message":null,
  "status":true,
  "totalCount":2,
  "transactionJournals":[
    {
      "ActivityDate":"2021-11-27T00:00:00Z",
      "BookedFareClass":"","
      "BookedRoomType":"","
      "BookingDate":"2020-10-23T00:00:00Z",
      "Brand":"Ikea",
      "Channel":"Offline",
      "Destination":"London",
      "Distance":0,
      "EndDate":"2021-10-23T00:00:00Z",
      "ErrorDescription":"","
      "EscrowPointsCreditDate":"2020-10-23",
      "Establishment":"Build",
      "ExternalTransactionNumber":"1",
      "FlightNumber":"123456",
      "Industry":"Wellness",
      "InvoiceDate":"2020-10-23",
      "JournalDate":"2021-12-25T00:00:00Z",
      "JournalReason":"","
      "JournalSubTypeId":"01ST10000000045MAA",
      "JournalTypeId":"01ET1000000009GMAQ",
      "LoyaltyProgramId":"01pT100000000HbIAI",
      "MemberId":"01MT100000000ZLMAY",
      "OrderId":"","
      "OrderItemId":"","
      "Origin":"NewYork",
      "PartnerId":"","
      "PaymentMethod":"","
      "ProductCategoryId":"0ZGT100000000AsOAI",

```

```

    "ProductId":"01tT1000000EQcXIAW",
    "PromotionId":"",
    "Quantity":1,
    "QuantityUnitOfMeasureId":"",
    "ReferredMemberId":"",
    "RelatedCorporateMembershipId":"",
    "RelatedJournalId":"",
    "StartDate":"2020-10-23T00:00:00Z",
    "TransactionAmount":45,
    "TransactionLocation":"",
    "TraveledFareClass":"",
    "VoucherCode":""
  },
  {
    "ActivityDate":"2021-11-27T00:00:00Z",
    "BookedFareClass":"",
    "BookedRoomType":"",
    "BookingDate":"2020-10-23T00:00:00Z",
    "Brand":"Ikea",
    "Channel":"Offline",
    "Destination":"London",
    "Distance":0,
    "EndDate":"2021-10-23T00:00:00Z",
    "ErrorDescription":"",
    "EscrowPointsCreditDate":"2020-10-23",
    "Establishment":"Build",
    "ExternalTransactionNumber":"1",
    "FlightNumber":"123456",
    "Industry":"Wellness",
    "InvoiceDate":"2020-10-23",
    "JournalDate":"2021-12-25T00:00:00Z",
    "JournalReason":"",
    "JournalSubTypeId":"01ST10000000045MAA",
    "JournalTypeId":"01ET1000000009GMAQ",
    "LoyaltyProgramId":"0lpT100000000HbIAI",
    "MemberId":"01MT100000000ZLMAY",
    "OrderId":"",
    "OrderItemId":"",
    "Origin":"NewYork",
    "PartnerId":"",
    "PaymentMethod":"",
    "ProductCategoryId":"0ZGT100000000AsOAI",
    "ProductId":"01tT1000000EQcXIAW",
    "PromotionId":"",
    "Quantity":2,
    "QuantityUnitOfMeasureId":"",
    "ReferredMemberId":"",
    "RelatedCorporateMembershipId":"",
    "RelatedJournalId":"",
    "StartDate":"2020-10-23T00:00:00Z",
    "TransactionAmount":55,
    "TransactionLocation":"",
    "TraveledFareClass":"",
    "VoucherCode":""
  }

```

```

    }
  ]
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
message	String	The message explaining why the request failed.	Small, 56.0	56.0
status	Boolean	Indicates whether the request succeeded (<code>true</code>) or failed (<code>false</code>).	Small, 56.0	56.0
totalCount	Integer	The total number of transaction journals that match the input criteria.	Small, 56.0	56.0
transactionJournals	Map<String, Object>[]	The details of the transaction journals.	Small, 56.0	56.0

Member Benefits Output

Output representation of all benefits available for a member.

JSON Example

```

{
  "result" : {
    "benefitId": "01B000000abc123",
    "benefitName": "15 Kg Luggage",
    "benefitTypeId": "01o0000000afja435",
    "benefitTypeName": "Carry On Luggage",
    "startDate": "23-05-2015",
    "endDate": "22-05-2035",
    "isActive": true,
    "createdRecordId": "570000000ak123",
    "createdRecordName": "15 Kg Luggage Carry On",
    "description": "This is a carry on luggage member benefit"
  },
  {
    "benefitId": "01B000000gsy88",
    "benefitName": "Priority Support",
    "benefitTypeId": "01o0000000pi23ii",
    "benefitTypeName": "Support Entitlements",
    "startDate": "16-08-2020",
    "endDate": "15-02-2021",
    "isActive": true,
    "createdRecordId": "550000000aj112",
    "createdRecordName": "Priority Support Entitlement",
    "description": "This is a support entitlement member benefit"
  }
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
memberBenefits	Member Benefit Output[]	List of all benefits available for a member.	Small, 51.0	51.0

Member Benefit Output

Output representation of an individual benefit available for a member.

Property Name	Type	Description	Filter Group and Version	Available Version
benefitId	String	The ID of the benefit available for the member.	Small, 51.0	51.0
benefitName	String	The name of the benefit available for the member.	Small, 51.0	51.0
benefitTypeId	String	The ID of the associated benefit type record.	Small, 51.0	51.0
benefitTypeName	String	The name of the associated benefit type.	Small, 51.0	51.0
createdRecordId	String	The ID of the record created during benefit assignment.	Small, 51.0	51.0
createdRecordName	String	The display name of the record created during benefit assignment.	Small, 51.0	51.0
description	String	The description of the benefit available to a loyalty program member.	Small, 58.0	58.0
endDate	String	The date the benefit expires.	Small, 51.0	51.0
isActive	Boolean	Indicates whether the benefit is active (<code>true</code>) or not (<code>false</code>).	Small, 51.0	51.0
startDate	String	The date from when the benefit applies to a member.	Small, 51.0	51.0

Member Currency Output

Represents the details of the loyalty program member's associated currencies.

Property Name	Type	Description	Filter Group and Version	Available Version
additionalLoyaltyMemberCurrencyFields	Map<String, Object>	The details of the custom fields added in the loyalty member currency object.	Small, 55.0	55.0
escrowPoints Balance	Double	The escrow points balance of the member for the currency.	Small, 55.0	55.0

Property Name	Type	Description	Filter Group and Version	Available Version
expirablePoints	Double	The number of points that the member hasn't redeemed for the currency.	Small, 55.0	55.0
lastAccrualProcessedDate	String	The date on which the member's accruals for the currency were last processed.	Small, 55.0	55.0
lastEscrowProcessedDate	String	The date on which the member's escrow points were last rolled over to their points balance for the currency.	Small, 55.0	55.0
lastExpirationProcessRunDate	String	The date on which the member's points were last expired for the currency.	Small, 55.0	55.0
lastPointsAggregationDate	String	The date on which the member's points for the fixed type non-qualifying currency were last aggregated.	Small, 55.0	55.0
lastPointsResetDate	String	The date on which the member's points for the currency were last reset.	Small, 55.0	55.0
loyaltyMemberCurrencyName	String	The name of the Loyalty Member Currency record.	Small, 55.0	55.0
loyaltyProgramCurrencyId	String	The ID of the loyalty program currency associated with the member.	Small, 55.0	55.0
loyaltyProgramCurrencyName	String	The name of the loyalty program currency associated with the member.	Small, 55.0	55.0
memberCurrencyId	String	The ID of the Loyalty Member Currency record.	Small, 55.0	55.0
nextQualifyingPointsResetDate	String	The date on which the member's points balance for the qualifying currency is reset next.	Small, 55.0	55.0
pointsBalance	Double	The points balance of the member for the currency.	Small, 55.0	55.0
qualifyingPointsBalanceBeforeReset	Double	The last point balance of the member before the member's points balance was reset to zero for the qualifying currency.	Small, 55.0	55.0
totalEscrowPointsAccrued	Double	The total number of points that the member has accrued in escrow for the currency.	Small, 55.0	55.0
totalEscrowRolloverPoints	Double	The total number of points that has rolled over from the member's escrow point balance to the member's points balance for the currency.	Small, 55.0	55.0
totalPointsAccrued	Double	The total number of points the member has accrued for the currency.	Small, 55.0	55.0

Property Name	Type	Description	Filter Group and Version	Available Version
totalPointsExpired	Double	The total number of points that have expired for the member.	Small, 55.0	55.0
totalPointsRedeemed	Double	The total number of points the member has redeemed for the currency.	Small, 55.0	55.0

Member Currency Change

Output representation of the member currency change.

Property Name	Type	Description	Filter Group and Version	Available Version
changeEscrowPointsBalance	Double	The change in escrow points balance.	Small, 54.0	54.0
changeInPointsBalance	Double	The change in points balance.	Small, 54.0	54.0
currencyName	String	The name of the member currency.	Small, 54.0	54.0
currencyType	String	The type of member currency. Possible values are: - NonQualifying - Qualifying	Small, 54.0	54.0
promotionId	String	The promotion ID for which the member is credited points.	Small, 54.0	54.0

Member Engagement Trail Output

Output representation of member engagement trail API.

Sample Response

 **Note:** The field `remainingResetCount` and `totalResetCount` may show null if the set limit is 1.

```
{
  "memberEngagementAttributeOutputRepresentations": [
    {
      "attributeStatus": "Active",
      "currentValue": null,
      "defaultValue": "0",
      "description": null,
      "endDate": "2025-01-31",
      "name": "Book A Cab",
      "startDate": "2024-11-01",
      "targetValue": "2.0"
    },
    {
      "attributeStatus": "Active",
```

```
    "currentValue": null,
    "defaultValue": "0",
    "description": null,
    "endDate": "2025-01-31",
    "name": "Book A Hotel Room",
    "startDate": "2024-11-01",
    "targetValue": "1.0"
  },
  {
    "attributeStatus": "Active",
    "currentValue": null,
    "defaultValue": "0",
    "description": null,
    "endDate": "2025-01-31",
    "name": "Book A Flight",
    "startDate": "2024-11-01",
    "targetValue": "1.0"
  }
],
"message": null,
"promotionName": "Engagement Trail",
"remainingResetCount": null,
"rewards": [
  {
    "badgeReward": [],
    "pointsRewards": [
      {
        "currencyName": "Reward Points",
        "points": 200
      },
      {
        "currencyName": "Tier Points",
        "points": 100
      }
    ]
  },
  {
    "voucherRewards": []
  }
],
"status": true,
"totalResetCount": null
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
status	Boolean	Indicates whether the request succeeded (true) or failed (false).	Small, 62.0	62.0
message	String	The message explaining why the request failed.	Small, 62.0	62.0
engagementAttribute		Indicates the event that is tracked for a member.	Small, 62.0	62.0

Property Name	Type	Description	Filter Group and Version	Available Version
totalResetCount	Integer	The maximum number of times a member can complete the engagement trail and earn rewards.	Small, 62.0	62.0
remainingResetCount	Integer	The number of times a member can still complete the engagement trail and earn rewards.	Small, 62.0	62.0

Member Engagement Attribute

Output representation of Member Engagement Attribute.

Property Name	Type	Description	Filter Group and Version	Available Version
name	String	The name of the engagement attribute.	Small, 62.0	62.0
description	String	The description of the engagement attribute.	Small, 62.0	62.0
startDate	DateTime	The date from when the engagement attribute is tracked for members.	Small, 62.0	62.0
endDate	DateTime	The date until which the engagement attribute is tracked for members.	Small, 62.0	62.0
currentValue	Integer	The current value of the engagement attribute.	Small, 62.0	62.0
targetValue	Integer	The value that members have to achieve to complete the engagement attribute's target.	Small, 62.0	62.0
attributeStatus	String	The status of the engagement attribute.	Small, 62.0	62.0
defaultValue	Integer	The default value of the engagement attribute.	Small, 62.0	62.0
pointsRewards	List	Indicates the rewards in the form of points or currencies.	Small, 62.0	62.0
currencyId	String	The ID of the currency.	Small, 62.0	62.0
currencyName	String	The name of the currency.	Small, 62.0	62.0
points	Integer	Number of points to give as a reward.	Small, 62.0	62.0
voucherRewards	List	Indicates the rewards in the form of vouchers.	Small, 62.0	62.0
voucherDefinitionId	String	The ID of the voucher definition.	Small, 62.0	62.0
voucherName	String	The name of the voucher.	Small, 62.0	62.0

Property Name	Type	Description	Filter Group and Version	Available Version
BadgeRewards	List	Indicates the rewards in the form of badges.	Small, 62.0	62.0
badgeId	String	The ID of the badge.	Small, 62.0	62.0
badgeName	String	The name of the badge.	Small, 62.0	62.0

Member Points Balance

Output representation of the member currency result.

Property Name	Type	Description	Filter Group and Version	Available Version
memberCurrencies	Member Currency []	List of currencies associated with the loyalty program member.	Small, 54.0	54.0
membershipNumber	String	The membership number of the loyalty program member who is associated with the transaction journal that's created.	Small, 54.0	54.0

Member Profile Output

Output representation of the details of a loyalty program member.

JSON example

```
{
  "programName": "baLoyalty",
  "associatedContactDetails": {
    "id": "003xx0000000003",
    "firstName": "Abcd",
    "lastName": "Xyz",
    "email": "abcd@salesforce.com"
  },
  "associatedAccountDetails": {
    "id": "001xx0000000003",
    "name": "Salesforce"
  },
  "membershipNumber": "1234",
  "memberId": "01Mxx0000A000001EAA",
  "memberStatus": "Active",
  "memberType": "Individual",
  "groupName": "Family Group",
  "groupCreatedByMember": "GM-001",
  "associatedCorporateMembershipNumber": "G-001",
  "referredBy": "H-001",
  "lastActivityDate": "2022-01-02",
  "statementLastGeneratedDate": "2022-09-09",
  "statementFrequency": "Monthly",
  "statementMethod": "Email",
}
```

```

"enrollmentDate": "2022-01-01",
"enrollmentChannel": "POS",
"canReceivePromotions": "True",
"canReceivePartnerPromotion": "True",
"membershipEndDate": "2023-01-01",
"membershipLastRenewalDate": "2021-01-01",
"memberCurrencies": [
  {
    "programCurrency": "NQP",
    "pointsBalance": 500,
    "escrowPointsBalance": 25
  }
],
"memberTiers": [
  {
    "tierGroupName": "QualifyingGroup",
    "tierName": "Gold",
    "sequenceNumber": "4",
    "effectiveDate": "2022-01-01",
    "expirationDate": "2022-01-10"
  }
],
"additionalMemberFieldValues": {
  "secondaryEmail__c": "abcd@salesforce.com"
},
"additionalMemberCurrencyFieldValues": {
  "conversionRate__c": "5"
},
"additionalMemberTierFieldValues": {
  "previousTier__c": "Silver"
}
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
additionalLoyaltyProgramMemberFields	Map<String, Object>	The details of the custom fields added in the loyalty program member object.	Small, 55.0	55.0
associatedAccount	Member Account Output	The details of the account record that is associated with the loyalty program member.	Small, 55.0	55.0
associatedContact	Member Contact Output	The details of the contact record that is associated with the loyalty program member.	Small, 55.0	55.0
canReceivePartnerPromotions	Boolean	Indicates whether the loyalty program member can receive partner promotions.	Small, 55.0	55.0
canReceivePromotions	Boolean	Indicates whether the loyalty program member can receive promotions (<code>true</code>) or not (<code>false</code>).	Small, 55.0	55.0

Property Name	Type	Description	Filter Group and Version	Available Version
enrollmentChannel	String	The channel used by the member to enroll in the loyalty program.	Small, 55.0	55.0
enrollmentDate	String	The date on which the member enrolled in the loyalty program.	Small, 55.0	55.0
groupCreatedByMember	String	The membership number of the loyalty program member who created the group.	Small, 55.0	55.0
groupName	String	The name of the loyalty program group to which the member belongs.	Small, 55.0	55.0
lastActivityDate	String	The date on which the member performed their last activity.	Small, 55.0	55.0
loyaltyProgramMemberId	String	The ID of the loyalty program member.	Small, 55.0	55.0
loyaltyProgramName	String	The name of the loyalty program.	Small, 55.0	55.0
memberCurrencies	Member Currency Output[]	The currencies associated with the loyalty program member.	Small, 55.0	55.0
memberStatus	String	The status of the loyalty program member.	Small, 55.0	55.0
memberTiers	Member Tier Output[]	The tiers associated with the loyalty program member.	Small, 55.0	55.0
memberType	String	The type of loyalty program member.	Small, 55.0	55.0
membershipEndDate	String	The date on which the membership ends.	Small, 55.0	55.0
membershipLastRenewalDate	String	The date on which the loyalty program member last renewed their membership.	Small, 55.0	55.0
membershipNumber	String	The membership number of the loyalty program member.	Small, 55.0	55.0
referredBy	String	The membership number of the loyalty program member who referred the member to the loyalty program.	Small, 55.0	55.0
relatedCorporateMembershipNumber	String	The membership number of the Corporate type loyalty program member associate with the member.	Small, 55.0	55.0
transactionJournalStatementFrequency	String	The frequency at which transaction journal statements must be delivered to the member. Possible values are: • Monthly • Quarterly	Small, 55.0	55.0

Property Name	Type	Description	Filter Group and Version	Available Version
~~transactionJournalStatementCreatedDate~~ transactionJournalStatementCreatedDate	String	The date on which the member's last transaction journal statement was generated.	Small, 55.0	55.0
~~transactionJournalStatementMethod~~ transactionJournalStatementMethod	String	The method used to deliver transaction journal statements to the member.	Small, 55.0	55.0

Member Currency

Output representation of the member currency details.

Property Name	Type	Description	Filter Group and Version	Available Version
currencyName	String	The name of the member currency.	Small, 54.0	54.0
currencyType	String	The type of member currency. Possible values are: - NonQualifying - Qualifying	Small, 54.0	54.0
~~escrowPointsBalance~~ escrowPointsBalance	Double	The number of points the member has in escrow for a currency.	Small, 54.0	54.0
pointsBalance	Double	The member's points balance for a currency.	Small, 54.0	54.0

Member Tier Output

Represents the details of the loyalty program member's associated tiers.

Property Name	Type	Description	Filter Group and Version	Available Version
~~additionalTierFields~~ additionalTierFields	Map<String, Object>	The details of the custom fields added in the Loyalty Member Tier object.	Small, 55.0	55.0
~~areTierBenefitsAssigned~~ areTierBenefitsAssigned	Boolean	Indicates whether the benefits associated with the tier have been assigned to the member (<code>true</code>) or not (<code>false</code>).	Small, 55.0	55.0
loyaltyMemberTierId	String	The ID of tier associated with the loyalty program member.	Small, 55.0	55.0
loyaltyMemberTierName	String	The name of the loyalty member tier.	Small, 55.0	55.0
tierChangeReason	String	The reason why the member's was assigned the current tier.	Small, 55.0	55.0

Property Name	Type	Description	Filter Group and Version	Available Version
tierChangeReasonType	String	The reason type why the member's was assigned the current tier.	Small, 55.0	55.0
tierEffectiveDate	String	The date on which the member was assigned the tier.	Small, 55.0	55.0
tierExpirationDate	String	The date on which the member's current tier expires.	Small, 55.0	55.0
tierGroupId	String	The ID of the tier group associated with the Loyalty Tier record.	Small, 55.0	55.0
tierGroupName	String	The name of the tier group.	Small, 55.0	55.0
tierId	String	The ID of the Loyalty Tier record.	Small, 55.0	55.0
tierSequenceNumber	Integer	The sequence number of the tier.	Small, 55.0	55.0

Member Points Balance Change

Output representation of the points balance change for a member.

Property Name	Type	Description	Filter Group and Version	Available Version
changeInMemberCurrencies	Member Currency Change[]	List of points balance changes for the member's currencies based on the transaction journal and promotions.	Small, 54.0	54.0
membershipNumber	String	The membership number of the loyalty program member who is associated with the transaction journal that's created.	Small, 54.0	54.0

Name And Id Output

Output representation of an object's name and ID.

Property Name	Type	Description	Filter Group and Version	Available Version
id	String	ID of the object.	Small, 60.0	60.0
name	String	Name of the object.	Small, 60.0	60.0

Participant Game Rewards List

Output representation of the participant game reward object.

Property Name	Type	Description	Filter Group and Version	Available Version
expirationDate	String	Date until which the game reward is valid.	Small, 60.0	60.0
~~gameParticipantRewardId~~	String	ID of the participant game reward object.	Small, 60.0	60.0
gameRewardId	String	ID of the game reward won. This field is null for a game yet to be played.	Small, 60.0	60.0
~~issuedRewardReference~~	String	Voucher ID, or the transaction journal ID associated with the voucher, or the loyalty points rewarded to the participant. This field is null for a game yet to be played.	Small, 60.0	60.0
sourceActivity	String	ID of the promotion, or the transaction journal that is associated with the activity because of which the participant was eligible to play the game. Examples of source activities include purchasing a product, and reviewing on social media. This field is null for a game yet to be played.	Small, 60.0	60.0
status	String	Status of the reward. Possible values are: - Expired - NoReward - Rewarded - YetToReward	Small, 60.0	60.0

Process Result

Output representation of results of the processed Transaction Journals records.

Property Name	Type	Description	Filter Group and Version	Available Version
~~memberPointsBalance~~	Member Points Balance[]	List of member currencies for which points balance has changed based on the Transaction Journal records that are created.	Small, 54.0	54.0
~~transactionJournalResult~~	Transaction Journal Result[]	Details of the Transaction Journal records that are created.	Small, 54.0	54.0

Points Change Output

Output representation of the points debited from or credited to the member's points balance by the transaction journal.

Property Name	Type	Description	Filter Group and Version	Available Version
changeInPoints	Double	Specifies the number of points that have been debited from or credited to the member by the transaction journal.	Small, 58.0	58.0
loyaltyMemberCurrency	String	Name of the loyalty member currency for which the transaction journal has debited or credited points.	Small, 58.0	58.0

Points Summary

Output representation of the point summary.

Property Name	Type	Description	Filter Group and Version	Available Version
changeInEscrowPoints Balance	Double	The change in the escrow points balance of the loyalty program member associated with the transaction journal.	Small, 55.0	55.0
changeInPointsBalance	Double	The change in the points balance of the loyalty program member associated with the transaction journal.	Small, 55.0	55.0
loyaltyProgramCurrency Name	String	The name of the loyalty program currency for which points are debited from or credited to the loyalty program member associated with the transaction journal.	Small, 55.0	55.0
loyaltyProgramCurrency Type	CurrencyType	The type of the loyalty program currency. Valid values are: - NonQualifying - Qualifying	Small, 55.0	55.0

Product Details

The details of a product associated with the recommended promotion.

Property Name	Type	Description	Filter Group and Version	Available Version
id	String	The ID of the product.	Big, 64.0	64.0
name	String	The name of the product.	Big, 64.0	64.0
personalization ContentId	String	The response ID of the product recommendation for the product.	Big, 64.0	64.0

Property Name	Type	Description	Filter Group and Version	Available Version
productImage Url	String	The URL of the image associated with the product.	Big, 64.0	64.0
sku	String	The SKU of the product.	Big, 64.0	64.0

Promotion Configuration Output

Output representation of the promotion’s details and its configuration.

JSON Example

```
{
  "apiExecutedDateTime": "2024-05-29T09:02:38.809Z",
  "promotionDetails": {
    "additionalFieldValues": {
      "attributes": {}
    },
    "displayName": "Tea Time",
    "endDateTime": "2024-07-31T08:58:00.000Z",
    "name": "Tea Time",
    "priorityNumber": 1,
    "promotionCode": "TEATIME",
    "promotionLimits": {},
    "startDateTime": "2024-05-01T08:58:00.000Z"
  },
  "promotionLastModifiedDateTime": "2024-05-29T09:01:51.000Z",
  "promotionRuleList": [
    {
      "apiName": "Eligibility_Rule_Unified_Promotions_0c8xx000000034L",
      "isEligibilityRule": true,
      "name": "Eligibility_Rule_Unified_Promotions_0c8xx000000034L_v1",
      "ruleActionList": [],
      "ruleFilterCriteriaList": [
        {
          "conditionLogic": "1",
          "name": "FilterCriteria1",
          "ruleActionList": [],
          "ruleConditionList": [
            {
              "operator": "Equals",
              "parameterName": "CurrencyISOCODE",
              "sequenceNumber": 1,
              "valueList": [
                "USD"
              ],
              "valueType": "Literal"
            }
          ],
          "sequenceNumber": 1,
          "type": "Exists"
        }
      ],
    }
  ],
}
```

```

{
  "conditionLogic": "1 AND 2",
  "name": "FilterCriteria2",
  "parentFilterCriteria": "FilterCriteria1",
  "ruleActionList": [],
  "ruleConditionList": [
    {
      "operator": "GreaterThanOrEquals",
      "parameterName": "ActivityStartDateTime",
      "sequenceNumber": 1,
      "valueList": [
        "2024-05-01 08:58:00"
      ],
      "valueType": "Literal"
    },
    {
      "operator": "LessThanOrEquals",
      "parameterName": "ActivityStartDateTime",
      "sequenceNumber": 2,
      "valueList": [
        "2024-07-31 08:58:00"
      ],
      "valueType": "Literal"
    }
  ],
  "sequenceNumber": 2,
  "type": "Exists"
},
{
  "conditionLogic": "1 AND 2",
  "name": "FilterCriteria3",
  "parentFilterCriteria": "FilterCriteria2",
  "ruleActionList": [],
  "ruleConditionList": [
    {
      "operator": "Equals",
      "parameterName": "ProductCatalog",
      "sequenceNumber": 1,
      "valueList": [
        "Beverages"
      ],
      "valueType": "Literal"
    },
    {
      "operator": "In",
      "parameterName": "EligibleCartLineProduct",
      "sequenceNumber": 2,
      "valueList": [
        "Tea"
      ],
      "valueType": "Literal"
    }
  ],
  "sequenceNumber": 3,

```

```

        "type": "Exists"
      }
    ],
    "ruleParameterList": [
      {
        "contextTag": "cartLineProductCatalogId",
        "dataType": "Text",
        "decimalPlaces": 0,
        "hasContextTagMapping": true,
        "isCollection": true,
        "isInput": true,
        "isOutput": false,
        "name": "ProductCatalog",
        "type": "Tag"
      },
      {
        "contextTag": "activityStartDate",
        "dataType": "DateTime",
        "decimalPlaces": 0,
        "hasContextTagMapping": true,
        "isCollection": true,
        "isInput": true,
        "isOutput": false,
        "name": "ActivityStartDateTime",
        "type": "Tag"
      },
      {
        "contextTag": "currencyISOCODE",
        "dataType": "Text",
        "decimalPlaces": 0,
        "hasContextTagMapping": true,
        "isCollection": true,
        "isInput": true,
        "isOutput": false,
        "name": "CurrencyISOCODE",
        "type": "Tag"
      },
      {
        "contextTag": "cartLineProductId",
        "dataType": "Text",
        "decimalPlaces": 0,
        "hasContextTagMapping": true,
        "isCollection": true,
        "isInput": true,
        "isOutput": false,
        "name": "EligibleCartLineProduct",
        "type": "Tag"
      }
    ],
    "sequenceNumber": 2147483647
  },
  {
    "apiName": "DiscountTea10Unified_Promotions_0c8xx000000034L",
    "isEligibilityRule": false,

```

```

"name": "DiscountTeal0Unified_Promotions_0c8xx000000034L_v1",
"ruleActionList": [],
"ruleFilterCriteriaList": [
  {
    "conditionLogic": "1",
    "name": "FinalFilterCriteria",
    "ruleActionList": [],
    "ruleConditionList": [
      {
        "operator": "GreaterThanOrEquals",
        "parameterName": "TransactionAmount",
        "sequenceNumber": 1,
        "valueList": [
          "25.0"
        ],
        "valueType": "Literal"
      }
    ],
    "sequenceNumber": 1,
    "type": "Exists"
  },
  {
    "conditionLogic": "1",
    "name": "NonLoyaltyMemberCriteria",
    "parentFilterCriteria": "BranchCriteria",
    "ruleActionList": [
      {
        "actionType": "SetValue",
        "name": "pricingReward1",
        "ruleActionParameterList": [
          {
            "name": "type",
            "sequenceNumber": 1,
            "value": "ProvideDiscount",
            "valueType": "Literal"
          },
          {
            "name": "discountLevel",
            "sequenceNumber": 2,
            "value": "Cart",
            "valueType": "Literal"
          },
          {
            "name": "discountValue",
            "sequenceNumber": 3,
            "value": "10.0",
            "valueType": "Literal"
          },
          {
            "name": "discountType",
            "sequenceNumber": 4,
            "value": "PercentageOff",
            "valueType": "Literal"
          }
        ]
      }
    ]
  }
]

```

```

        ],
        "sequenceNumber": 1
      }
    ],
    "ruleConditionList": [
      {
        "operator": "GreaterThanOrEquals",
        "parameterName": "ActivityStartDateTime",
        "sequenceNumber": 1,
        "valueList": [
          "1970-01-01 08:00:00"
        ],
        "valueType": "Literal"
      }
    ],
    "sequenceNumber": 3,
    "type": "Exists"
  },
  {
    "name": "BranchCriteria",
    "parentFilterCriteria": "FinalFilterCriteria",
    "ruleActionList": [],
    "ruleConditionList": [],
    "sequenceNumber": 2,
    "type": "Branch"
  }
],
"ruleParameterList": [
  {
    "contextTag": "transactionAmount",
    "dataType": "Currency",
    "decimalPlaces": 0,
    "hasContextTagMapping": true,
    "isCollection": true,
    "isInput": true,
    "isOutput": false,
    "name": "TransactionAmount",
    "type": "Tag"
  },
  {
    "dataType": "Text",
    "decimalPlaces": 0,
    "hasContextTagMapping": false,
    "isCollection": false,
    "isInput": false,
    "isOutput": true,
    "name": "discountLevel",
    "type": "Variable"
  },
  {
    "dataType": "Text",
    "decimalPlaces": 0,
    "hasContextTagMapping": false,
    "isCollection": false,

```

```

        "isInput": false,
        "isOutput": true,
        "name": "discountType",
        "type": "Variable"
    },
    {
        "contextTag": "activityStartDate",
        "dataType": "DateTime",
        "decimalPlaces": 0,
        "hasContextTagMapping": true,
        "isCollection": true,
        "isInput": true,
        "isOutput": false,
        "name": "ActivityStartDateTime",
        "type": "Tag"
    },
    {
        "dataType": "Text",
        "decimalPlaces": 0,
        "hasContextTagMapping": false,
        "isCollection": false,
        "isInput": false,
        "isOutput": true,
        "name": "type",
        "type": "Variable"
    },
    {
        "dataType": "Numeric",
        "decimalPlaces": 0,
        "hasContextTagMapping": false,
        "isCollection": false,
        "isInput": false,
        "isOutput": true,
        "name": "discountValue",
        "type": "Variable"
    }
],
"sequenceNumber": 1
},
{
    "apiName": "TeaAndCoffeeDiscount10Unified_Promotions_0c8xx000000034L",
    "isEligibilityRule": false,
    "name": "TeaAndCoffeeDiscount10Unified_Promotions_0c8xx000000034L_v1",
    "ruleActionList": [],
    "ruleFilterCriteriaList": [
        {
            "conditionLogic": "1",
            "name": "NonLoyaltyMemberCriteria",
            "parentFilterCriteria": "BranchCriteria",
            "ruleActionList": [
                {
                    "actionType": "SetValue",
                    "name": "pricingLineLevelReward1",
                    "ruleActionParameterList": [

```



```

    {
      "name": "type",
      "sequenceNumber": 1,
      "value": "ProvideDiscount",
      "valueType": "Literal"
    },
    {
      "name": "discountType",
      "sequenceNumber": 2,
      "value": "PercentageOff",
      "valueType": "Literal"
    },
    {
      "name": "discountLevel",
      "sequenceNumber": 3,
      "value": "Line",
      "valueType": "Literal"
    },
    {
      "name": "discountValue",
      "sequenceNumber": 4,
      "value": "10.0",
      "valueType": "Literal"
    },
    {
      "name": "discountProduct",
      "sequenceNumber": 5,
      "value": "Tea",
      "valueType": "Literal"
    },
    {
      "name": "discountProductId",
      "sequenceNumber": 6,
      "value": "Tea",
      "valueType": "Literal"
    },
    {
      "name": "discountAppliedAt",
      "sequenceNumber": 7,
      "value": "LineItem",
      "valueType": "Literal"
    }
  ],
  "sequenceNumber": 1
},
{
  "actionType": "SetValue",
  "name": "pricingLineLevelReward2",
  "ruleActionParameterList": [
    {
      "name": "type",
      "sequenceNumber": 1,
      "value": "ProvideDiscount",
      "valueType": "Literal"
    }
  ]
}

```

```

    },
    {
      "name": "discountType",
      "sequenceNumber": 2,
      "value": "PercentageOff",
      "valueType": "Literal"
    },
    {
      "name": "discountLevel",
      "sequenceNumber": 3,
      "value": "Line",
      "valueType": "Literal"
    },
    {
      "name": "discountValue",
      "sequenceNumber": 4,
      "value": "10.0",
      "valueType": "Literal"
    },
    {
      "name": "discountProduct",
      "sequenceNumber": 5,
      "value": "Coffee",
      "valueType": "Literal"
    },
    {
      "name": "discountProductId",
      "sequenceNumber": 6,
      "value": "Coffee",
      "valueType": "Literal"
    },
    {
      "name": "discountAppliedAt",
      "sequenceNumber": 7,
      "value": "LineItem",
      "valueType": "Literal"
    }
  ],
  "sequenceNumber": 2
}
],
"ruleConditionList": [
  {
    "operator": "GreaterThanOrEquals",
    "parameterName": "ActivityStartDateTime",
    "sequenceNumber": 1,
    "valueList": [
      "1970-01-01 08:00:00"
    ],
    "valueType": "Literal"
  }
],
"sequenceNumber": 4,
"type": "Exists"

```

```

    },
    {
      "conditionLogic": "1",
      "name": "ProductFilter1",
      "ruleActionList": [],
      "ruleConditionList": [
        {
          "operator": "In",
          "parameterName": "CartLineProduct",
          "sequenceNumber": 1,
          "valueList": [
            "Tea"
          ],
          "valueType": "Literal"
        }
      ],
      "sequenceNumber": 1,
      "type": "Loop"
    },
    {
      "name": "BranchCriteria",
      "parentFilterCriteria": "FinalFilterCriteria",
      "ruleActionList": [],
      "ruleConditionList": [],
      "sequenceNumber": 3,
      "type": "Branch"
    },
    {
      "conditionLogic": "1",
      "name": "FinalFilterCriteria",
      "ruleActionList": [],
      "ruleConditionList": [
        {
          "matchType": "Any",
          "operator": "GreaterThanOrEquals",
          "parameterName": "FilterProducts1",
          "sequenceNumber": 1,
          "valueList": [
            "6.0"
          ],
          "valueType": "Literal"
        }
      ],
      "sequenceNumber": 2,
      "type": "Exists"
    }
  ],
  "ruleParameterList": [
    {
      "dataType": "Text",
      "decimalPlaces": 0,
      "hasContextTagMapping": false,
      "isCollection": false,
      "isInput": false,

```

```

        "isOutput": true,
        "name": "discountLevel",
        "type": "Variable"
    },
    {
        "dataType": "Text",
        "decimalPlaces": 0,
        "hasContextTagMapping": false,
        "isCollection": false,
        "isInput": false,
        "isOutput": true,
        "name": "discountProduct",
        "type": "Variable"
    },
    {
        "dataType": "Text",
        "decimalPlaces": 0,
        "hasContextTagMapping": false,
        "isCollection": false,
        "isInput": false,
        "isOutput": true,
        "name": "discountProductId",
        "type": "Variable"
    },
    {
        "dataType": "Text",
        "decimalPlaces": 0,
        "hasContextTagMapping": false,
        "isCollection": false,
        "isInput": false,
        "isOutput": true,
        "name": "discountType",
        "type": "Variable"
    },
    {
        "contextTag": "activityStartDate",
        "dataType": "DateTime",
        "decimalPlaces": 0,
        "hasContextTagMapping": true,
        "isCollection": true,
        "isInput": true,
        "isOutput": false,
        "name": "ActivityStartDateTime",
        "type": "Tag"
    },
    {
        "contextTag": "cartLineProductId",
        "dataType": "Text",
        "decimalPlaces": 0,
        "hasContextTagMapping": true,
        "isCollection": true,
        "isInput": true,
        "isOutput": false,
        "name": "CartLineProduct",

```

```

        "type": "Tag"
      },
      {
        "dataType": "Text",
        "decimalPlaces": 0,
        "hasContextTagMapping": false,
        "isCollection": false,
        "isInput": false,
        "isOutput": true,
        "name": "type",
        "type": "Variable"
      },
      {
        "dataType": "Numeric",
        "decimalPlaces": 0,
        "hasContextTagMapping": false,
        "isCollection": false,
        "isInput": false,
        "isOutput": true,
        "name": "discountValue",
        "type": "Variable"
      },
      {
        "aggregateFunction": "Sum",
        "dataType": "Tuple",
        "decimalPlaces": 0,
        "groupByContextTag": "cartLineProductId",
        "hasContextTagMapping": true,
        "isCollection": true,
        "isInput": true,
        "isOutput": true,
        "name": "FilterProducts1",
        "relatedFilterCriteriaName": "ProductFilter1",
        "type": "Accumulate",
        "value": "cartLineItemQuantity"
      },
      {
        "dataType": "Text",
        "decimalPlaces": 0,
        "hasContextTagMapping": false,
        "isCollection": false,
        "isInput": false,
        "isOutput": true,
        "name": "discountAppliedAt",
        "type": "Variable"
      }
    ],
    "sequenceNumber": 2
  }
],
"relatedRuleLibrary": {
  "apiName": "GPMRuleLibrary",
  "description": "GPMRuleLibrary",
  "name": "GPMRuleLibrary_v1",

```

```

    "relatedContextDefinitionDeveloperName": "GPMContext"
  }
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
apiExecutedDateTime	String	The date and time on which the API was executed.	Small, 61.0	61.0
promotionDetails	Unified Promotions Output[]	The details of the promotion.	Small, 61.0	61.0
promotionLastModifiedTime	String	The date and time on which the promotion was last modified.	Small, 61.0	61.0
promotionRuleList	Promotion Rule[]	The rules of the promotion. The rules are represented as context rules in the response.	Small, 61.0	61.0
relatedRuleLibrary	Rule Library[]	The rule library that's associated with the promotion.	Small, 61.0	61.0

Promotion Coupon Availability

Output representation of the promotion coupon availability.

Property Name	Type	Description	Filter Group and Version	Available Version
coupon	Promotion Coupon on page 574	Details of the coupon that is eligible for the promotion.	Small, 60.0	60.0
couponAvailabilityMessage	String	Specifies reasons for coupon ineligibility for the promotion. Possible values are: • MultipleActiveCoupons: Multiple active coupons that are eligible for the cart on the activity date. • NoActiveCoupon: No coupons are active on the cart activity date • SingleActiveLimitReached: Only one coupon is active on the cart activity date but it has reached its usage limit.	Small, 60.0	60.0

Promotion Coupon

Output representation of the promotion coupon.

Property Name	Type	Description	Filter Group and Version	Available Version
couponCode	String	Unique code of the coupon.	Small, 60.0	60.0
endDateTime	DateTime	End date and time of the coupon.	Small, 60.0	60.0
startDateTime	DateTime	Start date and time of the coupon.	Small, 60.0	60.0
status	String	Status of the coupon.	Small, 60.0	60.0

Promotion Details

Output representation of the details of an eligible promotion.

Property Name	Type	Description	Filter Group and Version	Available Version
additionalPromotionFields	Map<String, Object>	Values of the promotion fields that are mapped in the context definition.	Big, 60.0	60.0
couponDetails	Promotion Coupon Availability on page 574	Provides the details of the coupon eligible for the recommended promotion or specifies the reason why there isn't an eligible coupon for the promotion.	Big, 60.0	60.0
currency IsoCode	String	Three-letter ISO 4217 code of the active monetary currency associated with the promotion.	Small, 60.0	60.0
description	String	The description of the promotion.	Big, 60.0	60.0
displayName	String	Name of the eligible promotion.	Small, 60.0	60.0
endDateTime	String	End date and time of the promotion.	Big, 60.0	60.0
isAutomatic	Boolean	Indicates whether the promotion is automatically applied for eligible carts (<code>true</code>) or not (<code>false</code>). The default value is <code>false</code> .	Small, 63.0	63.0
priorityNumber	Integer	Number that determines the priority of the promotion in a list of eligible promotions. Multiple promotions can have the same priority number.	Small, 60.0	60.0
promotionCode	String	Code of the eligible promotion.	Small, 60.0	60.0
promotionEligibleRules	Promotion Rules List []	List of promotion rules that are eligible for the customer's cart.	Big, 60.0	60.0
promotionId	String	ID of the eligible promotion.	Big, 60.0	60.0

Property Name	Type	Description	Filter Group and Version	Available Version
promotionLimits	Promotion Limit []	List of the promotion's cart level and cart item level limits.	Big, 60.0	60.0
startDateTime	String	Start date and time of the promotion.	Big, 60.0	60.0
termsAnd Condition	String	Terms and condition of the promotion.	Big, 60.0	60.0

Promotion Eligibility Output

Output representation of the promotion eligibility criteria.

Property Name	Type	Description	Filter Group and Version	Available Version
areAllLoyaltyMembersEligible	Boolean	Indicates whether all loyalty members are eligible (<code>true</code>) or not (<code>false</code>).	Small, 60.0	60.0
eligibleCampaign	String	A campaign whose members are eligible for the promotion.	Small, 60.0	60.0
eligibleChannels	Promotion Eligible Channel Output []	Channels that are eligible for the promotion.	Small, 60.0	60.0
eligibleCustomerEvents	Events Detail Output	Customer events that are eligible for the promotion.	Small, 60.0	60.0
eligibleEnrollmentPeriod	Promotion Eligible Enrollment Period Output	Dates within which customers can enroll for the promotion.	Small, 60.0	60.0
eligibleLoyaltyPartnerProducts	Name And Id Output []	List of loyalty partner products that are eligible for the promotion.	Small, 60.0	60.0
eligibleLoyaltyTiers	Eligible Loyalty Tiers Output []	Loyalty tiers eligible for the promotion.	Small, 60.0	60.0
eligibleProductCategories	Name And Id Output []	Product categories eligible for the promotion.	Small, 60.0	60.0
eligibleProducts	Name And Id Output []	Products eligible for the promotion.	Small, 60.0	60.0
eligibleSegments	Name And Id Output []	Segments eligible for the promotion.	Small, 60.0	60.0
productCatalog	String	Product catalog eligible for the promotion.	Small, 60.0	60.0

Promotion Eligible Channel Output

Output representation of the eligible channel for a promotion.

Property Name	Type	Description	Filter Group and Version	Available Version
channel	Name And Id Output []	ID or name of the channel eligible for the promotion. Possible names for the online channel type are: • App • Mobile • Web Possible names for the social channel type are: • Facebook • Instagram • Twitter • WhatsApp	Small, 60.0	60.0
channelType	String	Type of the channel eligible for the promotion. Possible values are: • Offline • Online • Social	Small, 60.0	60.0
retailLocationGroup	Name And Id Output []	ID or name of the retail location group eligible for the promotion. This field is applicable only for the offline channel type.	Small, 60.0	60.0
store	Name And Id Output []	ID or name of the retail store eligible for the promotion. This field is applicable only for the offline channel type.	Small, 60.0	60.0

Promotion Eligible Enrollment Period Output

Output representation of the enrollment eligibility details of a promotion.

Property Name	Type	Description	Filter Group and Version	Available Version
enrollmentEndDate	String	Date until which customers can enroll for the promotion.	Small, 60.0	60.0
enrollmentStartDate	String	Date from which customers can enroll for the promotion.	Small, 60.0	60.0

Property Name	Type	Description	Filter Group and Version	Available Version
isEnrollmentRequired	Boolean	Indicates whether customers must enroll for the promotion (<code>true</code>) or not (<code>false</code>).	Small, 60.0	60.0

Promotion Details

Output representation of the details of an eligible promotion.

Property Name	Type	Description	Filter Group and Version	Available Version
additionalPromotionFields	Map<String, Object>	Values of the promotion fields that are mapped in the context definition.	Big, 60.0	60.0
couponDetails	Promotion Coupon Availability on page 574	Provides the details of the coupon eligible for the recommended promotion or specifies the reason why there isn't an eligible coupon for the promotion.	Big, 60.0	60.0
currency IsoCode	String	Three-letter ISO 4217 code of the active monetary currency associated with the promotion.	Small, 60.0	60.0
description	String	The description of the promotion.	Big, 60.0	60.0
displayName	String	Name of the eligible promotion.	Small, 60.0	60.0
endDateTime	String	End date and time of the promotion.	Big, 60.0	60.0
isAutomatic	Boolean	Indicates whether the promotion is automatically applied for eligible carts (<code>true</code>) or not (<code>false</code>). The default value is <code>false</code> .	Small, 63.0	63.0
priorityNumber	Integer	Number that determines the priority of the promotion in a list of eligible promotions. Multiple promotions can have the same priority number.	Small, 60.0	60.0
promotionCode	String	Code of the eligible promotion.	Small, 60.0	60.0
promotionEligibleRules	Promotion Rules List []	List of promotion rules that are eligible for the customer's cart.	Big, 60.0	60.0
promotionId	String	ID of the eligible promotion.	Big, 60.0	60.0
promotionLimits	Promotion Limit []	List of the promotion's cart level and cart item level limits.	Big, 60.0	60.0
startDateTime	String	Start date and time of the promotion.	Big, 60.0	60.0

Property Name	Type	Description	Filter Group and Version	Available Version
termsAndCondition	String	Terms and condition of the promotion.	Big, 60.0	60.0

Promotion Limit

Output representation of the promotion limit of an eligible promotion.

Property Name	Type	Description	Filter Group and Version	Available Version
limitCurrentAttained	Double	Value of the promotion's limit that has been attained so far.	Small, 60.0	60.0
limitTarget	Double	Value beyond which the promotion can't be applied for customer carts.	Small, 60.0	60.0
limitType	String	Type of the promotion limit. Possible values are: • CartItemCount—Maximum number of cart items the promotion applies to. This value maps to Per Line Item in Salesforce Lightning. • OrderCount—Maximum number of carts the promotion applies to. This value maps to Per Cart in Salesforce Lightning.	Small, 60.0	60.0

Promotion Limit Output

Output representation of promotion limits.

Property Name	Type	Description	Filter Group and Version	Available Version
liabilityLimit	Double	Maximum financial liability the promotion can incur.	Small, 60.0	60.0
liabilityPerCustomerLimit	Double	Maximum financial liability the promotion can incur for a customer.	Small, 60.0	60.0
loyaltyProgramCurrency	String	Active loyalty program currency associated with the loyalty points.	Small, 60.0	60.0
perCartLimit	Integer	Maximum number of carts that the promotion can be applied for.	Small, 60.0	60.0

Property Name	Type	Description	Filter Group and Version	Available Version
perLineItemLimit	Integer	Maximum number of cart items that the promotion can be applied for.	Small, 60.0	60.0
pointsLimit	Double	Maximum number of points the promotion can credit.	Small, 60.0	60.0
pointsPerCustomerLimit	Integer	Maximum number of points the promotion can credit to a customer.	Small, 60.0	60.0
usePerCustomerLimit	Integer	Maximum number of times the promotion can be applied for a customer.	Small, 60.0	60.0
vouchersLimit	Integer	Maximum number of vouchers the promotion can issue.	Small, 60.0	60.0
vouchersPerCustomerLimit	Integer	Maximum number of vouchers the promotion can issue to a customer.	Small, 60.0	60.0

Promotion

Output representation of the promotion with rule configuration.

Property Name	Type	Description	Filter Group and Version	Available Version
promotionWarnings	String[]	List of warnings due to which the promotion wasn't created.	Small, 65.0	65.0
promotionDetails	Unified Promotions Output[]	Details of promotion that's created or updated.	Small, 64.0	64.0
rules	Template Configuration[]	List of template configuration.	Small, 64.0	64.0

Promotion Recommendations

The count and list of promotions recommended for the member or customer.

JSON Example

```
{
  "recommendationCount": 0,
  "personalizationId": "string",
  "recommendationList": [
    {
      "rank": 0,
      "promotionDetails": {
        "id": "string",
        "promotionCode": "string",
        "name": "string",
```

```
    "description": "string",
    "displayName": "string",
    "productDetails": {
      "personalizationContentId": "string",
      "name": "string",
      "id": "string",
      "sku": "string",
      "productImageUrl": "string"
    },
    "priority": 0,
    "currencyIsoCode": "string",
    "coupon": {
      "couponAvailabilityMessage": "string",
      "couponDetails": {
        "couponCode": "string",
        "startDateTime": "2025-03-06",
        "endDateTime": "2025-03-06",
        "status": "string"
      }
    },
    "isAutomatic": true,
    "startDateTime": "2025-03-06",
    "endDateTime": "2025-03-06",
    "promotionPageUrl": "string",
    "imageUrl": "string",
    "isEnrollmentRequired": true,
    "isEnrolled": true,
    "termsAndCondition": "string"
  }
},
"loyaltyProgramName": "string",
"membershipNumber": "string",
"loyaltyProgramMemberId": "string"
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
loyalty Program MemberId	String	Enter the Loyalty Program member ID.	Big, 64.0	64.0
loyalty ProgramName	String	Enter the name of the loyalty program.	Big, 64.0	64.0
membership Number	String	Enter the membership number.	Big, 64.0	64.0
personalization Id	String	Enter the response ID of the product recommendations.	Big, 64.0	64.0
recommendation Count	Integer	Enter the number of promotion recommendations provided.	Big, 64.0	64.0

Property Name	Type	Description	Filter Group and Version	Available Version
recommendationList	Recommendation Details on page 588	Enter the list of recommended promotions.	Big, 64.0	64.0

Promotion Reward Application

Output representation details of the transaction journals that are created and processed.

JSON example

This example shows a sample response.

```
{
  "transactionJournalList": [
    {
      "discountsAppliedList": [
        {
          "discountAmount": 15
        }
      ],
      "gamesAssignedList": [
        {
          "gameDefinitionName": "SpintheWheelGame",
          "gameExpirationDate": "2025-10-10T00:00:00.000Z",
          "gameParticipantRewardId": "3PRSB0000004LFZ"
        }
      ],
      "id": "01VSB00000E76ev",
      "voucherIssuedList": [
        {
          "effectiveDate": "2025-10-09T00:00:00.000Z",
          "expirationDate": "2025-10-10T00:00:00.000Z",
          "faceValue": 100,
          "promotionId": "0c8SB000000HASn",
          "voucherCode": "SALE-4F8G-RT2P",
          "voucherDefinitionName": "$100 off",
          "voucherId": "0kDSB0000000LNp"
        }
      ]
    }
  ]
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
transactionJournalList	Transaction Journal Details []	List of the transaction journals that are processed and created.	Big, 65.0	65.0

Promotion Reward Details

Output representation of the details of the reward offered by an eligible promotion rule.

Property Name	Type	Description	Filter Group and Version	Available Version
rewardDetails	Map<String, Object>	Details of the reward offered by the promotion rule.	Small, 60.0	60.0
rewardType	String	Type of the reward offered by the promotion rule. Possible values are: • AssignBadge—Assigns a badge to the loyalty program member. • AssignGame—Enrolls the customer in a game, playing which the customer can win rewards. • CreditFixedPoints—Credits a fixed number of loyalty points to the loyalty program member. • CreditMultiplierPoints—Credits loyalty points that are based on the multiplier and transaction journal fields to the loyalty program member. • GiveFreeProduct—provides the customer a free product on the cart or cart items. • IssueVoucher—Issues a voucher to the customer. • ProvideDiscount—Provides the customer a discount on the cart or cart items.	Small, 60.0	60.0

Promotion Rule

Output representation of the details of a promotion rule.

Property Name	Type	Description	Filter Group and Version	Available Version
apiName	String	The API name of the promotion rule.	Small, 61.0	61.0
description	String	The description of the promotion rule.	Small, 61.0	61.0
hasQualificationCriteria	Boolean	Indicates whether the promotion rule can be applied without any qualification criteria (true) or not (false).	Small, 65.0	65.0

Property Name	Type	Description	Filter Group and Version	Available Version
isEligibilityRule	Boolean	Indicates whether the rule is the promotion eligibility rule (<code>true</code>) or not (<code>false</code>).	Small, 61.0	61.0
isRuleApplicableToCategories	Boolean	Indicates whether the rule applies to all eligible categories associated with the promotion (<code>true</code>) or not (<code>false</code>).	Small, 65.0	65.0
isRuleApplicableToProducts	Boolean	Indicates whether the rule applies to all eligible products associated with the promotion (<code>true</code>) or not (<code>false</code>).	Small, 65.0	65.0
name	String	The name of the promotion rule. A promotion consists of two types of rules. The eligibility rule decides which products, categories, stores, store groups, and customers are eligible for the promotion. The template rules decide which customer activities are eligible for the promotion and the rewards available for each eligible customer activity.	Small, 61.0	61.0
ruleActionList	Rule Action []	The list of actions that belong to the rule.	Small, 61.0	61.0
ruleFilterCriteriaList	Rule Filter Criteria []	The list of filter criteria that belong to the rule.	Small, 61.0	61.0
ruleParameterList	Rule Parameter []	The list of parameters that belong to the rule.	Small, 61.0	61.0
sequenceNumber	Integer	The sequence number of the rule in the promotion rules list.	Small, 61.0	61.0

Promotion Rules List

Output representation of an eligible promotion rule.

Property Name	Type	Description	Filter Group and Version	Available Version
ruleName	String	Name of the eligible promotion rule.	Small, 60.0	60.0
rulePriority	Integer	Rule priority in the promotion which is eligible for rewards.	Big, 59.0	59.0
ruleRewards	Promotion Reward Details []	Rewards associated with the eligible rule.	Small, 60.0	60.0

Rule Action

Output representation of the details of a promotion rule action.

Property Name	Type	Description	Filter Group and Version	Available Version
actionType	String	The type of promotion rule action.	Small, 61.0	61.0
name	String	The name of the action associated with either a promotion rule or a promotion rule criteria.	Small, 61.0	61.0
ruleActionParameterList	Rule Action Parameter[]	The list of action parameters that belong to the rule or criteria action.	Small, 61.0	61.0
sequenceNumber	Integer	The sequence number of the action in either a promotion rule or a promotion rule criteria.	Small, 61.0	61.0

Rule Action Parameter

Output representation of the details of a promotion rule action parameter.

Property Name	Type	Description	Filter Group and Version	Available Version
name	String	The name of the parameter associated with the promotion rule action.	Small, 61.0	61.0
relatedFilterCriteriaName	String	The name of the rule filter criteria that's associated with the rule action parameter.	Small, 61.0	61.0
sequenceNumber	Integer	The sequence number of the promotion rule action parameter.	Small, 61.0	61.0
value	String	The value of the promotion rule action parameter.	Small, 61.0	61.0
valueType	String	The type of value.	Small, 61.0	61.0

Rule Condition

Output representation of the details of a promotion rule condition.

Property Name	Type	Description	Filter Group and Version	Available Version
matchType	String	Specifies whether the condition qualifies as true when all the values in the value list match the condition's requirements or when any one of the values in the value list match the condition's requirements.	Small, 61.0	61.0

Property Name	Type	Description	Filter Group and Version	Available Version
operator	String	The operator of the rule filter criteria condition.	Small, 61.0	61.0
parameterName	String	The name of the parameter that's used to define the rule filter criteria condition.	Small, 61.0	61.0
sequenceNumber	Integer	The sequence number of the rule filter criteria condition in the rule condition list.	Small, 61.0	61.0
valueList	String[]	The value that the rule filter criteria checks the parameter against.	Small, 61.0	61.0
valueType	String	Specifies the type of the value.	Small, 61.0	61.0

Rule Filter Criteria

Output representation of the details of a promotion rule filter criteria.

Property Name	Type	Description	Filter Group and Version	Available Version
conditionLogic	String	The condition logic of the promotion rule's filter criteria.	Small, 61.0	61.0
name	String	The name of the promotion rule's filter criteria.	Small, 61.0	61.0
parentFilterCriteria	String	The parent filter criteria that the promotion rule filter criteria belongs to.	Small, 61.0	61.0
ruleActionList	Rule Action []	The list of actions that belong to the promotion rule filter criteria.	Small, 61.0	61.0
ruleConditionList	Rule Condition []	The list of conditions that belong to the promotion rule filter criteria.	Small, 61.0	61.0
sequenceNumber	Integer	The sequence number of the promotion rule's filter criteria in the promotion rule.	Small, 61.0	61.0
type	String	Specifies the type of promotion rule filter criteria.	Small, 61.0	61.0

Rule Library

The details of the rule library associated with the promotion.

Property Name	Type	Description	Filter Group and Version	Available Version
apiName	String	The API name of the rule library associated with the promotion.	Small, 61.0	61.0
description	String	The description of the rule library associated with the promotion.	Small, 61.0	61.0
name	String	The name of the rule library associated with the promotion.	Small, 61.0	61.0
ruleDefinitionName	String	The developer name of the context definition that's associated with the promotion's rule library.	Small, 61.0	61.0

Rule Parameter

Output representation of the details of a promotion rule parameter.

Property Name	Type	Description	Filter Group and Version	Available Version
aggregateFunction	String	The aggregate function that's used in the rule parameter.	Small, 61.0	61.0
contextTag	String	The context tag that is used in the rule parameter. The context tag belongs to the context definition associated with the promotion's rule library.	Small, 61.0	61.0
dataType	String	The data type of the rule parameter.	Small, 61.0	61.0
decimalPlaces	Integer	The number of decimal places supported by a numeric type rule parameter.	Small, 61.0	61.0
groupByContextTag	String	The customer activity inputs are grouped by the context tag.	Small, 61.0	61.0
hasContextTagMapping	Boolean	Indicates whether the context tag has a corresponding context tag mapping. The context mapping is available in the active version of the rule library associated with the promotion.	Small, 61.0	61.0
isCollection	Boolean	Indicates whether the parameter stores a list of comma-separated values.	Small, 61.0	61.0
isInput	Boolean	Indicates whether the parameter stores a value that's used to process the rule.	Small, 61.0	61.0
isOutput	Boolean	Indicates whether the parameter stores a value that's provided after the rule is processed.	Small, 61.0	61.0

Property Name	Type	Description	Filter Group and Version	Available Version
name	String	The name of the parameter associated with the rule.	Small, 61.0	61.0
relatedFilterCriteriaName	String	The name of the rule filter criteria that's associated with the rule parameter.	Small, 61.0	61.0
type	String	The type of rule parameter.	Small, 61.0	61.0
value	String	The value of the rule parameter.	Small, 61.0	61.0

Recommendation Details

The rank and details of the recommended promotions.

Property Name	Type	Description	Filter Group and Version	Available Version
promotion Details	Promotion Details on page 578	The details of the recommended promotions.	Big, 64.0	64.0
rank	Integer	The priority sequence of how the recommended promotions are displayed.	Big, 64.0	64.0

Redeem Voucher Output

Represents the response for the redeem voucher POST method.

Sample Response: Reserve

```
{
  "reservationKey": "key1",
  "reservedValue": 16.34,
  "status": "Reserved"
  "currencyIsoCode": "EUR"
}
```

Sample Response: Reinstate - Reserved to Issued

```
{
  "remainingAmount": 98.32,
  "status": "Issued"
}
```

Sample Response: Reinstate - Reserved to Expired

```
{
  "status": "Expired"
}
```

Sample Response: Redeem

```
{
  "remainingAmount": 98.32,
  "status": "Redeemed",
  "currencyIsoCode": "EUR"
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
remainingAmount	Double	The voucher value after redemption.  Note: If the complete value of the voucher is redeemed, the remaining amount is zero.	Small, 53.0	53.0
currencyIsoCode	String	The ISO code of the currency.	Small, 53.0	53.0
status	VoucherStatus (Enum)	Specifies the status of the voucher.	Small, 62.0	62.0
reservationKey	String	Specifies the key used to redeem or reinstate a reserved voucher.	Small, 62.0	62.0
reservedValue	Currency	Specifies the amount reserved for partially redeemable vouchers.	Small, 62.0	62.0

Summary

Output representation of the execution summary of the transaction journals.

Property Name	Type	Description	Filter Group and Version	Available Version
pointsSummary	Points Summary []	A summary of points that are debited from or credited to the points balance or the escrow points balance of the loyalty program member associated with the transaction journal.	Small, 55.0	55.0
vouchersIssued Count	Integer	The number of vouchers that are issued to the loyalty program member associated with the transaction journal.	Small, 55.0	55.0

Template Configuration

Output representation of the template configuration.

Property Name	Type	Description	Filter Group and Version	Available Version
eventConfiguration	List<Map<String, Object>>	Criteria that decides when customers are eligible for the rewards of the promotion rule.	Small, 60.0	60.0
journalSubType	String	Journal sub-type of the rule in the promotion.	Small, 60.0	60.0
journalSubTypeName	String	Journal sub-type name of the rule in the promotion.	Small, 64.0	64.0
journalType	String	Journal type of the rule in the promotion.	Small, 60.0	60.0
priority	Integer	Priority number of the rule in the promotion.	Small, 60.0	60.0
rewardConfiguration	List<Map<String, Object>>	Rewards that members get when their activity is meet rule's eligibility criteria.	Small, 60.0	60.0
ruleName	String	Name of the promotion rule.	Small, 60.0	60.0
templateName	String	Name of the unified promotion template that's used in a promotion rule.	Small, 60.0	60.0

Template List Configuration

The output representation of the list template configuration

Property Name	Type	Description	Filter Group and Version	Available Version
rules	Template Configuration[]	The list of template configuration.	Small, 60.0	60.0

Transaction Journal Details

The details of the transaction journal that's created.

JSON example

This example shows a sample response.

```
{
  "discountsAppliedList": [
    {
      "discountType": "ProvideDiscount",
      "discountAmount": 15
    }
  ],
  "gamesAssignedList": [
    {
      "gameDefinitionName": "SpintheWheelGame",

```

```

        "gameExpirationDate": "2025-10-10T00:00:00.000Z",
        "gameParticipantRewardId": "3PRSB0000004LFZ"
    }
],
"id": "01VSB00000E76ev",
"voucherIssuedList": [
    {
        "discountPercent":10,
        "effectiveDate": "2025-10-09T00:00:00.000Z",
        "expirationDate": "2025-10-10T00:00:00.000Z",
        "faceValue": 100,
        "productId":"01txx0000006iTsAAI",
        "promotionId": "0c8SB000000HASn",
        "voucherCode": "SALE-4F8G-RT2P",
        "voucherDefinitionName": "$100 off",
        "voucherId": "0kDSB0000000LNP"
    }
]
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
discountsAppliedList	Discount Details[]	List of discounts applied to the cart for customers.	Big, 65.0	65.0
gamesAssignedList	Game Details[]	List of games assigned to the customers.	Big, 65.0	65.0
id	String	The ID of the transaction journal that is created.	Big, 65.0	65.0
voucherIssuedList	Voucher Details[]	List of vouchers issued to the customers.	Big, 65.0	65.0

Transaction Details Output

Output representation of the transaction journal ledger records.

Property Name	Type	Description	Filter Group and Version	Available Version
activityDate	String	The activity date of the transaction journal.	Small, 58.0	58.0
~~externalTransactionId~~	String	The unique transaction number from an external system.	Small, 58.0	58.0
journalSubTypeName	String	The name of the journal subtype associated with the transaction journal record.	Small, 58.0	58.0
journalTypeName	String	The name of the journal type associated with the transaction journal record.	Small, 58.0	58.0
pointsChange	Points Change Output	The details of the points debited from or credited to the member's points balance by the transaction journal.	Small, 58.0	58.0

Property Name	Type	Description	Filter Group and Version	Available Version
~~productCategoryName~~	String	The name of the product category associated with the transaction journal.	Small, 58.0	58.0
productName	String	The name of the product associated with the transaction journal.	Small, 58.0	58.0
transactionAmount	String	The amount that the member spent on the transaction journal.	Small, 58.0	58.0
transactionJournalId	String	The ID of the transaction journal record.	Small, 58.0	58.0
~~transactionJournalNumber~~	String	The unique number of the transaction journal record.	Small, 58.0	58.0

Transaction Journal Output

Output representation of the transaction journals.

Property Name	Type	Description	Filter Group and Version	Available Version
activityDate	String	The date and time of when transaction journal is created.	Small, 55.0	55.0
journalSubType	String	The journal subtype of the transaction journal.	Small, 55.0	55.0
journalType	String	The journal type of the transaction journal.	Small, 55.0	55.0
loyaltyProgram	String	The loyalty program related with the transaction journal.	Small, 55.0	55.0
~~loyaltyProgramMember~~	String	The membership number of the loyalty program related to the transaction journal.	Small, 55.0	55.0
referredMember	String	The membership number of the loyalty program member who has been referred by the member.	Small, 55.0	55.0
status	String	The status of the transaction journal.	Small, 55.0	55.0
~~transactionJournalId~~	String	The ID of the transaction journal that is created.	Small, 55.0	55.0

Transaction Journal Map Output

Output representation of the loyalty program process simulation result.

Property Name	Type	Description	Filter Group and Version	Available Version
transactionJournals	Transaction Journal Result[]	Represents the transaction journals on which the simulation was performed.	Small, 55.0	55.0

Transaction Journal Result

Output representation of the transaction journals result.

Property Name	Type	Description	Filter Group and Version	Available Version
errorMessage	String	Specifies the error message if the process has no rules or if no process qualifies for the transaction journal.	Small, 55.0	55.0
executionSummary	Summary	Represents the execution summary of the transaction journals.	Small, 55.0	55.0
processId	String	The ID of the loyalty program process used to process the transaction journals.	Small, 55.0	55.0
processName	String	The name of the loyalty program process used to process the transaction journals.	Small, 55.0	55.0
processRules	Loyalty Program Process Rule[]	The list of rules defined in the loyalty program process used to process the transaction journals.	Small, 55.0	55.0

Transaction Journal Process Result

Output representation of the Transaction Journal result.

Property Name	Type	Description	Filter Group and Version	Available Version
badgesAssigned	Badge Output[]	List of badges that are assigned to the member.	Small, 60.0	60.0
changeInMemberPoints	Member Points Balance Change[]	Details of points that are credited to or debited from the member's points balance and escrow points balance for all the currencies associated with the member.	Small, 54.0	54.0
gamesAssigned	Game Assignment Output[]	List of games that are assigned to the member.	Small, 60.0	60.0
id	String	Details of the Transaction Journal records that are created.	Small, 54.0	54.0
voucherResult	Voucher Result	List of vouchers that are issued or redeemed.	Small, 54.0	54.0

Transaction Ledgers Output

Output representation of the transaction journals and the ledgers associated with the specified loyalty program member.

Sample Response

```
{
  "message": null,
  "status": true,
  "transactionJournalCount": 2,
  "externalTransactionNumber" : "123123"
  "transactionJournals": [
    {
      "activityDate": "2023-03-01T08:00:00.000Z",
      "additionalAttributes": [
        {
          "fieldName": "additionalInfo__c",
          "value": "null"
        },
        {
          "fieldName": "Destination",
          "value": "null"
        }
      ],
      "journalSubTypeName": "Product Buy",
      "journalTypeName": "Accrual",
      "pointsChange": [],
      "transactionJournalId": "01Vxx00000002rREQ"
    },
    {
      "activityDate": "2023-02-01T20:00:00.000Z",
      "additionalAttributes": [
        {
          "fieldName": "additionalInfo__c",
          "value": "test123"
        },
        {
          "fieldName": "Destination",
          "value": "Jaipur"
        }
      ],
      "journalSubTypeName": "Member Enrollment",
      "journalTypeName": "Accrual",
      "pointsChange": [
        {
          "changeInPoints": "-100.0",
          "loyaltyMemberCurrency": "Tier Points"
        },
        {
          "changeInPoints": "90.0",
          "loyaltyMemberCurrency": "Reward Points"
        }
      ],
      "productCategoryName": "Gift",
      "productName": "New Member Gift Package",
    }
  ]
}
```

```

    "transactionAmount": "1000.0",
    "transactionJournalId": "01Vxx00000000H1EAI"
  }
]
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
message	String	The request response message.	Small, 58.0	58.0
status	Boolean	Indicates whether the request was successful (<code>true</code>) or not (<code>false</code>).	Small, 58.0	58.0
transactionJournalCount	Integer	The number of transaction journal records created for the member.	Small, 58.0	58.0
transactionJournals	Transaction Details Output	The details of the transaction journals.	Small, 58.0	58.0

Update Cumulative Usage Action Output

Output representation of the update cumulative usage action type.

Property Name	Type	Description	Filter Group and Version	Available Version
actionType	String	The type of action in the loyalty program process rule.	Small, 55.0	55.0
promotion	String	The name of promotion for which the member's cumulative usage is updated or reset.	Small, 55.0	55.0
promotionId	String	The ID of the promotion for which the member's cumulative usage is updated or reset.	Small, 55.0	55.0
stepName	String	The name of the step in the loyalty program process rule.	Small, 55.0	55.0
updatedCumulativePromotionUsage	Integer	The updated value of the loyalty program member's usage towards the cumulative promotion.	Small, 55.0	55.0

Unified Promotions

Output representation of the promotion details.

JSON example

```

{
  "additionalFieldValues": {

```

```

    "attributes":{
      "isUnifiedPromotion__c":"true"
    }
  },
  "description":"Buy 2 Get 1 free",
  "displayName":"Sample Promotion",
  "endTime":"2023-12-30T00:00:00.000Z",
  "id":"0c8SB0000000rCPYAY",
  "loyaltyProgram":{
    "id":"0lpSB00000001If",
    "name":"Program-001"
  },
  "name":"Free Beverage",
  "objective":"To increase sales",
  "priorityNumber":1,
  "promotionEligibility":{
    "eligibleChannels":[
      {
        "channelType":"Offline",
        "retailLocationGroup":{
          "id":"0Y2SB000000004r",
          "name":"test"
        },
        "store":{
          "id":"0YQSB000000006T",
          "name":"test Store"
        }
      }
    ],
    "eligibleCustomerEvents":{
      "event":"MemberBirthday",
      "eventPeriodUnit":"Date"
    },
    "eligibleEnrollmentPeriod":{
      "enrollmentEndDate":"2023-12-30T00:00:00.000Z",
      "enrollmentStartDate":"2023-06-30T00:00:00.000Z",
      "isEnrollmentRequired":true
    },
    "eligibleLoyaltyTiers":[
      {
        "tier":{
          "id":"0lgSB00000000srYAA",
          "name":"Silver"
        },
        "tierGroup":{
          "id":"0ltSB00000000T3YAI",
          "name":"Group-001"
        }
      }
    ],
    "eligibleProductCategories":[
      {
        "id":"0ZGSB00000002td7",
        "name":"Beverages"
      }
    ]
  }
}

```

```

    }
  ],
  "eligibleProducts": [
    {
      "id": "01tSB0000008Lth",
      "name": "Coke"
    }
  ]
},
"promotionLimits": {
  "liabilityLimit": 10000,
  "loyaltyProgramCurrency": {
    "id": "NQP",
    "name": "01cSB00000001Tx"
  },
  "perCartLimit": 5,
  "perLineItemLimit": 10,
  "pointsPerCustomerLimit": 5,
  "usePerCustomerLimit": 5,
  "vouchersLimit": 100,
  "vouchersPerCustomerLimit": 5
},
"startDateTime": "2023-06-30T00:00:00.000Z"
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
additionalFieldValues	Additional Attributes Output	Additional details of the promotion.	Small, 60.0	60.0
businessUnit	Name And Id Output []	ID or name of the Marketing Cloud business unit associated with the promotion.	Small, 60.0	60.0
currencyIsoCode	String	ISO code of the active currency associated with the promotion.	Small, 60.0	60.0
description	String	Description of the promotion.	Small, 60.0	60.0
displayName	String	Display name of the promotion.	Small, 60.0	60.0
endDateTime	String	Date and time until which the promotion can be applied for customers.	Small, 60.0	60.0
id	String	ID of the promotion.	Small, 60.0	60.0
isAutomatic	Boolean	Indicates whether the promotion is automatically applied for eligible carts (<code>true</code>) or not (<code>false</code>).	Small, 63.0	63.0
isEmailActivated	Boolean	Indicates whether the promotion's email has been activated (<code>true</code>) or not (<code>false</code>).	Small, 60.0	Small, 60.0
loyaltyProgram	String	Loyalty program associated with the promotion.	Small, 60.0	60.0

Property Name	Type	Description	Filter Group and Version	Available Version
name	String	Unique name of the promotion.	Small, 60.0	60.0
objective	String	Objective of the promotion.	Small, 60.0	60.0
priorityNumber	Integer	Number that determines the priority of the promotion in a list of eligible promotions.	Small, 60.0	60.0
promotionCode	String	Promotion code of the promotion.	Small, 60.0	60.0
promotionEligibility	Promotion Eligibility Output	Eligibility criteria of the promotion.	Small, 60.0	60.0
promotionLimits	Promotion Limit Output	Limits for the promotion.	Small, 60.0	60.0
ruleLibrary	String	Rule library associated with the promotion.	Small, 62.0	62.0
startDateTime	String	Start date and time of the promotion	Small, 60.0	60.0
termsAndConditions	String	Terms and conditions which are applied to the promotion.	Small, 63.0	63.0

Unified Promotion Templates

The output representation of the request to create a promotion.

The response varies based on the template selected for rule configuration.

Unlink Program Member Partner

Output representation of the program member and partner unlink request.

JSON example

```
{
  "message": null,
  "status": true
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
message	String	Message displayed when a loyalty program member is unlinked from a partner.	Big, 66.0	66.0
status	Boolean	Indicates whether the API request was successful (<code>true</code>) or not (<code>false</code>).	Big, 66.0	66.0

Voucher

Output representation of the details of a voucher.

Property Name	Type	Description	Filter Group and Version	Available Version
currencyIsoCode	String	The currency ISO code in which the member has redeemed the voucher.	Small, 58.0	58.0
description	String	The description of the voucher issued to the member.	Small, 58.0	58.0
discountPercent	Integer	The discount percentage that's offered to the member by the voucher.	Small, 58.0	58.0
effectiveDate	String	The date the member can redeem the voucher.	Small, 58.0	58.0
effectiveDateTime	DateTime	The date and time from when the voucher is available for redemption.	Small, 62.0	62.0
expirationDate	String	The date the voucher expires.	Small, 58.0	58.0
expirationDateTime	DateTime	The date and time when the voucher expires and is no longer available for redemption.	Small, 62.0	62.0
faceValue	Double	The value that the member can redeem using the voucher.	Small, 58.0	58.0
isVoucherDefinitionActive	Boolean	Indicates whether the voucher definition associated with the voucher is active (<code>true</code>) or not (<code>false</code>).	Small, 58.0	58.0
isVoucherPartiallyRedeemable	Boolean	Indicates whether the voucher can be redeemed across multiple transactions (<code>true</code>) or only one transaction (<code>false</code>).	Small, 58.0	58.0
hasTimeBasedVoucher	Boolean	Indicates whether the voucher has an effective date and time (<code>true</code>) or not (<code>false</code>).	Small, 62.0	62.0
partnerAccountName	String	The name of the partner associated with the voucher's voucher definition.	Small, 58.0	58.0
productCategoryId	String	The ID of product category associated with the voucher's voucher definition.	Small, 58.0	58.0
productCategoryName	String	The name of the product category associated with the voucher definition.	Small, 58.0	58.0
productId	String	The ID of the product associated with the voucher definition.	Small, 58.0	58.0
productName	String	The name of the product associated with the voucher definition.	Small, 58.0	58.0
promotionId	String	The ID of the promotion for which the voucher is issued.	Small, 58.0	58.0

Property Name	Type	Description	Filter Group and Version	Available Version
promotionName	String	The name of the promotion for which the voucher is issued.	Small, 58.0	58.0
redeemedValue	Double	The value that the member has already redeemed from a partially redeemable voucher.	Small, 58.0	58.0
remainingValue	Double	The remaining value of the voucher after partial redemption.	Small, 58.0	58.0
status	String	The status of the voucher.	Small, 58.0	58.0
type	String	The type of voucher issued to the member.	Small, 58.0	58.0
useDate	String	The date the voucher was redeemed.	Small, 58.0	58.0
voucherCode	String	The code of the voucher.	Small, 58.0	58.0
voucherDefinition	String	The name of the voucher definition used to issue the voucher to the member.	Small, 58.0	58.0
voucherId	String	The ID of the voucher record.	Small, 58.0	58.0
voucherImageUrl	String	The URL of the image associated with the voucher definition.	Small, 58.0	58.0
voucherNumber	String	The unique number of the voucher.	Small, 58.0	58.0

Voucher Details

The details of the voucher that's issued to the customer.

JSON example

This example shows a sample response.

```

"voucherIssuedList": [
  {
    "discountPercent":10,
    "effectiveDate": "2025-10-09T00:00:00.000Z",
    "expirationDate": "2025-10-10T00:00:00.000Z",
    "faceValue": 100,
    "productId":"01txx0000006iTsAAI",
    "promotionId": "0c8SB000000HASn",
    "voucherCode": "SALE-4F8G-RT2P",
    "voucherDefinitionName": "$100 off",
    "voucherId": "0kDSB0000000LNp"
  }
]

```


Property Name	Type	Description	Filter Group and Version	Available Version
discountPercent	Integer	Discount percentage associated with the voucher.	Big, 65.0	65.0
effectiveDate	String	Date when the voucher goes into effect.	Big, 65.0	65.0
expirationDate	String	Date on which the voucher expires.	Big, 65.0	65.0
faceValue	Double	Value of the voucher.	Big, 65.0	65.0
productId	String	The ID of the product in the voucher issued.	Big, 65.0	65.0
promotionId	String	The ID of the promotion associated with the voucher.	Big, 65.0	65.0
voucherCode	String	Code of the voucher issued.	Big, 65.0	65.0
voucherDefinitionName	String	Name of the voucher definition.	Big, 65.0	65.0
voucherId	String	The ID of the voucher issued.	Big, 65.0	65.0

Vouchers List

Output representation of the list of vouchers associated with the loyalty program member.

JSON example

```
{
  "voucherCount": 6,
  "vouchers": [
    {
      "effectiveDate": "2020-11-01",
      "effectiveDateTime": "2020-11-01T07:47:44.000Z",
      "expirationDate": "2021-11-01",
      "expirationDateTime": "2021-11-01T07:33:00.000Z",
      "faceValue": 100,
      "isVoucherDefinitionActive": true,
      "isVoucherPartiallyRedeemable": false,
      "hasTimeBasedVoucherPeriod": true,
      "redeemedValue": 0,
      "remainingValue": 100,
      "status": "Expired",
      "type": "FixedValue",
      "voucherCode": "J4XN26MT",
      "voucherDefinition": "Ajo",
      "voucherId": "0kDT1000000TUwsMAG",
      "voucherNumber": "00005852"
    },
    {
      "effectiveDate": "2020-11-01",
      "expirationDate": "2021-11-01",
      "faceValue": 100,
      "isVoucherDefinitionActive": true,
      "isVoucherPartiallyRedeemable": false,

```

```

    "hasTimeBasedVoucherPeriod": false,
    "redeemedValue": 0,
    "remainingValue": 100,
    "status": "Expired",
    "type": "FixedValue",
    "voucherCode": "6PM01LT8",
    "voucherDefinition": "Ajio",
    "voucherId": "0kDT1000000TUjEMAW",
    "voucherNumber": "00005006"
  },
  {
    "effectiveDate": "2020-11-01",
    "expirationDate": "2021-11-01",
    "faceValue": 100,
    "isVoucherDefinitionActive": true,
    "isVoucherPartiallyRedeemable": false,
    "hasTimeBasedVoucherPeriod": false,
    "redeemedValue": 0,
    "remainingValue": 100,
    "status": "Expired",
    "type": "FixedValue",
    "voucherCode": "LEEFAQGE",
    "voucherDefinition": "Ajio",
    "voucherId": "0kDT1000000TUQcMAO",
    "voucherNumber": "00003852"
  },
  {
    "effectiveDate": "2020-11-01",
    "expirationDate": "2021-11-01",
    "faceValue": 100,
    "isVoucherDefinitionActive": true,
    "isVoucherPartiallyRedeemable": false,
    "hasTimeBasedVoucherPeriod": false,
    "redeemedValue": 0,
    "remainingValue": 100,
    "status": "Expired",
    "type": "FixedValue",
    "voucherCode": "9WOYZK2G",
    "voucherDefinition": "Ajio",
    "voucherId": "0kDT1000000TUCyMAO",
    "voucherNumber": "00003006"
  },
  {
    "effectiveDate": "2020-11-01",
    "expirationDate": "2021-11-01",
    "faceValue": 100,
    "isVoucherDefinitionActive": true,
    "isVoucherPartiallyRedeemable": false,
    "hasTimeBasedVoucherPeriod": false,
    "redeemedValue": 0,
    "remainingValue": 100,
    "status": "Expired",
    "type": "FixedValue",
    "voucherCode": "6PLO64G1",

```

```

    "voucherDefinition": "Ajio",
    "voucherId": "0kDT1000000TU6yMAG",
    "voucherNumber": "00002634"
  },
  {
    "effectiveDate": "2020-11-01",
    "expirationDate": "2021-11-01",
    "faceValue": 100,
    "isVoucherDefinitionActive": true,
    "isVoucherPartiallyRedeemable": false,
    "hasTimeBasedVoucherPeriod": false,
    "redeemedValue": 0,
    "remainingValue": 100,
    "status": "Expired",
    "type": "FixedValue",
    "voucherCode": "K54Y02BK",
    "voucherDefinition": "Ajio",
    "voucherId": "0kDT1000000TTuKMAW",
    "voucherNumber": "00001850"
  }
]
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
voucherCount	Integer	The number of vouchers returned by the API request.	Small, 58.0	58.0
vouchers	Voucher	The details of the vouchers.	Small, 58.0	58.0

Voucher Result

Output representation of the voucher result.

Property Name	Type	Description	Filter Group and Version	Available Version
vouchersIssued	Voucher Output []	List of the vouchers that are issued to the associated loyalty program member.	Small, 54.0	54.0
 Note: This output is provided when the process contains an Issue Voucher action.				

Voucher Output

Output representation of the details about a voucher.

Property Name	Type	Description	Filter Group and Version	Available Version
discountPercent	Integer	The discount percentage associated with the voucher.	Small, 54.0	54.0
effectiveDate	String	The date the voucher goes into effect.	Small, 54.0	54.0
expirationDate	String	The date the voucher expires.	Small, 54.0	54.0
faceValue	Double	The value of the voucher.	Small, 54.0	54.0
productId	String	The ID of the product in the voucher issued.	Small, 54.0	54.0
promotionId	String	The ID of the promotion associated with the voucher.	Small, 54.0	54.0
voucherCode	String	The voucher code.	Small, 54.0	54.0
voucherId	String	The ID of the voucher issued.	Small, 54.0	54.0

Loyalty Management Connect API Interchangeable Request Fields

The Loyalty Management Business API supports interchangeable fields in JSON requests. For some of the request fields, you can use either an ID, number, or name. For example, if you don't want to store and use Salesforce IDs when accessing Loyalty Management-related data from an external system, simply use names and numbers instead.

Here's a list of Loyalty Management object fields that are interchangeable with alternative field names.

Field Name	Alternative Field Name
MemberId	MembershipNumber
VoucherId	VoucherCode or VoucherNumber Ensure that the voucher code is unique among all active vouchers.
LoyaltyProgramId	LoyaltyProgramName
JournalTypeId	JournalTypeName
JournalSubTypeId	JournalSubTypeName
PromotionId	PromotionName Ensure that the promotion name is unique among all active promotions.
ProductId	ProductCode or ProductName Ensure that the product name or code is unique among all active products.

Field Name	Alternative Field Name
ProductCategoryId	ProductCategoryName Ensure that the product name or code is unique among all active products.
PartnerId	PartnerName

JSON example with IDs

```
{
  "MemberId": "01MT10000004CfA",
  "JournalTypeId": "01ET10000004CQp",
  "JournalSubTypeId": "0pQS10000005CQm",
  "VoucherId": "0kdx000000000E245",
  "ProductId": "01Vxx00000000cj",
  "ProductCategoryId": "0c8RM0000004DnXYAU",
  "PromotionId": "0c8xx00000000BJEAY",
  "LoyaltyProgramId": "0lpT10000004CdN"
}
```

JSON example with names and numbers

```
{
  "MembershipNumber": "Member123",
  "JournalType": "Accrual",
  "JournalSubType": "Social",
  "VoucherCode": "15OFF",
  "ProductName": "Nike",
  "ProductCategoryName": "Apparel",
  "PromotionName": "Flat20",
  "LoyaltyProgramName": "NTO"
}
```

CHAPTER 7 Loyalty Management Standard Invocable Actions

In this chapter ...

- [Adjust Points](#)
- [Assign Member Tier Benefits](#)
- [Accrual Cancellation](#)
- [Credit Points](#)
- [Change Tier](#)
- [Change All Tiers or None](#)
- [Debit Points](#)
- [Generate Referral Code](#)
- [Get Member's Active Segments](#)
- [Get Points Balance](#)
- [Get Loyalty Promotions Based on Member's Data Cloud](#)
- [Get Promotions Based on Transaction Journal](#)
- [Get Tier](#)
- [Issue Voucher Action](#)
- [Map Traceable Points for Redemption Transactions Action](#)
- [Merge Memberships Action](#)
- [Process Member Benefit](#)
- [Redemption Cancellation](#)
- [Run Program Process for Transaction Journal](#)

Create and manage loyalty programs for your organization using invocable actions. Use tier groups to group members and change member tier based on their loyalty points, assign and process benefits based on their tier, manage loyalty points for a member, issue voucher for a member, and cancel redemption transactions for a member for Loyalty Management. For more information on standard invocable actions, see [REST API Developer Guide](#) and [Actions Developer Guide](#).

Loyalty Management Standard Invocable Actions

- Transfer Member Points to Groups
- Transfer Points
- Unmerge Memberships Action
- Update Member Progress for Cumulative Promotion

Adjust Points

Adjust loyalty points for a specified program member or journal transaction.

This action is available in API version 51.0 and later for users with Loyalty Management license.

Supported REST HTTP Methods

URI

/services/data/v~~XX~~.X/actions/standard/adjustPointsAction

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
AdjustmentCategory	Type String Description The category of adjustment.
AdjustmentType	Type string Description Required. The type of adjustment required.
JournalId	Type ID Description ID of the transaction journal.
LoyaltyProgramCurrencyId	Type ID Description ID of the loyalty program currency.
LoyaltyProgramMemberId	Type ID

Input	Details
	Description Required. ID of the loyalty program member to whom the points must be adjusted.
Points	Type Float Description Required. The points available for the loyalty program member.
PointsExpirationDate	Type string Description The date on which fixed-model non-qualifying points credited by the action expire. Points expiration date is only applicable when the adjustment is crediting points to the member. This input is available in API version 55.0 and later.
ReasonforChange	Type string Description Required. The reason for adjustment.

Outputs

Output	Details
AdjustmentTransactionJournalId	Type string Description The ID of the adjustment transaction journal generated after the request is successful.

Example

Sample Request

```
{
  "inputs": [{
    "LoyaltyProgramMemberId": "01mxx00004000001AAB",
    "LoyaltyProgramCurrencyId": "01pxx00004000001AAB",
    "Points": 500.0,
    "PointsExpirationDate": "2022-08-12",
    "AdjustmentType": "Credit",
    "ReasonForChange": "Compensation"
  }]
}
```

Sample Response

```
{
  "errors":null,
  "isSuccess":true,
  "outputValues":{
    "AdjustmentTransactionJournalId":"01vxx00004000001AAB"}
}
```

Assign Member Tier Benefits

Assign benefits to a loyalty program members based on their member's tier for benefits that are associated with a Benefit Action.

This action is available in API version 51.0 and later for users with Loyalty Management license.

Supported REST HTTP Methods

URI

/services/data/v**XX.X**/actions/standard/assignMemberTierBenefits

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
LoyaltyProgramMemberId	Type ID Description Required. ID of the loyalty program member to associate with the benefits.

Outputs

Output	Details
MemberBenefits	Type string[] Description The benefits associated with the member.

Example

Sample Request

```
{
  "inputs": [{
    "LoyaltyProgramMemberId": "01MR00000000001OAA"
  }]
}
```

Sample Response

```
{
  "actionName": "assignTierBenefits",
  "errors": null,
  "isSuccess": true,
  "outputValues": {
    "MemberBenefits": [
      '0hWR00000000001OAA',
      '0hWR00000000001OAB',
      '0hWR00000000001OAC'
    ]
  }
}
```

Accrual Cancellation

Cancel a specific set of accrual transactions.

This action is available in API version 51.0 and later for users with Loyalty Management license.

Supported REST HTTP Methods

URI

/services/data/v**xx.x**/actions/standard/cancelAccrual

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
CancellationReason	Type string

Input	Details
	Description Required. The reason for accrual cancellation.
ReversalCategory	Type string Description The category of accrual cancellation.
TransactionJournalId	Type string Description Required. ID of the transaction journal.

Outputs

Output	Details
CancellationTransactionJournalId	Type string Description The ID of the cancellation transaction journal generated after the request is successful.

Example

Sample Request

```
{
  "inputs": [{
    "TransactionJournalId": "0lvxx00004000001AAA",
    "CancellationReason": "Duplicate Accrual"
  }]
}
```

Sample Response

```
{
  "errors": null,
  "isSuccess": true,
  "outputValues": {
    "CancellationTransactionJournalId": "0lvxx00004000001AAB"
  }
}
```

Credit Points

Credit loyalty points to a specified loyalty program member's balance.

This action is available in API version 51.0 and later for users with Loyalty Management license.

Supported REST HTTP Methods

URI

/services/data/v~~XX.X~~/actions/standard/creditPoints

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
AdditionalNotes	Type String Description Additional information about the transactions performed by the loyalty program members.
CorrelationId	Type ID Description ID of any Salesforce entity.
FlowId	Type ID Description ID of the flow associated with credit points action.
JournalId	Type ID Description Required. ID of the transaction journal.
LoyaltyProgramMemberId	Type ID

Input	Details
	Description Required. ID of the loyalty program member.
Points	Type Float Description Required. The points available for the loyalty program member.
ProgramCurrencyName	Type string Description Required. The name of the program currency.
PointsExpirationDate	Type string Description The date on which fixed-model non-qualifying points credited by the action expire. This input is available in API version 55.0 and later.
ProgramName	Type string Description Required. The name of the loyalty program.
LoyaltyProgramCurrencySubtypeName	Type string Description Required. The name of the loyalty program currency subtype that is used to credit points from the member.

Outputs

Output	Details
JournalId	Type ID Description ID of the transaction journal.
LoyaltyProgramMemberId	Type string

Output	Details
	Description ID of the loyalty program member.
ProgramCurrencyName	Type string Description The name of the program currency.
ProgramName	Type string Description The name of the loyalty program.

Example

Sample Request

```
{
  "inputs": [{
    "LoyaltyProgramMemberId": "01Mxx0000A000001EAA",
    "ProgramCurrencyName": "Avios",
    "ProgramName": "baLoyaltyProgram",
    "Points": 500.0,
    "PointsExpirationDate": "12-08-2022",
    "CorrelationId": "01cxx00004000002AAA",
    "JournalId": "01cxx00204000001AAA",
    "FlowId": "01cxx00204000001AAA",
    "AdditionalNotes": "Any additional Information"
  }]
}
```

Sample Response

```
{
  "errors": null,
  "isSuccess": true,
  "outputValues": {
    "LoyaltyProgramMemberId": "01Mxx0000A000001EAA",
    "ProgramCurrencyName": "Avios",
    "ProgramName": "baLoyalty",
    "JournalId": "01Mxx0000A000001EAA"
  }
}
```

Change Tier

Change the tier for a specified loyalty program member.

This action is available in API version 51.0 and later for users with Loyalty Management license.

Supported REST HTTP Methods

URI

/services/data/v~~XX.X~~/actions/standard/changeTier

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
ChangeReason	Type string Description The reason for change in member tier.
LoyaltyProgramMemberId	Type ID Description Required. ID of the loyalty program member.
ProgramName	Type string Description Required. The loyalty program name associated with the tier.
TierGroupName	Type string Description Required. The name of the tier group.
TargetTierName	Type string Description Required. The new tier to be associated with the loyalty program member

Outputs

Output	Details
LoyaltyProgramMemberId	Type string Description ID of the loyalty program member.
ProgramName	Type string Description The name of the loyalty program.
TargetTierName	Type string Description The new tier associated with the loyalty program member.
TierGroupName	Type string Description The name of the tier group.

Example

Sample Request

```
{
  "inputs": [{
    "TierGroupName": "QualifyingGroup",
    "ProgramName": "baProgram",
    "TargetTierName": "Gold",
    "LoyaltyProgramMemberId": "01cxx00006000001AAA",
    "ChangeReason": "Reason to change the tier"
  }]
}
```

Sample Response

```
{
  "errors": null,
  "isSuccess": true,
  "outputValues": {
    "LoyaltyprogramMemberId": ".01Mxx0000A000001EAA",
    "TierGroupName": "QualifyingGroup",
    "ProgramName": "baProgram",
    "TargetTierName": "Gold"
  }
}
```

```
}  
}
```

Change All Tiers or None

Change tier for a specified loyalty program member only when all the input parameters meet the criteria.

This action is available in API version 51.0 and later for users with Loyalty Management license.

Supported REST HTTP Methods

URI

/services/data/v**XX.X**/actions/standard/changeAllTierOrNone

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
ChangeReason	Type string Description The reason for change in the tier.
ChangeReasonType	Type string Description The reason type for change in the tier.
LoyaltyProgramMemberId	Type ID Description Required. ID of the loyalty program member.
ProgramName	Type string Description Required. The name of the loyalty program.

Input	Details
TierGroupName	Type string Description Required. The name of the tier group.
TargetTierName	Type string Description Required. The new tier to be associated with the loyalty program member.

Outputs

Output	Details
LoyaltyProgramMemberId	Type ID Description ID of the loyalty program member.
ProgramName	Type string Description The name of the loyalty program.
TierGroupName	Type string Description The name of the tier group.
TargetTierName	Type string Description The new tier associated with the loyalty program.

Example

Sample Request

```
{
  "inputs": [{
    "TierGroupName": "Default Tier Group",
```

```

    "ProgramName": "Rewards program",
    "TargetTierName": "Gold",
    "LoyaltyProgramMemberId": "01cxx00006000001AAA",
    "ChangeReason": "Reason to change the tier"
  }]
}

```

Sample Response

```

{
  "actionName": "changeTierWhenNoErrors",
  "errors": null,
  "isSuccess": true,
  "outputValues": {
    "LoyaltyProgramMemberId": "01cxx00006000001AAA",
    "TierGroupName": "Default Tier Group",
    "ProgramName": "Rewards program",
    "TargetTierName": "Gold",
  }
}

```

Debit Points

Debit loyalty points from a specified loyalty program member's balance. You can debit loyalty points to redeem for a service or upgrade the member to a higher tier.

 **Note:** This action updates the point balance in the LoyaltyProgramMemberCurrency table in your Salesforce org.

This action is available in API version 51.0 and later for users with Loyalty Management license.

Supported REST HTTP Methods

URI

/services/data/v**XX.X**/actions/standard/debitPoints

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
AdditionalNotes	Type String

Input	Details
	Description Additional information about the transactions performed by the loyalty program members.
CorrelationId	Type ID Description ID of any Salesforce entity.
FlowId	Type ID Description ID of the flow associated with the debit points action.
JournalId	Type ID Description Required. ID of the transaction journal.
LoyaltyProgramMemberId	Type ID Description Required. ID of the loyalty program member to whom the points must be deducted.
Points	Type Float Description Required. The points available for the loyalty program member.
ProgramCurrencyName	Type string Description Required. The name of the loyalty program currency.
ProgramName	Type string Description Required. The name of the loyalty program.
LoyaltyProgramCurrencySubtypeName	Type string

Input	Details
	Description Required. The name of the loyalty program currency subtype that is used to debit points from the member.

Outputs

Output	Details
JournalId	Type ID Description ID of the transaction journal.
LoyaltyProgramMemberId	Type string Description ID of the loyalty program member.
PointsBalance	Type Integer Description The total points available for the loyalty program member.
ProgramCurrencyName	Type string Description The name of the program currency.
ProgramName	Type string Description The name of the loyalty program.

Example

Sample Request

```
{
  "inputs": [{
    "LoyaltyProgramMemberId": "01Mxx0000A000001EAA",
    "ProgramCurrencyName": "Avios",
    "ProgramName": "baLoyaltyProgram",
```

```

    "Points": 500.0,
    "CorrelationId": "01cxx00004000002AAA",
    "JournalId": "01cxx00204000001AAA",
    "AdditionalNotes": "Any additional Information"
  }
}

```

Sample Response

```

[ {
  "actionName" : "debitPoints",
  "errors" : null,
  "isSuccess" : true,
  "outputValues" : {
    "JournalId" : "01cxx00204000001AAA",
    "PointsBalance" : 130.0,
    "ProgramCurrencyName" : "exclusive points",
    "LoyaltyProgramMemberId" : "01Mxx0000A000001EAA",
    "ProgramName" : "sample program"
  }
} ]

```

Generate Referral Code

Generate a referral code for a loyalty program member.



Note: If a loyalty program member already has a referral code, generating another code replaces the existing one.

This action is available in API version 57.0 and later for users with the Loyalty Management license.

Supported REST HTTP Methods

URI

/services/data/v**XX.X**/actions/standard/generateMemberReferralCode

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
LoyaltyProgramMemberId	Type String

Input**Details****Description**

Required. ID of the loyalty program member to generate the referral code for.



Note: Specify an array of member IDs in the JSON request to generate referral codes for multiple members.

Outputs

None.

Example

Sample Request

```
{
  "inputs": [{
    "LoyaltyProgramMemberId": "01Mxx0000A000001EAA"
  }, {
    "LoyaltyProgramMemberId": "01Qxx0000B000001EBS"
  }]
}
```

Get Member's Active Segments

Retrieve active Data Cloud market segments that a loyalty program member is a part of.

For more information about how the `getMemberActiveSegments` action can be used to process transaction journals and apply promotions, see [Get a List of Member's Active Segments for a Member](#) in Salesforce Help.

This action is available in API version 55.0 and later for users in orgs where either B2C - Loyalty, B2C - Loyalty Plus, Loyalty Management - Growth, or Loyalty Management - Plus license and Data Cloud are enabled.

Supported REST HTTP Methods

URI

`/services/data/vXX.X/actions/standard/getMemberActiveSegments`

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
loyaltyProgramMemberID	Type ID Description Required. The ID of the loyalty program member to retrieve associated active segments for.
segmentIds	Type ID Description A comma-delimited list of segmentIds to retrieve for the specified member's associated segments.

Outputs

Output	Details
memberSegmentIDs	Type string Description A comma-delimited list of the active segment IDs that the specified loyalty program member is a part of. If no segments are specified in the request, the action returns segments from the default data space.

Example

Sample Request

```
{
  "inputs": [ {
    "loyaltyProgramMemberID": "01Vxx000000016GMAQ"
  } ]
}
```

Sample Response

```
{
  "errors": null,
  "isSuccess": true,
  "outputValues": {
    "memberSegmentIDs": "01vxx00004000001AAB"
  }
}
```

Get Points Balance

Retrieve the loyalty points balance for a specified loyalty program member.

This action is available in API version 51.0 and later for users with Loyalty Management license.

Supported REST HTTP Methods

URI

/services/data/v~~XX~~.X/actions/standard/getPointsBalance

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
LoyaltyProgramMemberId	Type string Description Required. ID of the loyalty program member to whom the points balance are to be fetched.
ProgramCurrencyName	Type string Description Required. The name of the program currency.
ProgramName	Type string Description Required. The name of the loyalty program.

Outputs

Output	Details
LoyaltyProgramMemberId	Type ID

Output	Details
	Description ID of the loyalty program member.
PointsBalance	Type integer Description The points available for the loyalty program member.
ProgramCurrencyName	Type string Description The name of the program currency.
ProgramName	Type string Description The name of the loyalty program.
EscrowPointsBalance	Type double Description The number points that are placed in escrow for the member. This field is available in API version 53.0 and later.

Example

Sample Request

```
{
  "inputs": [{
    "LoyaltyProgramMemberId": "01Mxx0000A000001EAA",
    "ProgramCurrencyName": "Avios",
    "ProgramName": "baLoyalty"
  }]
}
```

Sample Response

```
{
  "errors": null,
  "isSuccess": true,
  "outputValues": {
    "LoyaltyprogramMemberId": "01Mxx0000A000001EAA",
    "ProgramCurrencyName": "Avios",
    "ProgramName": "baLoyalty"
    "PointsBalance": 500.0
  }
}
```

```
    "EscrowPointsBalance": 25.00
  }
}
```

Get Loyalty Promotions Based on Member's Data Cloud

Retrieve the promotions that are eligible for a member based on their market segment.

For more information about how the `getLoyaltyPromotionBasedOnSalesforceCDP` action filters eligible promotions, see [Get a List of Promotions Applicable for a Member](#) in Salesforce Help.

This action is available in API version 53.0 and later.

Special Access Rules

This action is available in orgs where either B2C - Loyalty, B2C - Loyalty Plus, Loyalty Management - Growth, or Loyalty Management - Advanced license and Data Cloud are enabled.

Supported REST HTTP Methods

URI

`/services/data/v66.0/actions/standard/getLoyaltyPromotionBasedOnSalesforceCDP`

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
MemberId	Type ID Description Required. ID of the loyalty program member for whom promotions must be filtered. Make sure that the member is part of a Data Cloud market segment.
PromotionRecords	Type sObject Description Promotion records to filter the promotions that are applicable to the member's market segment.

Outputs

Output	Details
PromotionRecords	Type sObject Description Promotion records that the member is eligible for based on the member's market segment.

Example

POST

Here's a sample request for the Get Promotions Based on Member's Data Cloud Segment action.

```
{
  "inputs": [ {
    "MemberId": "01Mxx0000000EYAMA2"
  } ]
}
```

Here's a sample response for the Get Promotions Based on Member's Data Cloud Segment action.

```
[ {
  "actionName" : "getLoyaltyPromotionBasedOnSalesforceCDP",
  "errors" : null,
  "isSuccess" : true,
  "outputValues" : {
    "PromotionRecords" : [ {
      "attributes" : {
        "type" : "Promotion",
        "url" : "/services/data/v53.0/subjects/Promotion/0c8T10000004CBEIA2"
      },
      "Id" : "0c8xx0000004CBEIA2",
      "OwnerId" : "005xx000000HwDoIAK",
      "IsDeleted" : false,
      "Name" : "Sample Promotion",
      "CreatedDate" : "2021-07-21T04:13:15.000+0000",
      "CreatedById" : "005xx000000HwDoIAK",
      "LastModifiedDate" : "2021-07-21T04:13:15.000+0000",
      "LastModifiedById" : "005xx000000HwDoIAK",
      "SystemModstamp" : "2021-07-21T04:13:15.000+0000",
      "LastViewedDate" : "2021-07-21T04:15:21.000+0000",
      "LastReferencedDate" : "2021-07-21T04:15:21.000+0000",
      "StartDate" : "2021-07-20",
      "IsActive" : true,
      "LoyaltyPromotionType" : "Cumulative",
      "IsEnrollmentRequired" : false,
      "CumulativeUsageTarget" : 100,
      "FulfillmentAction" : "CreditPoints",
      "LoyaltyProgramId" : "01pxx0000000HWIAY",
      "TotalRewardPoints" : 100,
    } ]
  }
}
```

```
    "LoyaltyProgramCurrencyId" : "01cxx00000000XeIAI",
    "TransactionJournalType" : "Accrual",
    "IsPointsWithFactor" : false
  } ]
}
```

Get Promotions Based on Transaction Journal

Retrieve active promotions that are eligible for a transaction journal. The action can process accrual or redemption transactions.

For more information about how the `getLoyaltyPromotion` action filters eligible promotions, see [Get a List of Promotions Applicable for a Transaction](#) in Salesforce Help.

This action is available in API version 53.0 and later.

Special Access Rules

This action is available in orgs where either B2C - Loyalty, B2C - Loyalty Plus, Loyalty Management - Growth, or Loyalty Management - Advanced license is enabled.

Supported REST HTTP Methods

URI

`/services/data/v66.0/actions/standard/getLoyaltyPromotion`

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
JournalId	Type ID
	Description Required. ID of an Accrual or Redemption type Transaction Journal record.

Outputs

Output	Details
PromotionRecords	Type sObject Description Promotion records that are eligible for the transaction journal.

Example

POST

Here's a sample request for the Get Promotions Based on Transaction Journal action.

```
{
  "inputs": [ {
    "JournalId": "01Vxx000000016GMAQ"
  } ]
}
```

Here's a sample response for the Get Promotions Based on Transaction Journal action.

```
[ {
  "actionName" : "getLoyaltyPromotion",
  "errors" : null,
  "isSuccess" : true,
  "outputValues" : {
    "PromotionRecords" : [ {
      "attributes" : {
        "type" : "Promotion",
        "url" : "/services/data/v53.0/subjects/Promotion/0c8T10000004CBEIA2"
      },
      "Id" : "0c8xx0000004CBEIA2",
      "OwnerId" : "005xx000000HwDoIAK",
      "IsDeleted" : false,
      "Name" : "Sample Promotion",
      "CreatedDate" : "2021-07-21T04:13:15.000+0000",
      "CreatedById" : "005xx000000HwDoIAK",
      "LastModifiedDate" : "2021-07-21T04:13:15.000+0000",
      "LastModifiedById" : "005xx000000HwDoIAK",
      "SystemModstamp" : "2021-07-21T04:13:15.000+0000",
      "LastViewedDate" : "2021-07-21T04:15:21.000+0000",
      "LastReferencedDate" : "2021-07-21T04:15:21.000+0000",
      "StartDate" : "2021-07-20",
      "IsActive" : true,
      "LoyaltyPromotionType" : "Cumulative",
      "IsEnrollmentRequired" : false,
      "CumulativeUsageTarget" : 100,
      "FulfillmentAction" : "CreditPoints",
      "LoyaltyProgramId" : "01pxx00000000HWIAY",
      "TotalRewardPoints" : 100,
    } ]
  }
}
```

```
    "LoyaltyProgramCurrencyId" : "01cxx00000000XeIAI",
    "TransactionJournalType" : "Accrual",
    "IsPointsWithFactor" : false
  } ]
}
```

Get Tier

Retrieve the current tier for a specified loyalty program member.
This action is available in API version 51.0 and later for users with Loyalty Management license.

Supported REST HTTP Methods

- URI**
/services/data/vXX.X/actions/standard/getTier
- Formats**
JSON, XML
- HTTP Methods**
POST
- Authentication**
Authorization: Bearer *token*

Inputs

Input	Details
LoyaltyProgramMemberId	Type string Description Required. ID of the loyalty program member to whom the tier details needs to be fetched.
ProgramName	Type string Description Required. The loyalty program name associated with the tier.
TierGroupName	Type string Description Required. The name of the tier group.

Outputs

Output	Details
LoyaltyProgramMemberId	Type string Description The ID of the loyalty program member.
SequenceNumber	Type integer Description The ranking of the tier.
TierGroupName	Type string Description The name of the tier group.
TierName	Type string Description The name of the loyalty tier associated with the program member.

Example

Sample Request

```
{
  "inputs": [{
    "TierGroupName": "QualifyingGroup",
    "ProgramName": "baLoyalty",
    "LoyaltyProgramMemberId": "01cxx00004000001AAA"
  }]
}
```

Sample Response

```
{
  "errors": null,
  "isSuccess": true,
  "outputValues": {
    "LoyaltyProgramMemberId": "01Mxx0000A000001EAA",
    "TierGroupName": "QualifyingGroup",
    "TierName": "Gold",
    "ProgramName": "baLoyalty",
    "SequenceNumber": 4
  }
}
```

```
}  
}
```

Issue Voucher Action

Issue a voucher for a member or contact.

This action is available in API version 51.0 and later for users with Loyalty Management license.

Supported REST HTTP Methods

URI

/services/data/v`XX.X`/actions/standard/issueVoucher

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
AdditionalNotes	Type string Description Additional details related to the action, such as the reason for issuing the voucher.
AccountId	Type string Description The ID of the account that's used to issue the voucher. Provide <code>AccountId</code> only when you issue vouchers to a Business Account. If both <code>AccountId</code> and <code>ContactId</code> are provided, then an error is displayed.
BenefitId	Type ID Description The benefit associated with the voucher.
ContactId	Type ID

Input	Details
	Description The contact associated with the member. Provide <code>ContactId</code> only when you issue vouchers to a Person Account. If both <code>AccountId</code> and <code>ContactId</code> are provided, then an error is displayed.
DiscountPercent	Type percent Description The discount percentage provided by the voucher. Specify a value to override the discount percentage specified in the associated voucher definition. Specify this when the associated voucher definition is of the type Discount Percentage. This field is available in API version 53.0 and later.
FaceValue	Type double Description The amount of discount provided by the voucher. Specify a value to override the face value specified in the associated voucher definition. Specify this when the associated voucher definition is of the type Fixed Value. This field is available in API version 53.0 and later.
EffectiveDate	Type Date Description The date the voucher goes into effect.
ExpiryDate	Type Date Description The date the voucher expires.
JournalId	Type ID Description The transaction journal associated with the member voucher when the voucher results from accrual of points.
LoyaltyProgramMemberId	Type ID Description ID of the loyalty program member to issue the voucher.
PromotionId	Type ID

Input	Details
	Description The promotion associated with the voucher.
ProgramName	Type string Description The name of the loyalty program associated with this voucher.
VoucherCode	Type string Description Required. The voucher identifier generated by an external system.
VoucherDefinitionName	Type string Description Required. The name of the voucher definition associated with the voucher code.

Outputs

Output	Details
VoucherId	Type ID Description The ID of the voucher that's generated after the request is successful.

Example

Sample Request

Sample request to issue vouchers to a Person Account.

```
{
  "inputs": [{
    "LoyaltyProgramMemberId": "01Mxx0000A000001EAA",
    "ContactId": "01Mxx0000A000001EAA",
    "ProgramName": "Program 1",
    "VoucherDefinitionName": "Voucher1",
    "JournalId": "01Vxx0000A000001EAA",
    "VoucherCode": "Voucher123",
    "PromotionId": "01Mxx0000A000001EAA",
    "BenefitId": "01Mxx0000A000001EAA",
    "Reason": "Bonus",
```

```

        "EffectiveDate": "12-08-2021",
        "ExpiryDate": "12-09-2021"
        "DiscountPercent": "20"
    }
}

```

Sample request to issue vouchers to a Business Account.

```

{
  "inputs": [
    {
      "LoyaltyProgramMemberId": "01Mxx00000002HxEAI",
      "AccountId": "001xx000003H3NnAAK",
      "VoucherDefinitionName": "Voucher1",
      "VoucherCode": "Voucher123",
      "ProgramName": "Sip & Save Rewards"
    }
  ]
}

```

Sample Response

Sample request for the Issue Voucher action.

```

{
  "errors": null,
  "isSuccess": true,
  "outputValues": {
    "VoucherId": "01Yxx0000A000001EAA"
  }
}

```

Map Traceable Points for Redemption Transactions Action

Maps Debit type loyalty ledgers to Credit type loyalty ledgers for loyalty ledger traceability records that are pending.

This action is available in API version 66.0 and later.

Supported REST HTTP Methods

URI

/services/data/v**66.0**/actions/standard/mapTraceablePtForRedemTrxn

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Either `LoyaltyLedgerTraceabilityId` or `MemberId` is required. If both are provided and the `MemberId` doesn't belong to the specified `LoyaltyLedgerTraceabilityId`, an error is returned.

Input	Details
<code>loyaltyLedgerTraceabilityId</code>	Type string Description The ID of the traceable loyalty ledger. Processes all the pending traceable ledgers for the loyalty program member associated with the traceability ID.
<code>loyaltyProgramMemberId</code>	Type string Description The ID of the loyalty program member for whom you want to trace pending traceable ledgers.

Outputs

Output	Details
<code>loyaltyProgramMemberId</code>	Type string Description The ID of the loyalty program member whose pending traceable ledgers are processed.
<code>numberOfTracesProcessed</code>	Type integer Description The number of traceable loyalty ledgers processed for a loyalty program member.

Example

POST

Here's a sample request for the Map Traceable Points for Redemption Transactions action.

```
{
  "inputs": [
    {
      "loyaltyLedgerTraceabilityId": "0ltRM0000008Abc2AE",
      "loyaltyProgramMemberId": "0lMRM0000004Eqv2AE"
    }
  ]
}
```

```
  ]
}
```

Here's a sample response for the Map Traceable Points for Redemption Transactions action.

```
{
  "actionName": "mapTraceablePtForRedemTrxn",
  "errors": null,
  "isSuccess": true,
  "outputs": {
    "loyaltyProgramMemberId": "01MRM0000004Eqv2AE",
    "numberOfTracesProcessed": 5
  }
}
```

Merge Memberships Action

Merges two active loyalty program member records that belong to the same loyalty program. Out of the two memberships, one is source membership and the other is the target membership. After the merge is complete, the status of the source membership is changed to Merged and the status of the target membership remains active.

By default, all the transaction journals and points of the source member are transferred to the target member after the merge. Optionally, you can:

- Transfer all the vouchers of source member to the target member.
- Assign the higher current tier between the source and target members to the target member.

For more information about how the mergeLoyaltyProgramMembership action merges loyalty program memberships, see [Merge Memberships](#) in Salesforce Help.

This action is available in API version 56.0 and later for users in orgs where Loyalty Management is enabled.

Supported REST HTTP Methods

URI

/services/data/v~~XX~~.X/actions/standard/mergeLoyaltyProgramMembership

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
reasonForMergingMemberships	Type String

Input	Details
	Description Required. The reason why the source loyalty program member is to be merged with the target loyalty program member.
shouldSetHigherTierForTargetMember	Type Boolean Description Indicates whether the tier for the target record is set to the source record's tier when the source tier is higher than the target record's current tier across all groups (<code>true</code>) or not (<code>false</code>). The default value is <code>false</code>.
shouldTransferVouchers	Type Boolean Description Indicates whether all vouchers that belong to the source loyalty program member must be transferred to the target loyalty program member (<code>true</code>) or not (<code>false</code>). The default value is <code>false</code>.
sourceLoyaltyProgramMemberId	Type ID Description Required. The ID of the loyalty program member record to be merged with the target loyalty program member record. The status of the source loyalty program member record is set to <code>Merged</code> after the merge is complete.
targetLoyaltyProgramMemberId	Type ID Description Required. The ID of the loyalty program member record to remain active for the member after the merge is complete. The qualifying and non-qualifying points accrued by the source member are added to the respective points balances of the target member record.

Outputs

Output	Details
errors	Type List<String>

Output	Details
	Description A list of error messages if the operation fails.
isSuccess	Type Boolean Description Indicates whether the operation is successful or not.

Examples

Sample Request

```
{
  "inputs": [ {
    "reasonForMergingMemberships": "Test Merge",
    "shouldSetHigherCurrentTierForTargetMember": true,
    "shouldTransferVouchers": true,
    "sourceLoyaltyProgramMemberId": "01MT100000001JnMAK",
    "targetLoyaltyProgramMemberId": "01MT100000001JnMAI"
  } ]
}
```

Sample Response

```
[ {
  "actionName" : "mergeLoyaltyProgramMembership",
  "errors" : null,
  "isSuccess" : true
} ]
```

SEE ALSO:

[SOAP API Developer Guide: Core Data Types Used in API Calls](#)

Process Member Benefit

Process the benefit action associated with the benefit that is assigned to a loyalty program member.

This action is available in API version 51.0 and later for users with Loyalty Management license.

Supported REST HTTP Methods

URI

/services/data/v**xx.x**/actions/standard/processMemberBenefitAction

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
MemberBenefitId	Type ID Description Required. ID of the member associated with the benefit.

Example

Sample Request

```
{
  "inputs": [{
    "MemberBenefitId": "01Mxx0000A000001EAA"
  }]
}
```

Sample Response

```
{
  "actionName": "executeMemberBenefit",
  "errors": null,
  "isSuccess": true,
}
```

Redemption Cancellation

Revert a specific set of redemption transactions.

This action is available in API version 51.0 and later for users with Loyalty Management license.

Supported REST HTTP Methods

URI

/services/data/v**xx.x**/actions/standard/cancelRedemption

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
CancellationReason	Type string Description Required. The reason for cancellation of redemption transaction.
ExpiryDate	Type date Description The date of expiration.
ReversalCategory	Type string Description The category of reversal.
TransactionJournalId	Type string Description Required. ID of the transaction journal.

Outputs

Output	Details
CancellationTransactionJournalId	Type string Description The ID of the cancellation transaction journal generated after the request is successful.

Example

Sample Request

```
{
  "inputs": [{
    "TransactionJournalId": "01vxx00004000001AAA",
```

```

    "CancellationReason": "Order Returned by Member",
    "ExpiryDate": "2020-04-15"
  }]
}

```

Sample Response

```

{
  "errors": null,
  "isSuccess": true,
  "outputValues": {
    "CancellationTransactionJournalId": "01vxx00004000001AAB"
  }
}

```

Run Program Process for Transaction Journal

Run active loyalty program processes for pending Accrual or Redemption type transaction journals.

The action assesses transaction journals and runs the program processes whose selected journal type matches the transaction's journal type. If program processes have a journal subtype selected, then processes only run for a transaction journal when the transaction journal's journal subtype matches that processes' journal subtype.

For more information about this action, see [Run Program Process For Transaction Journal](#) in Salesforce Help.

This action is available in API version 54.0 and later for users in org's where either B2C - Loyalty, B2C - Loyalty Plus, Loyalty Management - Growth, or Loyalty Management - Advanced license are enabled.

Supported REST HTTP Methods

URI

/services/data/v**XX.X**/actions/standard/runProgramProcessForTransactionJournal

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
transactionJournalId	Type ID Description Required. ID of the transaction journal for which program processes must be run. The status of the Transaction Journal must be Pending, and the journal type must be specified for it.

Outputs

Output	Details
status	Type Boolean Description Indicates the status of the request. <code>True</code> indicates the program process was successfully run for the transaction journal.

Example

Sample Request

```
{
  "inputs": [
    {
      "transactionJournalId": "01Mxx0000A000001EAA"
    }
  ]
}
```

Sample Response

```
[ {
  "actionName" : "runProgramProcessForTransactionJournal",
  "errors" : null,
  "isSuccess" : true,
  "outputValues":{
    "status":"true"
  }
} ]
```

Transfer Member Points to Groups

Transfers Individual or Corporate type members' points contributions to the members' associated groups.

For more information about this action, see [Transfer Member Points to Groups](#) in Salesforce Help.

This action is available in API version 53.0 and later for users with Loyalty Management license.

Supported REST HTTP Methods

URI

/services/data/v`xx.x`/actions/standard/transferMemberPointsToGroups

Formats

JSON, XML

HTTP Methods

POST

AuthenticationAuthorization: Bearer *token*

Inputs

Input	Details
JournalId	Type ID Description Required. The transaction journal from which points must be transferred to groups. The transaction journal must be associated to either an Individual or a Corporate type member. The points are transferred to the member's associated groups.

Example

Sample Request

```
{
  "inputs": [
    {
      "JournalId": "01Mxx0000A000001EAA"
    }
  ]
}
```

Sample Response

```
{
  "actionName": "transferMemberPointsToGroups",
  "errors": null,
  "isSuccess": true
}
```

Transfer Points

Transfers points from a source loyalty program member to a target loyalty program member, or to a group that the member is a part of.

This action is available in API version 64.0 and later for users with Loyalty Management license.

Supported REST HTTP Methods

URI

/services/data/v**xx.x**/actions/standard/transferPoints

Formats

JSON, XML

HTTP Methods

POST

AuthenticationAuthorization: Bearer *token*

Inputs

Input	Details
programName	Type String Description Required. The name of the loyalty program that the source and target members are a part of.
programCurrencyName	Type ID Description The name of the non-qualifying currency for which points are transferred.
transferPoints	Type Integer Description The number of points that are transferred.
sourceProgramMembershipNumber	Type Integer Description The membership number of the loyalty program member from whom the points are transferred. Either the membership number or the member ID is mandatory.
sourceProgramMemberId	Type ID Description The ID of the loyalty program member from whom the points are transferred. Either the membership number or the member ID is mandatory.
targetProgramMembershipNumber	Type Integer Description The membership number of the loyalty program member to whom the points are transferred. Either the membership number or the member ID is mandatory.
targetProgramMemberId	Type ID

Input	Details
	Description The ID of the loyalty program member to whom the points are transferred. Either the membership number or the member ID is mandatory.
additionalInformation	Type ID Description The additional information about the transfer of points.

Outputs

Input	Details
programName	Type String Description Required. The name of the loyalty program that the source and target members are a part of.
programCurrencyName	Type ID Description The name of the non-qualifying currency for which points are transferred.
pointsTransferred	Type Integer Description The number of points that are transferred.
sourceProgramMemberId	Type ID Description The ID of the loyalty program member from whom the points are transferred.
targetProgramMemberId	Type ID Description The ID of the loyalty program member to whom the points are transferred.
sourceTransactionJournalId	Type ID

Input	Details
	Description The ID of the transaction journal that debited points from the source loyalty program member.
targetTransactionJournalId	Type ID
	Description The ID of the transaction journal that credited points to the target loyalty program member.

Sample Request

```
{
  "inputs": [
    {
      "programName": "NTORules",
      "sourceProgramMembershipNumber": "N01",
      "targetProgramMembershipNumber": "N02",
      "programCurrencyName": "NQP Fixed Negative Trace",
      "transferPoints": 5,
      "additionalInformation": "Tried Transfer Points testing"
    }
  ]
}
```

Sample Response

```
{
  "actionName": "transferPoints",
  "errors": null, "invocationId": null,
  "isSuccess": true, "outcome": null,
  "outputValues": {
    "targetTransactionJournalId": "01VLT0000001ATa",
    "programCurrencyName": "NQPFixed",
    "programName": "Program001",
    "pointsTransferred": 10,
    "sourceProgramMemberId": "01MLT00000000tZ",
    "sourceTransactionJournalId": "01VLT0000001ATZ",
    "targetProgramMemberId": "01MLT00000000vB"},
  "sortOrder": -1, "version": 1
}
```

Unmerge Memberships Action

Unmerges loyalty program member records have a Merged status. The action unmerges memberships that are in the Merged status from the membership that it was previously merged with. After the unmerge is complete, the status of the Loyalty Program Member Merge record that was created during the merge is changed to Unmerged. The members are reassigned to the tier that was assigned to them before the merge. They are also credited with the points and transferred the vouchers they had before the membership was merged.

For more information about how the unmergeLoyaltyProgramMembership action unmerges loyalty program memberships, see [Unmerge Memberships](#) in Salesforce Help.

This action is available in API version 56.0 and later for users in orgs where Loyalty Management is enabled.

Supported REST HTTP Methods

URI

/services/data/v~~XX~~.X/actions/standard/unmergeLoyaltyProgramMembership

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
mergedLoyaltyProgramMemberId	Type ID Description Required. The ID of the merged loyalty program member record to be unmerged.

Outputs

Output	Details
errors	Type List<String> Description A list of error messages if the operation fails.
isSuccess	Type Boolean Description Indicates whether the operation is successful or not.

Examples

Sample Request

```
{
  "inputs": [ {
    "mergedLoyaltyProgramMemberId": "01MT100000001JnMAK"
  } ]
}
```

Sample Response

```
[ {
  "actionName" : "unmergeLoyaltyProgramMembership",
  "errors" : null,
  "isSuccess" : true
} ]
```

SEE ALSO:

[SOAP API Developer Guide: Core Data Types Used in API Calls](#)

Update Member Progress for Cumulative Promotion

Update the progress made by a loyalty program member towards attaining a Cumulative type promotion. The action creates a Loyalty Program Member Promotion record if the promotion isn't associated with the member.

For more information about this action, see [Update the Member's Progress Towards Attaining a Cumulative Promotion](#) in Salesforce Help.

This action is available in API version 53.0 and later for users in orgs where either B2C - Loyalty, B2C - Loyalty Plus, Loyalty Management - Growth, or Loyalty Management - Advanced license is enabled.

Supported REST HTTP Methods

URI

/services/data/v**XX.X**/actions/standard/updateProgressForCumulativePromotionUsage

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
loyaltyProgramMemberId	Type ID

Input	Details
	Description Required. ID of the loyalty program member who progress towards attaining a cumulative promotion must be updated.
promotionId	Type ID Description Required. ID of the Cumulative type promotion for which member's progress must be updated.
usageIncrementForCumulativePromotion	Type integer Description Required. The value or quantity that must be added to member's usage towards attaining the promotion. For example, if the target of a promotion is 10 units and the member has previously purchased 1 unit. Now the member has purchased 4 units. In this scenario, the input must be 4 and not 5 (1+4).

Outputs

Output	Details
cumulativeUsageTarget	Type integer Description The target value or quantity which members must achieve to attain the promotion.
updatedCumulativeUsageCompleted	Type integer Description The updated value or quantity by which the member has progressed towards attaining the promotion. For example, if the target of a promotion is 10 units and the member has previously purchased 1 unit. Now the member has purchased 4 units. In this scenario, the output is 5.

Example

Sample Request

```
{
  "inputs" : [
```

```
{
  "loyaltyProgramMemberId" : "01MRxx0000006MfMAI",
  "promotionId": "0c8xx00000002MqIAI",
  "usageIncrementForCumulativePromotion": 4
}
```

Sample Response

```
[ {
  "actionName" : "updateProgressForCumulativePromotionUsage",
  "errors" : null,
  "isSuccess" : true,
  "outputValues" : {
    "cumulativeUsageTarget" : 10,
    "updatedCumulativeUsageCompleted" : 5
  }
} ]
```

CHAPTER 8 Loyalty Management Custom Invocable Actions

In this chapter ...

- [Issue Digital Pass](#)
- [Refresh Digital Pass](#)
- [Run Program Process](#)

Create and manage loyalty programs for your organization using custom invocable actions.

Issue Digital Pass

This action issues a digital pass to a specified party based on the configured digital pass template. The digital pass template can be of type Loyalty or Gift Card, and the available input parameters depend on the template type. You can trigger the issuance of a digital pass using its corresponding custom invocable action.

This action is available in API version 66.0 and later.

Supported REST HTTP Methods

URI

/services/data/v66.0/actions/custom/issueDigitalPass/*actionName*

 **Note:** <actionName> is the customer-defined name of the digital pass template. The action name is derived from the digital pass template. Spaces in the template name are URL-encoded (for example, replaced with %20).

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input parameters depend on the digital pass template type (Loyalty or Gift Card). At runtime, the system determines the template type and loads the corresponding parameters. These input parameters are applicable to both Loyalty and Gift Card digital passes.

Input	Details
partyId	Type string Description Required. The ID of the party (for example, Loyalty Program Member or Contact) to whom the digital pass is issued.
sourceRecordReferenceId	Type ID Description Required. Reference ID of the source record initiating the digital pass issuance (for example, a transaction, membership record, or promotion).

Inputs when the template type is Loyalty

These parameters are optional and are created by default when the loyalty digital pass template is activated. Users can modify these values as needed.

Input	Details
accountNameLabel	Type string Description The account name displayed on the loyalty digital pass.
accountIdLabel	Type string Description The account ID associated with the digital pass. Use this when issuing a digital pass for an account.
barcode	Type string Description The barcode value displayed on the loyalty digital pass.
rewardPoints	Type string Description The reward points balance displayed on the loyalty digital pass.
tier	Type string Description The tier of the loyalty program member displayed on the digital pass.
memberType	Type string Description The type of loyalty program member displayed on the digital pass.
expiryDate	Type date Description The expiry date of the loyalty digital pass.

Inputs when the template type is Gift Card

These parameters are optional and are created by default when the gift card digital pass template is activated. Users can modify these values as needed.

Input	Details
barcode	Type string Description The barcode value displayed on the gift card.
cardNumber	Type string Description The number displayed on the gift card.
value	Type string Description The monetary value of the gift card.
expiryDate	Type date Description The expiry date of the gift card.

Outputs

For Issue Digital Pass, the action returns a URL that can be used to create or add the digital pass to a supported wallet.

Output	Details
digitalPassUrl	Type List<String> Description The URL used to create the digital pass. This URL can be used to add the pass to a supported digital wallet.
errors	Type List<String> Description A list of error messages if the operation fails.
isSuccess	Type Boolean Description Indicates whether the operation is successful (<code>true</code>) or not (<code>false</code>).

Examples

This section has the sample request and response for issuing the digital pass.

Sample Resource

/services/data/v66.0/actions/custom/issueDigitalPass/*Gift%20Card*

Sample request

Here's an example POST request for Gift Card Digital Pass.

```
{
  "inputs": [
    {
      "partyId": "01MSG0000001Q8X2AU",
      "sourceRecordReferenceId": "01MSG0000001Q8X2AU",
      "barcode": "GC-45879234",
      "expiryDate": "2026-12-31"
    }
  ]
}
```

Sample response

Here's a sample response for the Issue Digital Pass action. The output structure is the same for both template types. If the digital pass template is inactive or misconfigured, the action returns an error.

```
{
  "actionName": "Gift_Card",
  "errors": null,
  "isSuccess": true,
  "outputValues": {
    "digitalPassUrl": "https://wallet.example.com/pass/refresh/9f3a8c7b"
  }
}
```

Refresh Digital Pass

This action refreshes a digital pass to a specified party based on the configured digital pass setup. You can trigger the digital pass refresh using its corresponding custom invocable action.

This action is available in API version 66.0 and later.

Supported REST HTTP Methods

URI

/services/data/v66.0/actions/custom/refreshDigitalPass/*actionName*



Note: <actionname> is the name of the custom action associated with the digital pass configuration.

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input parameters depend on the digital pass template type (Loyalty or Gift Card). At runtime, the system determines the template type and loads the corresponding parameters. These input parameters are applicable to both Loyalty and Gift Card digital passes.

Input	Details
partyId	Type string Description Required. The ID of the party (for example, Loyalty Program Member or Contact) to whom the digital pass is issued.
sourceRecordReferenceId	Type ID Description Required. Reference ID of the source record initiating the digital pass issuance (for example, a transaction, membership record, or promotion).

Inputs when the template type is Loyalty

These parameters are optional and are created by default when the loyalty digital pass template is activated. Users can modify these values as needed.

Input	Details
accountNameLabel	Type string Description The account name displayed on the loyalty digital pass.
accountIdLabel	Type string Description The account ID associated with the digital pass. Use this when issuing a digital pass for an account.
barcode	Type string Description The barcode value displayed on the loyalty digital pass.

Input	Details
rewardPoints	Type string Description The reward points balance displayed on the loyalty digital pass.
tier	Type string Description The tier of the loyalty program member displayed on the digital pass.
memberType	Type string Description The type of loyalty program member displayed on the digital pass.
expiryDate	Type date Description The expiry date of the loyalty digital pass.

Inputs when the template type is Gift Card

These parameters are optional and are created by default when the gift card digital pass template is activated. Users can modify these values as needed.

Input	Details
barcode	Type string Description The barcode value displayed on the gift card.
cardNumber	Type string Description The number displayed on the gift card.
value	Type string Description The monetary value of the gift card.

Input	Details
expiryDate	Type date Description The expiry date of the gift card.

Outputs

If a digital pass already exists for the specified `partyId` and `sourceRecordReferenceId`, the system refreshes the existing pass. If no digital pass exists for the specified party, the system silently exits without creating a new pass.

Output	Details
errors	Type List<String> Description A list of error messages if the operation fails.
isSuccess	Type Boolean Description Indicates whether the operation is successful or not.

Examples

This section has the sample request and response for refreshing the digital pass.

Sample Resource

`/services/data/v66.0/actions/custom/refreshDigitalPass/Gift%20Card`

Sample request

Here's sample request for the Refresh Digital Pass action. If a digital pass already exists for the specified `partyId` and `sourceRecordReference`, the system refreshes it. If no digital pass exists for the specified party, the system silently exits without creating a new pass.

```
{
  "inputs": [
    {
      "partyId": "01MSG0000001Q8X2AU",
      "sourceRecordReferenceId": "01MSG0000001Q8X2AU"
    }
  ]
}
```

Sample response

Here's a sample response for the successful refresh.

```
{
  "actionName": "Gift Card",
  "errors": null,
  "isSuccess": true
}
```

Run Program Process

This action lists custom actions associated with all the available loyalty program processes. You can trigger an active loyalty program process using its corresponding custom invocable action.

This action is available in API version 56.0 and later.

Supported REST HTTP Methods

URI

/services/data/v**56.0**/actions/custom/runProgramProcess/<actionName>



Note: <actionname> is the name of the custom action associated with an active loyalty program process. The custom action name consists of the name of the loyalty program that the process belongs to followed by a delimiter, and then the name of the process. The delimiter is `SLI`. The program name, the delimiter, and the process name are separated by underscores. If your loyalty program and process names have spaces, the spaces are replaced with underscores.

If the name of your loyalty program is `InnerCircle` and process name is `Issue Voucher`, then the action name is `InnerCircle_SLI_Issue_Voucher`. If the name of your loyalty program is `Inner Circle` and process name is `IssueVoucher`, then the action name is `Inner_Circle_SLI_IssueVoucher`.

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Vary depending on the Loyalty Program Process setup action.

Outputs

Standard invocable action result parameters, such as `isSuccess` and `errors` along with other parameters that vary depending on the inputs.

Output	Details
errors	Type List<String> Description A list of error messages if the operation fails.
isSuccess	Type Boolean Description Indicates whether the operation is successful or not.

Examples

This section has the sample request and response for triggering the Credit Points Program Process.

Sample Resource

/services/data/v56.0/actions/custom/runProgramProcess/NTO_SLI_CreditPoints

Sample request

Here's an example POST request that has the inputs, such as, Loyalty Program member ID, points to credit, and other transaction details:

```
{
  "inputs": [
    {
      "TransactionJournal": {
        "ActivityDate": "2022-11-27T00:00:00Z",
        "BookingDate": "2022-10-23",
        "EndDate": "2022-10-23T00:00:00Z",
        "ExternalTransactionNumber": "1",
        "InvoiceDate": "2022-10-23",
        "JournalTypeName": "Accrual",
        "JournalSubTypeName": "xyz",
        "LoyaltyProgramId": "0lpRM0000004D53YAE",
        "MemberId": "01MRM0000004Eqv2AE",
        "TransactionAmount": 45,
        "Status": "Processed"
      },
      "PointsToCredit": "100"
    }
  ]
}
```

Sample response

Here's an example response of a successful credit points program operation:

```
{
  "actionName": "NTO_SLI_CreditPoints",
  "errors": null,
  "isSuccess": true,
}
```

```
"outputValues": {  
  "PointsCredited": 100  
}  
}
```


CHAPTER 9 Loyalty Management Metadata API Types

In this chapter ...

- [AnalyticsDatasetDefinition](#)
- [BenefitAction](#)
- [Flow for Loyalty Management](#)
- [LoyaltyProgramSetup](#)
- [Settings](#)

Metadata API enables you to access some types and feature settings that you can customize in the user interface. For more information about Metadata API and to find a complete reference of existing metadata types, see [Metadata API Developer Guide](#).

AnalyticsDatasetDefinition

Represents the datasets used for analytics, such as those in CRM Analytics and Tableau.

 **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.

Parent Type

This type extends the [Metadata](#) metadata type and inherits its `fullName` field.

File Suffix and Directory Location

AnalyticsDatasetDefinition components have the suffix `.analyticsDatasetDefinition` and are stored in the `analyticsDatasetDefinitions` folder.

Version

AnalyticsDatasetDefinition components are available in API version 65.0 and later.

Special Access Rules

You must enable the Loyalty Analytics permission in your org to access this metadata.

Fields

Field Name	Description
datasetCategory	Field Type string Description Required. The type of metric that the dataset supports.
datasetName	Field Type string Description Required. Name of the dataset.
masterLabel	Field Type string

Field Name	Description
	Description Required. A user-friendly name for the analytics dataset definition, which is defined when the metadata component is created.
usageCategory	Field Type string Description Required. The category of usage that uniquely identifies an agent topic.

Declarative Metadata Sample Definition

The following is an example of an AnalyticsDatasetDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<AnalyticsDatasetDefinition xmlns="http://soap.sforce.com/2006/04/metadata">
  <datasetCategory>1</datasetCategory>
  <datasetName>analytics</datasetName>
  <masterLabel>sample</masterLabel>
  <usageCategory>PromotionPerformanceInsights</usageCategory>
</AnalyticsDatasetDefinition>
```

The following is an example package.xml that references the previous definition.

```
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
  <types>
    <members>*</members>
    <name>AnalyticsDatasetDefinition</name>
  </types>
  <version>[fctest]</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character * (asterisk) in the package.xml manifest file. For information about using the manifest file, see [Deploying and Retrieving Metadata with the Zip File](#).

BenefitAction

Represents details of an action that can be triggered automatically for a benefit. This type extends the Metadata metadata type and inherits its fullName field.



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.

File Suffix and Directory Location

BenefitAction components have the suffix `.benefitAction` and are stored in the `benefitActions` folder.

Version

BenefitAction components are available in API version 52.0 and later.

Special Access Rules

To use this metadata type, the Loyalty Management license is required

Fields

Field Name	Field Type	Description
<code>benefitActionParameters</code>	BenefitActionParameter on page 668[]	The collection of flow input parameters to be used in a benefit action.
<code>description</code>	string	The description of the benefit action.
<code>flowDefinition</code>	string	The ID of the flow associated with the benefit action.
<code>isActive</code>	boolean	Required. Indicates whether the benefit action is active (true) or inactive (false). The default value is <code>false</code> .
<code>isBenefitExpirationAllowed</code>	boolean	Indicates whether the benefit action is allowed to expire a benefit (true) or not (false). The default value is <code>false</code> .
<code>isBenefitUpdateAllowed</code>	boolean	Indicates whether the benefit action is allowed to update a benefit (true) or not (false). The default value is <code>false</code> .
<code>masterLabel</code>	string	Required. The user interface name of the benefit action.
<code>memberBenefitVariableName</code>	string	The name of the input variable that stores the member benefit ID in the flow specified for the benefit action.
<code>processType</code>	string	Required. The process type of the benefit action. The default value is <code>Loyalty</code> .
<code>type</code>	string	Required. The type of the benefit action.

BenefitActionParameter

Field Name	Field Type	Description
<code>dataType</code>	BenefitActionParameter (of type string)	Required. The data type of the parameter. Possible values are: • Boolean • Date

Field Name	Field Type	Description
		• DateTime • EntityPicklist • Number • Picklist • String
masterLabel	string	Required. The user interface name of the parameter.
name	string	The name of the parameter.
objectName	string	The object associated with the parameter when the data type is record.
parameterValues	BenefitActionParameterValue on page 669[]	A collection of predefined values for the benefit action parameter.
picklistName	string	The name of the picklist fields in the object associated with the parameter.

BenefitActionParameterValue

Field Name	Field Type	Description
parameterValue	string	Required. The predefined value of the parameter.

Declarative Metadata Sample Definition

The following is an example of a BenefitAction component.

```
<?xml version="1.0" encoding="UTF-8"?>
<BenefitAction xmlns="http://soap.sforce.com/2006/04/metadata">
  <description>Metadata API Test</description>
  <flowDefinition>test</flowDefinition>
  <isActive>true</isActive>
  <isBenefitExpirationAllowed>true</isBenefitExpirationAllowed>
  <isBenefitUpdateAllowed>true</isBenefitUpdateAllowed>
  <masterLabel>Test1</masterLabel>
  <processType>1</processType>
  <memberBenefitVariableName>test</memberBenefitVariableName>
  <benefitActionParameters>
    <name>test</name>
    <picklistName>Type</picklistName>
    <masterLabel>test</masterLabel>
    <dataType>Picklist</dataType>
    <objectName>Account</objectName>
    <parameterValues>
      <parameterValue>string</parameterValue>
    </parameterValues>
  </benefitActionParameters>
</BenefitAction>
```

```
<type>1</type>
</BenefitAction>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
  <types>
    <members>*</members>
    <name>BenefitAction</name>
  </types>
  <types>
    <members>test</members>
    <name>Flow</name>
  </types>
  <version>52.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character * (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see [Deploying and Retrieving Metadata with the Zip File](#).

Flow for Loyalty Management

Represents the metadata associated with a flow. Use Flow to create an application that takes users through a series of pages to query and update records in the database. You can also execute logic and provide branching capability based on user input to build dynamic applications.

FlowActionCall

Loyalty Management exposes additional `actionType` values for the `FlowActionCall` Metadata type.

Field Name	Field Type	Description
<code>actionType</code>	<code>InvokableActionType</code> (enumeration of type string)	Required. The action type. Additional valid values only for Loyalty Management include: <code>adjustPoints</code>—Adjusts loyalty points for a specified program member or journal transaction. This value is available in API version 51.0 and later. <code>assignTierBenefits</code>—Assigns Member Benefits to a member tier for benefits that are associated with a Benefit Action. This value is available in API version 51.0 and later. <code>cancelAccrual</code>—Cancels a specific set of accrual transactions.

Field Name	Field Type	Description
		• <code>creditPoints</code>—Credits loyalty points to a specified program member's balance. This value is available in API version 51.0 and later. • <code>cancelRedemption</code>—Reverts a specific set of redemption transactions. This value is available in API version 51.0 and later. • <code>changeTier</code>—Changes the tier for a specified program member. This value is available in API version 51.0 and later. • <code>changeTierWhenNoErrors</code>—Changes tier for a specified loyalty program member only when all the input parameters meet the criteria. This value is available in API version 51.0 and later. • <code>debitPoints</code>—Debits loyalty points to a specified program member's balance. This value is available in API version 51.0 and later. • <code>executeMemberBenefit</code>—Processes the benefit action associated with the benefit, which is assigned to a loyalty program member. This value is available in API version 51.0 and later. • <code>generateMemberReferralCode</code>—Generates a unique 8-character referral code for a loyalty program member. This value is available in API version 57.0 and later. • <code>getMemberActiveSegments</code>—Retrieve active Data Cloud market segments that a loyalty program member is a part of. • <code>getTier</code>—Gets the current tier for a specified program member. This value is available in API version 51.0 and later. • <code>getPointsBalance</code>—Gets the loyalty points balance for a specified program member. This value is available in API version 51.0 and later. • <code>getLoyaltyPromotion</code>—Gets active loyalty promotions based on a transaction journal. This value is available in API version 53.0 and later. • <code>getLoyaltyPromotionBasedOnSalesforceCDP</code>—Gets promotions for a member based on the market segment the member belongs to. This value is available in API version 53.0 and later. • <code>issueVoucher</code>—Issues a voucher for a member or contract. This value is available in API version 51.0 and later. • <code>mergeLoyaltyProgramMembership</code>—Merges two active loyalty program member records that both belong to the same loyalty program. This value is available in API version 56.0 and later. • <code>transferMemberPointsToGroups</code>—Transfers points from an individual member or a corporate member to the member's associated group. This value is available in API version 53.0 and later. • <code>transferPoints</code>—Transfers points from a source loyalty program member to a target loyalty program member, or to a group that the member is a part of. This value is available in API version 64.0 and later.

Field Name	Field Type	Description
		• <code>updateProgressForCumulativePromotionUsage</code>—Updates the progress a member has made towards attaining a cumulative type promotion. This value is available in API version 53.0 and later. • <code>unmergeLoyaltyProgramMembership</code>—Unmerges loyalty program member records that have a Merged status. The action unmerges memberships in the Merged status from the previously merged membership. This value is available in API version 56.0 and later. • <code>runProgramProcess</code>—Triggers an active loyalty program process. This value is available in API version 56.0 and later. • <code>runProgramProcessForTransactionJournal</code>—Triggers an active loyalty program process whose process type is TransactionJournal. This value is available in API version 54.0 and later. These values are reserved for internal use: • <code>addUpdateLimitsInPricingProm</code>—This value is available in API version 64.0 and later. • <code>createUpdatePricingPromotion</code>—This value is available in API version 64.0 and later. • <code>getEligibleLoyaltyPromotions</code>—This value is available in API version 64.0 and later. • <code>getExpiringPointsWithDate</code>—This value is available in API version 64.0 and later. • <code>getMemberBenefits</code>—This value is available in API version 64.0 and later. • <code>getPointsForNextTier</code>—This value is available in API version 64.0 and later.

LoyaltyProgramSetup

Represents the configuration of a loyalty program process including its parameters and rules. Program processes determine how new transaction journals are processed. When new transaction journals meet the criteria and conditions for a program process, actions that are set up in the process are triggered for the transaction journals.

 **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.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

LoyaltyProgramSetup components have the suffix `loyaltyProgramSetup` and are stored in the `loyaltyProgramSetups` folder.

Version

LoyaltyProgramSetup components are available in API version 54.0 and later for Loyalty Management and in API version 59.0 and later for Referral Marketing.

Special Access Rules

To use this metadata type, your org must have either B2C - Loyalty, B2C - Loyalty Plus, Loyalty Management - Growth, Loyalty Management - Advanced, or Referral Marketing license enabled.

Fields

Field Name	Description
<code>label</code>	Field Type string Description Name of the loyalty program that the program process is associated with. If a loyalty program or referral program with the specified name doesn't exist, a new LoyaltyProgram record is created. The name of a program must contain at least one alphanumeric character.
<code>programProcesses</code>	Field Type LoyaltyProgramProcess[] on page 673 Description Collection of loyalty program processes associated with a loyalty program or a referral program.

LoyaltyProgramProcess

Represents a collection of fields relating to a loyalty program process.

Field Name	Description
<code>description</code>	Field Type string Description The description of the loyalty program process.

Field Name	Description
executionType	Field Type LoyaltyPgmProcExecutionType (enumeration of type string) Description The mode of processing transaction journals by the loyalty program process. Possible values are: • Batch • BatchAndRealTime • RealTime
journalSubType	Field Type string Description The subtype of transaction journals processed by the loyalty program process.
journalType	Field Type string Description The type of transaction journal processed by the loyalty program process. Possible values for loyalty program: • Accrual • Redemption Possible value for referral program: • Referral
loyaltyTierGroup	Field Type string Description The tier group of a loyalty program. This field is available in API version 56.0 and later. This field isn't applicable for referral programs.
parameters	Field Type LoyaltyProgramProcessParameter[] on page 675 Description The parameters associated with the loyalty program process.
processName	Field Type string Description Required. The name of the loyalty program process.

Field Name	Description
processType	Field Type string Description Required. The type of records processed by the loyalty program process. For referral programs, the process type is <code>TransactionJournal</code>.
rules	Field Type LoyaltyProgramProcessRule[] on page 679 Description The rules associated with the loyalty program process.
status	Field Type LoyaltyPgmProcStatus (enumeration of type string) Description The status of the loyalty program process. Possible values are: • Active • Draft • Inactive  Note: Only active program processes can process transaction journals.

LoyaltyProgramProcessParameter

Represents a collection of fields relating to a parameter that's associated with the program process. Parameters are dynamic or fixed values that are used in rule. You can define the value of a parameter based on its type and data type.

Field Name	Description
condition	Field Type LoyaltyProgramProcessCondition on page 677 Description The filter condition that decides which records are stored in the parameter.
dataType	Field Type LoyaltyPgmProcParmDataType (enumeration of type string) Description The data type of the parameter. Determines the type of value that can be stored in the parameter. Possible values are:

Field Name	Description
	• Boolean • Date • DateTime • Numeric • Subject • Text
decimalPlaces	Field Type int Description The number of decimal places supported by the parameter when it is of the type Variable and data type Numeric.
description	Field Type string Description The description of the parameter.
isCollection	Field Type boolean Description Indicates whether the parameter can store multiple values when it is of the type Variable.
isInput	Field Type boolean Description Indicates whether a parameter can be used as an input outside the loyalty program process.
isOutput	Field Type boolean Description Indicates whether a parameter can be used as an output outside the loyalty program process.
objectName	Field Type string Description Name of the object whose records are stored by the parameter when it is of the type Variable and data type sObject.

Field Name	Description
parameterName	Field Type string Description Required. The name of the parameter.
parameterType	Field Type LoyaltyPgmProcParmType (enumeration of type string) Description The type of value the parameter can store. Possible values are: • Constant • Formula • Variable
value	Field Type string Description The value of the parameter when it is of the type Variable or Formula and isn't of the data type sObject.

LoyaltyProgramProcessCondition

Represents a collection of fields relating to a condition. Conditions filter records that parameters store or check whether child actions must be triggered for a transaction journal.

Field Name	Description
conditionCriteria	Field Type string Description Required. The criteria that determine when the condition is met by a record or by a transaction journal.
conditionFilterCriteria	Field Type LoyaltyProgramProcessConditionFilterCriteria[] on page 678 Description The filter criteria that determines which records or transaction journals are filtered.

Field Name	Description
conditionName	Field Type string Description Required. The name of the condition.

LoyaltyProgramProcessConditionFilterCriteria

Represents a collection of fields relating to a filter criteria that's part of a condition. Multiple filter criteria can be added for a condition. Filter criteria determine which records are filtered by related condition.

Field Name	Description
operator	Field Type LoyaltyPgmProcCondOperator (enumeration of type string) Description Required. The operator of the filter criteria. Possible values are: • Contains • DoesNotContain • EndsWith • Equals • GreaterThan • GreaterThanOrEquals • IsNotNull • IsNull • LessThan • LessThanOrEquals • NotEquals • StartsWith
sequence	Field Type int Description Required. The sequence number of the filter criteria within a condition.

Field Name	Description
sourceFieldName	Field Type string Description Required. The name of the field used in the filter criteria.
value	Field Type string Description The value of the filter criteria.
valueType	Field Type LoyaltyPgmProcCondType (enumeration of type string) Description Required. The type of value specified in the filter criteria. Possible values are: • Formula • Literal • Lookup • Parameter

LoyaltyProgramProcessRule

Represents a collection of fields relating to a rule. A rule consists of a set of conditions and actions.

Field Name	Description
actions	Field Type LoyaltyProgramProcessAction[] on page 681 Description The actions associated with the rule.
conditions	Field Type LoyaltyProgramProcessCondition[] on page 677 Description The conditions associated with the rule.
description	Field Type string

Field Name	Description
	Description The description of the rule.
endDate	Field Type date Description The date until which the rule processes transaction journals.
previousRule	Field Type string Description The rule that processes new transaction journals before the current rule. The current rule is triggered when the previous rule completes processing transaction journals.
promotion	Field Type string Description The promotion associated with the rule. When a promotion is associated with a rule, the start date, end date, and status of the promotion determines the corresponding fields of the rule.
ruleName	Field Type string Description Required. The name of the rule.
startDate	Field Type date Description The date from which the rule starts processing transaction journals.
status	Field Type LoyaltyPgmProcRuleStatus (enumeration of type string) Description The status of the rule. Possible values are: • Active • Draft • Inactive

Field Name	Description
stepMappings	Field Type LoyaltyProgramProcessRuleStepMapping[] on page 685 Description The list of step mappings associated with rule.

LoyaltyProgramProcessAction

Represents a collection of fields relating to an action.

Field Name	Description
actionName	Field Type string Description Required. The name of the action.
actionParameters	Field Type LoyaltyProgramProcessActionParameter[] on page 684 Description The parameters of the action.
actionType	Field Type LoyaltyPgmProcActionType (enumeration of type string) Description Required. The type of action. Possible values are: • These values are used in Loyalty Management: – <code>AssignParameterValues</code>—Assigns values to parameters. – <code>AssignBadgeToMember</code>—Assigns a badge to a loyalty program member. This value is available in API version 56.0 and later. – <code>Crud</code>—Creates or updates records in the target object. This value is available in API version 56.0 and later. – <code>CheckMemberBadgeAssignment</code>—Checks whether a badge is assigned to a loyalty program member. This value is available in API version 56.0 and later. – <code>ChangeMemberTier</code>—Changes the tier of a loyalty program member. This value is available in API version 56.0 and later. – <code>CreditPoints</code>—Credits points to the loyalty program member associated with the transaction journal that's processed by the rule.

Field Name	Description
	- <code>DebitPoints</code>—Debits points from the points balance of the loyalty program member associated with the transaction journal that's processed by the rule. - <code>GetMemberAttributesValues</code>—Gets the details of a loyalty program member's attribute value for the selected engagement attribute. This value is available in API version 55.0 and later. - <code>GetMemberPointBalance</code>—Gets the points balance of a loyalty program member. - <code>GetMemberPromotions</code>—Get promotions of a loyalty program member. This value is available in API version 56.0 and later. - <code>GetMemberTier</code>—Gets the tier details of a loyalty program member. - <code>GetOutputsFromDecisionTable</code>—Gets outputs provided by a decision table. This value is available in API version 56.0 and later. - <code>IncreaseUsageForCumulativePromotion</code>—Increases a loyalty program member's usage for a cumulative promotion. - <code>IssueVoucher</code>—Issues a voucher to the loyalty program member associated with the transaction journal that's processed by the rule. - <code>RedeemVoucher</code>—Redeems a voucher for the loyalty program member associated with the transaction journal that's processed by the rule. This value is available in API version 58.0 and later. - —Updates the loyalty program member's usage towards achieving a cumulative promotion by a specified value. - <code>RunFlow</code>—Runs a flow. - <code>RunProgramProcess</code>—Runs an active loyalty program process as a subprocess. This value is available in API version 56.0 and later. - <code>SendMail</code>—Sends emails to the loyalty program member for whom the process is run. This value is available in API version 59.0 and later. - <code>UpdateCurrentValueForMemberAttribute</code>—Updates the loyalty program member's current attribute value for the selected engagement attribute. This value is available in API version 55.0 and later. - <code>UpdatePointBalance</code>—Updates the points balance of the loyalty program member associated with the transaction journal that's processed by the rule. - <code>UpdateUsageForCumulativePromotion</code>—Updates a loyalty program member's usage for a cumulative promotion. • These values are used in Referral Marketing: - <code>AssignParameterValues</code>—Assigns values to parameters. - <code>Crud</code>—Creates or updates records in the target object. - <code>GetMemberAttributesValues</code>—Gets the details of an advocate's attribute value for the selected engagement attribute. - <code>GetMemberPromotions</code>—Gets the promotions of an advocate.

Field Name	Description
	- GetOutputsFromDecisionTable—Gets outputs provided by a decision table. - IssueExtendedReward—Issues an extended reward to an advocate or a referred friend. This value is available in API version 64.0 and later. - IssueVoucher—Issues a voucher to an advocate or a referred friend. - RedeemVoucher—Redeems a voucher for an advocate or a friend. - SendMail—Sends emails to a referral program's advocates and referrals. - UpdateCurrentValueForMemberAttribute—Updates an advocate's current attribute value for the selected engagement attribute. - These values are reserved for internal use: - GetCustomerPromotionAttrValue—This value is available in API version 64.0 and later. - UpdateCustomerPromotionAttrValue—This value is available in API version 64.0 and later.
crudActionType	Field Type LoyaltyPgmProcCrudActType (enumeration of type string) Description The type of operation to perform on target object records by the action. This field is available from API version 56.0 and later.  Note: This field is required when the <code>actionType</code> field value is CRUD. Possible values are: - create - update
decisionTable	Field Type string Description The decision that's invoked by the action for the transaction journal that's processed by the rule.
decisionTableDatasetLink	Field Type string Description The dataset link associated with the selected decision table.
entityApiName	Field Type string Description The API name of the target object. This field is available from API version 56.0 and later.

Field Name	Description
	 Note: This field is required when the <code>actionType</code> field value is <code>CRUD</code> .
<code>flowDefinition</code>	Field Type string Description The flow that's run by the action for the transaction journal that's processed by the rule. The selected flow must be of the type <code>LoyaltyManagementFlow</code> .
<code>loyaltyProgramProcess</code>	Field Type string Description The subprogram processes that's run by the action. This field is available from API version 56.0 and later.  Note: This field is required when the <code>actionType</code> field value is <code>RunProgramProcess</code> .

LoyaltyProgramProcessActionParameter

Represents a collection of fields relating to an action parameter. A parameter is either an input or an output for the action. Input parameters store the values used by the action. Output parameters store the result of the action.

Field Name	Description
<code>operator</code>	Field Type <code>LoyaltyPgmProcActParamOper</code> (enumeration of type string) Description The type of operator used in the action. This field is available in API version 56.0 and later. Possible value is: • <code>Equals</code>
<code>parameterName</code>	Field Type string Description Required. The name of parameter. The parameter name must be the same as the input or the output field that's supported depending on the associated action's type.
<code>sequenceNumber</code>	Field Type int

Field Name	Description
	Description The sequence number of the parameter in the action. This field is available in API version 56.0 and later.
value	Field Type string Description Required. The value of the parameter.
valueType	Field Type LoyaltyPgmProcActParamType (enumeration of type string) Description The type of value to provide in the parameter. This field is available in API version 56.0 and later. Possible values are: • <code>Literal</code>—A constant value. • <code>Parameter</code>—A runtime value passed using a parameter.

LoyaltyProgramProcessRuleStepMapping

Represents a collection of fields relating to a step mapping. Map conditions with child actions or map an action without a parent step.

Field Name	Description
associatedStep	Field Type string Description Required. The action that's associated with the mapping.
parentStep	Field Type string Description The condition that contains one or more child actions.
sequence	Field Type int Description Required.

Field Name**Description**

The sequence number of the mapping within a rule.

Declarative Metadata Sample Definition

The following is an example of a LoyaltyProgramSetup component.

```
<?xml version="1.0" encoding="UTF-8"?>
<LoyaltyProgramSetup xmlns="http://soap.sforce.com/2006/04/metadata">
  <label>Cloud Kicks Inner Circle</label>
  <programProcesses>
    <executionType>RealTime</executionType>
    <parameters>
      <dataType>Numeric</dataType>
      <decimalPlaces>0</decimalPlaces>
      <isCollection>>false</isCollection>
      <isInput>>false</isInput>
      <isOutput>>false</isOutput>
      <parameterName>VoucherValue</parameterName>
      <parameterType>Constant</parameterType>
      <value>50</value>
    </parameters>
    <processName>Issue Vouchers</processName>
    <processType>Transaction Journal</processType>
    <rules>
      <actions>
        <actionName>Issue High Transaction Value Voucher</actionName>
        <actionParameters>
          <operator>Equals</operator>
          <parameterName>VoucherDefinitionName</parameterName>
          <sequenceNumber>1</sequenceNumber>
          <value>Voucher for High Value Transactions</value>
          <valueType>Literal</valueType>
        </actionParameters>
        <actionParameters>
          <operator>Equals</operator>
          <parameterName>VoucherCode</parameterName>
          <sequenceNumber>2</sequenceNumber>
          <value>{!TransactionJournal.Order.Id}</value>
        </actionParameters>
        <actionParameters>
          <operator>Equals</operator>
          <parameterName>VoucherEffectiveDate</parameterName>
          <sequenceNumber>3</sequenceNumber>
          <value>DATEVALUE(&quot;2021-11-21 00:00:00&quot;)</value>
        </actionParameters>
        <actionParameters>
          <operator>Equals</operator>
          <parameterName>VoucherExpirationDate</parameterName>
          <sequenceNumber>4</sequenceNumber>
          <value>DATEVALUE(&quot;2022-01-01 00:00:00&quot;)</value>
        </actionParameters>
      </actions>
    </rules>
  </programProcesses>
</LoyaltyProgramSetup>
```

```

        <actionParameters>
            <operator>Equals</operator>
            <parameterName>VoucherFaceValue</parameterName>
            <sequenceNumber>5</sequenceNumber>
            <value>{!VoucherValue}</value>
        </actionParameters>
        <actionType>IssueVoucher</actionType>
    </actions>
    <conditions>
        <conditionCriteria>1</conditionCriteria>
        <conditionFilterCriteria>
            <operator>GreaterThanOrEquals</operator>
            <sequence>1</sequence>
            <sourceFieldName>TransactionJournal.TransactionAmount</sourceFieldName>

            <value>100</value>
            <valueType>Literal</valueType>
        </conditionFilterCriteria>
        <conditionName>New Condition</conditionName>
        <conditionType>Condition</conditionType>
    </conditions>
    <endDate>2022-01-01</endDate>
    <ruleName>Issue Voucher for Transactions Above $100</ruleName>
    <startDate>2021-11-21</startDate>
    <status>Draft</status>
    <stepMappings>
        <associatedStep>New Condition</associatedStep>
        <sequence>1</sequence>
    </stepMappings>
    <stepMappings>
        <associatedStep>Issue High Transaction Value Voucher</associatedStep>
        <parentStep>New Condition</parentStep>
        <sequence>1</sequence>
    </stepMappings>
</rules>
<status>Draft</status>
</programProcesses>
</LoyaltyProgramSetup>

```

The following is an example package.xml that references the previous definition.

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
    ~ Copyright 2020 Salesforce, Inc.
    ~ All Rights Reserved
    ~ Company Confidential
-->
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>LoyaltyProgramSetup</name>
    </types>
    <version>54.0</version>
</Package>

```

 **Note:** To retrieve metadata specific to any loyalty program, mention the loyalty program name in the `<members>` tag. The generated file contains all the information regarding that loyalty program.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see [Deploying and Retrieving Metadata with the Zip File](#).

Settings

Represents the organization settings to control the update of non-qualifying points balance for members when points are credited. For more information, see [Settings](#).

[IndustriesGamificationSettings](#)

Represents the settings to enable games in Loyalty Management.

[IndustriesLoyaltySettings](#)

Represents the settings to enable capabilities of Loyalty Management.

[IndustriesUnifiedPromotionsSettings](#)

Represents the settings to Global Promotions Management.

IndustriesGamificationSettings

Represents the settings to enable games in Loyalty Management.

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See [Settings](#) for more details.

File Suffix and Directory Location

IndustriesGamificationSettings values are stored in a single file named `IndustriesGamification.settings` in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

IndustriesGamificationSettings components are available in API version 60.0 and later.

Fields

Field Name	Field Type	Description
<code>enableGamification</code>	boolean	Indicates whether the capability to offer games to loyalty program members as program reward is enabled (<code>true</code>) or disabled (<code>false</code>) for your org. The default value is <code>false</code> .

Declarative Metadata Sample Definition

The following is an example of a `IndustriesGamificationSettings` component.

```
<?xml version="1.0" encoding="UTF-8"?>
<IndustriesGamificationSettings xmlns="http://soap.sforce.com/2006/04/metadata">
  <enableGamification>true</enableGamification>
</IndustriesGamificationSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
  <types>
    <members>IndustriesGamification</members>
    <name>Settings</name>
  </types>
  <version>60.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see [Settings](#). For information about using the manifest file, see [Deploying and Retrieving Metadata with the Zip File](#).

IndustriesLoyaltySettings

Represents the settings to enable capabilities of Loyalty Management.

This type extends the `Metadata` metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the `Settings` name. See [Settings](#) for more details.

File Suffix and Directory Location

`IndustriesLoyaltySettings` values are stored in a single file named `IndustriesLoyalty.settings` in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

`IndustriesLoyaltySettings` components are available in API version 53.0 and later.

Fields

Field Name	Field Type	Description
<code>enableAutomaticTierAssessmentSelection</code>	boolean	Indicates whether the capability that automatically selects members as eligible for tier assessment when members' qualifying points balance changes is enabled (<code>true</code>) or disabled (<code>false</code>) for your org. The default value is <code>false</code> . This field is available in API version 58.0 or later.

Field Name	Field Type	Description
<code>enableAutomaticVoucherCodeGeneration</code>	boolean	Indicates whether the capability to automatically generate voucher codes is enabled (<code>true</code>) or not (<code>false</code>). This field is available in API version 57.0 and later.
<code>enableConfigureClubs</code>	boolean	Indicates whether the capability to create and manage clubs is enabled (<code>true</code>) or not (<code>false</code>). The default value is <code>false</code> . This field is available in API version 64.0 or later.
<code>enableFixedTypeNQPAggregation</code>	boolean	Indicates whether the capability to aggregate and expire fixed type non-qualifying points in batches is enabled (<code>true</code>) or disabled (<code>false</code>) for your org. The default value is <code>true</code> . This field is available in API version 54.0 or later.
<code>enableLoyaltyApiAccessForExternalSiteUsers</code>	boolean	Indicates whether the capability that allows customers to join and leave loyalty programs and allows members to join and opt out of promotions from Experience Cloud sites is enabled (<code>true</code>) or disabled (<code>false</code>) for your org. The default value is <code>false</code> . This field is available in API version 58.0 or later.
<code>enableLoyaltyGenerativeAi</code>	boolean	Indicates whether the Einstein AI capabilities of Loyalty Management are enabled (<code>true</code>) or not (<code>false</code>). The default value is <code>false</code> . This field is available in API version 62.0 or later.
<code>enableLoyaltyRedeemPointsExpirationInfoPref</code>	boolean	Indicates whether the capability that automatically calculates and adds the expiration date of points credited back to members for canceled redemptions is enabled (<code>true</code>) or disabled (<code>false</code>) for your org. The default value is <code>true</code> . This field is available in API version 55.0 or later.
<code>enableLoyaltyRulesVerifyForMemberSegment</code>	boolean	Indicates whether the capability that allows: • Loyalty program process rules to process transaction journals only when the loyalty program member is part of a Data Cloud segment associated with the rule's promotion is enabled (<code>true</code>) or disabled (<code>false</code>) for your org. • The Promotion Eligibility component on Loyalty Program Member record page to categorize promotions based on whether the member belongs to the promotion's campaign or Data Cloud segment is enabled (<code>true</code>) or disabled (<code>false</code>) for your org. The default value is <code>false</code>. This field is available in API version 55.0 or later.
<code>enableLoyaltyServiceExcellence</code>	boolean	Indicates whether Service Console for Loyalty Management is enabled (<code>true</code>) or disabled (<code>false</code>) for your org. The default value is <code>false</code> . This field is available in API version 57.0 or later.
<code>enableNegativePointBalance</code>	boolean	Indicates whether the capability that lets loyalty program members to hold negative point balances for non-qualifying currencies is enabled (<code>true</code>) or disabled (<code>false</code>) for your org. The default value is <code>false</code> . This field is available in API version 61.0 or later.

Field Name	Field Type	Description
<code>enableNonQualifyingPointsConsolidation</code>	boolean	Indicates whether the capability to aggregate and expire fixed type non-qualifying points in real time is enabled (<code>true</code>) or disabled (<code>false</code>) for your org. The default value is <code>true</code> . This field is available in API version 58.0 or later.
<code>enablePointsLifecycleTracking</code>	boolean	Indicates whether the capability that allows the company to trace how members redeem their accrued fixed-model non-qualifying points is enabled (<code>true</code>) or disabled (<code>false</code>) for your org. The default value is <code>false</code> . This field is available in API version 62.0 or later.
<code>enablePromSetupPromRuleStatusInheritance</code>	boolean	Indicates whether the capability, which allows Promotion Setup processes and rules to be deployed to target orgs in the same status as the source orgs when using the <code>LoyaltyProgramSetup</code> metadata type, is enabled (<code>true</code>) or disabled (<code>false</code>). The default value is <code>false</code> . This field is available in API version 62.0 or later.
<code>enableQPREalTimePointBalance</code>	boolean	Indicates whether the real time qualifying points balance update capability is enabled (<code>true</code>) or disabled (<code>false</code>) for your org. The default value is <code>true</code> . This field is available in API version 55.0 or later.
<code>enableNQPREalTimePointBalance</code>	boolean	Indicates whether the capability to update the non-qualifying point balance of members in real time is enabled (<code>true</code>) or disabled (<code>false</code>) for your org. The default value is <code>true</code> .
<code>enableSegmentQueryApiMultipleDataSpace</code>	boolean	Indicates whether the capability that allows Query API to verify the Data Cloud segments that customers are part of across multiple data spaces is enabled (<code>true</code>) or not (<code>false</code>). The default value is <code>false</code> . This field is available in API version 64.0 or later.
<code>enableSegmentQueryByDataGraph</code>	boolean	Indicates whether the capability, which uses Data Cloud data graphs to query for Data Cloud segments that customers are part of, is enabled (<code>true</code>) or disabled (<code>false</code>). The default value is <code>false</code> . This field is available in API version 62.0 or later.
<code>enableTransferPointsToMemberGroupRealtime</code>	boolean	Indicates whether the capability that automates real-time transfer of a member's non-qualifying points to associated group is enabled (<code>true</code>) or not (<code>false</code>). The default value is <code>false</code> . This field is available in API version 64.0 or later.
<code>enableUsePromPtyUsageForEngmtTrail</code>	boolean	Indicates whether the capability that uses Promotion Party Usage records to store attributes for Engagement Trail promotions is enabled (<code>true</code>) or not (<code>false</code>). The default value is <code>true</code> . This field is available in API version 64.0 or later.

Declarative Metadata Sample Definition

The following is an example of a `IndustriesLoyaltySettings` component.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<IndustriesLoyaltySettings xmlns="http://soap.sforce.com/2006/04/metadata">
  <enableNQPPRealTimePointBalance>true</enableNQPPRealTimePointBalance>
  <enableFixedTypeNQPAggregation>true</enableFixedTypeNQPAggregation>
  <enableQPPRealTimePointBalance>true</enableQPPRealTimePointBalance>

  <enableLoyaltyRulesVerifyCdpMemberSegment>false</enableLoyaltyRulesVerifyCdpMemberSegment>

  <enableLoyaltyRedeemedPointsExpirationInfoPref>true</enableLoyaltyRedeemedPointsExpirationInfoPref>

  <enableLoyaltyServiceExcellence>true</enableLoyaltyServiceExcellence>
  <enableAutomaticVoucherCodeGeneration>true</enableAutomaticVoucherCodeGeneration>

  <enableLoyaltyApiAccessForExternalSiteUsers>false</enableLoyaltyApiAccessForExternalSiteUsers>

  <enableAutomaticMemberTierAssessmentSelection>false</enableAutomaticMemberTierAssessmentSelection>

  <enableNonQualifyingPointsConsolidation>false</enableNonQualifyingPointsConsolidation>

  <enablePointsLifecycleTracking>false</enablePointsLifecycleTracking>
  <enableNegativePointBalance>false</enableNegativePointBalance>
  <enableSegmentQueryByDataGraph>false</enableSegmentQueryByDataGraph>

  <enablePromSetupProcRuleStatusInheritDplymt>false</enablePromSetupProcRuleStatusInheritDplymt>

  <enableLoyaltyGenerativeAi>false</enableLoyaltyGenerativeAi>

  <enableTransferPointsToMemberGroupsRealtime>false</enableTransferPointsToMemberGroupsRealtime>

  <enableUsePromPtyUsageForEngmtTrail>false</enableUsePromPtyUsageForEngmtTrail>
  <enableSegmentQueryApiMultipleDataSpace>true</enableSegmentQueryApiMultipleDataSpace>

  <enableConfigureClubs>true</enableConfigureClubs>
</IndustriesLoyaltySettings>

```

The following is an example `package.xml` that references the previous definition.

```

<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
  <types>
    <members>IndustriesLoyalty</members>
    <name>Settings</name>
  </types>
  <version>53.0</version>
</Package>

```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see [Settings](#). For information about using the manifest file, see [Deploying and Retrieving Metadata with the Zip File](#).

IndustriesUnifiedPromotionsSettings

Represents the settings to Global Promotions Management.

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for more details.

File Suffix and Directory Location

IndustriesUnifiedPromotionsSettings values are stored in a single file named `IndustriesUnifiedPromotions.settings` in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

IndustriesUnifiedPromotionsSettings components are available in API version 60.0 and later.

Fields

Field Name	Field Type	Description
<code>enableGlobalPromotionsEvaluation</code>	boolean	Indicates whether promotions evaluation is enabled (<code>true</code>) or disabled (<code>false</code>) in your org. The default value is <code>false</code> . When enabled, the system evaluates cart details and returns the adjusted price of the cart and line items after applying eligible promotions. Available in API version 65.0 or later.
<code>enableGlobalPromotionsForRevenueCloud</code>	boolean	Indicates whether global promotions is enabled (<code>true</code>) or disabled (<code>false</code>) for Revenue Cloud in your org. The default value is <code>false</code> . Available in API version 66.0 or later.
<code>enableGlobalPromotionsProductCatalogManagement</code>	boolean	Indicates whether Product Catalog Management is enabled (<code>true</code>) or disabled (<code>false</code>) in your org. The default value is <code>false</code> . Available in API version 62.0 or later.
<code>enableUnifiedPromotions</code>	boolean	Indicates whether Global Promotions Management is enabled (<code>true</code>) or disabled (<code>false</code>) for your org. The default value is <code>true</code> .

Declarative Metadata Sample Definition

The following is an example of a IndustriesGamificationSettings component.

```
<?xml version="1.0" encoding="UTF-8"?><IndustriesUnifiedPromotionsSettings
xmlns="http://soap.sforce.com/2006/04/metadata">
  <enableUnifiedPromotions>true</enableUnifiedPromotions>

  <enableGlobalPromotionsProductCatalogManagement>true</enableGlobalPromotionsProductCatalogManagement>

  <enableGlobalPromotionsEvaluation>true</enableGlobalPromotionsEvaluation>
</IndustriesUnifiedPromotionsSettings>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
  <types>
    <members>IndustriesUnifiedPromotions</members>
    <name>Settings</name>
  </types>
  <version>66.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character * (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see [Settings](#). For information about using the manifest file, see [Deploying and Retrieving Metadata with the Zip File](#).

CHAPTER 10 Loyalty Management Tooling API Objects

In this chapter ...

- [BenefitAction](#)

Tooling API exposes metadata used in developer tooling that you can access through REST or SOAP. Tooling API's SOQL capabilities for many metadata types allow you to retrieve smaller pieces of metadata. For more information about Tooling API objects and to find a complete reference of all the supported objects, see [Introducing Tooling API](#).

BenefitAction

Represents details of an action that can be triggered for a benefit. This object is available in API version 51.0 and later.



Note: 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.

Supported SOAP API Calls

`create()`, `delete()`, `describeSObjects()`, `query()`, `retrieve()`, `update()`, `upsert()`

Supported REST API Methods

`DELETE`, `GET`, `HEAD`, `PATCH`, `POST`, `Query`

Special Access Rules

To use this object, the Loyalty Management license is required.

Fields

Field	Details
Description	Type textarea Properties Filter, Group, Nillable, Sort Description The description of the benefit action.
DeveloperName	Type string Properties Filter, Group, Sort Description The unique name of the object in the API. This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. In managed packages, this field prevents naming conflicts on package installations. With this field, a developer can change the object's name in a managed package and the changes are reflected in a subscriber's organization. This field is automatically generated, but you can supply your own value if you create the record using the API. When creating large sets of data, always specify a unique <code>DeveloperName</code> for each record. If no <code>DeveloperName</code> is specified, performance may slow while Salesforce generates one for each record.

Field	Details
FlowDefinition	Type picklist Properties Filter, Group, Nillable, Restricted picklist, Sort Description The ID of the flow associated with the benefit action. The possible values depend on the flows defined in the org.
FlowInstance	Type string Properties Filter, Group, Nillable, Sort Description The flow associated with the benefit action.
FullName	Type string Properties Create, Group, Nillable Description The full name of the associated in Metadata API. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
IsActive	Type boolean Properties Defaulted on create, Filter, Group, Sort Description Indicates whether the benefit action is active. The default value is <code>false</code>.
IsBenefitExpirationAllowed	Type boolean Properties Defaulted on create, Filter, Group, Sort Description Indicates whether the benefit action is allowed to expire the benefit. The default value is <code>false</code>.

Field	Details
IsBenefitUpdateAllowed	Type boolean Properties Defaulted on create, Filter, Group, Sort Description Indicates whether the benefit action is allowed to update the benefit. The default value is <code>false</code>.
Language	Type picklist Properties Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort Description The language of the benefit action.
ManageableState	Type picklist Properties Filter, Group, Nillable, Restricted picklist, Sort Description Indicates the manageable state of the specified component that is contained in a package. Possible values are: • <code>beta</code>—Managed-Beta • <code>deleted</code>—Managed-Proposed-Deleted • <code>deprecated</code>—Managed-Proposed-Deprecated • <code>deprecatedEditable</code>—SecondGen-Installed-Deprecated • <code>installed</code>—Managed-Installed • <code>installedEditable</code>—SecondGen-Installed-Editable • <code>released</code>—Managed-Released • <code>unmanaged</code>—Unmanaged
MasterLabel	Type string Properties Filter, Group, Sort Description The UI label of the benefit action.
MemberBenefitVariableName	Type string

Field	Details
	Properties Filter, Group, Nillable, Sort Description The name of the input variable that stores the member benefit ID in the flow specified for the benefit action. The field is available in API version 52.0 and later.
Metadata	Type BenefitAction Properties Create, Nillable, Update Description Provides access to the associated metadata type and related fields in Tooling API. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
NamespacePrefix	Type string Properties Filter, Group, Nillable, Sort Description The namespace prefix that is associated with this object. Each Developer Edition org that creates a managed package has a unique namespace prefix. Limit: 15 characters. You can refer to a component in a managed package by using the <i>namespacePrefix__componentName</i> notation.
ProcessType	Type picklist Properties Defaulted on create, Filter, Group, Restricted picklist, Sort Description The process type of the benefit action. Possible value is: • Loyalty The default value is Loyalty.
Type	Type picklist Properties Defaulted on create, Filter, Group, Restricted picklist, Sort

Field	Details
	Description The type of the benefit action. Possible value is: Flow The default value is Flow.

CHAPTER 11 Loyalty Management Mobile SDK

In this chapter ...

- [What to Expect from Loyalty Management Mobile SDK](#)
- [Supported Versions of Tools and Components](#)
- [Supported DateTime Format](#)
- [Getting Started with Loyalty Management Mobile SDK](#)
- [Loyalty Management Mobile SDK for iOS Development](#)
- [Loyalty Management Mobile SDK for Android Development](#)
- [Loyalty Management Sample App for Mobile SDK](#)
- [Receipt Scanning with Loyalty Management Mobile SDK](#)
- [Loyalty Management Mobile SDK and Sample App Support](#)
- [Upgrade Loyalty Management Mobile SDK for iOS](#)

Elevate your customer experience by providing Loyalty Management on iOS and Android mobile devices. Use Loyalty Management Mobile SDK to engage members or business customers with the brand on iOS and Android mobile channels and enhance brand engagement. The standalone is lightweight and provides customer flexibility, immersive mobile user experience, and a ready to use sample app embedded with Loyalty Management features.

Loyalty Management Mobile SDK delivers customized user experiences and provides developers with the tools to build iOS and Android mobile applications. Developers can use the SDK to produce standalone custom apps that they can distribute the app through the App Store or Google Play Store or embed the SDK in an existing app. If you're building an app from scratch, you can integrate the SDK during the development process. If you have an existing app and want to embed the SDK, modify your code to include the functionality of the SDK. To demonstrate the capabilities of Loyalty Management Mobile SDK, a sample app, which is a reference implementation of the SDK is included with the SDK. Developers can refer to these sample apps to understand the usage and best practices while building their own apps. Developers can also reuse the sample app source code that's included within the SDK and extend the code as required. The SDK is designed to work with your existing mobile app infrastructure, and provides a smooth integration process. Whether you build a new app or embed the SDK in an existing app, the SDK helps you create an effective mobile application for your loyalty program.

SEE ALSO:

[Salesforce Help: Receipt Scanning with Loyalty Management Mobile SDK](#)

[Salesforce Help: Receipt Scanning with Loyalty Management Mobile SDK](#)

EDITIONS

Available in: all editions that have Loyalty Management enabled

What to Expect from Loyalty Management Mobile SDK

Loyalty Management Mobile SDK helps a loyalty program to engage with their customers through a mobile app. Use the mobile SDK to manage members' profiles, to track transactions, to reward members, and to manage promotions and vouchers. Manage your loyalty rewards and engage with the brand.

The Loyalty Management Mobile SDK comes along with a built-in MyNTORewards sample app, which is a reference implementation of the SDK. With the sample app, you can:

- Enroll in the loyalty program.
- Review and opt in to active promotional offers, or opt out of a promotion.
- View your membership details, points summary, and tier details.
- View your transaction history.
- View and redeem your active vouchers.
- View the benefits of the loyalty program.
- View badges assigned to you.

EDITIONS

Available in: all editions that have Loyalty Management enabled

Supported Versions of Tools and Components

Loyalty Management Mobile SDK works on iOS 15.0+ and Android Studio Electric Eel 2022.1.1+ versions. Install the tools and components required to set up Loyalty Management Mobile SDK

EDITIONS

Available in: all editions that have Loyalty Management enabled

iOS

Tool or Component	Supported Version	Installation Details
Swift Version	5.7+	Installed by Xcode
Xcode	14.0+	Install from the App Store
iOS SDK	15.0+	Installed by Xcode
Swift Package Manager	5.7+	Included in Swift

Android

Tool or Component	Supported Version	Installation Details
Android Studio Electric Eel	2022.1.1+	Install from the Android Developer site
Android SDK and APIs	8.0+ (API level 26+)	Install from the Android Developer site
Android SDK Tools	8.0+	Install from the Android Developer site
Android Virtual Device	8.0+	Install from Android Studio
Kotlin	1.8.0+	Installed by Android Studio

Supported DateTime Format

Use the `yyyy-MM-dd'T'HH:mm:ss.SSSZ` format to specify dateTime fields. Currently, the Loyalty Management Mobile SDK doesn't support defining custom datetime formats.

- `yyyy` is the four-digit year
- `MM` is the two-digit month (01-12)
- `dd` is the two-digit day (01-31)
- `T` is a separator indicating that time-of-day follows
- `HH` is the two-digit hour (00-23)
- `mm` is the two-digit minute (00-59)
- `ss` is the two-digit seconds (00-59)
- `SSS` is the optional three-digit milliseconds (000-999)
- `Z` is the reference UTC timezone

EDITIONS

Available in: all editions that have Loyalty Management enabled

Getting Started with Loyalty Management Mobile SDK

Prepare to install the Loyalty Management Mobile SDK, to create a connected app, and then to set up your experience cloud users' access to the Loyalty Management Mobile SDK.

[Prepare to Install Loyalty Management Mobile SDK](#)

Prepare your org before you install Loyalty Management Mobile SDK. Before you install Loyalty Management Mobile SDK, ensure you have a Salesforce org that has Loyalty Management and Experience Cloud enabled.

[Create an External Client App to Integrate with Loyalty Management](#)

Create an external client app for Loyalty Management Mobile SDK to request access to Salesforce APIs and to enable secure integration with the APIs. To request access, integrate the app with the Salesforce API by using the OAuth 2.0 protocol.

[Prepare Experience Cloud Users for Loyalty Management Mobile SDK](#)

Prepare your Experience Cloud user before you install Loyalty Management Mobile SDK. To register or log in to the sample app for Loyalty Management Mobile SDK, it's recommended that you set up Experience Cloud for your loyalty program member profile.

EDITIONS

Available in: all editions that have Loyalty Management enabled

Prepare to Install Loyalty Management Mobile SDK

Prepare your org before you install Loyalty Management Mobile SDK. Before you install Loyalty Management Mobile SDK, ensure you have a Salesforce org that has Loyalty Management and Experience Cloud enabled.

1. Get a Salesforce [Developer Edition organization](#) that has Loyalty Management enabled.
2. To use another program, modify the program name in the Loyalty Management Mobile SDK configuration file.
3. Create an Experience Cloud site user profile for your loyalty program. See [Create Community User Profile](#).
4. [Create a connected app](#) on page 704 and enable an external application to integrate with Salesforce by using APIs and standard protocols.
5. Create and activate loyalty program processes and process rules for GetMemberPromotions, EnrollInPromotion, and OptOutFromPromotion APIs. See [Create Loyalty Program Processes by Using Promotion Setup](#).
6. Access the [Loyalty Management Mobile SDK API References for iOS](#) on page 722 and [Loyalty Management Mobile SDK API References for Android](#) on page 764.
7. Review iOS and Android Native requirements, see [Development Prerequisites for iOS and Android](#).
8. Install the required iOS and Android software. See [iOS Basic Requirements](#) and [Native Android Requirements](#).
9. To set up the Loyalty Management Sample App for Mobile SDK in Android Studio, see [Setting Up Sample Projects in Android Studio](#).

EDITIONS

Available in: all editions that have Loyalty Management enabled

USER PERMISSIONS

To enable experience cloud:

- Customize Application

To read, create, update, or delete connected apps:

- Customize Application

AND

Modify All Data

OR

Manage Connected Apps

To create a loyalty program process:

- Loyalty Management

Create an External Client App to Integrate with Loyalty Management

Create an external client app for Loyalty Management Mobile SDK to request access to Salesforce APIs and to enable secure integration with the APIs. To request access, integrate the app with the Salesforce API by using the OAuth 2.0 protocol.

1. Create an [External Client App](#).
2. Under API (Enable OAuth Settings), select **Enable OAuth Settings**.
3. In the Callback URL, enter a URL where Salesforce can send OAuth2 responses.
4. Add these scopes to the Selected OAuth Scopes list.
 - Access the identity URL service (id, profile, email, address, phone)
 - Access Visualforce applications (visualforce)
 - Manage user data via APIs (api)
 - Perform requests at any time (refresh_token, offline_access)
5. Select **Enable Authorization Code and Credential Flow**, and deselect **Require Proof Key for Code Exchange (PKCE) Extension for Supported Authorization Flows**, **Require Secret for Web Server Flow**, and **Require Secret for Refresh Token Flow**.
6. Save the app.

EDITIONS

Available in: all editions that have Loyalty Management enabled

USER PERMISSIONS

To read, create, update, or delete connected apps:

- Customize Application

AND

Modify All Data

OR

Manage Connected Apps

7. From Setup, in the Quick Find box, enter *External Client Apps Manager*, and then select **External Client Apps Manager**.
8. From the actions list for the external client app, select **Edit Policies**.
9. Under OAuth Policies, from the Permitted Users dropdown, select **Admin approved users are Pre-authorized** as the permitted users.
10. Select **Relax IP restrictions** as the IP relaxation.
11. Select **Refresh token is valid until revoked** as the refresh token policy.
12. Save your changes.

From the App Manager, for the external client app, click **Manage Consumer Details**, and copy the consumer key and consumer secret. Use these credentials in mobile applications to authenticate with Salesforce APIs.

SEE ALSO:

[Mobile SDK Development Guide: Create a Connected App](#)

[Mobile SDK Development Guide: Create a Connected App](#)

Prepare Experience Cloud Users for Loyalty Management Mobile SDK

Prepare your Experience Cloud user before you install Loyalty Management Mobile SDK. To register or log in to the sample app for Loyalty Management Mobile SDK, it's recommended that you set up Experience Cloud for your loyalty program member profile.

Create an external client app and an Experience Cloud site user for your loyalty program.

1. Enable your Experience Cloud site user to access the external client app.
 - a. From Setup, in the Quick Find box, enter *External Client Apps Manager*, and then select **External Client Apps Manager**.
 - b. From the actions list for the external client app, select **Edit Policies**.
 - c. Select your Experience Cloud site user profile.
 - d. Save your changes.
2. Enable the person account record type to the Experience Cloud site user profile.
 - a. From Setup, click **Users > Profiles**.
 - b. Click your Experience Cloud site user profile.
 - c. In Record Type Settings, find the Accounts entity, and click **Edit**.
 - d. Move the Person Account record type from Available Record Types to Selected Record Types.
 - e. Save your changes.
3. Create Loyalty Program Processes for the GetMemberPromotions, EnrollInPromotion, and OptOutFromPromotion APIs.
4. Provide the Experience Cloud site user the permission to join or to opt out of promotions from the Experience Cloud site.
 - a. From Setup, in the Quick Find box, enter *Loyalty Management*, and then select **Loyalty Management Settings**.
 - b. Turn on Customers and Members Join and Opt Out from Experience Cloud Site.

EDITIONS

Available in: all editions that have Loyalty Management enabled

USER PERMISSIONS

To read, create, update, or delete connected apps:

- Customize Application

AND

Modify All Data

OR

Manage Connected Apps

To create, edit, and delete profiles:

- Manage Profiles

AND

Permission Sets

5. Enable API access for Experience Cloud users.
 - a. From Setup, click **Users > Profiles**.
 - b. Click your Experience Cloud site user profile.
 - c. Under Administrative Permissions, select **API Enabled**.
 - d. Save your changes.
6. Add permissions for Salesforce Objects.
 - a. From Setup, click **Users > Profiles**.
 - b. Click your Experience Cloud site user profile.
 - c. Under Standard Object Permissions, enable permissions for standard objects.
 - d. Save your changes.
7. Add the sharing rules.
 - a. From Setup, in the Quick Find box, enter *Sharing*, and then select **Sharing Settings**.
 - b. Under Sharing Rules, click **New**.
 - c. Add rules for Journal Type, Loyalty Program, Loyalty Program Process, Promotion, Voucher, Voucher Definition, and Voucher Code.
 - d. Save your changes.

Supported Standard Objects for Experience Cloud Site User Setup

To set up Experience Cloud for your loyalty program member profile, enable permissions for these standard objects.

SEE ALSO:

[Salesforce Help: Create a Loyalty Program Process](#)

[Supported Standard Objects for Experience Cloud Site User Setup](#)

[Salesforce Help: Create a Loyalty Program Process](#)

[Supported Standard Objects for Experience Cloud Site User Setup](#)

Supported Standard Objects for Experience Cloud Site User Setup

To set up Experience Cloud for your loyalty program member profile, enable permissions for these standard objects.

Standard Object	Read	Create	Edit
Accounts	✓	✓	✓
Contacts	✓	✓	✓
Journal SubTypes	✓	✗	✗
Journal Types	✓	✗	✗
Loyalty Ledgers	✓	✗	✗

EDITIONS

Available in: all editions that have Loyalty Management enabled

Standard Object	Read	Create	Edit
Loyalty Member Currency	✓	✗	✗
Loyalty Member Tiers	✓	✗	✗
Loyalty Program Currencies	✓	✗	✗
Loyalty Program Processes	✓	✗	✗
Loyalty Tiers	✓	✗	✗
Loyalty Tier Groups	✓	✗	✗
Member Benefits	✓	✗	✗
Promotions	✓	✗	✗
Transaction Journals	✓	✗	✗
Vouchers	✓	✗	✗
Voucher Definitions	✓	✗	✗

Loyalty Management Mobile SDK for iOS Development

Loyalty Management Mobile SDK for iOS delivers libraries and sample Xcode project to develop mobile apps on iOS. To create the iOS application, Loyalty Management Mobile SDK for iOS uses the ForceSwift library, which uses SwiftUI to build the user interfaces in iOS apps while interacting with Salesforce.

EDITIONS

Available in: all editions that have Loyalty Management enabled

[Clone and Import Loyalty Management Mobile SDK for iOS](#)

Clone the Loyalty Management Mobile SDK GitHub repository for iOS to create a local copy of the repository on your development machine. The repository contains the code for Loyalty Management Mobile SDK for iOS and the MyNTORewards sample app.

[Loyalty Management Mobile SDK for iOS Classes](#)

Loyalty Management Mobile SDK for iOS classes organize the functionalities and data structures provided by the SDK. When building applications using the SDK, classes provide components, services, or features that you can interact with.

[Loyalty Management Mobile SDK API References for iOS](#)

These API references describe the functions, classes, methods, parameters, and return values of the Loyalty Management Mobile SDK. If your admin has enabled Receipt Make API requests to Salesforce from the POS system to enroll members, view member details, view transactions, manage vouchers, create orders, and accrue and redeem points and vouchers.

Clone and Import Loyalty Management Mobile SDK for iOS

Clone the Loyalty Management Mobile SDK GitHub repository for iOS to create a local copy of the repository on your development machine. The repository contains the code for Loyalty Management Mobile SDK for iOS and the MyNTORewards sample app.

EDITIONS

Available in: all editions that have Loyalty Management enabled

1. [Clone Mobile SDK for iOS GitHub Repository](#)

Clone the GitHub repository to copy the code base and revision history of the repository to your local computer.

2. [Import SDK into iOS Swift Projects](#)

After you clone the mobile SDK for iOS repository, import Loyalty Management Mobile SDK in Xcode to automatically download and manage external dependencies.

Clone Mobile SDK for iOS GitHub Repository

Clone the GitHub repository to copy the code base and revision history of the repository to your local computer.

EDITIONS

Available in: all editions that have Loyalty Management enabled

1. Go to the [Loyalty Management Mobile SDK iOS repository](#).
2. To copy the repository URL to your clipboard, click **Code**, and then click .
3. Open Terminal, and go to the directory where you want to clone the repository.
4. Enter `git clone https://github.com/salesforce-misc/LoyaltyMobileSDK-iOS`
5. Press Enter and wait for the repository to clone.

You can now set up your own development environment.

Import SDK into iOS Swift Projects

After you clone the mobile SDK for iOS repository, import Loyalty Management Mobile SDK in Xcode to automatically download and manage external dependencies.

EDITIONS

Available in: all editions that have Loyalty Management enabled

1. Open your existing app project in Xcode or create an app project.
2. With your app project open in Xcode, select **File > Swift Packages > Add Package Dependency**.
3. Enter the repository URL:
`https://github.com/salesforce-misc/LoyaltyMobileSDK-iOS`
4. Select a version, and click **Add Package** and on the confirmation window, click **Add Package**.
5. To import the LoyaltyMobileSDK module into your Swift file, open the swift file where you want to save LoyaltyMobileSDK, and to the first line of code, add `import LoyaltyMobileSDK`.
6. Press Enter or return.

You can now use the classes and structures in the LoyaltyMobileSDK module and API references to manage the features in your application.

[Create an App Project in Xcode](#)

If you don't have an existing app project in Xcode, create an app project to import the Loyalty Management Mobile SDK for iOS.

SEE ALSO:

[Create an App Project in Xcode](#)[Create an App Project in Xcode](#)

Create an App Project in Xcode

If you don't have an existing app project in Xcode, create an app project to import the Loyalty Management Mobile SDK for iOS.

1. Open Xcode, and click **Create a new Xcode project**.
2. Select **iOS > App**, and then click **Next**.
3. Enter the product name.
4. Select a team.
5. Select **Storyboard** as the interface and **Swift** as the language.
6. Select **Include Tests**, and then click **Next**.

EDITIONS

Available in: all editions that have Loyalty Management enabled

Loyalty Management Mobile SDK for iOS Classes

Loyalty Management Mobile SDK for iOS classes organize the functionalities and data structures provided by the SDK. When building applications using the SDK, classes provide components, services, or features that you can interact with.

EDITIONS

Available in: all editions that have Loyalty Management enabled

[ForceClient](#)

Create authentication network requests by using the ForceClient class. ForceClient uses ForceAuthenticator, which is a protocol that defines the required methods to handle access tokens in Salesforce API.

[Logger](#)

The `Logger` class provides a standardized way to log error, warning, or info messages. Trace the code execution and identify error or debug.

[LoyaltyAPIManager](#)

Manage requests related to loyalty program and interactions between the application and the APIs.

[NetworkManager](#)

The `NetworkManager` class manages network requests and processing of data. The NetworkManager class conforms to the NetworkManagerProtocol, which defines the set of methods and properties that are used to manage network requests and responses.

ForceClient

Create authentication network requests by using the ForceClient class. ForceClient uses ForceAuthenticator, which is a protocol that defines the required methods to handle access tokens in Salesforce API.

Use this signature to define the ForceClient class.

```
public class ForceClient
```

Define Authentication

Handle the authentication process and encapsulate the logic to obtain and refresh the access token. Use this resource and signature to define the auth variable.

```
auth
```

```
public var auth: ForceAuthenticator
```

Define the Network Manager

Make HTTP requests to the Salesforce server and manage the responses. Use this resource and signature to define the forceNetworkManager variable.

ForceNetworkManager

```
public var forceNetworkManager: NetworkManagerProtocol
```

Create an Instance from Force Authenticator

Create an instance from the ForceAuthenticator, and defines the required methods to handle access tokens in Salesforce API. Use this resource and signature to define the public initializer for the ForceClient class.

```
init(auth:forceNetworkManager:)
```

```
public init(auth: ForceAuthenticator, forceNetworkManager: NetworkManagerProtocol =  
NetworkManager.shared
```

Use the parameters in this table to define an authentication instance.

Parameters	Description
auth	A ForceAuthenticator instance that defines the required methods to handle access tokens in Salesforce API.
forceNetworkManager	A NetworkManagerProtocol instance that defines the requirements of a network manager.

Fetch Authentication Requests

Get all the REST requests for authentication. This request returns a decoded JSON response result. Use this resource and signature to define the fetch request.

```
fetch(type:with:urlSession:)
```

```
func fetch<T>(type: T.Type, with request: URLRequest, urlSession: URLSession = .shared)  
async throws -> T where T : Decodable
```

Use the parameters in this table to create a fetch request.

Parameters	Description
type	A type or model of data that you want to decode from the JSON format.

Parameters	Description
request	A <code>URLRequest</code> , which is executed by <code>URLSession</code> .

Logger

The `Logger` class provides a standardized way to log error, warning, or info messages. Trace the code execution and identify error or debug.

Use this signature to define the `Logger` class.

```
public class Logger
```

Log an Informational Message

Use the message parameter to log an informational message by using this resource and signature.

info

```
public static func info(_ message: String)
```

Log a Debug Message

Use the message parameter to log a debug message by using this resource and signature.

debug(_:)

```
public static func debug(_ message: String)
```

Log an Error Message

Use the message parameter to log an error message by using this resource and signature.

error(_:)

```
public static func error(_ message: String)
```

Log a Faulty Message

Use the message parameter to log a faulty message by using this resource and signature.

fault(_:)

```
public static func fault(_ message: String)
```

LoyaltyAPIManager

Manage requests related to loyalty program and interactions between the application and the APIs.

Use this signature to define the `LoyaltyAPIManager` class.

```
public class LoyaltyAPIManager
```

Define Authentication

Create an instance of `ForceAuthenticator` for authentication by using this resource and signature.

`auth`

```
public var auth: ForceAuthenticator
```

Create a Loyalty Program Name

Create a loyalty program name variable by using this resource and signature.

`loyaltyProgramName`

```
public var loyaltyProgramName: String
```

Create an Instance URL

Provide the base URL of the loyalty program API by using this resource and signature. Base URL is the root address of your Salesforce instance. For example, if the URL in your browser's address bar is `https://na35.salesforce.com/home/home.jsp`, your base URL is `https://na35.salesforce.com`.

`instanceURL`

```
public var instanceURL: String
```

Initialize the LoyaltyAPIManager Class

Initialize a `LoyaltyAPIManager` that contains the necessary parameters such as `auth`, `loyaltyProgramName`, `instanceURL`, and `forceClient` by using this resource and signature.

`init(auth:loyaltyProgramName:instanceURL:forceClient:)`

```
public init(auth: ForceAuthenticator, loyaltyProgramName: String, instanceURL: String, forceClient: ForceClient)
```

Sort Requests

Sort the requests by a specific field by using this resource and signature.

`SortBy`

```
public init(auth: ForceAuthenticator, loyaltyProgramName: String, instanceURL: String, forceClient: ForceClient)
```

Use the parameters in this table to create a sort request.

Parameter	Definition	Description
<code>expirationDate</code>	<code>case expirationDate</code>	Sorts the requests by the expiration date of the authentication token.
<code>effectiveDate</code>	<code>case effectiveDate</code>	Sorts the requests by the effective date of the authentication token.

Parameter	Definition	Description
createdDate	<code>case createdDate</code>	Sorts the requests by the created date of the authentication token.

Sort Results

Specify if you want to view the sort results in ascending or descending order by using this resource and signature.

```
public enum SortOrder : String
```

Use the parameters in this table to create a sort order request.

Parameter	Definition	Description
ascending	<code>case ascending</code>	Sorts the results in ascending order.
descending	<code>case descending</code>	Sorts the results in descending order.

Define Voucher Status

Define the possible statuses of a voucher by using this resource and signature.

VoucherStatus

```
public enum VoucherStatus : String
```

Use the parameters in this table to define the voucher statuses.

Parameter	Definition	Description
issued	<code>case issued</code>	Represents the vouchers that are issued.
cancelled	<code>case cancelled</code>	Represents the vouchers that are cancelled.
expired	<code>case expired</code>	Represents the vouchers that have expired.

Identify Resource Type

Identify the type of resource to request by using this resource and signature. Use these resources to create secure and scalable mobile applications.

Resource

```
public enum Resource
```

Use the methods in this table to get resources for a loyalty program.

Method	Definition	Description
<code>individualEnrollment(programName:version:)</code>	<code>case individualEnrollment(programName: String, version: String)</code>	Creates an individual enrollment for the loyalty program. The <code>individualEnrollment</code> method takes two parameters: <code>programName</code> ,

Method	Definition	Description
		which represents the name of the loyalty program for which the individual enrollment is being created, and <code>version</code> , which represents the API version.
<code>getMemberBenefits(memberId:version:)</code>	case <code>getMemberBenefits(memberId: String, version: String)</code>	Retrieves benefits related to the member ID. The <code>getMemberBenefits</code> method takes two parameters: <code>memberId</code> , which represents the unique identifier of the member whose benefits are being retrieved, and <code>version</code> , which represents the API version.
<code>getMemberProfile(programName:version:)</code>	<code>case getMemberProfile(programName: String, version: String)</code>	Retrieves program name related to the member ID. The <code>getMemberProfile</code> method takes two parameters: <code>programName</code> , which represents the program membership name, and <code>version</code> , which represents the API version.
<code>getTransactionHistory(programName:memberId:version:)</code>	case <code>getTransactionHistory(programName: String, memberId: String, version: String)</code>	Retrieves the transaction history associated with the member ID. The <code>getTransactionHistory</code> method takes three parameters: <code>programName</code> , which represents the loyalty program name, <code>memberId</code> , which represents the unique identifier of the member whose transaction history is being retrieved, and <code>version</code> , which represents the API version.
<code>getPromotions(programName:version:)</code>	case <code>getPromotions(programName: String, version: String)</code>	Retrieves a member's existing promotions. The <code>getPromotions</code> method takes two parameters: <code>programName</code> , which represents the name of the loyalty program, and <code>version</code> , which represents the API version.
<code>enrollInPromotion(programName:version:)</code>	case <code>enrollInPromotion(programName: String, version: String)</code>	Enrolls the member in a promotion. The <code>enrollInPromotion</code> method takes two parameters: <code>programName</code> , which represents the name of the loyalty program, and <code>version</code> , which represents the API version.
<code>unenrollPromotion(programName:version:)</code>	case <code>unenrollPromotion(programName: String, version: String)</code>	Opts out a member from an available promotion. The <code>unenrollInPromotion</code> method takes two parameters: <code>programName</code> , which represents the name of the loyalty

Method	Definition	Description
<code>getVouchers(programName: String, membershipNumber: String, version: String)</code>	<pre>case getVouchers(programName: String, membershipNumber: String, version: String)</pre>	program, and <code>version</code>, which represents the API version. Retrieves vouchers available to the member ID. The <code>getVouchers</code> method takes three parameters: <code>programName</code>, which represents the name of the loyalty program, <code>membershipNumber</code>, which represents the unique identifier of the member whose vouchers are being retrieved, and <code>version</code>, which represents the API version.

Get API Resource Path

Get the path for the API resource by using this resource and signature. The function uses a resource parameter as input and returns the URL path of the resource as a string.

`getPath(for:)`

```
public func getPath(for resource: Resource) -> String
```

Get Member Benefits

Retrieve the benefits associated with a loyalty program member for the specified API version, while asynchronously requesting data from the Member Benefits API. Use this resource and signature to get member benefits.

`getMemberBenefits(for:version:devMode:)`

```
public func getMemberBenefits(
    for memberId: String,
    version: String = LoyaltyAPIVersion.defaultVersion,
    devMode: Bool = false) async throws -> [BenefitModel]
```

The `getMemberBenefits` method takes three parameters: `memberId`, which represents the unique identifier of the member whose benefits are being retrieved, `version`, which represents the API version, and the `devMode` parameter, which determines if the request is run in developer mode.

Get Member Profile

Get the profile details of a loyalty program member, while asynchronously requesting data from the Member Profile API. Use this resource and signature to get member profile details.

`getMemberProfile(for:version:devMode:)`

```
public func getMemberProfile(
    for memberId: String,
    version: String = LoyaltyAPIVersion.defaultVersion,
    devMode: Bool = false) async throws -> ProfileModel
```

The `getMemberProfile` method takes three parameters: `memberId`, which represents the unique identifier of the member whose profile is being retrieved, `version`, which represents the API version, and the `devMode` parameter, which determines if the request is run in developer mode.

Get Community Member Profile

Retrieve the Experience Cloud user profile associated with a loyalty program member for the specified API version, while asynchronously requesting data from Salesforce. Use this resource and signature to get the Experience Cloud user profile details.

`getCommunityMemberProfile(version:devMode:)`

```
public func getCommunityMemberProfile(
    version: String = LoyaltyAPIVersion.defaultVersion,
    devMode: Bool = false) async throws -> ProfileModel
```

The `getCommunityMemberProfile` method takes two parameters: `version`, which represents the default version of the mobile SDK, and the `devMode` parameter, which determines if the request is run in developer mode.

Enroll a Member

Enroll a member to a loyalty program, while asynchronously sending data to the Individual Member Enrollments API. Use this resource and signature to enroll a member.

`postEnrollment(membershipNumber:firstName:lastName:email:phone:emailNotification:version:)`

```
public func postEnrollment(
    membershipNumber: String,
    firstName: String,
    lastName: String,
    email: String, phone: String,
    emailNotification: Bool,
    version: String = LoyaltyAPIVersion.defaultVersion) async throws -> EnrollmentOutputModel
```

This table shows the parameters in the method.

Parameter	Description
<code>membershipNumber</code>	Membership number of the member being enrolled.
<code>firstName</code>	First name of the member being enrolled.
<code>lastName</code>	Last name of the member being enrolled.
<code>email</code>	Email address of the member being enrolled.
<code>phone</code>	Phone number of the member being enrolled.
<code>emailNotification</code>	Indicates whether the member wants to receive email notifications.
<code>version</code>	The default API version.

After the enrollment, the request returns the membership number and program member ID.

Opt In For a Promotion

Enroll a loyalty member to a promotion, while asynchronously sending data to the Enroll for Promotions API. Use this resource and signature to enroll a member.

`enrollIn(promotion:for:version:)`

```
public func enrollIn(
    promotion promotionName: String,
    for membershipNumber: String,
    version: String = LoyaltyAPIVersion.defaultVersion) async throws
```

The `enrollIn` method takes three parameters: `promotionName`, which represents the promotion name the member is being enrolled to, `membershipNumber`, which represents the membership number of the member being enrolled, and `version`, which represents the API version.

Use Promotion ID to Opt Out of a Promotion

Opt out a loyalty member from a promotion by using the promotion ID. Use this resource and signature to opt out of a promotion.

`unenroll(promotionId:for:version:devMode:)`

```
public final func unenroll(
    promotionId: String,
    for membershipNumber: String,
    version: String = LoyaltyAPIVersion.defaultVersion,
    devMode: Bool = false) async throws
```

The `unenroll` method takes two parameters: `promotionID`, which represents the promotion ID from which the member is being unenrolled, `membershipNumber`, which represents the membership number of the member being unenrolled, and `version`, which represents the API version.

Use Promotion Name to Opt Out of a Promotion

Opt out a loyalty member from a promotion by using the promotion name. Use this resource and signature to opt out of a promotion.

`unenroll(promotionName:for:version:devMode:)`

```
public final func unenroll(
    promotionName: String,
    for membershipNumber: String,
    version: String = LoyaltyAPIVersion.defaultVersion,
    devMode: Bool = false) async throws
```

This table shows the parameters in the `unenroll` method.

Parameter	Description
<code>promotionName</code>	Name of the promotion from which the member is being unenrolled.
<code>membershipNumber</code>	Membership number of the member being unenrolled.
<code>version</code>	Version of the mobile SDK.
<code>devMode</code>	Determines if the request is run in developer mode.

Use Member ID to Get Enrolled Promotions

Get the promotions that the member is enrolled to, by using the loyalty program member ID while asynchronously requesting data from the Enroll for Promotions API. Use this resource and signature to get the enrolled promotions.

`getPromotions(memberId:version:devMode:)`

```
public func getPromotions(
    memberId: String,
    version: String = LoyaltyAPIVersion.defaultVersion,
    devMode: Bool = false) async throws -> PromotionModel
```

The `getPromotions` method takes three parameters: `memberID`, which represents the unique identifier of the member whose promotions are being retrieved, `version`, which represents the default API version, and `devMode`, which determines if the request is run in developer mode.

Use Membership Number to Get Enrolled Promotions

Get the promotions that the member is enrolled to, by using the loyalty program membership number while asynchronously requesting data from the Enroll for Promotions API. Use this resource and signature to get the enrolled promotions.

`getPromotions(membershipNumber:version:devMode:)`

```
public func getPromotions(
    membershipNumber: String,
    version: String = LoyaltyAPIVersion.defaultVersion,
    devMode: Bool = false) async throws -> PromotionModel
```

The `getPromotions` method takes three parameters: `membershipNumber`, which represents the membership number of the member whose promotions are being retrieved, `version`, which represents the default API version, and `devMode`, which determines if the request is run in developer mode.

Get Transaction Journals

Get transaction journals of a loyalty member. Obtain either accrual or redemption transaction journal records. Use this resource and signature to get transactions.

`getTransactions(for:pageNumber:journalTypeName:journalSubTypeName:periodStartDate:periodEndDate:version:devMode:)`

```
public func getTransactions(
    for membershipNumber: String,
    pageNumber: Int? = nil,
    journalTypeName: String? = nil,
    journalSubTypeName: String? = nil,
    periodStartDate: String? = nil,
    periodEndDate: String? = nil,
    version: String = LoyaltyAPIVersion.defaultVersion,
    devMode: Bool = false) async throws -> [TransactionJournal]
```

The `getTransactions` method takes the `pageNumber` parameter that represents the number of the page you want returned. If you don't specify a value, the first page is returned. Each page contains 200 transaction journals and the transaction journals are sorted based on the activity date.

Get Vouchers

Retrieve the vouchers issued to a loyalty member by using this resource and signature.

```
getVouchers(membershipNumber:voucherStatus:pageNumber:productId:productCategoryId:productName:productCategoryName:sortBy:sortOrder:version:devMode:)
```

```
public func getVouchers(
    membershipNumber: String,
    voucherStatus: [VoucherStatus]? = nil,
    pageNumber: Int? = nil,
    productId: [String]? = nil,
    productCategoryId: [String]? = nil,
    productName: [String]? = nil,
    productCategoryName: [String]? = nil,
    sortBy: SortBy? = nil,
    sortOrder: SortOrder? = nil,
    version: String = LoyaltyAPIVersion.defaultVersion,
    devMode: Bool = false) async throws -> [VoucherModel]
```

This table shows the parameters in the `getVouchers` method.

Parameter	Description
<code>membershipNumber</code>	Membership number of the member whose vouchers are being retrieved.
<code>voucherStatus</code>	The status of a voucher.
<code>pageNumber</code>	Number of the page you want returned. If you don't specify a value, the first page is returned. By default, each page contains 200 vouchers, which are sorted based on the voucher creation date.
<code>productId</code>	The ID of the product related to the vouchers that you want to retrieve. You can specify the ID of up to 20 products.
<code>productCategoryId</code>	The ID of the product categories related to the vouchers that you want to retrieve. You can specify the ID of up to 20 product categories.
<code>productName</code>	The product name related to the voucher.
<code>productCategoryName</code>	The name of the product categories related to the vouchers that you want to retrieve. You can specify the ID of up to 20 product category names.
<code>sortBy</code>	Specifies the field used to sort the vouchers you want to retrieve. The default value is <code>ExpirationDate</code> .
<code>sortOrder</code>	Specifies the sort order of the vouchers you want to retrieve. The default value is <code>Ascending</code> .
<code>version</code>	Version of the API.
<code>devMode</code>	Determines if the request is run in developer mode.

Create an Instance of LoyaltyAPIManager Class

Create an instance of `LoyaltyAPIManager` class and interact with the Loyalty Management APIs.

Create an Instance of LoyaltyAPIManager Class

Create an instance of `LoyaltyAPIManager` class and interact with the Loyalty Management APIs.

1. Create an instance of `LoyaltyAPIManager` by using this signature.

```
let loyaltyAPIManager = LoyaltyAPIManager(auth: forceAuthenticator, loyaltyProgramName:
    "YourLoyaltyProgramName", instanceURL: "YourInstanceURL", forceClient: forceClient)
```

2. Call the appropriate methods to interact with the Loyalty Management API by using this signature.

```
import LoyaltyMobileSDK

let instanceURL = URL(string: "https://your_salesforce_instance_url")!
let loyaltyProgramName = "YourLoyaltyProgramName"
let forceClient = ForceClient(clientId: "your_client_id", clientSecret:
    "your_client_secret", redirectURI: "your_redirect_uri")

let loyaltyAPIManager = LoyaltyAPIManager(instanceURL: instanceURL, loyaltyProgramName:
    loyaltyProgramName, forceClient: forceClient)
```

3. To enroll an individual member, use this signature.

```
let enrollmentOutput = try await loyaltyAPIManager.postEnrollment(
    membershipNumber: membershipNumber,
    firstName: firstName,
    lastName: lastName,
    email: email,
    phone: phone,
    emailNotification: emailNotification)
```

For example:

```
let membershipNumber = "1234567890"
let firstName = "John"
let lastName = "Doe"
let email = "john.doe@example.com"
let phone = "4157891234"
let emailNotification = true
```

4. To retrieve the member benefits, use this signature.

```
let benefits = try await loyaltyAPIManager.getMemberBenefits(for: "memberId")
```

5. To retrieve the member profile, use this signature.

```
let profile = try await loyaltyAPIManager.getMemberProfile(for: "memberId")
```

6. To retrieve the Experience Cloud user profile, use this signature.

```
let communityProfile = try await loyaltyAPIManager.getCommunityMemberProfile()
```


7. To opt a member in to a promotion, use this signature.

```
try await loyaltyAPIManager.enrollIn(promotion: "PromotionName", for: "1234567890")
```

8. To opt out a member from a promotion using promotion ID or promotion name, use this signature.

```
try await loyaltyAPIManager.unenroll(promotionId: "promotionId", for: "1234567890")
try await loyaltyAPIManager.unenroll(promotionName: "PromotionName", for: "1234567890")
```

9. To retrieve loyalty member transactions, use this signature.

```
let transactions = try await loyaltyAPIManager.getTransactions(for: "1234567890")
```

10. To retrieve promotions for a loyalty member, use this signature.

```
let promotions = try await loyaltyAPIManager.getPromotions(membershipNumber: "1234567890")
```

11. To retrieve vouchers for a loyalty member, use this signature.

```
let vouchers = try await loyaltyAPIManager.getVouchers(membershipNumber: "1234567890")
```

NetworkManager

The `NetworkManager` class manages network requests and processing of data. The `NetworkManager` class conforms to the `NetworkManagerProtocol`, which defines the set of methods and properties that are used to manage network requests and responses.

Use this signature to define the `NetworkManager` class.

```
public class NetworkManager : NetworkManagerProtocol
```

Create a Shared Instance

Create a shared instance of the `NetworkManager` class by using this signature.

shared

```
public static let shared: NetworkManager
```

Get REST Requests

Use `Async/Await` to get all the REST requests asynchronously by using this signature.

`fetch(type:request:urlSession:)`

```
public func fetch<T>(type: T.Type, request: URLRequest, urlSession: URLSession =
URLSession.shared) async throws -> T where T : Decodable
```

Use the parameters in this table to create a fetch request.

Parameters	Description
type	A type or model of data that you want to decode from the JSON format.
request	The <code>URLRequest</code> , which is executed by <code>URLSession</code> .

Parameters	Description
urlSession	An instance of <code>URLSession</code> , which is used to execute the request and manage HTTP requests and responses. The default instance is <code>URLSession.shared</code> .

Loyalty Management Mobile SDK API References for iOS

These API references describe the functions, classes, methods, parameters, and return values of the Loyalty Management Mobile SDK. If your admin has enabled Receipt Make API requests to Salesforce from the POS system to enroll members, view member details, view transactions, manage vouchers, create orders, and accrue and redeem points and vouchers.

EDITIONS

Available in: all editions that have Loyalty Management enabled.

[Individual Member Enrollments](#)

Enroll an individual member in a loyalty program, while asynchronously sending data to the Individual Member Enrollments API.

[Enroll for Promotions](#)

Enroll loyalty program member to a promotion, while asynchronously sending data to the Enroll for Promotions API.

[Expense Analysis](#)

Stores the product and information from a receipt submitted for review by a member. To get the receipt information, access the `LoyaltyProgramMember` object reference.

[Get Member Promotions](#)

Retrieve a list of promotions that the member is eligible for, while asynchronously requesting data from the Get Member Promotions loyalty program process API.

[Member Benefits](#)

Retrieve a list of benefits for a member, while asynchronously requesting data from the Member Benefits API.

[Member Profile](#)

Get the profile details of a loyalty program member, while asynchronously requesting data from the Member Profile API.

[Member Vouchers](#)

Retrieve a list of vouchers issued to the member, while asynchronously requesting data from the Member Vouchers API.

[Receipt Status Update](#)

Update the status of the receipt, reward points, and create a transaction journal for each line item in the receipt.

[Receipt Upload](#)

Upload and scan a receipt with a receipt scanning app that uses optical character recognition to read and interpret receipts. For the unlocked package, this API uploads the image into the Amazon S3 bucket.

[Transaction History](#)

Retrieve member's transaction history, while asynchronously requesting data from the Transaction History API.

[Opt Out from a Promotion](#)

Opt out a loyalty program member from an available promotion.

Individual Member Enrollments

Enroll an individual member in a loyalty program, while asynchronously sending data to the Individual Member Enrollments API.

Resource

```
postEnrollment(membershipNumber:firstName:lastName:email:phone:emailNotification:version:)
```

Signature

```
public func postEnrollment(
    membershipNumber: String,
    firstName: String,
    lastName: String,
    email: String,
    phone: String,
    emailNotification: Bool,
    version: String = LoyaltyAPIVersion.defaultVersion) async throws -> EnrollmentOutputModel
```

Input Parameters

Parameter	Type	Description	Required or Optional
email	String	Email address of the member being enrolled.	Required
emailNotification	Boolean	Indicates whether the member wants to receive email notifications ((true)) or not ((false)).	Required
firstName	String	First name of the member being enrolled.	Required
lastName	String	Last name of the member being enrolled.	Required
membershipNumber	String	Membership number of the member being enrolled.	Required
phone	String	Phone number of the member being enrolled.	Required
version	String	Version of the API. The request body is available in version 55.0 and later.	Optional

Return Value

An EnrollmentOutputModel instance of type:

```
public struct EnrollmentOutputModel : Codable
```

Sample Output

```
{
  "contactId": "0034x00001JdPg6",
  "loyaltyProgramMemberId": "01M4x000000LOrM",
  "loyaltyProgramName": "NTO Insider",
  "membershipNumber": "Member2",
  "transactionJournals": [
    {
      "activityDate": "2022-01-01T08:00:00.000Z",
      "journalSubType": "Member Enrollment",
      "journalType": "Accrual",
      "loyaltyProgram": "NTO Insider",
      "loyaltyProgramMember": "Member2",
      "referredMember": null,
      "status": "Pending",
      "transactionJournalId": "01V4x000000CcBF"
    }
  ]
}
```

Output Parameters

Parameter	Type	Description
activityDate	String	The date and time of when transaction journal is created.
contactId	String	The ID of the loyalty program member's person account.
journalSubType	String	The journal subtype of the transaction journal.
journalType	String	The journal type of the transaction journal.
loyaltyProgram	String	The loyalty program related with the transaction journal.
loyaltyProgramMember	String	The membership number of the loyalty program related to the transaction journal.
loyaltyProgramMemberId	String	The ID of the loyalty program member record created for the member.
loyaltyProgramName	String	The name of the loyalty program into which the member is enrolled.
membershipNumber	String	The membership number of the enrolled member.
referredMember	String	The membership number of the loyalty program member who has been referred by the member.

Parameter	Type	Description
status	String	The status of the transaction journal.
transactionJournalId	String	The ID of the transaction journal that is created.

Enroll for Promotions

Enroll loyalty program member to a promotion, while asynchronously sending data to the Enroll for Promotions API.

Resource

```
enrollIn(promotion:for:version:)
```

Signature

```
public func enrollIn(  
    promotionName: String,  
    membershipNumber: String,  
    version: String = LoyaltyAPIVersion.defaultVersion) async throws
```

Input Parameters

Parameter	Type	Description	Required or Optional
promotionName	String	Name of the promotion that the member wants to enroll into.	Required
membershipNumber	String	Unique identifier of the member.	Required
version	String	Version of the API. The request body is available in version 55.0 and later.	Optional

Sample Output

```
{  
  "message": null,  
  "outputParameters": {  
    "outputParameters": {  
      "results": [  
        {  
          "MemberId": "11p4x00000002ffAAA"  
        }  
      ]  
    }  
  },  
  "simulationDetails": {},  
  "status": true
```

```
}
```

Output Parameter

Parameter	Type	Description
memberId	String	The ID of the member enrolled for the promotion.

Expense Analysis

Stores the product and information from a receipt submitted for review by a member. To get the receipt information, access the LoyaltyProgramMember object reference.

Resource

```
services/apexrest/expense-analysis
```

Rest Method Type

POST

Signature

```
curl --location --request POST
'https://yourInstance.salesforce.com/services/apexrest/expense-analysis?filename=8925687d-7489-7eb-5307-19d483b650.jpeg&membershipnumber=12434566'
\
--header 'Authorization: Bearer
00DB000000FX0aR!ARQAQCyXQGgMxjFvjshnPCDJEZyPavtzxM26XvjDU.gr56iTykGp_EQCZ0yyilQZzILDqOlmy3KpJqhKAOC56o4jqx6CxU'
\
--header 'Cookie: BrowserId=MZ4W4WebEe6bX0_gtfh0Tw; CookieConsentPolicy=0:1;
LSKey-c$CookieConsentPolicy=0:1'
```

Request Parameters

Parameter	Type	Description	Required or Optional
filename	String	The file that contains the image of the receipt.	Required
membershipnumber	String	The unique identifier of the member.	Required

Sample Output

```
{
  "storeName" : "alpine nutrition",
  "storeAddress" : "625 Market Street,\n6th floor,\nSan Francisco,\nCA, 94105",
```

```

"receiptSFDCId" : "a0PHp00000tstqeMAA",
"receiptNumber" : "#0529-0002",
"receiptDate" : "2020-01-10",
"memberShipNumber" : "124345696",
"lineItem" : [ {
  "quantity" : "5",
  "productName" : "Alpine-Bran Cereal",
  "price" : "$7.99",
  "lineItemPrice" : "$39.95",
  "isEligible" : false
}, {
  "quantity" : "2",
  "productName" : "Alpine-Oat Cereal",
  "price" : "$7.99",
  "lineItemPrice" : "$15.98",
  "isEligible" : false
} ],
"confidenceStatus" : "Partial"
}

```

Output Parameters

Parameter	Type	Description
confidenceStatus	String	The receipt analysis status based on the confidence value defined in the app. Possible values are: • Success • Failure • Partial
isEligible	Boolean	Indicates whether the item is eligible for accrual of loyalty points (<code>true</code>) or not (<code>false</code>). This value is displayed only if the admin has opted to display ineligible items in the receipt.
lineItem	String	The details of the receipt line item.
lineItemPrice	String	The total price of the receipt line item.
memberShipNumber	String	The unique identifier of the member.
price	Currency	The unit price of the receipt line item.
productName	String	The name of the product purchased.
quantity	Integer	The quantity of the product purchased.
receiptDate	dateTime	The date when the receipt is generated.

Parameter	Type	Description
receiptNumber	String	The receipt number.
receiptSFDCId	Reference	The unique ID of the receipt.
storeAddress	String	The store address.
storeName	String	The store name.

SEE ALSO:

[Salesforce Help: Set Up Amazon Web Services](#)

[Salesforce Help: Set Up Salesforce Org With Loyalty Management Receipt Scanner Kit Unlocked Package](#)

[Salesforce Help: Set Up Amazon Web Services](#)

[Salesforce Help: Set Up Salesforce Org With Loyalty Management Receipt Scanner Kit Unlocked Package](#)

Get Member Promotions

Retrieve a list of promotions that the member is eligible for, while asynchronously requesting data from the Get Member Promotions loyalty program process API.

Resource

Here's the request when the input parameter is member ID.

```
getPromotions(memberId:version:devMode:)
```

Signature

```
public func getPromotions(  
    memberId: String,  
    version: String = LoyaltyAPIVersion.defaultVersion,  
    devMode: Bool = false) async throws -> PromotionModel
```

Here's the request when the input parameter is membership number.

```
getPromotions(memberNumber:version:devMode:)
```

Signature

```
public func getPromotions(  
    membershipNumber: String,  
    version: String = LoyaltyAPIVersion.defaultVersion,  
    devMode: Bool = false) async throws -> PromotionModel
```


Input Parameters

Parameter	Types	Description	Required or Optional
devMode	Boolean	Indicates whether the request is run in developer mode ((true)) or not ((false))..	Optional
memberId	String	Identifier of the member whose promotions are being retrieved.	Required
membershipNumber	String	The membership number of the member whose promotions are being retrieved.	Required
version	String	Version of the API. The request body is available in version 55.0 and later.	Optional

Sample Output

```
{
  "message": null,
  "outputParameters": {
    "outputParameters": {
      "results": [
        {
          "loyaltyPromotionType": "STANDARD",
          "maximumPromotionRewardValue": 0,
          "totalPromotionRewardPointsVal": 0,
          "loyaltyProgramCurrency": "01c4x000000L1bmAAC",
          "memberEligibilityCategory": "Eligible",
          "promotionEnrollmentRqr": true,
          "fulfillmentAction": "CREDIT_POINTS",
          "promotionName": "Anniversary promotion",
          "promotionId": "0c84x000000CoNhAAK",
          "startDate": "2020-01-01",
          "endDate": "2022-12-31",
          "description": "Double point promotion on member purchases on their anniversary",
          "imageName": "offer2"
        },
        {
          "loyaltyPromotionType": "CUMULATIVE",
          "maximumPromotionRewardValue": 0,
          "totalPromotionRewardPointsVal": 200,
          "loyaltyProgramCurrency": "01c4x000000L1bmAAC",
          "memberEligibilityCategory": "Eligible",
          "promotionEnrollmentRqr": false,
          "fulfillmentAction": "CREDIT_POINTS",
          "promotionName": "Referral Promotions",
          "promotionId": "0c84x000000CoNiAAK",
          "startDate": "2021-01-02",

```

```

        "description": "Additional points for member referral",
        "imageName": "promotion1"
    },
    {
        "promEnrollmentStartDate": "2022-11-01",
        "memberEligibilityCategory": "EligibleButNotEnrolled",
        "promotionEnrollmentEndDate": "2022-11-30",
        "promotionEnrollmentRqr": true,
        "promotionName": "New Year Promotion Premium",
        "promotionId": "0c84x000000CtzRAAS",
        "startDate": "2022-11-01",
        "description": "New Year Promotion Premium",
        "imageName": "meal1"
    },
    {
        "promEnrollmentStartDate": "2022-11-01",
        "memberEligibilityCategory": "Eligible",
        "promotionEnrollmentEndDate": "2022-11-30",
        "promotionEnrollmentRqr": true,
        "promotionName": "Christmas Promotion",
        "promotionId": "0c84x000020CtzWAAS",
        "startDate": "2022-12-01",
        "description": "Christmas Promotion - Happy holidays, enjoy a joyful
holiday season with this promotion",
        "imageName": "promotion3"
    },
    {
        "loyaltyPromotionType": "STANDARD",
        "maximumPromotionRewardValue": 0,
        "totalPromotionRewardPointsVal": 0,
        "loyaltyProgramCurrency": "01c4x000000L1bmAAC",
        "memberEligibilityCategory": "Eligible",
        "promotionEnrollmentRqr": true,
        "fulfillmentAction": "CREDIT_POINTS",
        "promotionName": "Anniversary promotion",
        "promotionId": "0c84x004000CoNhAAK",
        "startDate": "2020-01-01",
        "description": "Double point promotion on member purchases on their
anniversary",
        "imageName": "offers"
    },
    {
        "loyaltyPromotionType": "CUMULATIVE",
        "maximumPromotionRewardValue": 0,
        "totalPromotionRewardPointsVal": 200,
        "loyaltyProgramCurrency": "01c4x000000L1bmAAC",
        "memberEligibilityCategory": "Eligible",
        "promotionEnrollmentRqr": false,
        "fulfillmentAction": "CREDIT_POINTS",
        "promotionName": "Referral Promotions",
        "promotionId": "0c84x000006CoNiAAK",
        "startDate": "2021-01-02",
        "description": "Additional points for member referral",
        "imageName": "shoes1"
    }

```

```

    },
    {
      "promEnrollmentStartDate": "2022-11-01",
      "memberEligibilityCategory": "EligibleButNotEnrolled",
      "promotionEnrollmentEndDate": "2022-11-30",
      "promotionEnrollmentRqr": true,
      "promotionName": "Thanksgiving Promotion Premium",
      "promotionId": "0c84x070000CtzRAAS",
      "startDate": "2022-11-01",
      "description": "Thanksgiving Promotion Premium",
      "imageName": "jean1"
    },
    {
      "promEnrollmentStartDate": "2022-11-01",
      "memberEligibilityCategory": "Ineligible",
      "promotionEnrollmentEndDate": "2022-11-30",
      "promotionEnrollmentRqr": true,
      "promotionName": "New Year Promotion Premium",
      "promotionId": "0c84x000000CtzWAAS",
      "startDate": "2022-11-01",
      "description": "New Year Promotion Premium - Ineligible",
      "imageName": "coffeel"
    }
  ]
},
{
  "simulationDetails": {},
  "status": true
}

```

Output Parameters

Parameter	Type	Description
description	String	The description of the promotion.
endDate	String	The end date of the promotion.
fulfillmentAction	String	Specifies the fulfillment action of the related promotion. Possible values are: Credit_Points and Issue_Voucher
imageName	String	The name for the promotion image.
loyaltyProgramCurrency	String	The loyalty program currency associated with the promotion.
loyaltyPromotionType	String	The type of promotion. Possible values are: - Cumulative - Standard - Joint

Parameter	Type	Description
maximumPromotionRewardValue	Double	The value that can be accumulated for the promotion.
memberEligibilityCategory	String	Specifies whether the loyalty program member is eligible for the promotion. Possible values are: • Eligible • EligibleButNotEnrolled • Ineligible
message	String	The request response message.
promEnrollmentStartDate	String	The date when the enrollment in the related promotion starts.
promotionEnrollmentEndDate	String	The date when the enrollment in the related promotion ends.
promotionEnrollmentRqr	Boolean	Indicates whether the related promotion requires member enrollment.
promotionId	String	The ID related to the promotion.
promotionImageURL	String	The URL for the promotion image. In the promotion record, enter an image URL and ensure that the webpage that hosts the image is publicly accessible.
promotionName	String	The name of the promotion.
startDate	String	The start date of the related promotion.
status	Boolean	Indicates whether the member is eligible to enroll for the promotion.
totalPromotionRewardPointsVal	Double	The total points credited to the loyalty program member for the related promotion.

Member Benefits

Retrieve a list of benefits for a member, while asynchronously requesting data from the Member Benefits API.

Resource

```
getMemberBenefits (for:version:devMode:)
```

Signature

```
public func getMemberBenefits(
    for memberId: String,
    version: String = LoyaltyAPIVersion.defaultVersion,
    devMode: Bool = false) async throws -> [BenefitModel]
```

Input Parameters

Parameter	Type	Description	Required or Optional
devMode	Boolean	Indicates whether the request is run in developer mode (true) or not (false).	Optional
memberId	String	Identifier of the member whose benefits are being retrieved.	Required
version	String	Version of the API. The request body is available in version 55.0 and later.	Optional

Return Value

A BenefitModel instance of type:

```
public struct BenefitModel : Identifiable, Codable
```

Sample Output

```
{
  "memberBenefits": [
    {
      "benefitId": "0ji4x0000008REdAAM",
      "benefitName": "Extended Returns (7 Days)",
      "benefitTypeId": "0jh4x0000008REQAA2",
      "benefitTypeName": "Extended Returns",
      "createdRecordId": null,
      "createdRecordName": null,
      "endDate": null,
      "isActive": true,
      "memberBenefitStatus": null,
      "startDate": null
    },
    {
      "benefitId": "0ji4x0000008REjAAM",
      "benefitName": "Standard Support",
      "benefitTypeId": "0jh4x0000008REXAA2",
      "benefitTypeName": "Customer Support",
      "createdRecordId": null,
      "createdRecordName": null,
      "endDate": null,
    }
  ]
}
```

```

        "isActive": true,
        "memberBenefitStatus": null,
        "startDate": null
    },
    {
        "benefitId": "0ji4x0000008REeAAM",
        "benefitName": "Free Subscription",
        "benefitTypeId": "0jh4x0000008REXAA2",
        "benefitTypeName": "Free Subscription",
        "createdRecordId": null,
        "createdRecordName": null,
        "endDate": null,
        "isActive": true,
        "memberBenefitStatus": null,
        "startDate": null
    },
    {
        "benefitId": "0ji4x0000008REZAA2",
        "benefitName": "Free Shipping",
        "benefitTypeId": "0jh4x0000008REQAA2",
        "benefitTypeName": "Free Shipping",
        "createdRecordId": null,
        "createdRecordName": null,
        "endDate": null,
        "isActive": true,
        "memberBenefitStatus": null,
        "startDate": null
    },
    {
        "benefitId": "0ji4x0000008REiAAM",
        "benefitName": "Tier Exclusive Offer",
        "benefitTypeId": "0jh4x0000008REXAA2",
        "benefitTypeName": "Tier Exclusive Offer",
        "createdRecordId": null,
        "createdRecordName": null,
        "endDate": null,
        "isActive": true,
        "memberBenefitStatus": null,
        "startDate": null
    },
    {
        "benefitId": "0ji4x0000008REfAAM",
        "benefitName": "Complimentary Voouchers",
        "benefitTypeId": "0jh4x0000008REQAA2",
        "benefitTypeName": "Complimentary Voouchers",
        "createdRecordId": null,
        "createdRecordName": null,
        "endDate": null,
        "isActive": true,
        "memberBenefitStatus": null,
        "startDate": null
    }
]
}

```

Output Parameters

Parameter	Type	Description
benefitId	String	The ID of the benefit available for the member.
benefitName	String	The name of the benefit available for the member.
benefitTypeId	String	The ID of the associated benefit type record.
benefitTypeName	String	The name of the associated benefit type.
createdRecordId	String	The ID of the record created during benefit assignment.
createdRecordName	String	The display name of the record created during benefit assignment.
description	String	The description of the benefit available to a loyalty program member.
endDate	String	The date the benefit expires.
isActive	Boolean	Indicates whether the benefit is active (<code>true</code>) or not (<code>false</code>).
memberBenefitStatus	String	The status of the benefit available for the member.
startDate	String	The date from when the benefit applies to a member.

Member Profile

Get the profile details of a loyalty program member, while asynchronously requesting data from the Member Profile API.

Resource

```
getMemberProfile(for:version:devMode:)
```

Signature

```
public func getMemberProfile(  
    for memberId: String,  
    version: String = LoyaltyAPIVersion.defaultVersion,  
    devMode: Bool = false) async throws -> ProfileModel
```

Input Parameters

Parameter	Type	Description	Required or Optional
devMode	Boolean	Indicates whether the request is run in developer mode (true) or not (false).	Optional
memberId	String	Identifier of the member whose benefits are being retrieved.	Required
version	String	Version of the API. The request body is available in version 55.0 and later.	Optional

Return Value

A ProfileModel instance of type:

```
public struct ProfileModel : Codable
```

Sample Output

```
{
  "additionalLoyaltyProgramMemberFields": {
    "City__c": null,
    "Age__c": null,
    "Anniversary__c": null,
    "ReferredMemberPromotion__c": null,
    "DateOfBirth__c": null,
    "Hobbies__c": null,
    "Gender__c": null,
    "SeedData__c": false,
    "Income__c": null,
    "State__c": null,
    "B2CRetail_Most_Recent_Survey_Email_Date__c": null
  },
  "associatedAccount": null,
  "associatedContact": {
    "contactId": "0034x00001EhiXSAAZ",
    "email": "ab@gmail.com",
    "firstName": "Aman",
    "lastName": "Bindal"
  },
  "canReceivePartnerPromotions": true,
  "canReceivePromotions": true,
  "enrollmentChannel": "Email",
  "enrollmentDate": "2022-01-01",
  "groupCreatedByMember": null,
  "groupName": null,
  "lastActivityDate": null,
  "loyaltyProgramMemberId": "01M4x000000LECAEA4",
  "loyaltyProgramName": "NTO Insider",
```



```

"memberCurrencies": [
  {
    "additionalLoyaltyMemberCurrencyFields": {},
    "escrowPointsBalance": 0.0,
    "expirablePoints": 0.0,
    "lastAccrualProcessedDate": null,
    "lastEscrowProcessedDate": null,
    "lastExpirationProcessRunDate": null,
    "lastPointsAggregationDate": null,
    "lastPointsResetDate": null,
    "loyaltyMemberCurrencyName": "Reward Points",
    "loyaltyProgramCurrencyId": "01c4x000000L1bmAAC",
    "loyaltyProgramCurrencyName": null,
    "memberCurrencyId": "01z4x000000LC3nAAG",
    "nextQualifyingPointsResetDate": null,
    "pointsBalance": 17850,
    "qualifyingPointsBalanceBeforeReset": 0.0,
    "totalEscrowPointsAccrued": 0.0,
    "totalEscrowRolloverPoints": 0.0,
    "totalPointsAccrued": 0.0,
    "totalPointsExpired": 0.0,
    "totalPointsRedeemed": 0.0
  },
  {
    "additionalLoyaltyMemberCurrencyFields": {},
    "escrowPointsBalance": 0.0,
    "expirablePoints": 0.0,
    "lastAccrualProcessedDate": null,
    "lastEscrowProcessedDate": null,
    "lastExpirationProcessRunDate": null,
    "lastPointsAggregationDate": null,
    "lastPointsResetDate": null,
    "loyaltyMemberCurrencyName": "Tier Points",
    "loyaltyProgramCurrencyId": "01c4x000000L1bnAAC",
    "loyaltyProgramCurrencyName": null,
    "memberCurrencyId": "01z4x000000LC3oAAG",
    "nextQualifyingPointsResetDate": "2023-01-01",
    "pointsBalance": 4000,
    "qualifyingPointsBalanceBeforeReset": 0.0,
    "totalEscrowPointsAccrued": 0.0,
    "totalEscrowRolloverPoints": 0.0,
    "totalPointsAccrued": 0.0,
    "totalPointsExpired": 0.0,
    "totalPointsRedeemed": 0.0
  }
],
"memberStatus": "Active",
"memberTiers": [
  {
    "additionalLoyaltyMemberTierFields": {},
    "areTierBenefitsAssigned": false,
    "loyaltyMemberTierId": "01y4x000000btVzAAI",
    "loyaltyMemberTierName": "Silver",
    "tierChangeReason": null,

```

```

        "tierChangeReasonType": null,
        "tierEffectiveDate": "2022-01-01",
        "tierExpirationDate": "2023-08-31",
        "tierGroupId": "01t4x000000L1bXAAS",
        "tierGroupName": null,
        "tierId": "01g4x000000L1bmAAC",
        "tierSequenceNumber": 10
    },
    ],
    "memberType": "Individual",
    "membershipEndDate": "2023-01-01",
    "membershipLastRenewalDate": null,
    "membershipNumber": "Member1",
    "referredBy": null,
    "relatedCorporateMembershipNumber": null,
    "transactionJournalStatementFrequency": "Monthly",
    "transactionJournalStatementLastGeneratedDate": null,
    "transactionJournalStatementMethod": "Mail"
}

```

Output Parameters

Property Name	Type	Description
additionalLoyaltyProgramMemberFields	Map<String, Object>	The details of the custom fields added in the loyalty program member object.
associatedAccount	Codable	The details of the account record that is associated with the loyalty program member.
associatedContact	Codable	The details of the contact record that is associated with the loyalty program member.
canReceivePartnerPromotions	Boolean	Indicates whether the loyalty program member can receive partner promotions.
canReceivePromotions	Boolean	Indicates whether the loyalty program member can receive promotions (<code>true</code>) or not (<code>false</code>).
enrollmentChannel	String	The channel used by the member to enroll in the loyalty program.
enrollmentDate	String	The date on which the member enrolled in the loyalty program.
groupCreatedByMember	String	The membership number of the loyalty program member who created the group.
groupName	String	The name of the loyalty program group to which the member belongs.
lastActivityDate	String	The date on which the member performed their last activity.
loyaltyProgramMemberId	String	The ID of the loyalty program member.
loyaltyProgramName	String	The name of the loyalty program.

Property Name	Type	Description
memberCurrencies	Codable	The currencies associated with the loyalty program member.
memberStatus	String	The status of the loyalty program member.
memberTiers	Codable	The tiers associated with the loyalty program member.
memberType	String	The type of loyalty program member.
membershipEndDate	String	The date on which the membership ends.
membershipLastRenewalDate	String	The date on which the loyalty program member last renewed their membership.
membershipNumber	String	The membership number of the loyalty program member.
referredBy	String	The membership number of the loyalty program member who referred the member to the loyalty program.
relatedCorporateMembershipNumber	String	The membership number of the Corporate type loyalty program member associate with the member.
transactionJournalStatementFrequency	String	The frequency at which transaction journal statements must be delivered to the member. Possible values are: - Monthly - Quarterly
transactionJournalStatementLastGeneratedDate	String	The date on which the member's last transaction journal statement was generated.
transactionJournalStatementMethod	String	The method used to deliver transaction journal statements to the member.

Member Vouchers

Retrieve a list of vouchers issued to the member, while asynchronously requesting data from the Member Vouchers API.

Resource

```
getVouchers(membershipNumber:voucherStatus:pageNumber:productId:productCategoryId:productName:productCategoryName:sortBy:sortOrder:version:devMode:)
```

Signature

```
public func getVouchers(
    membershipNumber: String,
    voucherStatus: [VoucherStatus]? = nil,
    pageNumber: Int? = nil,
```

```

productId: [String]? = nil,
productCategoryId: [String]? = nil,
productName: [String]? = nil,
productCategoryName: [String]? = nil,
sortBy: SortBy? = nil,
sortOrder: SortOrder? = nil,
version: String = LoyaltyAPIVersion.defaultVersion,
devMode: Bool = false) async throws -> [VoucherModel]

```

Input Parameters

Parameter Name	Type	Description	Required or Optional
membershipNumber	String	Unique identifier of the member whose vouchers are being retrieved.	Required
devMode	Boolean	Indicates whether the request is run in developer mode (<code>true</code>) or not (<code>false</code>).	Optional
pageNumber	Integer	The page number to be returned. If you don't specify a value, the first page is returned. Each page contains 200 vouchers, sorted by their date of creation.	Optional
productCategoryId	String	The ID of a product category associated with the member vouchers to be retrieved. You can specify up to 20 product category IDs.	Optional
productCategoryName	String	The product category name associated with the vouchers to be retrieved. You can specify up to 20 product categories.	Optional
productId	String	The product ID associated with the vouchers to be retrieved. You can specify up to 20 product IDs.	Optional
productName	String	The product name associated with the vouchers to be retrieved. You can specify up to 20 product names.	Optional
voucherStatus	String	The status of the member vouchers to filter the API response. Possible values are: • Issued • Cancelled • Expired	Optional
sortBy	String	The voucher field for sorting the API response. The default value is <code>ExpirationDate</code> . Possible values are: • CreatedDate • EffectiveDate • ExpirationDate	Optional

Parameter Name	Type	Description	Required or Optional
sortOrder	String	The sort order of the API response. The default value is <code>Ascending</code> . Possible values are: - Ascending - Descending	Optional
version	String	Version of the API. The request body is available in version 55.0 and later.	Optional

Sample Output

```
{
  "voucherCount": 4,
  "vouchers": [
    {
      "description": "20% discount on children group party at selected event centers.",
      "discountPercent": 50,
      "effectiveDate": "2023-01-01",
      "expirationDate": "2023-04-01",
      "isVoucherDefinitionActive": true,
      "isVoucherPartiallyRedeemable": false,
      "promotionId": "0c8RO0000000ILdYAM",
      "promotionName": "Birthday Promotion",
      "redeemedValue": 0.0,
      "remainingValue": 0.0,
      "status": "Issued",
      "type": "DiscountPercentage",
      "voucherCode": "VOU10101011",
      "voucherDefinition": "Birthday Discount Voucher",
      "voucherId": "0kDRO00000000Hk2AI",
      "voucherImageUrl":
"https://thumbs.dreamstime.com/b/10-percent-discount-24471598.jpg",
      "voucherNumber": "00000001"
    },
    {
      "description": "20% discount on children group party at selected event centers.",
      "discountPercent": 40,
      "effectiveDate": "2023-01-01",
      "expirationDate": "2023-04-01",
      "isVoucherDefinitionActive": true,
      "isVoucherPartiallyRedeemable": false,
      "promotionId": "0c8RO0000000ILdYAM",
      "promotionName": "Birthday Promotion",
      "redeemedValue": 0.0,
      "remainingValue": 0.0,
      "status": "Redeemed",
      "type": "DiscountPercentage",
      "voucherCode": "VOU10101012",
      "voucherDefinition": "Birthday Discount Voucher",

```

```

        "voucherId": "0kDRO00000000Hp2AI",
        "voucherImageUrl":
"https://thumbs.dreamstime.com/b/10-percent-discount-24471598.jpg",
        "voucherNumber": "00000002"
    },
    {
        "description": "20% discount on children group party at selected event centers.",

        "discountPercent": 40,
        "effectiveDate": "2023-01-01",
        "expirationDate": "2023-06-01",
        "isVoucherDefinitionActive": true,
        "isVoucherPartiallyRedeemable": false,
        "promotionId": "0c8RO00000000ILdYAM",
        "promotionName": "Birthday Promotion",
        "redeemedValue": 0.0,
        "remainingValue": 0.0,
        "status": "Redeemed",
        "type": "DiscountPercentage",
        "voucherCode": "VOU10101012",
        "voucherDefinition": "Birthday Discount Voucher",
        "voucherId": "0kDRO00000000Hp2BI",
        "voucherImageUrl":
"https://thumbs.dreamstime.com/b/10-percent-discount-24471598.jpg",
        "voucherNumber": "00000003"
    },
    {
        "description": "20% discount on children group party at selected event centers.",

        "discountPercent": 40,
        "effectiveDate": "2023-01-01",
        "expirationDate": "2023-05-01",
        "isVoucherDefinitionActive": true,
        "isVoucherPartiallyRedeemable": false,
        "promotionId": "0c8RO00000000ILdYAH",
        "promotionName": "Birthday Promotion",
        "redeemedValue": 0.0,
        "remainingValue": 0.0,
        "status": "Expired",
        "type": "DiscountPercentage",
        "voucherCode": "VOU10101012",
        "voucherDefinition": "Birthday Discount Voucher",
        "voucherId": "0kD43872679RO00000000Hp2AI",
        "voucherImageUrl":
"https://thumbs.dreamstime.com/b/10-percent-discount-24471598.jpg",
        "voucherNumber": "00000004"
    }
]
}

```

Output Parameters

Property Name	Type	Description
attributesUrl	String	The URL of the attribute associated with the voucher.
currencyIsoCode	String	The currency ISO code in which the member has redeemed the voucher.
description	String	The description of the voucher issued to the member.
discountPercent	Integer	The discount percentage that's offered to the member by the voucher.
effectiveDate	String	The date the member can redeem the voucher.
expirationDate	String	The date the voucher expires.
faceValue	Double	The value that the member can redeem using the voucher.
id	String	The ID of the voucher record.
isVoucherDefinitionActive	Boolean	Indicates whether the voucher definition associated with the voucher is active (<code>true</code>) or not (<code>false</code>).
isVoucherPartiallyRedeemable	Boolean	Indicates whether the voucher can be redeemed across multiple transactions (<code>true</code>) or only one transaction (<code>false</code>).
partnerAccount	String	The name of the partner associated with the voucher's voucher definition.
product	String	The name of the product associated with the voucher definition.
productCategory	String	The product category associated with the voucher's voucher definition.
productCategoryId	String	The ID of product category associated with the voucher's voucher definition.
productId	String	The ID of the product associated with the voucher definition.
promotionId	String	The ID of the promotion for which the voucher is issued.
promotionName	String	The name of the promotion for which the voucher is issued.
redeemedValue	Double	The value that the member has already redeemed from a partially redeemable voucher.
remainingValue	Double	The remaining value of the voucher after partial redemption.
status	String	The status of the voucher.
type	String	The type of voucher issued to the member.
useDate	String	The date the voucher was redeemed.
voucherCode	String	The code of the voucher.

Property Name	Type	Description
<code>voucherDefinition</code>	String	The name of the voucher definition used to issue the voucher to the member.
<code>voucherImageUrl</code>	String	The URL of the image associated with the voucher definition. In the promotion record, enter an image URL and ensure that the webpage that hosts the image is publicly accessible.
<code>voucherNumber</code>	String	The unique number of the voucher.

Receipt Status Updation

Update the status of the receipt, reward points, and create a transaction journal for each line item in the receipt.

Resource

```
services/apexrest/receipt-status-updation
```

HTTP methods

PUT

Signature

```
curl --location --request PUT
'https://yourInstance.salesforce.com/services/apexrest/receipt-status-updation/' \
--header 'Content-Type: text/plain' \
--header 'Authorization: Bearer
00DB000000FX0aR!ARQACyXQGgMxjFvjslhnPCDJEZyPavtzxM26XvjDU.gr56iTkGp_EQCZ0yyilQZzILDqOlmy3KpJqhKA0c56o4jqx6CxU'
\
--header 'Cookie: BrowserId=MZ4W4WebEe6bX0_gtfh0Tw; CookieConsentPolicy=0:1;
LSKey-c$CookieConsentPolicy=0:1' \
--data '{
  "receiptId":"a0N1Q00000d0Rv4UAE",
  "status":"In Progress",
  "comments":""
}'
```

Input Parameters

Parameter	Type	Description	Required or Optional
<code>comments</code>	String	The comments related to the receipt status change.	Required
<code>receiptId</code>	String	The ID of the receipt whose status is to be updated.	Required

Parameter	Type	Description	Required or Optional
status	String	The status that's assigned to the receipt after the update. Valid status values are: • Approved • Cancelled • Draft • In Progress • Manual Review • Pending • Processed • Rejected	Required

Sample Output

```
{
  "storeName" : "alpine nutrition",
  "storeAddress" : "625 Market Street,\n6th floor,\nSan Francisco,\nCA, 94105",
  "receiptSFDCId" : "a0PHp00000tstqeMAA",
  "receiptNumber" : "#0529-0002",
  "receiptDate" : "2020-01-10",
  "memberShipNumber" : "124345696",
  "lineItem" : [ {
    "quantity" : "5",
    "productName" : "Alpine-Bran Cereal",
    "price" : "$7.99",
    "lineItemPrice" : "$39.95",
    "isEligible" : false
  }, {
    "quantity" : "2",
    "productName" : "Alpine-Oat Cereal",
    "price" : "$7.99",
    "lineItemPrice" : "$15.98",
    "isEligible" : false
  } ],
  "confidenceStatus" : "Partial"
}
```

Output Parameter

Parameter	Type	Description
confidenceStatus	String	The receipt analysis status based on the confidence value defined in the app. Possible values are: Success, Failure, or Partial.

Parameter	Type	Description
<code>isEligible</code>	Boolean	Indicates whether the item is eligible for accrual of loyalty points (<code>true</code>) or not (<code>false</code>). This value is displayed only if the admin has opted to display ineligible items in the receipt.
<code>lineItem</code>	String	The details of the receipt line item.
<code>lineItemPrice</code>	String	The total price of the receipt line item.
<code>memberShipNumber</code>	String	The unique identifier of the member.
<code>price</code>	Currency	The unit price of the receipt line item.
<code>productName</code>	String	The name of the product purchased.
<code>quantity</code>	Integer	The quantity of the product purchased.
<code>receiptDate</code>	dateTime	The date when the receipt is generated.
<code>receiptNumber</code>	String	The receipt number.
<code>receiptSFDCId</code>	Reference	The unique ID of the receipt.
<code>storeAddress</code>	String	The store address.
<code>storeName</code>	String	The store name.

SEE ALSO:

[Salesforce Help: Set Up Amazon Web Services](#)

[Salesforce Help: Set Up Salesforce Org With Loyalty Management Receipt Scanner Kit Unlocked Package](#)

[Salesforce Help: Set Up Amazon Web Services](#)

[Salesforce Help: Set Up Salesforce Org With Loyalty Management Receipt Scanner Kit Unlocked Package](#)

Receipt Upload

Upload and scan a receipt with a receipt scanning app that uses optical character recognition to read and interpret receipts. For the unlocked package, this API uploads the image into the Amazon S3 bucket.

Resource

```
services/apexrest/upload-receipt
```

Rest Method Type

PUT

Signature

```
curl --location --request PUT
'https://yourInstance.salesforce.com/services/apexrest/upload-receipt/?membershipnumber=124345696'
\
--header 'Content-Type: image/jpeg' \
--header 'Authorization: Bearer
00DB000000FX0aR!ARQAQCyXQGgMxxjFvjslhnPCDJEZyPavtzxM26XvjDU.gr56iTgkGp_EQCZ0yyilQZzILDqOlmy3KpJqhKAOc56o4jqx6CxU'
\
--header 'Cookie: BrowserId=MZ4W4WebEe6bX0_gtfh0Tw; CookieConsentPolicy=0:1;
LSKey-c$CookieConsentPolicy=0:1' \
--data '@1jFvSuLyR/Alpine receipt (4) (1) (2).jpg'
```

Input Parameters

Parameter	Type	Description	Required or Optional
membershipnumber	String	The unique identifier of the member.	Required

Sample Output

```
{
  "status" : "Success",
  "message" : "b9746ab0-fea3-b9ef-dc9c-ea5b905a822e.jpeg",
  "errorCode" : ""
}
```

Output Parameter

Parameter	Type	Description
errorCode	Integer	The error code if the transaction fails for any reason.
message	String	The error message if the transaction fails for any reason.
status	String	The status of the receipt upload.

SEE ALSO:

[Salesforce Help: Set Up Amazon Web Services](#)

[Salesforce Help: Set Up Salesforce Org With Loyalty Management Receipt Scanner Kit Unlocked Package](#)

[Salesforce Help: Set Up Amazon Web Services](#)

[Salesforce Help: Set Up Salesforce Org With Loyalty Management Receipt Scanner Kit Unlocked Package](#)

Transaction History

Retrieve member's transaction history, while asynchronously requesting data from the Transaction History API.

Resource

```
getTransactions(for:pageNumber:journalTypeName:journalSubTypeName:periodStartDate:periodEndDate:version:devMode:)
```

Signature

```
public func getTransactions(
    for membershipNumber: String,
    pageNumber: Int? = nil,
    journalTypeName: String? = nil,
    journalSubTypeName: String? = nil,
    periodStartDate: String? = nil,
    periodEndDate: String? = nil,
    version: String = LoyaltyAPIVersion.defaultVersion,
    devMode: Bool = false) async throws -> [TransactionJournal]
```

Input Parameters

Parameter Name	Type	Description	Required or Optional
devMode	Boolean	Indicates whether the request is run in developer mode (<code>true</code>) or not (<code>false</code>).	Optional
journalSubTypeName	String	The subtype of the transaction journal record.	Optional
journalTypeName	String	The type of the transaction journal record. Possible values are: • <code>Accrual</code> • <code>Redemption</code>	Optional
membershipNumber	String	Unique identifier of the member whose transactions are being retrieved.	Required
pageNumber	Integer	The page number to be returned. If you don't specify a value, the first page is returned. Each page contains 200 transactions, sorted by their date of creation.	Optional
periodStartDate	String	Retrieve transaction journals starting on this date.	Optional
periodEndDate	String	Retrieve transaction journals until this date.	Optional
version	String	Version of the API. The request body is available in version 55.0 and later.	Optional

Sample Output

```
{
  "message": null,
  "status": true,
  "transactionJournalCount": 4,
  "transactionJournals": [
```

```

    {
      "activityDate": "2023-04-08T04:59:40.000Z",
      "additionalTransactionJournalAttributes": [],
      "journalTypeName": "Manual Points Adjustment",
      "pointsChange": [
        {
          "changeInPoints": 100.0,
          "loyaltyMemberCurrency": "Reward Points"
        }
      ],
      "transactionJournalId": "01VRO00000002og2AA",
      "transactionJournalNumber": "00000035"
    },
    {
      "activityDate": "2023-04-08T04:58:07.000Z",
      "additionalTransactionJournalAttributes": [],
      "journalTypeName": "Manual Points Adjustment",
      "pointsChange": [
        {
          "changeInPoints": 300.0,
          "loyaltyMemberCurrency": "Reward Points"
        }
      ],
      "transactionJournalId": "01VRO00000002ob2AA",
      "transactionJournalNumber": "00000034"
    },
    {
      "activityDate": "2023-04-08T04:56:38.000Z",
      "additionalTransactionJournalAttributes": [],
      "journalTypeName": "Manual Points Adjustment",
      "pointsChange": [
        {
          "changeInPoints": -100.0,
          "loyaltyMemberCurrency": "Reward Points"
        }
      ],
      "transactionJournalId": "01VRO00000002oW2AQ",
      "transactionJournalNumber": "00000033"
    },
    {
      "activityDate": "2022-03-08T04:53:19.000Z",
      "additionalTransactionJournalAttributes": [],
      "journalTypeName": "Manual Points Adjustment",
      "pointsChange": [
        {
          "changeInPoints": 100.0,
          "loyaltyMemberCurrency": "Reward Points"
        }
      ],
      "transactionJournalId": "01VRO00000002oR2AQ",
      "transactionJournalNumber": "00000032"
    }
  ]
}

```

Output Parameters

Parameter Name	Type	Description
externalTransactionNumber	String	The transaction journal record ID from an external source system.
message	String	The message explaining why the request failed.
status	Boolean	Indicates whether the request succeeded (<code>true</code>) or failed (<code>false</code>).
transactionJournalCount	Integer	The total number of transaction journals that match the input criteria.
transactionJournals	Map<String, Object>[]	The details of the transaction journals.

Opt Out from a Promotion

Opt out a loyalty program member from an available promotion.

Resource to Opt Out of a Promotion by Using the Promotion ID

```
unenroll(promotionId:for:version:devMode:)
```

Signature

```
public final func unenroll(  
    promotionId: String,  
    for membershipNumber: String,  
    version: String = LoyaltyAPIVersion.defaultVersion,  
    devMode: Bool = false) async throws
```

Resource to Opt Out of a Promotion by Using the Promotion Name

```
unenroll(promotionName:for:version:devMode:)
```

Signature

```
public final func unenroll(  
    promotionName: String,  
    for membershipNumber: String,  
    version: String = LoyaltyAPIVersion.defaultVersion,  
    devMode: Bool = false) async throws
```

Input Parameters

Parameter	Type	Description	Required or Optional
devMode	Boolean	Indicates whether the request is run in developer mode (<code>true</code>) or not (<code>false</code>).	Optional
membershipNumber	String	Unique identifier of the member who wants to opt out of a promotion	Required
promotionId	String	ID of the promotion that the member wants to opt out of.	Required
promotionName	String	Name of the promotion that the member wants to opt out of.	Required
version	String	Version of the API. The request body is available in version 55.0 and later.	Optional

Sample Output

```
{
  "message" : null,
  "outputParameters" : {
    "outputParameters" : {
      "results" : [ {
        "LoyaltyProgramMbrPromotionId" : "1lpT100000000h4IAA"
      } ]
    }
  },
  "simulationDetails" : { },
  "status" : true
}
```

Output Parameter

Parameter	Type	Description
LoyaltyProgramMbrPromotionId	String	The ID of the promotion that the member opts out of.

Loyalty Management Mobile SDK for Android Development

Loyalty Management Mobile SDK for Android delivers source code, build scripts, and native sample app to develop mobile apps on Android.

To upgrade the Loyalty Management Mobile SDK for Android, see [Update your tools with the SDK Manager](#).

EDITIONS

Available in: all editions that have Loyalty Management enabled

[Clone and Import Loyalty Management Mobile SDK for Android](#)

Clone the Loyalty Management Mobile SDK GitHub repository for Android to create a local copy of the repository on your development machine. The repository contains the code for the Loyalty Management Mobile SDK for Android and the MyNTORewards sample app. To create the Android application, Loyalty Management Mobile SDK for Android uses Android Studio to build the user interfaces in Android apps while interacting with Salesforce.

[Loyalty Management Mobile SDK for Android Classes](#)

The Android classes organize the Loyalty Management Mobile SDK's functionalities and data structures. When developing mobile apps using the SDK, these classes provide components, services, or features with which you can interact.

[Loyalty Management Mobile SDK API References for Android](#)

The Loyalty Management Mobile SDK for Android interacts with Salesforce Loyalty Management APIs and retrieves member benefits, transactions, profiles, promotions, vouchers, enrolls an individual member, opts a member out of a promotion.

Clone and Import Loyalty Management Mobile SDK for Android

Clone the Loyalty Management Mobile SDK GitHub repository for Android to create a local copy of the repository on your development machine. The repository contains the code for the Loyalty Management Mobile SDK for Android and the MyNTORewards sample app. To create the Android application, Loyalty Management Mobile SDK for Android uses Android Studio to build the user interfaces in Android apps while interacting with Salesforce.

EDITIONS

Available in: all editions that have Loyalty Management enabled

[Clone Mobile SDK for Android GitHub Repository](#)

Clone the GitHub repository to copy the code base and revision history of the repository to your local computer.

[Import SDK into Android Studio](#)

After you clone the mobile SDK for Android repository, import Loyalty Management Mobile SDK in Android Studio to automatically download and manage external dependencies.

Clone Mobile SDK for Android GitHub Repository

Clone the GitHub repository to copy the code base and revision history of the repository to your local computer.

1. Go to the [Loyalty Management Mobile SDK Android repository](#).
2. To copy the repository URL to your clipboard, click **Code**, and then click .
3. Open Terminal, and go to the directory where you want to clone the repository.
4. Enter `git clone https://github.com/salesforce-misc/LoyaltyMobileSDK-Android`
5. Press Enter and wait for the repository to clone.

You can now set up your own development environment.

EDITIONS

Available in: all editions that have Loyalty Management enabled

Import SDK into Android Studio

After you clone the mobile SDK for Android repository, import Loyalty Management Mobile SDK in Android Studio to automatically download and manage external dependencies.

1. Unzip the `LoyaltyMobileSDK-Android` package that you downloaded.
2. Open Android Studio.
3. Go to **File > New > Import Project**, and then select the `LoyaltyMobileSDK-Android` folder.
4. Click **Open**.
5. Open the `SampleApps/MyNTORewards/build.gradle` file and under Dependencies block, to the first line of Dependencies, add:


```
{implementation fileTree(dir: 'libs', include: ['*.jar']) implementation project(path: ':Sources') }
```
6. Open the `SampleApps/MyNTORewards/connectedapp.properties.sample` file and modify the properties of the connected app as per your org.
7. Rename the `connectedapp.properties.sample` file to `connectedapp.properties`.
8. Click **Sync Now**.
Android Studio automatically downloads and manages external dependencies for the Mobile SDK and the Sample App.
9. Open the SDK files and ensure that the files include the appropriate SDK packages.

 **Note:** If the sync fails, disconnect the VPN to ensure there's no blocker when downloading the dependencies, and then resync.

10. Add the code snippet to add the appropriate SDK package:

```
import com.salesforce.loyalty.mobile.sources.forceUtils.* import
com.salesforce.loyalty.mobile.sources.loyaltyAPI.* import
com.salesforce.loyalty.mobile.sources.loyaltyModels.* import
com.salesforce.loyalty.mobile.sources.loyaltyExtensions.*
```

You can now use the classes and structures in the `LoyaltyMobileSDK-Android` module and API references to manage the features in your application.

Loyalty Management Mobile SDK for Android Classes

The Android classes organize the Loyalty Management Mobile SDK's functionalities and data structures. When developing mobile apps using the SDK, these classes provide components, services, or features with which you can interact.

Logger

The `Logger` class provides a standardized way to save debug, verbose, info, or error log messages.

LoyaltyClient

Create authentication network requests by using the `LoyaltyClient` class. `LoyaltyClient` uses `ForceAuthenticator`, which is an interface that defines the methods to handle access tokens in the Salesforce API. `LoyaltyClient` creates a retrofit instance to invoke Force APIs.

EDITIONS

Available in: all editions that have Loyalty Management enabled

EDITIONS

Available in: all editions that have Loyalty Management enabled

[LoyaltyAPIManager for Android](#)

Manage requests related to loyalty program and interactions between the application and the APIs. The `LoyaltyAPIManager` class manages requests related to loyalty programs using the `Force` API and manages authentication by using the instance of `ForceAuthenticator` that was injected at the time of initialization. Use this class to manage asynchronous requests by using Retrofit and Kotlin coroutines.

[LoyaltyAPIInterface](#)

The `LoyaltyApiInterface` class defines the interface for creating, formatting, and sending REST requests to the Salesforce service. This class stores REST API call definitions for all Loyalty API.

Logger

The `Logger` class provides a standardized way to save debug, verbose, info, or error log messages.

Use this signature to define the `Logger` class.

```
object Logger
```

Log a Debug Message

Use the `message` and `tag` parameters to log an informational message by using this code.

```
final Unit d(String tag, String message)
```

Log an Error Message

Use the `message` parameter to log an error message by using this code.

```
final Unit e(String tag, String message, Throwable throwable)
```

Log an Information Message

Use the `message` parameter to add an information log by using this code.

```
final Unit i(String tag, String message)
```

Add a Verbose Log

Use the `message` parameter to log a verbose message by using this code.

```
final Unit v(String tag, String message)
```

LoyaltyClient

Create authentication network requests by using the `LoyaltyClient` class. `LoyaltyClient` uses `ForceAuthenticator`, which is an interface that defines the methods to handle access tokens in the Salesforce API. `LoyaltyClient` creates a retrofit instance to invoke Force APIs.

Use this signature to define the `LoyaltyClient` class.

```
public final class LoyaltyClient  
extends NetworkClient
```

EDITIONS

Available in: all editions that have Loyalty Management enabled

getNetworkClient Method

Gives an implementation of the `retrofit` interface using an instance of `retrofit`.

```
LoyaltyApiInterface getNetworkClient()
```

LoyaltyAPIManager for Android

Manage requests related to loyalty program and interactions between the application and the APIs. The `LoyaltyAPIManager` class manages requests related to loyalty programs using the `Force` API and manages authentication by using the instance of `ForceAuthenticator` that was injected at the time of initialization. Use this class to manage asynchronous requests by using Retrofit and Kotlin coroutines.

Use this signature to define the `LoyaltyAPIManager` class.

```
public final class LoyaltyAPIManager
```

Method Summary

Method	Description
<code>enrollInPromotions</code>	Enroll a loyalty member to a promotion, while asynchronously sending data to the Enroll for Promotions API.
<code>getEligiblePromotions</code>	Get promotions that a member is eligible to enroll for.
<code>getMemberBenefits</code>	Retrieve the benefits associated with a loyalty program member for the specified API version, while asynchronously requesting data from the Member Benefits API.
<code>getMemberProfile</code>	Get the profile details of a loyalty program member by using this code.
<code>getTransactions</code>	Get transaction journals of a loyalty member. Obtain either accrual or redemption transaction journal records.
<code>getVouchers</code>	Retrieve the vouchers issued to a loyalty member by using this code.
<code>postEnrollment</code>	Enroll a member to a loyalty program, while asynchronously sending data to the Individual Member Enrollments API.
<code>unEnrollPromotion</code>	Opt out a loyalty member from a promotion.

enrollInPromotions Method

Enroll a loyalty member to a promotion, while asynchronously sending data to the Enroll for Promotions API. Use this code to enroll a member.

```
final Result<EnrollPromotionsResponse> enrollInPromotions(String membershipNumber, String promotionName)
```

Parameter	Type	Description	Required or Optional
memberShipNumber	String	The membership number of the loyalty program member.	Required
promotionName	String	The promotion name the member is being enrolled to	Required

getEligiblePromotions Method

Get promotions that a member is eligible to enroll for. Use this code to get eligible promotions.

```
final Result<PromotionsResponse> getEligiblePromotions(String membershipNumber, String memberId)
```

Parameter	Type	Description	Required or Optional
memberId	Boolean	The identifier of the member whose eligible promotions are being retrieved.	Optional
memberShipNumber	String	The membership number of the loyalty program member.	Required

getMemberBenefits Method

Retrieve the benefits associated with a loyalty program member for the specified API version, while asynchronously requesting data from the Member Benefits API. Use this code to get member benefits.

```
final Result<MemberBenefitsResponse> getMemberBenefits(String memberId, String membershipNumber)
```

Parameter	Type	Description	Required or Optional
memberId	Boolean	Identifier of the member whose benefits are being retrieved.	Optional
memberShipNumber	String	The membership number of the loyalty program member.	Required

getMemberProfile Method

Get the profile details of a loyalty program member by using this code.

```
final Result<MemberProfileResponse> getMemberProfile(String memberId, String memberShipNumber, String programCurrencyName)
```

Parameter	Type	Description	Required or Optional
memberId	Boolean	Identifier of the member whose profile is being retrieved.	Optional

Parameter	Type	Description	Required or Optional
membershipNumber	String	The membership number of the loyalty program member.	Required
programCurrencyName	String	The name of the loyalty program currency associated with the member.	Optional

getTransactions Method

Get transaction journals of a loyalty member. Obtain either accrual or redemption transaction journal records. Use this code to get transactions.

```
final Result<MemberBenefitsResponse> getMemberBenefits(String memberId, String membershipNumber)
```

Parameter	Type	Description	Required or Optional
journalSubTypeName	String	The subtype of the transaction journal record.	Optional
journalTypeName	String	The type of the transaction journal record. Possible values are: • Accrual • Redemption	Optional
pageNumber	Integer	The page number to be returned. If you don't specify a value, the first page is returned. Each page contains 200 transactions, sorted by their date of creation.	Optional
membershipNumber	String	The membership number of the loyalty program member.	Required
periodStartDate	String	Retrieve transaction journals starting on this date.	Optional
periodEndDate	String	Retrieve transaction journals until this date.	Optional

getVouchers Method

Retrieve the vouchers issued to a loyalty member by using this code.

```
final Result<VoucherResult> getVouchers(String membershipNumber, Array<String> voucherStatus, Integer pageNumber, Array<String> productId, Array<String> productCategoryId, Array<String> productName, Array<String> productCategoryName)
```

This table shows the parameters in the `getVouchers` method.

Parameter	Description
membershipNumber	Membership number of the member whose vouchers are being retrieved.
voucherStatus	The status of a voucher.
pageNumber	Number of the page you want returned. If you don't specify a value, the first page is returned. By default, each page contains 200 vouchers, which are sorted based on the voucher creation date.
productId	The ID of the product related to the vouchers that you want to retrieve. You can specify the ID of up to 20 products.
productCategoryId	The ID of the product categories related to the vouchers that you want to retrieve. You can specify the ID of up to 20 product categories.
productName	The product name related to the voucher.
productCategoryName	The name of the product categories related to the vouchers that you want to retrieve. You can specify the ID of up to 20 product category names.

postEnrollment Method

Enroll a member to a loyalty program, while asynchronously sending data to the Individual Member Enrollments API. Use this code to enroll a member.

```
final Result<EnrollmentResponse> postEnrollment(String firstName, String lastName, String
    email, Map<String, Object> additionalContactAttributes, Boolean emailNotification,
    MemberStatus memberStatus, Boolean createTransactionJournals,
    TransactionalJournalStatementFrequency transactionalJournalStatementFrequency,
    TransactionalJournalStatementMethod transactionalJournalStatementMethod, EnrollmentChannel
    enrollmentChannel, Boolean canReceivePromotions, Boolean canReceivePartnerPromotions)
```

This table shows the parameters in the `postEnrollment` method.

Parameter	Description
additionalContactAttributes	The person account record details of the individual type member.
canReceivePartnerPromotions	Indicates whether the member can receive partner promotions for the program (<code>true</code>) or not (<code>false</code>).
canReceivePromotions	Indicates whether the loyalty program member can receive promotions.
createTransactionJournals	Indicates whether to create the transaction journal records <code>true</code> or not (<code>false</code>).
email	The email address of the member being enrolled.
emailNotification	Indicates whether the member wants to receive email notifications.

Parameter	Description
enrollmentChannel	The channel used to by loyalty program members to enroll in to the loyalty program. Possible values are: • Pos • Web • Email • CallCenter • Social • Mobile • Store • Franchise • Partner • Print
firstName	The first name of the member being enrolled.
lastName	The last name of the member being enrolled.
memberStatus	The status of the loyalty program member. Possible values are: • Active • Inactive • Custom
transactionalJournalStatementFrequency	The frequency at which transaction journal statements must be delivered to the member. Possible values are: • Monthly • Quarterly
transactionalJournalStatementMethod	The method used to deliver transaction journal statements to the member. Possible values are: • Mail • Email

unEnrollPromotion Method

Opt out a loyalty member from a promotion. Use this code to opt out of a promotion.

```
final Result<UnenrollPromotionResponse> unEnrollPromotion(String membershipNumber, String
promotionName)
```

Parameter	Type	Description	Required or Optional
membershipNumber	String	Unique identifier of the member who wants to opt out of a promotion	Required
promotionName	String	Name of the promotion that the member wants to opt out of.	Required

[Create an Instance of LoyaltyAPIManager Class for Android](#)

Create an instance of the `LoyaltyAPIManager` class and interact with the Loyalty Management APIs.

Create an Instance of LoyaltyAPIManager Class for Android

Create an instance of the `LoyaltyAPIManager` class and interact with the Loyalty Management APIs.

1. Create an instance of `LoyaltyAPIManager` by using this signature.

```
val loyaltyAPIManager = LoyaltyAPIManager(
    auth = forceAuthManager,
    instanceUrl = mInstanceUrl,
    loyaltyClient = LoyaltyClient(forceAuthManager, mInstanceUrl)
)``
```

2. To interact with the Loyalty Management API, call the appropriate methods by using this signature.

```
import LoyaltyMobileSDK

val forceAuthManager = ForceAuthManager(applicationContext)
val mInstanceUrl =
    forceAuthManager.getInstanceUrl() ?:
    AppSettings.DEFAULT_FORCE_CONNECTED_APP.instanceUrl
val loyaltyAPIManager = LoyaltyAPIManager(
    auth = forceAuthManager,
    instanceUrl = mInstanceUrl,
    loyaltyClient = LoyaltyClient(forceAuthManager, mInstanceUrl)
)
```

3. To enroll an individual member, use this signature.

```
let membershipNumber = "1234567890"
let firstName = "John"
let lastName = "Doe"
let email = "john.doe@example.com"
let phone = "4157891234"
let emailNotification = true

loyaltyAPIManager.postEnrollment(
    firstName = firstName,
    lastName = lastName,
    email = email,
    additionalContactAttributes = mapof("phone" to phone),
    emailNotification = emailNotification,
```



```

        memberStatus = MemberStatus.ACTIVE,
        createTransactionJournals = true,
        transactionJournalStatementFrequency =
TransactionalJournalStatementFrequency.MONTHLY,
        transactionJournalStatementMethod =
TransactionalJournalStatementMethod.EMAIL,
        enrollmentChannel = EnrollmentChannel.EMAIL,
        canReceivePromotions = true,
        canReceivePartnerPromotions = true
    )

```

4. To retrieve the member benefits, use this signature.

```
loyaltyAPIManager.getMemberBenefits(memberId = null, membershipKey = "1234567890")
```

5. To retrieve the member profile, use this signature.

```
loyaltyAPIManager.getMemberProfile(memberId = null, membershipKey = "1234567890",
programCurrencyName = null)
```

6. To opt a member into a promotion, use this signature.

```
loyaltyAPIManager.enrollInPromotions(membershipNumber = "1234567890", promotionName =
"PromotionName")
```

7. To opt out a member from a promotion using promotion ID or promotion name, use this signature.

```
loyaltyAPIManager.unEnrollPromotion(membershipNumber = "1234567890", promotionName =
"PromotionName")
```

8. To retrieve loyalty member transactions, use this signature.

```
loyaltyAPIManager.getTransactions(membershipNumber = "1234567890", pageNumber = null,
journalTypeName = null, journalSubTypeName = null, periodStartDate = null,
periodEndDate = null)
```

9. To retrieve promotions for a loyalty member, use this signature.

```
loyaltyAPIManager.getEligiblePromotions(membershipKey = "1234567890", memberId = null)
```

10. To retrieve vouchers for a loyalty member, use this signature.

```
loyaltyAPIManager.getVouchers(
    membershipKey = "1234567890", vouchreStatus = null, pageNumber = 1,
productId = null,
    productCategoryId = null, productName = null, productCategoryName = null
)
```

LoyaltyAPIInterface

The `LoyaltyApiInterface` class defines the interface for creating, formatting, and sending REST requests to the Salesforce service. This class stores REST API call definitions for all Loyalty API.

Use this signature to define the `LoyaltyAPIInterface` class.

```
public interface LoyaltyAPIInterface
```

Method Summary

Modifier and Type	Method	Method Detail	Description
abstract Result<EnrollmentResponse>	postEnrollment(@Url() String url, @Body() EnrollmentRequest json)	postEnrollment @POST() @Headers({ "Content-Type": "application/json" }) @Body() EnrollmentRequest json)	Creates a POST request for an individual enrollment for the loyalty program.
abstract Result<MemberProfileResponse>	getMemberProfile(@Url() String url, @Query(value = "memberId") String memberId, @Query(value = "relationshipName") String relationshipName, @Query(value = "pageNumber") Integer pageNumber)	getMemberProfile @GET() @Headers({ "Content-Type": "application/json" }) @Query(value = "memberId") String memberId, @Query(value = "relationshipName") String relationshipName, @Query(value = "pageNumber") Integer pageNumber)	Retrieves profile details related to the member ID.
abstract Result<MemberBenefitsResponse>	getMemberBenefits(@Url() String url, @Query(value = "relationshipName") String relationshipName)	getMemberBenefits @GET() @Headers({ "Content-Type": "application/json" }) @Query(value = "relationshipName") String relationshipName)	Retrieves benefits related to the member ID.
abstract Result<TransactionsResponse>	getTransactions(@Url() String url, @Query(value = "pageNumber") Integer pageNumber, @Query(value = "journalType") String journalType, @Query(value = "journalStyle") String journalStyle, @Query(value = "periodStartDate") String periodStartDate, @Query(value = "periodEndDate") String periodEndDate)	getTransactions @GET() @Headers({ "Content-Type": "application/json" }) @Query(value = "pageNumber") Integer pageNumber, @Query(value = "journalType") String journalType, @Query(value = "journalStyle") String journalStyle, @Query(value = "periodStartDate") String periodStartDate, @Query(value = "periodEndDate") String periodEndDate)	Retrieves transactions related to the member.
abstract Result<PromotionsResponse>	getEligiblePromotions(@Url() String url, @Body() PromotionsRequest requestBody)	getEligiblePromotions @POST() @Headers({ "Content-Type": "application/json" }) @Body() PromotionsRequest requestBody)	Retrieves promotions that a promotion is eligible to enroll for.

Modifier and Type	Method	Method Detail	Description
		<code>@Body() PromotionsRequest requestBody)</code>	
<code>abstract Result<EnrollPromotionResponse></code>	<code>enrollInPromotion(@Url() String url, @Body() PromotionsRequest requestBody)</code>	enrollInPromotion <code>@POST() abstract Result<EnrollPromotionResponse></code> <code>enrollInPromotion(@Url() String url, @Body() PromotionsRequest requestBody)</code>	Enrolls member in a promotion.
<code>abstract Result<UnenrollPromotionResponse></code>	<code>unenrollPromotion(@Url() String url, @Body() PromotionsRequest requestBody)</code>	unenrollPromotion <code>@POST() abstract Result<UnenrollPromotionResponse></code> <code>unenrollPromotion(@Url() String url, @Body() PromotionsRequest requestBody)</code>	Opts out a member from an available promotion.
<code>abstract Result<VoucherResult></code>	<code>getVouchers(@Url() String url, @Query(value = "voucherStatus") String voucherStatus, @Query(value = "pageNumber") Integer pageNumber, @Query(value = "productId") String productId, @Query(value = "productCategoryId") String productCategoryId, @Query(value = "productName") String productName, @Query(value = "productCategoryName") String productCategoryName)</code>	getVouchers <code>@GET() abstract Result<VoucherResult></code> <code>getVouchers(@Url() String url, @Query(value = "voucherStatus") String voucherStatus, @Query(value = "pageNumber") Integer pageNumber, @Query(value = "productId") String productId, @Query(value = "productCategoryId") String productCategoryId, @Query(value = "productName") String productName, @Query(value = "productCategoryName") String productCategoryName)</code>	Retrieves vouchers that are available to the program member.

Loyalty Management Mobile SDK API References for Android

The Loyalty Management Mobile SDK for Android interacts with Salesforce Loyalty Management APIs and retrieves member benefits, transactions, profiles, promotions, vouchers, enrolls an individual member, opts a member out of a promotion.

Interact with the Loyalty Management APIs, including:

Resource	Description
/loyalty-programs/{LoyaltyProgramName}/individual-member-enrollments on page 393	Individual Member Enrollments
/connect/loyalty/programs/{ProgramName}/program-process/enroll-for-promotion on page 425	Enroll for Promotions
services/apexrest/expense-analysis on page 726	Expense Analysis
/connect/loyalty/programs/{ProgramName}/program-process/get-member-promotions	Get Member Promotions
/connect/loyalty/member/{memberId}/member-benefits	Member Benefits
/loyalty-programs/{LoyaltyProgramName}/members	Member Profile
/loyalty/programs/{ProgramName}/members/{memberId}/vouchers	Member Vouchers
services/apexrest/receipt-status-updation on page 744	Receipt Status Updation
services/apexrest/upload-receipt on page 746	Receipt Upload
/connect/loyalty/programs/{ProgramName}/transaction-history	Transaction History
/connect/loyalty/programs/{ProgramName}/program-process/opt-out-member	Opt Out from a Promotion

EDITIONS

Available in: all editions that have Loyalty Management enabled

Loyalty Management Sample App for Mobile SDK

Loyalty Management Sample App for Mobile SDK is a configurable, native iOS and Android mobile app that delivers a personalized loyalty experience with every iOS and Android mobile device. Build the sample app for your loyalty program members to experience the loyalty and reward capabilities. The app provides a quick turnaround as opposed to a loyalty app that is built from scratch.

Loyalty Management Sample App for Mobile SDK offers a range of customization options. Customize the look and feel of your loyalty program, set up reward structures, and define the accrual and redemption rules.

EDITIONS

Available in: all editions that have Loyalty Management enabled

Use the Loyalty Management Mobile SDK Sample App

The MyNTORewards sample apps are a reference implementation of Loyalty Management Mobile SDK capabilities, and include the SDK's code, structure, and implementation. You can either implement the source code provided with the sample app in your existing code as-is or extend the capabilities of the code by modifying the code.

Walkthrough of the Sample App for Loyalty Management Mobile SDK

Adam can enroll as a loyalty program member and can experience Loyalty Management on the mobile device. Let's look at how Adam uses the sample app for Loyalty Management Mobile SDK.

Customize the Android SDK

After you build the Loyalty Management Mobile SDK for Android, you can customize the colors, fonts, and strings.

Customize the iOS SDK

After you build the Loyalty Management Mobile SDK for iOS, you can customize the colors and fonts to suit your requirements. You can also replace the existing images and logos included in the SDK for the sample app.

Use the Loyalty Management Mobile SDK Sample App

The MyNTORewards sample apps are a reference implementation of Loyalty Management Mobile SDK capabilities, and include the SDK's code, structure, and implementation. You can either implement the source code provided with the sample app in your existing code as-is or extend the capabilities of the code by modifying the code.

After ensuring you have the necessary development tools, dependencies, and the SDK installed, integrate the Loyalty Management Mobile SDK sample app code in your iOS or Android project:

1. Go to the location from where you downloaded the iOS or Android SDK, and select **SampleApps > MyNTORewards**.
2. From the Sample Apps project, copy the files that you want to reuse in your project.
3. Open an existing or new project in Xcode or Android Studio, and create a class or file.
4. Paste the files that you copied from the MyNTORewards sample apps.
5. Import any necessary dependencies that are used within.
6. Modify the code as per your requirements.
7. Test your app after you add the sample app code in iOS or Android.



Example: For example, consider the source code for the class, `CommunityMemberModel`:

```
package com.salesforce.loyalty.mobile.myntorewards.utilities

data class CommunityMemberModel(
    val firstName: String?,
    val lastName: String?,
    val email: String?,
    val loyaltyProgramMemberId: String?,
    val loyaltyProgramName: String?,
    val membershipNumber: String?
)
```

You can use the `CommunityMemberModel` class to represent the community member information as per your app. For example, you can create instances of this class to store member data:

```
val member = CommunityMemberModel(
    firstName = "Adam",
    lastName = "Smith",
    email = "adamsmith@example.com",
    loyaltyProgramMemberId = "123456",
    loyaltyProgramName = "NTO Insider Loyalty Program",
    membershipNumber = "78190"
)
```

You can modify the code provided with the sample app in the SDK to extract and store member data:

```
val firstName = member.firstName
val lastName = member.lastName
```

```
val email = member.email
val loyaltyProgramName = member.loyaltyProgramName
```

Walkthrough of the Sample App for Loyalty Management Mobile SDK

Adam can enroll as a loyalty program member and can experience Loyalty Management on the mobile device. Let's look at how Adam uses the sample app for Loyalty Management Mobile SDK.

Capability	Adam's Journey
Join as loyalty program	Opens the sample app on a mobile device and signs up as a member of the NTO Rewards loyalty program. Adam uses their Experience Cloud site email address to enroll in the NTO Insider program.
Review the membership details on the home page	Checks their current points balance, available promotions, vouchers, and badges.
Opt in for promotions	Reviews the list of promotions that are available to join and joins the Gold Tier Rejuvenation promotion. As a promotion enrollment bonus, 500 reward points are credited to Adam's points balance.
Add items to a cart and place an order You can extend the checkout workflow based on your mobile app's requirements.	Adds a red jacket to the cart, proceeds to the checkout page, enters the payment details, and places the order.
Track order rewards	Views the transactions list on the home page to verify the points that have been credited for the placed order.
View vouchers received for orders	Views the list of vouchers to verify that the Holiday Season voucher has been issued as a reward for the placed order.
Scan receipts for third-party store purchases	Uses the QR code scanner that's available on the My Profile tab to scan the receipt of a purchase made at a third-party store.
View and play assigned games	Goes to the Game Zone tab, opens an assigned game, and plays the game. After Adam plays the game, a game reward of 100 reward points is provided to Adam.

Customize the Android SDK

After you build the Loyalty Management Mobile SDK for Android, you can customize the colors, fonts, and strings.

1. Customize the colors.

- a. With your Sample Apps project open in Android Studio, open the **MyNTORewards > src > main > java > com > salesforce > loyalty > mobile > myntorewards > ui > theme > Color.kt** file.

- b. To add a color to the Android UI theme or as another app color, edit the `Color.kt` file.
For example, to add blue as one of the app colors, open the `Color.kt` file, and under `//App Colors`, add the `val Blue = Color(0x0000FF33)` code.
- c. Save the `Color.kt` file.
- d. To add the new color in any of the component definitions, add an entry or replace an entry with the new color.
For example, to replace the vibrant purple 40 color with blue, in any of the files, replace `color = VibrantPurple40` with `color = Blue`.
- e. Clean and rebuild your project in Android Studio.

2. Add a font to the Android UI theme.

- a. With your Sample Apps project open in Android Studio, go to the **MyNTORewards > src > main > res > font** folder.
- b. In the font folder, paste a .ttf file with the font script that you want to add.
- c. Open the **MyNTORewards > src > main > java > com > salesforce > loyalty > mobile > myntorewards > ui > theme > Type.kt** file.
- d. Edit the `Type.kt` file, and add the file that you added in the font folder.
For example, to add Arial as one of the fonts, edit the `Type.kt` file to add the sample code:

```
val font_arial = FontFamily(
    Font(R.font.arial)
)
```

- e. Save the `Type.kt` file.
 - f. Add this font in any of the component definitions.
For example, to replace the `sf_pro` font on the My Transactions page with Arial, replace `fontFamily = font_sf_pro` with `fontFamily = font_arial`.
 - g. Clean and rebuild your project in Android Studio.
- ## 3. Customize the UI labels or strings used in your Sample App.
- a. Open the **MyNTORewards > src > main > res > values > strings.xml** file.
 - b. Edit any text included in the `<string>` tag.
For example, to change the text Membership Number to Membership ID, replace all the strings that contain Membership Number with Membership ID.
 - c. Save the `strings.xml` file.
 - d. Clean and rebuild your project in Android Studio.
- ## 4. To replace an image, go to **MyNTORewards > src > main > res > drawable**, and replace the image.

Customize the iOS SDK

After you build the Loyalty Management Mobile SDK for iOS, you can customize the colors and fonts to suit your requirements. You can also replace the existing images and logos included in the SDK for the sample app.

1. Customize the colors.

- a. With your Sample Apps project open in Xcode, select **MyNTORewards > MyNTORewards > Assets > Colors**.
- b. To modify an existing colorset, click **Any Appearance**, and under Color, edit the Hex value.
- c. To add the dark color, click **Dark**, and from Content, select a color.
- d. To add a color theme, open the **MyNTORewards > MyNTORewards > Misc > Color** file, and add a color theme to the sample code.

For example, to add blue color to the theme, use this code:

```
struct ColorTheme {let blue = Color("BlueColor")}
```

- e. To use a color in the app, use these sample codes as view or within view modifier:

```
ZStack {
    Color.theme.[color]
}
```

Or `background(Color.theme.[color])`

For example, to add Blue color to the theme, use this code as view or within view modifier:

```
ZStack {
    Color.theme.blue
}
```

Or `background(Color.theme.blue)`

2. Add your font to the iOS app.
 - a. [Add your font to the Xcode project](#).
 - b. Add your font to the **MyNTORewards > MyNTORewards > Info.plist** file.
3. Use the font that you added in the app.
 - a. With your Sample Apps project open in Xcode, select **MyNTORewards > MyNTORewards > UICompnents > FontStyles.swift** file.
 - b. Edit the file and add the location where you want to use the font. For example, to customize a footer text, use this sample code:


```
static var footerText: Font {
    return Font.custom("SFPro-Regular", size: 16)
}
```
 - c. In the Text view, call the font style as a new modifier with the sample code:


```
Text("Footer")
    .font(.footerText)
```
 - d. Save your changes.
4. To replace an image, go to **MyNTORewards > MyNTORewards > Assets > Images**. For example, to replace a logo, go to the **MyNTORewards > MyNTORewards > Assets > Icons** folder, and modify the logo file.

Receipt Scanning with Loyalty Management Mobile SDK

Members of your loyalty program can scan and upload the purchase receipts of third-party store or site transactions on the Loyalty Management iOS or Android mobile app. After members upload their receipts, members are rewarded for eligible purchases, they can also view the accrued points on the Loyalty Management mobile app.

Watch this video to understand how you can enable members to earn rewards by purchasing your company's products from multibrand and third-party stores.

 [Watch a video](#)

If you aren't able to watch the video in full screen, open the video on a new tab: [Reward Members for Shopping at Multibrand \(Third-Party\) Stores.](#)

[How Receipt Scanning Works](#)

Integrate Loyalty Management with receipt scanning to enhance your brand's member experience and streamline how members can earn loyalty points for their store purchases. The receipt scanning workflow is designed to automatically extract and organize essential information from scanned receipts and reward members for their store purchases.

[Loyalty Management Receipt Scanner Kit Unlocked Package](#)

The Loyalty Management Receipt Scanner Kit unlocked package allows members to scan and submit receipts to receive loyalty points that they're purchases are eligible for. After the receipt is submitted, the package automatically creates transaction journals for the eligible receipt line items. The unlocked package contains a set of components, and developers can select the modular components that they want to use in their implementation. Unlocked packages are customizable and help you easily manage and reuse components across different projects.

[Set Up Amazon Web Services](#)

To integrate the Loyalty Management Receipt Scanning Kit unlocked package with Amazon Web Services (AWS), create an S3 bucket that stores the receipt images, and create roles, policies, and users. Then, get the access key and secret ID to authenticate the access to AWS resources.

[Set Up Salesforce Org With Loyalty Management Receipt Scanner Kit Unlocked Package](#)

To ensure that members can submit their receipts by using their Loyalty Management mobile app and get rewarded for eligible receipt line items, install and set up the Loyalty Management Receipt Scanner Kit Unlocked Package in your Salesforce org. Before you install the unlocked package, ensure that you have a loyalty program already set up in your org.

[Receipt Scan and Upload on iOS and Android](#)

After you set up the Loyalty Management Receipt Scanner Kit unlocked package and integrate the Loyalty Management org with an Optical Character Recognition (OCR) software, members can upload and scan receipts by using the loyalty program's mobile app. Members are rewarded for with loyalty points for eligible receipt line items.

How Receipt Scanning Works

Integrate Loyalty Management with receipt scanning to enhance your brand's member experience and streamline how members can earn loyalty points for their store purchases. The receipt scanning workflow is designed to automatically extract and organize essential information from scanned receipts and reward members for their store purchases.

[Why Integrate Loyalty Management With a Receipt Scanner?](#)

Brands that don't own their sales channels can't accurately track customer purchases made in on third-party stores or sites. In such scenarios, the brand's loyalty program needs a receipt-scanning solution that can scan the items from a receipt, identify eligible items, and reward the points to the members.

Receipt Scanning Workflow

The receipt scanning workflow describes how a program member uploads and scans a receipt, verifies the scanned result, and earns loyalty points.

Receipt Evaluation

Members use their mobile phones to upload their purchase receipts. After a member uploads a receipt, they can review the receipt's scanned line items, and submit the scanned receipt. After a receipt is submitted, the eligible line items are processed as transaction journals and members are rewarded. You can define how a receipt is evaluated when you enable Receipt Scanning.

Why Integrate Loyalty Management With a Receipt Scanner?

Brands that don't own their sales channels can't accurately track customer purchases made in on third-party stores or sites. In such scenarios, the brand's loyalty program needs a receipt-scanning solution that can scan the items from a receipt, identify eligible items, and reward the points to the members.

For Loyalty Management's receipt scanning capabilities, install the Loyalty Management Receipt Scanner Kit package and then integrate the package with any optical character recognition (OCR) and document processing software such as Amazon Textract AnalyzeExpense. The integration helps convert text from receipts into a valid format.

EDITIONS

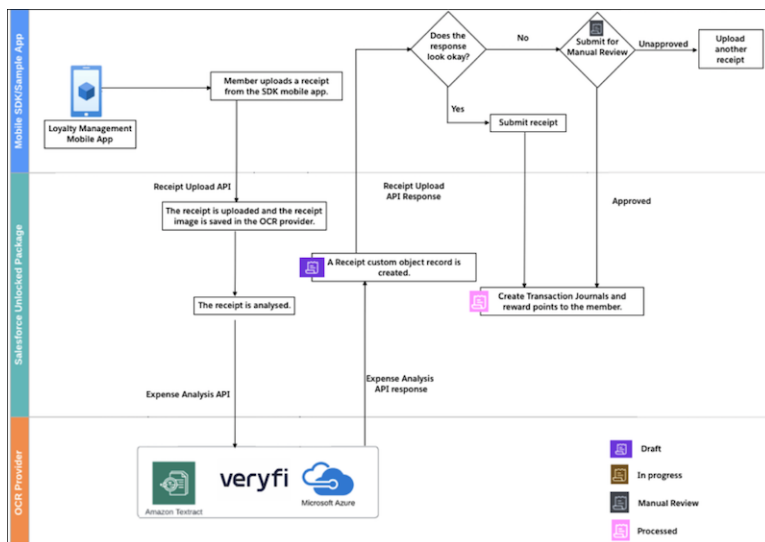
Available in: all editions that have Loyalty Management enabled



Example: Member shops at a shopping mall for their loyalty products. Then, the Member logs in to the mobile app and submits the receipt to earn points. The receipts are verified and the member is rewarded for the purchase of the eligible items.

Receipt Scanning Workflow

The receipt scanning workflow describes how a program member uploads and scans a receipt, verifies the scanned result, and earns loyalty points.



- After a member uploads a receipt by using the Loyalty Management iOS or Android mobile app, the Receipt Upload API is automatically run to encode the receipt.
- The AnalyzeExpense API is then automatically executed for the encoded receipt, and the response creates a record to store the receipt data in an Amazon S3 bucket. The API scans the uploaded receipt and provides the details of the receipt as a response. The

response is saved in the custom object, Receipts, in the Loyalty Management app. The member then previews the scanned results on the mobile app, and submits the receipt. A transaction journal is created for each line item in the receipt.

- If the member submits the receipt for a manual review, the loyalty program manager or admin reviews, and approves or rejects the receipt.

If the program manager approves the receipt, the Receipt Status Updation API updates the status of the receipt to Processed, updates the reward points, and creates a transaction journal for each line item in the receipt. The RulesEngineResponseWrapper Apex class shows the response returned from Transaction Journal Execution API that includes the transaction journal details.

Receipt Evaluation

Members use their mobile phones to upload their purchase receipts. After a member uploads a receipt, they can review the receipt's scanned line items, and submit the scanned receipt. After a receipt is submitted, the eligible line items are processed as transaction journals and members are rewarded. You can define how a receipt is evaluated when you enable Receipt Scanning.

Here are the factors for evaluating.

EDITIONS

Available in: all editions that have Loyalty Management enabled

Confidence Score

Define a confidence score to assess if the scanned receipt meets the criteria to extract the receipt information. A confidence value higher than the defined threshold value typically implies greater certainty in the accuracy of the receipt scanning result, while a lower value suggests more uncertainty in reading the receipt.

To understand how a receipt is evaluated based on the confidence score, consider these scenarios:

Business Scenario	Result	What Happens in this Scenario?
A member uploads a clean receipt and the receipt content is readable.	The confidence value for the receipt is higher than or equal to the threshold value.	Submit the receipt and get rewards for eligible items.
A member uploads a crumpled receipt or a receipt with poor image quality, and the receipt lines are partially readable.	The confidence value for the crumpled receipt is lower than the threshold value.	Submit the receipt for a manual review or capture the image again to ensure that it's readable.
A member uploads a crumpled or a torn receipt, and the mandatory fields on the receipt such as receipt number, date, and store name aren't readable.	The confidence value for the crumpled receipt is lower than the threshold value.	The receipt is rejected because the information required to process the receipt wasn't extracted.

Ineligible Receipt Items

Configure the receipt scanning to show members the items from their receipt that aren't eligible for loyalty points. Typically, items that don't meet the criteria for earning loyalty points include certain excluded categories, products, or expired promotions.

Required Receipt Fields

If any mandatory information is missing in a receipt uploaded by a member, the receipt is rejected. When you enable receipt scanning, Receipt Date is a mandatory field to be included in every receipt. You can select the other mandatory fields from these options:

- Receipt Number

- Store Name
- Postal Code

You can also configure additional mandatory fields on a receipt.

Eligible Item Identifier Fields

When you enable receipt scanning, you can select fields that uniquely determine eligible items from a receipt. For example, if you select product name and product code as the eligible item identifier fields, the receipt is evaluated to determine the eligible items, and the results show the products that match the filter criteria with product name and product code.

Consider this example to understand how an eligible item is determined. Adam Smith submits a receipt by using the Loyalty Management mobile app.



The receipt is uploaded using the Receipt Upload API, and the Expense Analysis API provides the scanned results by using this JSON response:

```
{
  "ExpenseDocuments": [
    {
      "SummaryFields": [
        {
          "ValueDetection": {
            "Text": "FROSTFLAKES SNOWFLAKES",
            "Type": {
              "Text": "NAME"
            }
          }
        },
        {
          "ValueDetection": {
            "Text": "260.00",
            "Type": {
              "Text": "AMOUNT_PAID"
            }
          }
        },
        {
          "ValueDetection": {
            "Text": "19-04-2018",
            "Type": {
              "Text": "INVOICE_RECEIPT_DATE"
            }
          }
        },
        {
          "ValueDetection": {
            "Text": "38",
            "Type": {
              "Text": "INVOICE_RECEIPT_ID"
            }
          }
        },
        {
          "ValueDetection": {
            "Text": "23BBNCA2901C1Z6",
            "Type": {
              "Text": "TAX_PAYER_ID"
            }
          }
        },
        {
          "ValueDetection": {
            "Text": "23BBNCA2901C1Z6",
            "Type": {
              "Text": "VENDOR_GST_NUMBER"
            }
          }
        },
        {
          "ValueDetection": {
            "Text": "247.62",
            "Type": {
              "Text": "SUBTOTAL"
            }
          }
        },
        {
          "ValueDetection": {
            "Text": "6.19",
            "Type": {
              "Text": "TAX"
            }
          }
        },
        {
          "ValueDetection": {
            "Text": "6.19",
            "Type": {
              "Text": "TAX"
            }
          }
        },
        {
          "ValueDetection": {
            "Text": "260.00",
            "Type": {
              "Text": "TOTAL"
            }
          }
        },
        {
          "ValueDetection": {
            "Text": "FROSTFLAKES SNOWFLAKES",
            "Type": {
              "Text": "VENDOR_NAME"
            }
          }
        },
        {
          "ValueDetection": {
            "Text": "2001",
            "Type": {
              "Text": "OTHER"
            }
          }
        },
        {
          "ValueDetection": {
            "Text": "01",
            "Type": {
              "Text": "OTHER"
            }
          }
        },
        {
          "ValueDetection": {
            "Text": "06:17:34 PM",
            "Type": {
              "Text": "OTHER"
            }
          }
        },
        {
          "ValueDetection": {
            "Text": "Q38",
            "Type": {
              "Text": "OTHER"
            }
          }
        },
        {
          "ValueDetection": {
            "Text": "Cashier 1",
            "Type": {
              "Text": "OTHER"
            }
          }
        },
        {
          "ValueDetection": {
            "Text": "QUICK SERVICE",
            "Type": {
              "Text": "OTHER"
            }
          }
        },
        {
          "ValueDetection": {
            "Text": "Cash",
            "Type": {
              "Text": "OTHER"
            }
          }
        }
      ]
    }
  ]
}
```

```

"LineItemGroups": [
  "LineItems": [
    {
      "LineItemExpenseFields": [
        {
          "ValueDetection": {
            "Text": "THE PHAL",
            "Type": {
              "Text": "PRODUCT_NAME"
            }
          },
        },
        {
          "ValueDetection": {
            "Text": "95.24",
            "Type": {
              "Text": "UNIT_PRICE"
            }
          },
        },
        {
          "ValueDetection": {
            "Text": "1.00",
            "Type": {
              "Text": "QUANTITY"
            }
          },
        },
        {
          "ValueDetection": {
            "Text": "95.24",
            "Type": {
              "Text": "PRICE"
            }
          },
        },
        {
          "ValueDetection": {
            "Text": "THE PHAL 95.24 1.00 95.24",
            "Type": {
              "Text": "EXPENSE_ROW"
            }
          }
        ]
      },
      "LineItemExpenseFields": [
        {
          "ValueDetection": {
            "Text": "NATURAL",
            "Type": {
              "Text": "PRODUCT_NAME"
            }
          },
        },
        {
          "ValueDetection": {
            "Text": "76.19",
            "Type": {
              "Text": "UNIT_PRICE"
            }
          },
        },
        {
          "ValueDetection": {
            "Text": "1.00",
            "Type": {
              "Text": "QUANTITY"
            }
          },
        },
        {
          "ValueDetection": {
            "Text": "76.19",
            "Type": {
              "Text": "PRICE"
            }
          },
        },
        {
          "ValueDetection": {
            "Text": "NATURAL 76.19 1.00 76.19",
            "Type": {
              "Text": "EXPENSE_ROW"
            }
          }
        ]
      },
      "LineItemExpenseFields": [
        {
          "ValueDetection": {
            "Text": "FIGS",
            "Type": {
              "Text": "PRODUCT_NAME"
            }
          },
        },
        {
          "ValueDetection": {
            "Text": "76.19",
            "Type": {
              "Text": "UNIT_PRICE"
            }
          },
        },
        {
          "ValueDetection": {
            "Text": "1.00",
            "Type": {
              "Text": "QUANTITY"
            }
          },
        },
        {
          "ValueDetection": {
            "Text": "76.19",
            "Type": {
              "Text": "PRICE"
            }
          },
        },
        {
          "ValueDetection": {
            "Text": "FIGS 76.19 1.00 76.19",
            "Type": {
              "Text": "EXPENSE_ROW"
            }
          }
        ]
      }
    ]
  ]
}

```

The loyalty program admin has selected Price as the eligible item identifier field, and so the receipt line item's response shows eligible items by comparing the line item's price with the product price. The eligible items in the receipt are: The Phal, Natural, and Figs.

Loyalty Management Receipt Scanner Kit Unlocked Package

The Loyalty Management Receipt Scanner Kit unlocked package allows members to scan and submit receipts to receive loyalty points that they're purchases are eligible for. After the receipt is submitted, the package automatically creates transaction journals for the eligible receipt line items. The unlocked package contains a set of components, and developers can select the modular components that they want to use in their implementation. Unlocked packages are customizable and help you easily manage and reuse components across different projects.

EDITIONS

Available in: all editions that have Loyalty Management enabled

[Loyalty Management Receipt Scanner Kit Unlocked Package Components](#)

The Loyalty Management Receipt Scanner Kit unlocked package contains a set of components such as custom objects, fields, Apex classes, Lightning web components, and more to facilitate easier distribution of these bundled components.

Loyalty Management Receipt Scanner Kit Unlocked Package Components

The Loyalty Management Receipt Scanner Kit unlocked package contains a set of components such as custom objects, fields, Apex classes, Lightning web components, and more to facilitate easier distribution of these bundled components.

EDITIONS

Available in: all editions that have Loyalty Management enabled

Custom Objects

The package consists of custom objects that store the details of the scanned receipts.

Object	What the Object Stores
Receipt	Saves the details of the scanned and submitted receipts
Receipt Scanning Error	Saves the details of the errors that occur while scanning receipts.

Permission Set, List View, Page Layout, and Tab

The Receipt Scanning permission set enables loyalty program members to access the Receipt Scanning feature.

Loyalty program managers can review all submitted receipts on the Receipts tab on the Loyalty Management app. In Setup, the Receipt Scanning setting enables your loyalty program members to scan receipts on their Loyalty Management iOS and Android mobile apps, and to earn loyalty points for eligible items.

Custom Settings

The package consists of a custom setting that stores the values that are required to connect Loyalty Management with an Optical Character Recognition (OCR) software.

Setting	What the Setting Stores
API Endpoint Details	The API Gateway endpoint URL that is used to access and interact with • The APIs • The bucket details that store the objects used in the AWS endpoint • The API Named Credential that provides the API Gateway endpoint's URL • The authentication settings for Salesforce to make callouts to AWS

Custom Label

The package consists of a custom label that stores the values that are required to reward non-qualifying points.

Setting	What the Setting Stores
PrimaryNQPCurrencyName	This label stores the primary NQP currency name. To replace this currency name with your program's primary NQP currency, from the Setup page, search and find Custom Label , and then edit and replace the label.

Apex Classes

The unlocked package contains Apex classes to run the Expense Analysis API and Receipt Status Updation API for submitted receipts, and to show the scan results on the Loyalty Management mobile app.

Apex Class	What the Class Does
DMLManager	Manages Data Manipulation Language (DML) statements to insert, update, merge, delete, and restore data in Salesforce.
ExpenseAnalysis	Manages the request to process the uploaded receipt.
ProcessTransactionJournal	Creates transaction journals for each line item of a scanned receipt and calls the Rules Engine API to process the transaction journals.
ReceiptScanningConfigController	Manages the receipt scanning configuration.
ReceiptScanningControllerUtils	Manages the common functionality methods.
ReceiptScanningResponseWrapper	Analyzes the eligible and ineligible items in the receipt after a receipt is uploaded and returns a response of line items for the scanned receipt.
ReceiptStatusUpdate	Updates the status of receipts by sending REST requests to a specific location or endpoint.
ReceiptUpload	Manages the request to upload the receipt and creates a request to analyze the uploaded receipt.
ReceiptWrapper	Represents the data structure of an AWS response.
RulesEngineResponseWrapper	Analyzes the response returned from the Rules Engine API that includes transaction journal details.

Custom Metadata Types

The unlocked package contains custom metadata types to create and store the objects and fields related to receipt scanning.

Custom Metadata Type	What the Custom Metadata Type Does	Examples
Accepted Date Format	Represents the configuration of the accepted date format and the store name fields related to the receipt.	26/02/2022
Required Receipt Fields	Represents the configuration of the mandatory fields to be captured from a scanned receipt.	Receipt Date
Receipt Scanning	Represents the configuration details related to the receipt scanning feature.	Confidence score, named credential, receipt scanning activity, and show ineligible receipt items

Apex Trigger

The unlocked package contains apex triggers to update the status of a receipt and to automatically assign the Receipt Scanning permission set to Experience Cloud users.

Apex Trigger	What the Apex Trigger Does
ReceiptUpdateTrigger	Creates transaction journals, and processes the journals by using the Rule Engine API.
UserReceiptScanTrigger	Automatically assigns the Receipt Scanning permission set to all the Experience Cloud profile users, after the admin installs the unlocked package. For existing loyalty program members, the admin has to manually assign the Receipt Scanning permission set to the members' Experience Cloud users.

Customizations

Admins can customize their receipt scanning implementation by extending the component provided by the Loyalty Management Receipt Scanner Kit unlocked package. The admin can add additional checks that decide which elements must be available in a receipt for it to be eligible to be rewarded.



Example: For example, if the loyalty program manager wants to add a filter logic that decides which products details must be available in a receipt, the admin can customize these package components:

- Apex classes
 - ReceiptScanningResponseWrapper
 - ReceiptScanningControllerUtils
 - receiptScanningDetail
 - Eligible Item Identifier Field custom metadata type

Set Up Amazon Web Services

To integrate the Loyalty Management Receipt Scanning Kit unlocked package with Amazon Web Services (AWS), create an S3 bucket that stores the receipt images, and create roles, policies, and users. Then, get the access key and secret ID to authenticate the access to AWS resources.

1. [Create Your Amazon S3 Bucket](#)

Create a bucket to store your receipt images and any other data in AWS.

2. [Create an IAM Role](#)

Create your Identity and Access Management (IAM) role to control the access to your Amazon S3 bucket. The IAM role ensures the security and functioning of AWS resources.

3. [Attach a Policy to the IAM Role](#)

After you create an Identity and Access Management (IAM) role for your S3 bucket, create a policy and attach the policy to the IAM role. Policies define the actions for IAM roles and users.

4. [Get Your AWS ARN](#)

After the Identity and Access Management (IAM) role is created, and a policy is attached to the role, you can copy the unique identifiers of AWS resources, Amazon Resource Names (ARN). Use the ARN when you create the API gateway.

5. [Create an IAM User](#)

An Identity and Access Management (IAM) user manages the access and permissions for the users and applications within your AWS account. Create an IAM user and assign permissions to the IAM user to independently maintain security and access control in your AWS Management Console.

6. [Create Access Key and Secret](#)

After you create an IAM user, create an access key for the user and download the access key and secret. Use the access keys and secrets to authenticate and authorize programmatic access to AWS resources.

7. [Create an API Gateway](#)

Create an API Gateway that serves as an entry point for all incoming API requests.

8. [Get the Invoke URL](#)

After you create the API and select a deployment stage, deploy the API so that members can upload their receipts and get loyalty points. This invoke URL is used to create the API Endpoint Details custom setting and represents the Remote Site URL for your AWS API Gateway endpoint.

Create Your Amazon S3 Bucket

Create a bucket to store your receipt images and any other data in AWS.

1. Log in to your AWS Management Console.
2. From the Services menu, find and select **S3**.
3. Click **Create bucket**.
4. In AWS Region, select the region where you want the bucket to reside.
5. Enter a name for your bucket.
6. Click **Create bucket**.

EDITIONS

Available in: all editions that have Loyalty Management enabled

Create an IAM Role

Create your Identity and Access Management (IAM) role to control the access to your Amazon S3 bucket. The IAM role ensures the security and functioning of AWS resources.

1. Log in to your AWS Management Console.
2. From the Services menu, find and select **IAM**.
3. From the navigation menu, click **Roles**.
4. Click **Create role**.
5. From the Service or use case dropdown list, select **API Gateway**, and then click **Next**.
6. On the Add permissions page, click **Next**.
7. Enter the role name and description, and then click **Create role**.

EDITIONS

Available in: all editions that have Loyalty Management enabled

Attach a Policy to the IAM Role

After you create an Identity and Access Management (IAM) role for your S3 bucket, create a policy and attach the policy to the IAM role. Policies define the actions for IAM roles and users.

EDITIONS

Available in: all editions that have Loyalty Management enabled

1. Log in to your AWS Management Console.
2. From the Services menu, find and select **IAM**.
3. From the navigation menu, click **Roles**.
4. Click the IAM role that you created.
5. Under Permissions, from Add permissions, select **Attach policies**.
If you see the Create policy button instead of the Attach policies button, click Create policy and follow the subsequent steps.
6. From the navigation menu, under Access management, click **Policies**, and then click **Create policy**.
7. From Service, select **S3**.
8. In the Actions allowed list, find and select the PutObject write action.
9. Under Resources, click **Add ARNs**.
10. Specify the ARNs.
 - a. Enter your S3 bucket name as the resource bucket name.
 - b. In Resource object name, select **Any object name**.
 - c. Click **Add ARNs**.
11. Click **Next**.
12. On the Review and create page, enter your policy name and description, and then click **Create policy**.
The policy is created, you can now attach this policy to the IAM role.
13. On the Policies page, find and select your policy.
14. Click **Entities Attached**.
15. Under Attached as a permissions policy, click **Attach**, and select the IAM role that you created.
16. Click **Attach policy**.
17. From the navigation menu, under Access management, click **Roles**, and then select the role that you created.
18. Under Permissions, from Add permissions, select **Attach Policies**.
19. Select the **AdministratorAccess** option, and then click **Add permissions**.

Get Your AWS ARN

After the Identity and Access Management (IAM) role is created, and a policy is attached to the role, you can copy the unique identifiers of AWS resources, Amazon Resource Names (ARN). Use the ARN when you create the API gateway.

EDITIONS

Available in: all editions that have Loyalty Management enabled

1. Log in to your AWS Management Console.
2. From the Services menu, find and select **IAM**.
3. From the navigation menu, click **Roles**.
4. Click the IAM role that you created.
5. Under Summary, click  to copy the ARN.

Create an IAM User

An Identity and Access Management (IAM) user manages the access and permissions for the users and applications within your AWS account. Create an IAM user and assign permissions to the IAM user to independently maintain security and access control in your AWS Management Console.

1. Log in to your AWS Management Console.
2. From the Services menu, find and select **IAM**.
3. From the navigation menu, click **Users**.
4. Click **Create user**.
5. Enter a username, and then click **Next**.
6. Under Permissions options, click **Attach policies directly**.
7. Under Permissions policies, enter *textract*.
8. Select **AmazonTextractFullAccess**, and find and select **AmazonS3ReadOnlyAccess**.
9. Click **Next**.
10. Click **Create user**.

EDITIONS

Available in: all editions that have Loyalty Management enabled

Create Access Key and Secret

After you create an IAM user, create an access key for the user and download the access key and secret. Use the access keys and secrets to authenticate and authorize programmatic access to AWS resources.

1. Log in to your AWS Management Console.
2. From the Services menu, find and select **IAM**.
3. From the navigation menu, click **Users**.
4. Select the user that you created.
5. Under Summary, click **Create access key**.
6. Select **Application running outside AWS**, and then click **Next**.
7. Enter a description to define the purpose of the key, and then click **Create access key**.
8. To ensure that you don't lose the access credentials, download the CSV file that contains the access key and secret.

EDITIONS

Available in: all editions that have Loyalty Management enabled

Create an API Gateway

Create an API Gateway that serves as an entry point for all incoming API requests.

1. Log in to your AWS Management Console.
2. From the Services menu, find and select **API Gateway**.
3. Click **Create API**.
4. Under REST API, click **Build**.
5. Enter a name and description for the new API, and then click **Create API**.
6. Click **Create resource**.
7. Enter *{bucket}* as the resource name.

EDITIONS

Available in: all editions that have Loyalty Management enabled

8. Select **CORS (Cross Origin Resource Sharing)**, and then click **Create resource**.
9. To create another resource for bucket, select **bucket**, and then click **Create resource**.
10. Enter `{filename}` as the resource name.
11. Select **CORS (Cross Origin Resource Sharing)**, and then click **Create resource**.
12. To create a method under filename, select **filename**, and then click **Create method**.
13. For the Method type, select **PUT**.
14. Select **AWS service** as the integration type.
15. From AWS Region, select the region of your S3 bucket.
16. From AWS service, select **Simple Storage Service (S3)**.
17. Select **PUT** as the HTTP method.
18. Select **Use path override** as the action type.
19. In Execution role, paste the ARN for the IAM role that you created earlier.
20. Click **Create method**.
21. On the PUT-Method Execution page, click **Integration request**.
22. Click **Edit**.
23. Expand URL path parameters, and click **Add path**, and enter these details:
 - Name: `bucket`
 - Mapped from: `method.request.path.bucket`
24. Save your changes.
25. Under Integration request, click **Edit**, and then expand URL path parameters.
26. Click **Add path** again, and enter these details:
 - Name: `filename`
 - Mapped from: `method.request.path.filename`
27. Save your changes.
28. From the navigation menu, click **API settings**.
29. To allow all the media types, under Binary media types, click **Manage media types**.
30. Click **Add binary media type**, and enter `*/*`.
31. Save your changes.
32. Similarly, create the Get and Delete methods under `/ {bucket} / {filename}`.

Get the Invoke URL

After you create the API and select a deployment stage, deploy the API so that members can upload their receipts and get loyalty points. This invoke URL is used to create the API Endpoint Details custom setting and represents the Remote Site URL for your AWS API Gateway endpoint.

1. Log in to your AWS Management Console.
2. From the Services menu, find and select **API Gateway**.
3. Click the name of the API that you created.

EDITIONS

Available in: all editions that have Loyalty Management enabled

4. Click **Deploy API**.
5. Select the deployment stage.
If you don't have a stage, select **[New Stage]** and enter the stage name and the deployment description.
6. Click **Deploy**.
The deployment takes a couple of minutes to be completed.
7. Under Stage details, click , and copy the Invoke URL.

Set Up Salesforce Org With Loyalty Management Receipt Scanner Kit Unlocked Package

To ensure that members can submit their receipts by using their Loyalty Management mobile app and get rewarded for eligible receipt line items, install and set up the Loyalty Management Receipt Scanner Kit Unlocked Package in your Salesforce org. Before you install the unlocked package, ensure that you have a loyalty program already set up in your org.

1. [Install the Loyalty Management Receipt Scanner Kit Unlocked Package](#)
Install the Loyalty Management Receipt Scanner Kit unlocked package in the org that contains the loyalty program for members to upload their purchase receipts by using the Loyalty Management mobile app.
2. [Assign Permission Set to Existing Users](#)
After admins install the Loyalty Management Receipt Scanner Kit unlocked package in their org, new users are automatically assigned the Receipt Scanning permission set. The Receipt Scanning permission set provides users access to manage the capabilities of the Loyalty Management Receipt Scanner Kit unlocked package. Assign existing users who need the permissions to manage and work with the capabilities of the package the Receipt Scanning permission set.
3. [Assign Permission Set for Members to Access Receipt Scanner](#)
For your members to access the Receipt Scanner feature on their mobile devices, assign the Receipt Scanning permission set to the Experience Cloud users. Assign permission set to the members to ensure that Apex classes are automatically run when members upload a receipt. After you assign the permission set, modify the Receipts object to provide read and edit permissions.
4. [Create a Named Credential](#)
Connect Amazon Web Services (AWS) with the Loyalty Management Receipt Scanner Kit unlocked package. Make callout requests to AWS from Salesforce after a receipt is uploaded. The AnalyzeExpense API then scans the receipts to analyze and interpret the information from the receipt.
5. [Create an External Client App](#)
Create an external client app for Amazon Web Services (AWS) to connect with the Loyalty Management Receipt Scanner Kit unlocked package. This external client app enables secure OAuth authentication so that the Expense Analysis API runs after the receipts are uploaded and creates transaction journals for the submitted receipts.
6. [Add a Loyalty Program Name](#)
The Loyalty Management Receipt Scanner Kit unlocked package has a custom label to store the name of your loyalty program.
7. [Add API Endpoint Details in Custom Setting](#)
The Loyalty Management Receipt Scanner Kit unlocked package has a custom setting to store the API Gateway endpoint URL and bucket details from your Amazon Web Services (AWS) account.
8. [Create a Remote Site Setting](#)
Before any Apex callout can call the API Gateway endpoint, register the API Gateway endpoint on the Remote Site Settings page. Remote site settings authorize your org to access external websites.

9. [Enable Receipt Scanning](#)

Enable receipt scanning in the org where the Loyalty Management Receipt Scanner Kit unlocked package is installed. Members can then use their mobile device to upload receipts and get loyalty points.

Install the Loyalty Management Receipt Scanner Kit Unlocked Package

Install the Loyalty Management Receipt Scanner Kit unlocked package in the org that contains the loyalty program for members to upload their purchase receipts by using the Loyalty Management mobile app.

1. From your loyalty org, click [Loyalty Management Receipt Scanner Kit Unlocked Package](#).
2. Enter your Salesforce credentials.
3. Select the audience for your package.
4. Click **Install**.

After the unlocked package is installed, the Receipt Scanning permission set is automatically assigned to the Experience Cloud users' of new loyalty program members and Apex classes are automatically run when members upload a receipt. Members can now use the Loyalty Management app on their mobile devices to submit receipts and earn loyalty points.

For existing loyalty program members, you've to manually assign the Receipt Scanning permission set to the members' Experience Cloud users.

EDITIONS

Available in: all editions that have Loyalty Management and Receipt Scanning enabled

Assign Permission Set to Existing Users

After admins install the Loyalty Management Receipt Scanner Kit unlocked package in their org, new users are automatically assigned the Receipt Scanning permission set. The Receipt Scanning permission set provides users access to manage the capabilities of the Loyalty Management Receipt Scanner Kit unlocked package. Assign existing users who need the permissions to manage and work with the capabilities of the package the Receipt Scanning permission set.

1. From Setup, in the Quick Find box, enter *Permission Sets*, and then select **Permission Sets**.
2. Click **Receipt Scanning > Manage Assignments > Add Assignment**.
3. Select the existing users must manage and work with the receipt scanning capabilities.
4. Click **Next > Assign**.

EDITIONS

Available in: all editions that have Loyalty Management and Receipt Scanning enabled

USER PERMISSIONS

To assign a permission set:

- Assign Permission Sets

To assign a permission set license:

- Manage Users

Assign Permission Set for Members to Access Receipt Scanner

For your members to access the Receipt Scanner feature on their mobile devices, assign the Receipt Scanning permission set to the Experience Cloud users. Assign permission set to the members to ensure that Apex classes are automatically run when members upload a receipt. After you assign the permission set, modify the Receipts object to provide read and edit permissions.

1. From Setup, in the Quick Find box, enter *User*, and then select **Users**.
2. Select the user record of the Loyalty Customer Community User.
3. Click **Permission Set Assignments > Edit Assignments**.
4. Move the **Receipt Scanning** permission set to the Enabled Permission Sets list.
5. Save your changes.
6. From Setup, in the Quick Find box, enter *Permission Sets*, and then select **Permission Sets**.
7. Click **Receipt Scanning**.
8. Click **Object Settings**.
9. From the object settings, click **Receipts**, and then click **Edit**.
10. For the ReceiptScanner__ReceiptID__c field, select the Read and Edit permissions.
11. Save your changes.

EDITIONS

Available in: all editions that have Loyalty Management and Receipt Scanning enabled

USER PERMISSIONS

To assign a permission set:

- Assign Permission Sets

To assign a permission set license:

- Manage Users

Create a Named Credential

Connect Amazon Web Services (AWS) with the Loyalty Management Receipt Scanner Kit unlocked package. Make callout requests to AWS from Salesforce after a receipt is uploaded. The AnalyzeExpense API then scans the receipts to analyze and interpret the information from the receipt.

1. From Setup, in the Quick Find box, enter *Named*, and then select **Named Credentials**.
2. From the dropdown, select **New Legacy**.
3. Enter these details:
 - Label: Enter a name for the legacy named credential that's shown in your org.
 - Name: Enter a unique identifier that's used to refer to this legacy-named credential from callout definitions and through the API.
 - URL: Enter `https://textract.[REGION].amazonaws.com`. For example, `https://textract.ap-south-1.amazonaws.com/`.
4. Select **Named Principal** as the identity type.
5. Select **AWS Signature Version 4** as the authentication protocol, and enter these details from the CSV file that's downloaded from AWS:
 - AWS Access Key ID: Enter the access key ID. For example, `AKIAQO45CVNMD4NEU3VB`.
 - Access Secret: Enter the secret access key. For example, `51rC7gCfPF+RfF8GWxZnjBw++ACT3ESm5BYwH+jS`
 - AWS Region: Enter the region that you selected in AWS. For example, `ap-south-1`.
 - AWS Service: Enter `textract` as the AWS service.
6. Save your changes.

EDITIONS

Available in: all editions that have Loyalty Management and Receipt Scanning enabled

USER PERMISSIONS

To view named credentials:

- View Setup and Configuration

To create, edit, or delete named credentials:

- Manage Named Credentials OR Customize Applications

Create an External Client App

Create an external client app for Amazon Web Services (AWS) to connect with the Loyalty Management Receipt Scanner Kit unlocked package. This external client app enables secure OAuth authentication so that the Expense Analysis API runs after the receipts are uploaded and creates transaction journals for the submitted receipts.

1. From Setup, in the Quick Find box, enter *certificate*, and then select **Certificate and Key Management**.
2. Under Certificates, click **Create Self-Signed Certificate**.
3. Enter a label and unique name for the certificate.
4. Save the new certificate.
5. Click **Download Certificate**.
6. Create an [External Client App](#).
7. Under OAuth Settings, enable **Enable OAuth**.
8. In the Callback URL, enter

`https://login.salesforce.com/services/oauth2/token/callback`.

9. Select **Use digital signatures**.
10. Click **Choose file**, and upload the certificate that you downloaded.
11. In the Selected OAuth Scopes section, move the **Manage user data via APIs (api)** and **Perform requests at any time (refresh_token, offline_access)** values to Selected OAuth Scopes.
12. Save the external client app.
13. From Setup, in the Quick Find box, enter *External Client Apps Manager*, and then select **External Client Apps Manager**.
14. From the actions list for the external client app, select **Edit Policies**.
15. Under OAuth Policies, from the Permitted Users dropdown, select **Admin approved users are Pre-authorized** as the permitted users.
16. From Timeout Value, select **24 hours**.
17. Save your changes. After you turn on this option, manage access using [profiles and permission sets](#). Select the profile or permission set that should have access by editing the external client app policies or the OAuth policies.
18. In the Quick Find box, enter *custom*, and then select **Custom Settings**.
19. For the Manage Authenticate Setting whose namespace prefix is ReceiptScanner, select **Manage**, and then select **New**.
20. Under Manage Authenticate Setting Information, enter these values.
 - Name: Enter *Manage Authentication*
 - Audience: Based on your environment, enter one of the values: *`https://test.salesforce.com/`* or *`https://login.salesforce.com/`*
 - Certificate: Enter the certificate name, without the extension
 - Consumer Key: Enter the consumer key. To get the consumer key, from the App Manager setup, for the Manage Authentication app, select **View**. Under API (Enable OAuth Settings), click **Manage Consumer Details**, enter your credentials, and then copy the consumer key.
 - Subscriber: Enter the username of the Salesforce admin.

EDITIONS

Available in: all editions that have Loyalty Management enabled

USER PERMISSIONS

To create external client apps:

- Customize Application
- AND
- Modify All Data OR
Manage External Client Apps

- Token Endpoint: Based on your environment, enter one of the values:
`https://test.salesforce.com/services/oauth2/token` or
`https://login.salesforce.com/services/oauth2/token`

21. Save your changes.

Add a Loyalty Program Name

The Loyalty Management Receipt Scanner Kit unlocked package has a custom label to store the name of your loyalty program.

 **Note:** If you installed the Loyalty Management Receipt Scanner Kit unlocked package in a B2C Retail Industry Template org, then the loyalty program name is already set up.

1. In the Quick Find box, enter `custom_label`, and then select **Custom Label**.
2. Next to Loyalty Program Name, click **Edit**.
3. Enter your loyalty program name.
4. Save your changes.

Add API Endpoint Details in Custom Setting

The Loyalty Management Receipt Scanner Kit unlocked package has a custom setting to store the API Gateway endpoint URL and bucket details from your Amazon Web Services (AWS) account.

1. In the Quick Find box, enter `custom_settings`, and then select **Custom Settings**.
2. Next to API Endpoint Details, click **Manage**.
3. Click **New**.
4. Enter these values.
 - API Gateway Endpoint: From your AWS Account, enter the Invoke URL until the stage name.
For example, if the Invoke URL is

`https://baznrtuyv3.execute-api.us-east-2.amazonaws.com/abc`, where `abc` is the stage name, enter the endpoint URL as `https://baznrtuyv3.execute-api.us-east-2.amazonaws.com`.

- Bucket Name: From your AWS account, enter the S3 bucket name.
 - Select Disable Duplicate Check if you want members to submit the same receipt multiple times.
5. Save your changes.
The location of the endpoint is automatically selected.

EDITIONS

Available in: all editions that have Loyalty Management enabled

USER PERMISSIONS

Create, edit, or delete custom labels:

- Customize Application

EDITIONS

Available in: all editions that have Loyalty Management and Receipt Scanning enabled

USER PERMISSIONS

To manage, create, edit, and delete custom settings:

- Customize Application

Create a Remote Site Setting

Before any Apex callout can call the API Gateway endpoint, register the API Gateway endpoint on the Remote Site Settings page. Remote site settings authorize your org to access external websites.

1. In the Quick Find box, enter *Remote*, and then select **Remote Site Settings**.
2. Click **New Remote Site**.
3. Enter *AWS Gateway Remote Site* as the remote site name.
4. Enter the Invoke URL of your Amazon Web Services account.
For example, a sample Invoke URL looks like this:
`https://*****.execute-api.us-east-2.amazonaws.com`
5. Save your changes.

EDITIONS

Available in: all editions that have Loyalty Management and Receipt Scanning enabled

USER PERMISSIONS

To configure remote settings:

- Customize Application or Modify All Data

Enable Receipt Scanning

Enable receipt scanning in the org where the Loyalty Management Receipt Scanner Kit unlocked package is installed. Members can then use their mobile device to upload receipts and get loyalty points.

1. From Setup, in the Quick Find box, enter *Lightning*, and then select **Lightning App Builder**.
2. Next to the Receipt Scanning app, click **Edit**.
3. From the Setup menu, click **Edit Page**.
4. Click **Activation**.
5. To change the icon, click **Change**, and then assign another logo for Receipt Scanning.
6. Click the Lightning Experience tab, and then click **Loyalty Management**.
7. In the Loyalty Management section, drag the Receipt Scanning app based on where you want the app to appear.
8. Save your changes.
9. From the App Launcher, enter *Receipt*, and then select **Receipt Scanning**.
10. Under Named Credential, click **Assign Named Credential**.
11. Select the named credential that you created for the Loyalty Management Receipt Scanning Kit unlocked package to connect with AWS.
12. Read and agree to the terms and conditions, and then save your changes.
13. Turn on Receipt Scanning.
14. Enter the confidence value at which the receipt is eligible for submission.
15. To show ineligible items for a member after a receipt is scanned and analyzed, turn on Ineligible Receipt Items.
16. Under Required Receipt Fields, select the fields that you want to include in a scanned receipt.
Select the Receipt Number field if you want to uniquely identify each receipt that's uploaded.
17. Under Eligible Item Identifier Fields, select the fields that identify the eligible items from a receipt.
18. Save your changes.

EDITIONS

Available in: all editions that have Loyalty Management and Receipt Scanning enabled

USER PERMISSIONS

To create and save Lightning Pages in the Lightning App Builder:

- Customize Application

To view Lightning Pages in the Lightning App Builder:

- View Setup and Configuration



Example: If you select Receipt Number as a required receipt field, the members can upload only the receipts that contain a receipt number. If you select Product Name as an eligible item identifier field, you earn loyalty points only for the items with matching product names in the Products object.

Receipt Scan and Upload on iOS and Android

After you set up the Loyalty Management Receipt Scanner Kit unlocked package and integrate the Loyalty Management org with an Optical Character Recognition (OCR) software, members can upload and scan receipts by using the loyalty program's mobile app. Members are rewarded for with loyalty points for eligible receipt line items.

Members can use the loyalty program's mobile app to upload purchase receipts from third-party partner stores and sites. To get the latest mobile app SDK, developers can clone and import the latest Loyalty Management Mobile SDKs for iOS and Android.

Considerations Before You Upload a Receipt

Members can upload receipts of products purchased at third-party stores, and get rewards for eligible purchases.

Upload and Submit a Receipt

Members can upload receipts from their third-party store purchases that are less than 5 MB, and get loyalty points for eligible receipt line items. After the receipt is scanned, members can review the line items detected by the Expense Analysis API and then submit the receipt.

View Submitted Receipts

Loyalty program members can view the submitted receipts and line items from their loyalty management mobile app after uploading the receipts, and view the loyalty points credit for a processed receipt.

Considerations Before You Upload a Receipt

Members can upload receipts of products purchased at third-party stores, and get rewards for eligible purchases.

To ensure the receipt upload and scan process is seamless, check these conditions before you upload a scanned receipt.

- The receipt is legible, and not torn or crumpled.
- The size of the receipt file is less than 5 MB.
- The receipt is valid and accurate, and contains this data:
 - purchase date
 - receipt number
 - store name and address,
 - product information
 - quantity purchased
 - price of a unit
 - total price
 - any other information that's mandatory to process the receipt.
- The receipt isn't duplicated.
- The receipt's date format matches the store's date format that's specified in the Accepted Date Format custom metadata type.
The custom metadata type supports these date formats:

EDITIONS

The Loyalty Management mobile app is available in: all editions that have Loyalty Management and Receipt Scanning enabled

- mm-dd-yyyy
- dd-mm-yyyy
- yyyy-mm-dd
- mm.dd.yyyy
- dd.mm.yyyy
- yyyy.mm.dd
- mm/dd/yyyy
- dd/mm/yyyy
- yyyy/mm/dd

Upload and Submit a Receipt

Members can upload receipts from their third-party store purchases that are less than 5 MB, and get loyalty points for eligible receipt line items. After the receipt is scanned, members can review the line items detected by the Expense Analysis API and then submit the receipt.

1. On the home screen of the iOS or the Android Loyalty Management mobile app, tap .
2. Tap **New**, and provide access to your photo library.
3. Tap the camera, and click the photo of a receipt or tap the gallery icon and upload a receipt from your mobile device.
4. Tap **Upload**.
The receipt is processed and the results show eligible and ineligible items from the receipt.
5. Select an option:
 - If the processed receipt results are accurate, tap **Submit**, and then tap **Done**.
The receipt is submitted and the screen shows the number of loyalty points credited for the receipt.
 - To rescan or upload another receipt, tap **Try Again**.
 - If your receipt doesn't list all eligible items, tap **Request a Manual Review**.
Enter a reason for the manual request, and tap **Submit**.

EDITIONS

The Loyalty Management mobile app is available in: all editions that have Loyalty Management and Receipt Scanning enabled

What Happens When Members Upload Receipts

Members can upload receipts of their third-party store purchases and after they upload the receipts, the Expense Analysis API scans the receipts and reviews the eligibility of the receipt.

What Happens When Members Submit Receipts

After members submit receipts, they can view the submitted receipts and the line items for which they are rewarded from their loyalty management mobile app.

What Happens When Members Submit Receipts for Manual Review

If the eligible items of a receipt don't show all items or if there's an incorrect evaluation of loyalty points for the receipt, members can submit the receipt for a manual review.

What Happens When Members Upload Receipts

Members can upload receipts of their third-party store purchases and after they upload the receipts, the Expense Analysis API scans the receipts and reviews the eligibility of the receipt.

Here's what happens in the Loyalty Management Receipt Scanner Kit unlocked package org when members upload their receipts.

- The ReceiptUpload Apex class runs the Receipt Upload API to upload the receipt.
- Then, the ReceiptScanningControllerUtils Apex class runs the Expense Analysis API to scan receipts uploaded by members.
- The ExpenseAnalysis class runs the Expense Analysis API to analyze the receipt.
- The API scans the receipts and provides a response, which is stored in the ReceiptScanningResponseWrapper class. The response consists of the store details, receipt ID, number, and date, and the detected line item details. The response also shows a confidence value, if applicable.

EDITIONS

Available in: all editions that have Loyalty Management and Receipt Scanning enabled

What Happens When Members Submit Receipts

After members submit receipts, they can view the submitted receipts and the line items for which they are rewarded from their loyalty management mobile app.

Here's what happens in your Loyalty Management Receipt Scanner Kit unlocked package org after members submit their receipts.

- The ReceiptStatusUpdate Apex class runs the Receipt Status Updation API, to update the status of the receipt to Processed.
- The ProcessTransactionJournal Apex class runs the [Transaction Journal Execution Business API](#), which creates and processes transaction journals.
- The RulesEngineResponseWrapper class deserializes the response from the Rules Engine API that includes transaction journal details. The response provides loyalty points, which are shown in the mobile app.
- The Receipt Status Updation API creates and processes receipt records for each line item.

EDITIONS

The Loyalty Management mobile app is available in: all editions that have Loyalty Management and Receipt Scanning enabled

What Happens When Members Submit Receipts for Manual Review

If the eligible items of a receipt don't show all items or if there's an incorrect evaluation of loyalty points for the receipt, members can submit the receipt for a manual review.

Here's what happens when the receipt is submitted for manual review.

- The ReceiptStatusUpdate Apex class runs the Receipt Status Updation API, to update the status of the receipt to Manual Review.
- The ReceiptUpdateTrigger updates the receipt record.

To view the receipt in the Loyalty Management app, loyalty program managers go to the App Launcher, and find and select **Receipts**. After the admin or loyalty program manager manually completes the review, the Receipts screen on the mobile app shows the status of the receipt. If the receipt is manually approved, the member gets loyalty points for the eligible items.

EDITIONS

The Loyalty Management mobile app is available in: all editions that have Loyalty Management and Receipt Scanning enabled



Tip: Admins can also add the Receipts related list on the Loyalty Program Member record. This allows program managers to review all the receipts that are pending review for a loyalty program member.

View Submitted Receipts

Loyalty program members can view the submitted receipts and line items from their loyalty management mobile app after uploading the receipts, and view the loyalty points credit for a processed receipt.

1. On the home screen of the iOS or the Android Loyalty Management mobile app, tap .
2. On the Receipts screen, view these details:
 - Receipt number
 - Receipt date
 - Receipt total
 - Loyalty points credited for a processed receipt
 - Receipt status for an unprocessed receipt
3. Tap the receipt and view the details of each eligible and ineligible detected line item including:
 - Item name
 - Quantity purchased
 - Unit price
 - Total price
 - Receipt image
4. To download the receipt image, tap **Receipt Image**, and then tap **Download Original Receipt**.

EDITIONS

The Loyalty Management mobile app is available in: all editions that have Loyalty Management and Receipt Scanning enabled

Loyalty Management Mobile SDK and Sample App Support

Loyalty Management Mobile SDK is an open-source product. If you have any questions or feedback on the mobile SDK, post your questions or feedback on the Salesforce Stack Exchange site and use the `loyaltymobile` hashtag. If you find a bug with either the SDK or the sample app, raise a ticket on GitHub.

Upgrade Loyalty Management Mobile SDK for iOS

For a major upgrade, clone and import the latest version of the SDK. For minor or incremental patch updates, update the Loyalty Management Mobile SDK to the latest package version.

- To update the Loyalty Management Mobile SDK for iOS to the latest package version, in Xcode, click **File > Swift Packages > Update to Latest Package Versions**.
- For a major upgrade of Loyalty Management Mobile SDK for iOS, clone and import the latest version of the Loyalty Management Mobile SDK for iOS.

EDITIONS

Available in: all editions that have Loyalty Management enabled

SEE ALSO:

[Clone and Import Loyalty Management Mobile SDK for iOS](#)

[Clone and Import Loyalty Management Mobile SDK for iOS](#)

CHAPTER 12 Gamification Mobile SDK

In this chapter ...

- [Supported Versions of Tools and Components](#)
- [Gamification Mobile SDK for iOS](#)
- [Gamification Mobile SDK for Android](#)

Enhance brand engagement by providing gamification features on Android and iOS mobile devices. Use the Gamification mobile Software Development Kit (SDK) for Android and iOS to build custom mobile applications with unique game reward experiences. Or, use the SDK as an external dependency in your existing Loyalty Management Mobile SDK, and extend the `MyNTORewards` sample app to include games that a loyalty member can play to win rewards.

Watch this video to understand how to design and run games.



[Watch a video](#)

If you aren't able to watch the video in full screen, open the video in a new tab: [📺 Gamify Member Engagement with Loyalty Management](#).

Supported Versions of Tools and Components

The Gamification Mobile SDK works on iOS 15.0+ and Android 8.0+ versions. Install the tools and components required to set up the Gamification Mobile SDK.

EDITIONS

Available in: all editions that have Loyalty Management enabled

iOS

Tool or Component	Supported Version	Installation Details
Swift Version	5.7+	Installed by Xcode
Xcode	14.0+	Install from the App Store
iOS SDK	15.0+	Installed by Xcode
Swift Package Manager	5.7+	Included in Swift

Android

Tool or Component	Supported Version	Installation Details
Android Studio Electric Eel	2022.1.1+	Install from the Android Developer site
Android SDK and APIs	8.0+ (API level 26+)	Install from the Android Developer site
Android SDK Tools	8.0+	Install from the Android Developer site
Android Virtual Device	8.0+	Install from Android Studio
Kotlin	1.8.0+	Installed by Android Studio

Gamification Mobile SDK for iOS

Clone and import the Gamification Mobile iOS SDK to integrate the gamification feature in a mobile app.

EDITIONS

Available in: all editions that have Loyalty Management enabled

[Clone and Import Gamification Mobile SDK for iOS](#)

Clone the Gamification Mobile iOS SDK GitHub repository to copy the code base and revision history of the repository to your local computer. After you clone the repository, import Gamification Mobile iOS SDK in an Xcode project.

[Integrate Gamification Mobile iOS SDK into Loyalty Management Mobile iOS SDK](#)

Import the Gamification Mobile iOS SDK into the existing Loyalty Management Mobile iOS SDK by adding it as an external dependency.

[Gamification Mobile SDK iOS Classes](#)

The iOS classes organize the Gamification Mobile SDK's functionalities and data structures. When developing mobile apps using the SDK, these classes provide components, services, or features with which you can interact.

[Gamification Mobile SDK API Reference for iOS](#)

The Gamification Mobile iOS SDK interacts with the Salesforce Gamification APIs, and retrieves games assigned to a loyalty member, and the rewards won by the member after playing the game.

[Game Zone UI Customization for iOS](#)

After you build the Gamification Mobile SDK for iOS, you can customize the game zone UI components.

Clone and Import Gamification Mobile SDK for iOS

Clone the Gamification Mobile iOS SDK GitHub repository to copy the code base and revision history of the repository to your local computer. After you clone the repository, import Gamification Mobile iOS SDK in an Xcode project.

1. Go to the [Gamification Mobile SDK iOS repository](#).
2. To copy the repository URL to your clipboard, click **Code**, and then click .
3. Open Terminal, and go to the directory where you want to clone the repository.
4. Enter `git clone https://github.com/loyaltysampleapp/GamificationMobileSDK-iOS`.
5. Press Enter and wait for the repository to clone.
6. Open the project where you want to import `GamificationMobileSDK-iOS` in Xcode.
7. Go to **File > Swift Packages > Add Package Dependency**.
8. Click **Add Local**, and then select the local `GamificationMobileSDK-iOS` folder.
9. Select a version, and click **Add Package** and on the confirmation window, click **Add Package**.
10. To import the `GamificationMobileSDK-iOS` module into your Swift file, open the swift file where you want to use `GamificationMobileSDK-iOS`, and to the first line of code, add:

```
import GamificationMobileSDK-iOS
```

You can now use the classes, structures, and protocols in the `GamificationMobileSDK-iOS` module to manage and customize the gamification features in your mobile application.

EDITIONS

Available in: all editions that have Loyalty Management enabled

Integrate Gamification Mobile iOS SDK into Loyalty Management Mobile iOS SDK

Import the Gamification Mobile iOS SDK into the existing Loyalty Management Mobile iOS SDK by adding it as an external dependency.

1. Open the `LoyaltyMobileSDK-iOS` project in Xcode.
2. Go to **File > Swift Packages > Add Package Dependency**.
3. Enter the repository

URL: `https://github.com/loyaltysampleapp/GamificationMobileSDK-iOS`, or if you've cloned the repository, click **Add Local**, and then select the local `GamificationMobileSDK-iOS` folder.

4. Select a version, and click **Add Package** and on the confirmation window, click **Add Package**.
5. To import the `GamificationMobileSDK-iOS` module into your Swift file, open the swift file where you want to use `GamificationMobileSDK-iOS`, and to the first line of code, add:

```
import GamificationMobileSDK-iOS
```

EDITIONS

Available in: all editions that have Loyalty Management enabled

You can now use the classes, structures, and protocols in the `GamificationMobileSDK-iOS` module to extend and manage the gamification features in the `MyNTORewards` sample app.

Gamification Mobile SDK iOS Classes

The iOS classes organize the Gamification Mobile SDK’s functionalities and data structures. When developing mobile apps using the SDK, these classes provide components, services, or features with which you can interact.

EDITIONS

Available in: all editions that have Loyalty Management enabled

[APIManager](#)

The `APIManager` class manages requests related to gamification, and invokes the gamification REST APIs. Use this class to interact with the Salesforce gamification APIs and retrieve member games, and game rewards.

[GamificationForceClient](#)

The `GamificationForceClient` class handles network requests using the `GamificationForceAuthenticator` protocol.

[GamificationForceAuthenticator](#)

The `GamificationForceAuthenticator` protocol defines the necessary methods for handling access tokens in the Salesforce API.

APIManager

The `APIManager` class manages requests related to gamification, and invokes the gamification REST APIs. Use this class to interact with the Salesforce gamification APIs and retrieve member games, and game rewards.

The Gamification Mobile iOS SDK requires a valid access token to interact with the Salesforce gamification APIs. To provide a valid access token, pass an instance of `GamificationForceAuthenticator` protocol in your `APIManager` class at the time of initialization. The `APIManager` class manages authentication using the provided instance of `GamificationForceAuthenticator`.

Use this initializer to initialize the `GamificationRemoteRepository` class.

```
public init(auth: GamificationForceAuthenticator, instanceURL: String, forceClient: GamificationForceClient) {
    self.auth = auth
    self.instanceURL = instanceURL
    self.forceClient = forceClient
}
```

Parameter	Type	Description
auth	GamificationForceAuthenticator	An instance of the GamificationForceAuthenticator protocol.
instanceUrl	String	The URL of your Salesforce org.
forceClient	GamificationForceClient	An instance of the GamificationForceClient class.

Methods

getGames

Retrieve games for the specified participant ID. This method returns an instance of the `GameModel` object.

```
public func getGames(
    participantId: String,
    version: String = APIVersion.defaultVersion,
    devMode: Bool = false,
    mockFileName: String = "GetGames_Success"
) async throws -> GameModel
```

Parameter	Type	Description
participantId	String	The ID of a loyalty program member, an account, or a contact whose eligible games are to be retrieved.
version	String	The API version.
mockFileName	String	The name of the file that has the sample response.
devMode	Boolean	Indicates whether the method is run in the developer mode (<code>true</code>) or not (<code>false</code>).

getGameReward

Retrieve the details of the reward that's provided to a participant after the game play. This method returns an instance of the `PlayGameModel` object.

```
public func playGame(
    gameParticipantRewardId: String,
    version: String = APIVersion.defaultVersion,
    devMode: Bool = false,
    mockFileName: String = "PlayGame_Success"
) async throws -> PlayGameModel
```

Parameter	Type	Description
gameParticipantRewardId	String	The ID of the participant's game reward won by the participant whose details are to be fetched.
version	String	The API version.
mockFileName	String	The name of the file that has the sample response.
devMode	Boolean	Indicates whether the method is run in the developer mode (<code>true</code>) or not (<code>false</code>).

GamificationForceClient

The `GamificationForceClient` class handles network requests using the `GamificationForceAuthenticator` protocol.

This class uses a `NetworkManager` instance to handle network requests and data processing.

Use this initializer to initialize the `GamificationForceClient` class.

```
public init(auth: GamificationForceAuthenticator, forceNetworkManager: NetworkManagerProtocol = NetworkManager.shared)
```

Parameter	Type	Description
auth	GamificationForceAuthenticator	An instance of the <code>GamificationForceAuthenticator</code> protocol.
forceNetworkManager	NetworkManagerProtocol	A <code>NetworkManager</code> instance.

Methods

fetchLocalJson

Returns a decoded JSON response result from the given local JSON file.

```
public func fetchLocalJson<T: Decodable>(type: T.Type, file: String, bundle: Bundle = Bundle.publicModule)
```

Parameter	Type	Description
type	T.Type	The model to be used by the JSON decoder.
file	String	The file name of a local JSON file.
bundle	Bundle	The app bundle name.

fetch

Returns a decoded JSON response result.

```
public func fetch<T: Decodable>(type: T.Type, with request: URLRequest, urlSession: URLSession = .shared)
```

Parameter	Type	Description
type	T.Type	The model to be used by the JSON decoder.
request	URLRequest	The URL request to be executed by the URL session.
urlSession	URLSession	An instance of <code>URLSession</code> to execute the request.

GamificationForceAuthenticator

The `GamificationForceAuthenticator` protocol defines the necessary methods for handling access tokens in the Salesforce API.

Methods

`getAccessToken()`

Returns a valid access token as a string if available, otherwise null.

```
func getAccessToken()
```

`grantAccessToken`

Returns a refreshed or newly granted access token as a string.

```
func grantAccessToken()
```

Gamification Mobile SDK API Reference for iOS

The Gamification Mobile iOS SDK interacts with the Salesforce Gamification APIs, and retrieves games assigned to a loyalty member, and the rewards won by the member after playing the game.

[Assigned Games](#)

Get a list of eligible, expired, and played games, such as Spin the Wheel, or Scratch card, for a loyalty program member, an account, or a contact.

[Game reward](#)

Get the details of the reward that's provided to a participant after the participant plays a Spin the Wheel, or Scratch Card game.

EDITIONS

Available in: all editions that have Loyalty Management enabled.

Assigned Games

Get a list of eligible, expired, and played games, such as Spin the Wheel, or Scratch card, for a loyalty program member, an account, or a contact.

Resource

```
getGames(participantId:version:devMode:mockFileName:)
```

Signature

```
public func getGames(  
    participantId: String,  
    version: String = APIVersion.defaultVersion,  
    devMode: Bool = false,  
    mockFileName: String = "GetGames_Success"  
) async throws -> GameModel
```

Input Parameters

Parameter	Type	Description	Required or Optional
devMode	Boolean	Indicates whether the method is run in the developer mode (<code>true</code>) or not (<code>false</code>).	Optional
mockFileName	String	The name of the file that has the sample response.	Optional
participantId	String	The ID of a loyalty program member, an account, or a contact whose eligible games are to be retrieved.	Required
version	String	Version of the API. The request body is available in version 60.0 and later.	Optional

Return Value

A `GameModel` instance.

```
public struct GameModel : Codable
```

Sample Output

```
{
  "message":null,
  "gameDefinitions":[
    {
      "description":"Play Spin The Wheel to get amazing rewards",
      "endDate":"2024-07-31T19:00:00.000Z",
      "gameRewards":[
        {
          "color":"FF5733",
          "description":"Get 10% Off Voucher",
          "imageUrl":null,
          "name":"10 % Off",
          "rewardType":"Voucher",
          "rewardValue":null
        },
        {
          "color":"000000",
          "description":"Loyalty Points",
          "imageUrl":null,
          "name":"1000 Reward Points",
          "rewardType":"LoyaltyPoints",
          "rewardValue":"100.0"
        }
      ]
    }
  ]
}
```

```

        "color":"762E2E",
        "description":"Better Luck Next Time",
        "imageUrl":null,
        "name":"No Reward",
        "rewardType":"NoReward",
        "rewardValue":null
    }
],
"participantGameRewards":[
    {
        "expirationDate":"2024-01-04T20:00:00.000Z",
        "gameParticipantRewardId":"3PRSB0000000hjJ4AQ",
        "gameRewardId":null,
        "issuedRewardReference":null,
        "sourceActivity":null,
        "status":"YetToReward"
    }
],
"name":"Spin The Wheel",
"startDate":"2023-10-01T19:00:00.000Z",
"timeoutDuration":10,
"type":"SpintheWheel"
},
{
    "description":"Play Scratch card to get amazing rewards",
    "endDate":"2024-07-31T19:00:00.000Z",
    "gameRewards":[
        {
            "color":"",
            "description":"Get 10% Off Voucher",
            "imageUrl":null,
            "name":"10 % Off",
            "rewardType":"Voucher",
            "rewardValue":null
        },
        {
            "color":"",
            "description":"Better Luck Next Time",
            "imageUrl":null,
            "name":"No Reward",
            "rewardType":"NoReward",
            "rewardValue":null
        }
    ]
},
"participantGameRewards":[
    {
        "expirationDate":"2024-01-04T20:00:00.000Z",
        "gameParticipantRewardId":"2ABSB0000000hjJ3AM",
        "gameRewardId":"2GRSB0000000Caf4AB",
        "issuedRewardReference":"01VSB000000032j2AA",
        "sourceActivity":"0c85i00000009c6D",
        "status":"Rewarded"
    }
],

```

```

        "name": "Scratch Card",
        "startDate": "2023-10-01T19:00:00.000Z",
        "timeoutDuration": 20,
        "type": "Scratchcard"
    }
],
"status": true
}

```

Output Parameters

Property Name	Type	Description	Filter Group and Version	Available Version
message	String	Message that provides the reason why a list of games weren't found for the participant.	Small, 60.0	60.0
gameDefinitions	GameDefinition[]	List of game definitions associated with the games assigned to the member. A game definition has the details of a game, including: - The list of possible rewards for the game. - The participant game reward records that contain details of the game reward the participant has won for each game play, or can win after playing the game.	Small, 60.0	60.0
status	Boolean	Indicates whether the request succeeded (<code>true</code>) or failed (<code>false</code>).	Small, 60.0	60.0

Table 1: GameDefinition

Property Name	Type	Description	Filter Group and Version	Available Version
description	String	Description of the game.	Small, 60.0	60.0
endDate	String	Date until which participants are eligible to play the game.	Small, 60.0	60.0
gameDefinitionId	String	ID of the game definition associated with the game.	Small, 60.0	60.0
gameRewards	GameReward []	List of possible rewards for a game.	Small, 60.0	60.0
name	String	Name of the game definition.	Small, 60.0	60.0
participantGameRewards	ParticipantGameReward[]	List of participant game reward records that contain the details of the game reward the	Small, 60.0	60.0

Property Name	Type	Description	Filter Group and Version	Available Version
		participant has won for each game play, or can win after playing the game.		
startDate	String	Date from which participants are eligible to play the game.	Small, 60.0	60.0
timeoutDuration	Integer	Duration of the game in seconds before the game session times out.	Small, 60.0	60.0
type	String	Type of the game. Possible values are: - Scratchcard - SpintheWheel	Small, 60.0	60.0

Table 2: GameReward

Property Name	Type	Description	Filter Group and Version	Available Version
color	String	Color of the segment in the Spin the Wheel game that represents the game reward.	Small, 60.0	60.0
description	String	Description of the game reward.	Small, 60.0	60.0
imageUrl	String	Image URL of the segment in the Spin the Wheel game that represents the game reward.	Small, 60.0	60.0
name	String	Name of the game reward.	Small, 60.0	60.0
gameRewardId	String	ID of the game reward.	Small, 60.0	60.0
rewardType	String	Type of the game reward. Possible values are: - CustomReward - LoyaltyPoints - NoReward - Raffle - Voucher	Small, 60.0	60.0
rewardValue	String	Value of the voucher, or the number of points to credit as a game reward to participants.	Small, 60.0	60.0

Table 3: ParticipantGameReward

Property Name	Type	Description	Filter Group and Version	Available Version
expirationDate	String	Date until which the game reward is valid.	Small, 60.0	60.0
gameParticipantRewardId	String	ID of the participant game reward object.	Small, 60.0	60.0
gameRewardId	String	ID of the game reward won. This field is null for a game yet to be played.	Small, 60.0	60.0
issuedRewardReference	String	Voucher ID, or the transaction journal ID associated with the voucher, or the loyalty points rewarded to the participant. This field is null for a game yet to be played.	Small, 60.0	60.0
sourceActivity	String	ID of the promotion, or the transaction journal that is associated with the activity because of which the participant was eligible to play the game. Examples of source activities include purchasing a product, and reviewing on social media. This field is null for a game yet to be played.	Small, 60.0	60.0
status	String	Status of the reward. Possible values are: - Expired - NoReward - Rewarded - YetToReward	Small, 60.0	60.0

Game reward

Get the details of the reward that's provided to a participant after the participant plays a Spin the Wheel, or Scratch Card game.

Resource

```
playGame(gameParticipantRewardId:version:devMode:mockFileName:)
```

Signature

```
public func playGame(
    gameParticipantRewardId: String,
    version: String = APIVersion.defaultVersion,
    devMode: Bool = false,
    mockFileName: String = "PlayGame_Success"
) async throws -> PlayGameModel
```

Input Parameters

Parameter	Type	Description	Required or Optional
devMode	Boolean	Indicates whether the method is run in the developer mode (<code>true</code>) or not (<code>false</code>).	Optional
mockFileName	String	The name of the file that has the sample response.	Optional
gameParticipantRewardId	String	The ID of the participant's game reward won by the participant whose details are to be fetched.	Required
version	String	Version of the API. The request body is available in version 60.0 and later.	Optional

Return Value

A `PlayGameModel` instance.

```
public struct PlayGameModel: Codable
```

Sample Output

```
{
  "message":null,
  "gameReward":[
    {
      "color":"01CD6C",
      "description":"Loyalty Points",
      "gameRewardId":"3GRSB0000000Caf4AE",
      "imageUrl":null,
      "issuedRewardReference":"01VSB000000032j2AA",
      "name":"1000 Reward Points",
      "rewardType":"LoyaltyPoints",
      "rewardValue":"100.0"
    }
  ],
  "status":true
}
```

Output Parameters

Property Name	Type	Description	Filter Group and Version	Available Version
message	String	Error message that provides the reason why the details of the participant reward weren't found.	Small, 60.0	60.0
gameReward	PlayGameReward []	Details of the game reward won by the participant.	Small, 60.0	60.0
status	Boolean	Indicates whether the request succeeded (<code>true</code>) or failed (<code>false</code>).	Small, 60.0	60.0

Table 4: PlayGameReward

Property Name	Type	Description	Filter Group and Version	Available Version
color	String	Color of the segment in the Spin the Wheel game that represents the game reward.	Small, 60.0	60.0
description	String	Description of the game reward.	Small, 60.0	60.0
gameRewardId	String	ID of the game reward won by the participant.	Small, 60.0	60.0
imageUrl	String	Image URL of the segment in the Spin the Wheel game that represents the game reward.	Small, 60.0	60.0
issuedRewardReference	String	Voucher ID, or the transaction journal ID associated with the voucher, or the loyalty points rewarded to the participant.	Small, 60.0	60.0
name	String	Name of the game reward.	Small, 60.0	60.0
rewardType	String	Type of the game reward. Possible values are: • CustomReward • LoyaltyPoints • NoReward • Raffle • Voucher	Small, 60.0	60.0
rewardValue	String	Value of the voucher, or the number of points to credit as a game reward to participants.	Small, 60.0	60.0

Game Zone UI Customization for iOS

After you build the Gamification Mobile SDK for iOS, you can customize the game zone UI components.

The game zone UI components are available in the

`LoyaltyMobileSDK-iOS/SampleApps/MyNTORewards/MyNTORewards/GameZone` package.

Spinner Customization

Customize the spinner components, such as the wheel size, frame width and color, wheel segments, spin button, and more in the `FortuneWheelView.swift` file. Here are all the parameters you can modify.

```
VStack {
    HStack {
        Button {
            // Invalidating the timer to avoid unintended navigation because
of the timer

            timer?.invalidate()
            isSpinning ? nil : dismiss()
        } label: {
            Image("ic-backarrow")
        }
        .padding(.leading, 20)
        .padding(.bottom, 10)
        .frame(width: 30, height: 30)

        Spacer()
    }
    ZStack {
        Color.theme.accent
        VStack {
            VStack(spacing: 10) {
                Text(gameDefinitionModel?.name ??
StringConstants.Gamification.spinaWheelHeaderLabel)
                    .font(.gameHeaderTitle)

                Text(gameDefinitionModel?.description ??
StringConstants.Gamification.spinaWheelSubHeaderLabel)
                    .font(.gameHeaderSubTitle)
            }
            .padding(30)

            Spacer()
            ZStack {
                // Fortune Wheel Segments
                ZStack {
                    if let colors: [Color] = viewModel.getWheelColors(gameModel:
gameDefinitionModel),
                        let labels: [LocalizedStringKey] =
gameDefinitionModel?.gameRewards.map({LocalizedStringKey($0.name)}) {
                        ForEach(0..

```

```

Double(index + 1)) - 90.0

                WheelSegment(startAngle: startAngle,
                              endAngle: endAngle,
                              color: colors[index],
                              label: labels[index])
            }
        }
    }
    .overlay {
        ZStack {
            Circle()
                .strokeBorder(Color.theme.fortuneWheelStrokeColor,
                              lineWidth: 5)
                .frame(width: 316, height: 316)
            Circle()
                .strokeBorder(Color.theme.fortuneWheelSecondaryStrokeColor, lineWidth: 1)
                .frame(width: 307, height: 307)
        }
        .rotationEffect(Angle(degrees: rotationAngle))
        .animation(isSpinning ? Animation.easeOut(duration:
gameDefinitionModel?.timeoutDuration ?? 15)
            .delay(0)
            .repeatForever(autoreverses: false) : .default, value:
isSpinning)

        // Triangle Arrow Indicator
        ZStack {
            Triangle()
                .fill(Color.white)
                .frame(width: 70, height: 70) // Increase these
dimensions by the stroke width
                .offset(y: -40)

            Triangle()
                .fill(Color.theme.wheelIndicatorBackground)
                .frame(width: 60, height: 60)
                .offset(y: -35)
        }

        // Spin Button
        Button {
            playGame()
        } label: {
            userStartedSpinning ?

Text(StringConstants.Gamification.tapSpinButtonLabel).foregroundColor(Color.theme.wheelIndicatorBackground)
:

Text(StringConstants.Gamification.tapSpinButtonLabel).foregroundColor(.white)

        }
        .frame(width: 70, height: 70)

```

```

        .background(Color.theme.wheelIndicatorBackground)
        .clipShape(Circle())
        .overlay(Circle().stroke(Color.white, lineWidth: 5))
        .disabled(userStartedSpinning) // Disable the button when
spinning

    }
    .frame(width: 300, height: 300)

    Spacer()

    VStack(spacing: 20) {
        Text((StringConstants.Gamification.tapSpinaWheeltoPlayLabel))

        .font(.gameDescTitle)
        Text(StringConstants.Gamification.spinaWheelBodyLabel)
        .font(.gameDescText)
        .multilineTextAlignment(.center)
        .frame(width: 258)
    }
    Spacer()
}
.foregroundColor(.white)
}
.cornerRadius(15, corners: [.topLeft, .topRight])
.edgesIgnoringSafeArea(.bottom)
.fullScreenCover(isPresented: $showAlertForError) {
    Spacer()
    ProcessingErrorView(message: "Oops! Something went wrong while processing
the request. Try again.")
    Spacer()
    Button {
        timer?.invalidate()
        isSpinning ? nil : dismiss()
    } label: {
        Text(StringConstants.Receipts.tryAgainButton)
        .frame(maxWidth: .infinity)
    }
    .buttonStyle(.borderedProminent)
    .longFlexibleButtonStyle()
}
}
}
}

```

For example, to change the spin wheel outer frame width and height, open the

LoyaltyMobileSDK-iOS/SampleApps/MyNTORewards/MyNTORewards/GameZone/Views/FortuneWheel/FortuneWheelView.swift file in Xcode, and replace `Circle() frame(width: 316, height: 316)` with `Circle() frame(width: 300, height: 300)`.

```

overlay {
    ZStack {
        Circle()
        .strokeBorder(Color.theme.fortuneWheelStrokeColor,
lineWidth: 5)
    }
}

```

```

                .frame(width: 316, height: 316)
                Circle()

            .strokeBorder(Color.theme.fortuneWheelSecondaryStrokeColor, lineWidth: 1)
                .frame(width: 307, height: 307)
            }
        }
    }
}

```

To change the spin button background color to blue, open the `LoyaltyMobileSDK-iOS/SampleApps/MyNTORewards/MyNTORewards/Misc/Color.swift` file, and inside the `ColorTheme` structure, replace `let wheelIndicatorBackground = Color("wheelIndicatorBackgroundColor") // #23475F` with `let wheelIndicatorBackground = Color.blue`.

Scratch Card Customization

You can customize the scratch card components, such as the box, canvas, wrapper text, and more in the `ScratchCanvasView.kt` file. Here are all the properties you can customize:

```

private var backButton: some View {
    HStack {
        Button {
            // Invalidating the timer to avoid unintended navigation because of the
timer
            timer?.invalidate()
            dismiss()
        } label: {
            Image("ic-backarrow")
        }
        .padding(.leading, 20)
        .padding(.bottom, 10)
        .frame(width: 30, height: 30)

        Spacer()
    }
}

private var titleView: some View {
    VStack(spacing: 10) {
        Text(gameDefinitionModel?.name ??
StringConstants.Gamification.scratchCardTitleLabel)
        .font(.gameHeaderTitle)
        Text(gameDefinitionModel?.description ??
StringConstants.Gamification.scratchCardSubTitleLabel)
        .font(.gameHeaderSubTitle)
    }
    .padding(30)
}

private var loadingView: some View {
    Text("Loading...")
        .foregroundColor(.white)
        .font(.system(size: 24))
        .fontWeight(.bold)
}

```



```

        .frame(width: cardSize.width, height: cardSize.height)
        .background(Color.theme.accent)
    }

    private var rewardText: some View {
        Text(playGameViewModel.playedGameRewards?.first?.name ?? "--")
            .font(.largeTitle)
            .bold()
            .foregroundColor(.white)
            .frame(width: cardSize.width, height: cardSize.height)
            .background(Color.theme.accent)
            .zIndex(finishedScratching ? 10 : 0)
    }

    private var scratchCardWrapperText: some View {
        Text(String(repeating: (StringConstants.Gamification.scratchCardLabel), count:
70))
            .font(Font.scratchText)
            .multilineTextAlignment(.center)
            .foregroundColor(Color.theme.scratchCardText)
            .rotationEffect(Angle(degrees: -45))
            .mask {
                Rectangle()
                    .frame(width: cardSize.width, height: cardSize.height)
                    .cornerRadius(10)
                    .opacity(finishedScratching ? 0 : 1)
            }
            .opacity(finishedScratching ? 0 : 1)
    }

    private var scratchCardContentView: some View {
        ZStack {
            // Purple background with postage stamp border
            DottedBorderRectangle(width: backgroundSize.width,
                                height: backgroundSize.height,
                                color: Color.theme.accent)

            // Grey scratch card
            Rectangle()
                .fill(Color.theme.scratchCardBackground)
                .frame(width: cardSize.width, height: cardSize.height)
                .cornerRadius(10)
                .opacity(finishedScratching ? 0 : 1)

            // Text overlay
            scratchCardWrapperText
        }
    }

    private var scratchCardGame: some View {
        ScratchCardGame(cursorSize: 30, cardSize: cardSize, gameModel: gameDefinitionModel,
onFinish: $finishedScratching) {
            scratchCardContentView
        } overlayView: {

```

```

        // Reward text
        switch playGameViewModel.state {
        case .loaded:
            rewardText
        default:
            loadingView
        }
    }
    .animation(.linear(duration: 0.5), value: playGameViewModel.state)
    .environmentObject(playGameViewModel)
}

private var descriptionView: some View {
    VStack(spacing: 20) {
        Text((StringConstants.Gamification.scratchCardBodyLabel))
            .font(.gameDescText)
            .multilineTextAlignment(.center)
            .frame(width: 258)
    }
}
}

```

For example, to change the angle of the scratch card wrapper text inside the gray box, open the `LoyaltyMobileSDK-iOS/SampleApps/MyNTORewards/MyNTORewards/GameZone/Views/ScratchCard/ScratchCardView.swift` file in Xcode, and replace `rotationEffect (Angle(degrees: -45))` with `rotationEffect (Angle(degrees: -25))`. To change the number of times the text appears inside the gray box, replace `Text (String (repeating: (StringConstants.Gamification.scratchCardLabel), count: 70))` with `Text (String (repeating: (StringConstants.Gamification.scratchCardLabel), count: 50))`.

```

private var scratchCardWrapperText: some View {
    Text (String (repeating: (StringConstants.Gamification.scratchCardLabel), count:
50))

    .font (Font.scratchText)
    .multilineTextAlignment (.center)
    .foregroundColor (Color.theme.scratchCardText)
    .rotationEffect (Angle (degrees: -25))
    .mask {
        Rectangle ()
            .frame (width: cardSize.width, height: cardSize.height)
            .cornerRadius (10)
            .opacity (finishedScratching ? 0 : 1)
    }
    .opacity (finishedScratching ? 0 : 1)
}

```

Gamification Mobile SDK for Android

Clone and import the Gamification Mobile Android SDK to integrate the gamification feature in a mobile app.

EDITIONS

Available in: all editions that have Loyalty Management enabled

Clone and Import Gamification Mobile SDK for Android

Clone the GitHub repository to copy the code base and revision history of the repository to your local computer. After you clone the mobile SDK for Android repository, import Gamification Mobile Android SDK in Android Studio to automatically download and manage external dependencies.

Integrate Gamification Mobile Android SDK into Loyalty Management Mobile Android SDK

To integrate the Gamification Mobile Android SDK into the existing Loyalty Management Mobile Android SDK, add it as an external dependency.

Gamification Mobile SDK Android Classes

The Android classes organize the Gamification Mobile SDK's functionalities and data structures. When developing mobile apps using the SDK, these classes provide components, services, or features with which you can interact.

Gamification Mobile Android SDK API Reference

The Gamification Mobile Android SDK interacts with the Salesforce Gamification APIs, and retrieves games assigned to a loyalty member, and the rewards won by the member after playing the game.

Game Zone UI Customization for Android

After you build the Gamification Mobile SDK for Android, you can customize the game zone UI components.

Clone and Import Gamification Mobile SDK for Android

Clone the GitHub repository to copy the code base and revision history of the repository to your local computer. After you clone the mobile SDK for Android repository, import Gamification Mobile Android SDK in Android Studio to automatically download and manage external dependencies.

EDITIONS

Available in: all editions that have Loyalty Management enabled

1. Go to the [Gamification Mobile SDK Android repository](#).
2. To copy the repository URL to your clipboard, click **Code**, and then click .
3. Open Terminal, and go to the directory where you want to clone the repository.
4. Enter `git clone https://github.com/loyaltysampleapp/GamificationMobileSDK-Android`.
5. Press Enter and wait for the repository to clone.
6. Unzip the `GamificationMobileSDK-Android` package that you downloaded.
7. Open Android Studio.
8. Go to **File > New > Import Project**, and then select the `GamificationMobileSDK-Android` folder.
9. Click **Open**.
10. Open the SDK files and ensure that the files include the appropriate SDK packages.



Note: If the sync fails, disconnect the VPN to ensure there's no blocker when downloading the dependencies, and then resync.

You can now use the classes and interfaces in the `GamificationMobileSDK-Android` module to manage and customize the gamification features in your application.

Integrate Gamification Mobile Android SDK into Loyalty Management Mobile Android SDK

To integrate the Gamification Mobile Android SDK into the existing Loyalty Management Mobile Android SDK, add it as an external dependency.

EDITIONS

Available in: all editions that have Loyalty Management enabled

1. Open Terminal, and go to the `LoyaltyMobileSDK-Android` folder.

2. Run these commands:

```
git submodule add
git@github.com:loyaltysampleapp/GamificationMobileSDK-Android.git
git submodule sync
git submodule update
cd GamificationMobileSDK-Android
git submodule update --init --recursive --remote
```

3. Open the `LoyaltyMobileSDK-Android` project in Android Studio.

4. Click **Open**.

5. Open the `SampleApps/MyNTORewards/build.gradle` file, and under Dependencies block, to the first line of Dependencies, add:

```
implementation(project(":GamificationMobileSDK-Android:GamificationMobileSDK"))
```

6. Open the `SampleApps/MyNTORewards/connectedapp.properties.sample` file and modify the properties of the connected app as per your org.

7. Click **Sync Now**.

8. Open the SDK files and ensure that the files include the appropriate SDK packages.



Note: If the sync fails, disconnect the VPN to ensure there's no blocker when downloading the dependencies, and then resync.

9. To add the gamification SDK package, include this code snippet in your code.

```
`import com.salesforce.gamification.api.*
import com.salesforce.gamification.model.*
import com.salesforce.gamification.repository.*
import com.salesforce.gamification.utilities.*`
```

You can now use the classes and interfaces in the `GamificationMobileSDK-Android` module to extend and manage the gamification features in the `MyNTORewards` sample app.

Gamification Mobile SDK Android Classes

The Android classes organize the Gamification Mobile SDK's functionalities and data structures. When developing mobile apps using the SDK, these classes provide components, services, or features with which you can interact.

EDITIONS

Available in: all editions that have Loyalty Management enabled

[GamificationRemoteRepository](#)

The `GamificationRemoteRepository` class invokes and manages gamification-related REST API requests. Use this class to interact with the Salesforce gamification APIs and retrieve member games, and play a game.

[GameAPIClient](#)

The `GameAPIClient` class creates a retrofit instance to invoke REST APIs.

[GameAuthenticator](#)

The `GameAuthenticator` interface defines the necessary methods for handling access tokens in the Salesforce API.

GamificationRemoteRepository

The `GamificationRemoteRepository` class invokes and manages gamification-related REST API requests. Use this class to interact with the Salesforce gamification APIs and retrieve member games, and play a game.

The Gamification Mobile Android SDK requires a valid access token to interact with the Salesforce gamification APIs. To provide a valid access token, implement the `GameAuthenticator` interface in your `GamificationRemoteRepository` class at the time of initialization. This class manages authentication using the provided instance of `GameAuthenticator`.

Use this constructor to initialize the `GamificationRemoteRepository` class.

```
class GamificationRemoteRepository constructor(
    auth: GameAuthenticator,
    instanceUrl: String,
    gameClient: NetworkClient
)
```

Parameter	Type	Description	Required or Optional
<code>auth</code>	<code>GameAuthenticator</code>	An instance of the <code>GameAuthenticator</code> interface.	Required
<code>instanceUrl</code>	<code>String</code>	The URL of your Salesforce org.	Required
<code>gameClient</code>	<code>NetworkClient</code>	An instance of the <code>GameAPIClient</code> class.	Required

Methods

getGames

Retrieve games for the specified participant ID.

```
final Result<Games> getGames(String participantId, String gameParticipantRewardId, Boolean mockResponse)
```

Parameter	Type	Description	Required or Optional
<code>participantId</code>	<code>String</code>	The ID of a loyalty program member, an account, or a	Required

Parameter	Type	Description	Required or Optional
		contact whose eligible games are to be retrieved.	
gameParticipantRewardId	String	The ID of the participant's game reward associated with the game whose details are to be fetched.	Optional
mockResponse	Boolean	Indicates whether to include a sample response (<code>true</code>) or not (<code>false</code>).	Optional

getGameReward

Retrieve the details of the reward that's provided to a participant after the game play.

```
final Result<GameRewardResponse> getGameReward(String gameParticipantRewardId, Boolean mockResponse)
```

Parameter	Type	Description	Required or Optional
gameParticipantRewardId	String	The ID of the participant's game reward won by the participant whose details are to be fetched.	Required
mockResponse	Boolean	Indicates whether to include a sample response (<code>true</code>) or not (<code>false</code>).	Optional

GameAPIClient

The `GameAPIClient` class creates a retrofit instance to invoke REST APIs.

This class extends the `NetworkClient` abstract class.

Use this constructor to initialize the `GameAPIClient` class.

```
class GameAPIClient constructor(auth: GameAuthenticator, instanceUrl: String) :
    NetworkClient(auth, instanceUrl)
```

Parameter	Type	Description	Required or Optional
auth	GameAuthenticator	An instance of the <code>GameAuthenticator</code> interface.	Required
instanceUrl	String	The URL of your Salesforce org.	Required

Methods

getOkHttpClientBuilder

Retrieve an instance of `OkHttpClient`.

```
final OkHttpClient getOkHttpClientBuilder()
```

getNetworkClient

Override the `getNetworkClient` method of the `NetworkClient` abstract class to get an instance of `GameAPIInterface`.

```
final GameAPIInterface getNetworkClient()
```

GameAuthenticator

The `GameAuthenticator` interface defines the necessary methods for handling access tokens in the Salesforce API.

Methods

getAccessToken()

Returns a valid access token as a string if available, otherwise null.

```
final String getAccessToken()
```

grantAccessToken

Returns a refreshed or newly granted access token as a string.

```
final String grantAccessToken()
```

Gamification Mobile Android SDK API Reference

The Gamification Mobile Android SDK interacts with the Salesforce Gamification APIs, and retrieves games assigned to a loyalty member, and the rewards won by the member after playing the game.

Interact with the Gamification APIs, including:

Resource	Description
/game/participant/\${participantId}/games on page 434	Get games assigned to a participant.
/game/participant/\${participantId}/game-reward on page 435	Get details of the reward won after the member plays the game.

EDITIONS

Available in: all editions that have Loyalty Management enabled

Game Zone UI Customization for Android

After you build the Gamification Mobile SDK for Android, you can customize the game zone UI components.

The game zone UI components are available in the

`/LoyaltyMobileSDK-Android/SampleApps/MY Rewards/src/main/java/com/salesforce/loyalty/mobile/myrewards/views/gamezone` package.

Spinner Customization

Customize the spinner components, such as the wheel size, frame width and color, and more in the `SpinnerConfiguration.kt` file. Here are all the parameters you can modify.

```
class SpinnerConfiguration {

    companion object {
        //preferences
        val WHEEL_SIZE = 280.dp
        val WHEEL_FRAME_WIDTH = 4.dp
        val WHEEL_FRAME_COLOUR = SpinnerFrameColour
        val WHEEL_BORDER_WIDTH = 12.dp
        val WHEEL_SEGMENT_BACKGROUND = Color.White //same colour will apply on border
        const val START_DEGREE = 0f
        //spinner default loading position degree 30f represents when spinner will load
        it will be at 30 degree
        const val ROTATION_PER_SECOND = 3f
        const val PADDING_ANGLE_BETWEEN_WHEEL_SEGMENT = 2f //padding between two segments
        of wheel
        const val ROTATION_DURATION = 3000 // duration of wheel rotate after the API
        responds
        const val INITIAL_ROTATION_DURATION = 20000 // duration of wheel rotate till API
        responds
        const val WHEEL_TEXT_INITIAL_ROTATION = 0f // we can configure the orientation of
        text
        //based on this angle,we can make wheel content straight or with angle inside the
        wheel
    }
}
```

For example, to change the spinner frame color to blue, open the

`/LoyaltyMobileSDK-Android/SampleApps/MY Rewards/src/main/java/com/salesforce/loyalty/mobile/myrewards/views/gamezone/spinner/SpinnerConfiguration.kt` file in Android Studio, and change the value of `val WHEEL_FRAME_COLOUR` from `SpinnerFrameColour` to `Color.Blue`.

To increase the padding between two segments of the wheel, change the value of `const val PADDING_ANGLE_BETWEEN_WHEEL_SEGMENT` from `2f` to `4f`.

Scratch Card Customization

You can customize the scratch card components, such as the box and canvas in the `ScratchCanvasView.kt` file.

For example, to change the scratch card background color to white, open the

`/LoyaltyMobileSDK-Android/SampleApps/MY Rewards/src/main/java/com/salesforce/loyalty/mobile/myrewards/views/gamezone/spinner/ScratchCanvasView.kt` file in Android Studio, and inside the `CanvasForScratching` method, replace `Box( modifier = modifier.background(VibrantPurple40).testTag(TEST_TAG_SCRATCH_CARD)` with `Box( modifier = modifier.background(Color.White).testTag(TEST_TAG_SCRATCH_CARD)`.

CHAPTER 13 Data Processing Engine, Batch Management, and Monitor Workflow Services

In this chapter ...

- [Data Model](#)
- [Common Tooling API Object](#)
- [Common Platform Event](#)
- [Common Business APIs](#)
- [Data Processing Engine](#)
- [Batch Management](#)
- [Monitor Workflow Services](#)

Data Processing Engine and Batch Management help automate your business processes. Use objects, APIs, Platform Events, and invocable actions to define, run, and review Data Processing Engine definitions and Batch Management jobs.

Here's how both these features can automate your business processes:

- **Data Processing Engine:** Transform data that's available in your Salesforce org and write back the transformation results as new or updated records. You can transform the data for standard and custom objects.
- **Batch Management:** Automate the processing of records in scheduled flows. You can process a high volume of standard and custom object records.

Once a Data Processing Engine definition is run or a Batch Management job is run, you can view the progress of the run and the results of the run using Monitor Workflow Services.

EDITIONS

Available in: Lightning Experience

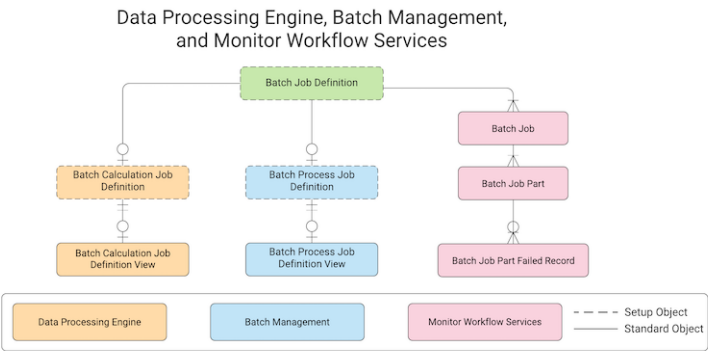
Available in: Data Processing Engine is available with Enterprise, Unlimited, and Performance Editions with the Financial Services Cloud, Loyalty Management, Manufacturing Cloud, Rebate Management, Accounting Subledger, or Provider Search in Health Cloud.

Available in: Batch Management is available with Enterprise, Unlimited, and Performance Editions with the Loyalty Management, Manufacturing Cloud, Rebate Management, or Accounting Subledger

Data Model

Data Processing Engine, Batch Management, and Monitor Workflow Services share a data model. Let's learn about the objects and relationships in this shared data model.

Here's the data model:



EDITIONS

Available in: Lightning Experience

Available in: Data Processing Engine is available with Enterprise, Unlimited, and Performance Editions with the Financial Services Cloud, Loyalty Management, Manufacturing Cloud, Rebate Management, Accounting Subledger, or Provider Search in Health Cloud.

Available in: Batch Management is available with Enterprise, Unlimited, and Performance Editions with the Loyalty Management, Manufacturing Cloud, Rebate Management, or Accounting Subledger

Common Tooling API Object

BatchJobDefinition is a common Tooling API object that is shared between Data Processing Engine and Batch Management.

Tooling API objects let you interact with metadata for declarative development. For example, you can create your own version of Setup.

BatchJobDefinition

Represents the definition of a batch job. This object is available in API version 51.0 and later.

BatchJobDefinition

Represents the definition of a batch job. This object is available in API version 51.0 and later.

 **Note:** 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.

Supported SOAP API Calls

`describeSObjects()`, `query()`, `retrieve()`

Supported REST API Methods

`GET`, `HEAD`, `Query`

Fields

Field	Details
Description	Type textarea Properties Filter, Group, Nillable, Sort Description The description of the batch job definition.
DeveloperName	Type string Properties Filter, Group, Sort Description The developer name of the batch job.
Language	Type picklist

EDITIONS

Available in: Lightning Experience

Available in: Data Processing Engine is available with Enterprise, Unlimited, and Performance Editions with the Financial Services Cloud, Loyalty Management, Manufacturing Cloud, Rebate Management, Accounting Subledger, or Provider Search in Health Cloud.

Available in: Batch Management is available with Enterprise, Unlimited, and Performance Editions with the Loyalty Management, Manufacturing Cloud, Rebate Management, or Accounting Subledger

Field	Details
	Properties Filter, Group, Restricted picklist, Sort
	Description The language in which the batch job is created. Possible values are: - af—Afrikaans - am—Amharic - ar—Arabic - ar_AE—Arabic (United Arab Emirates) - ar_BH—Arabic (Bahrain) - ar_DZ—Arabic (Algeria) - ar_EG—Arabic (Egypt) - ar_IQ—Arabic (Iraq) - ar_JO—Arabic (Jordan) - ar_KW—Arabic (Kuwait) - ar_LB—Arabic (Lebanon) - ar_LY—Arabic (Libya) - ar_MA—Arabic (Morocco) - ar_OM—Arabic (Oman) - ar_QA—Arabic (Qatar) - ar_SA—Arabic (Saudi Arabia) - ar_SD—Arabic (Sudan) - ar_SY—Arabic (Syria) - ar_TN—Arabic (Tunisia) - ar_YE—Arabic (Yemen) - bg—Bulgarian - bn—Bengali - bs—Bosnian - ca—Catalan - cs—Czech - cy—Welsh - da—Danish - de—German - de_AT—German (Austria) - de_BE—German (Belgium) - de_CH—German (Switzerland) - de_LU—German (Luxembourg) - el—Greek

Field	Details
	• en_AU—English (Australian) • en_CA—English (Canadian) • en_GB—English (UK) • en_HK—English (Hong Kong) • en_IE—English (Ireland) • en_IN—English (Indian) • en_MY—English (Malaysian) • en_NZ—English (New Zealand) • en_PH—English (Phillipines) • en_SG—English (Singapore) • en_US—English • en_ZA—English (South Africa) • eo—Esperanto (Pseudo) • es—Spanish • es_AR—Spanish (Argentina) • es_BO—Spanish (Bolivia) • es_CL—Spanish (Chile) • es_CO—Spanish (Colombia) • es_CR—Spanish (Costa Rica) • es_DO—Spanish (Dominican Republic) • es_EC—Spanish (Ecuador) • es_GT—Spanish (Guatemala) • es_HN—Spanish (Honduras) • es_MX—Spanish (Mexico) • es_NI—Spanish (Nicaragua) • es_PA—Spanish (Panama) • es_PE—Spanish (Peru) • es_PR—Spanish (Puerto Rico) • es_PY—Spanish (Paraguay) • es_SV—Spanish (El Salvador) • es_US—Spanish (United States) • es_UY—Spanish (Uruguay) • es_VE—Spanish (Venezuela) • et—Estonian • eu—Basque • fa—Farsi • fi—Finnish • fr—French

Field	Details
	• fr_BE—French (Belgium) • fr_CA—French (Canadian) • fr_CH—French (Switzerland) • fr_LU—French (Luxembourg) • ga—Irish • gu—Gujarati • hi—Hindi • hr—Croatian • hu—Hungarian • hy—Armenian • in—Indonesian • is—Icelandic • it—Italian • it_CH—Italian (Switzerland) • iw—Hebrew • iw_EO—Esperanto RTL (Pseudo) • ja—Japanese • ka—Georgian • km—Khmer • kn—Kannada • ko—Korean • lb—Luxembourgish • lt—Lithuanian • lv—Latvian • mi—Te reo • mk—Macedonian • ml—Malayalam • mr—Marathi • ms—Malay • mt—Maltese • my—Burmese • nl_BE—Dutch (Belgium) • nl_NL—Dutch • no—Norwegian • pl—Polish • pt_BR—Portuguese (Brazil) • pt_PT—Portuguese (European) • rm—Romansh

Field	Details
	• ro—Romanian • ro_MD—Romanian (Moldova) • ru—Russian • sh—Serbian (Latin) • sh_ME—Montenegrin • sk—Slovak • sl—Slovene • sq—Albanian • sr—Serbian (Cyrillic) • sv—Swedish • sw—Swahili • ta—Tamil • te—Telugu • th—Thai • tl—Tagalog • tr—Turkish • uk—Ukrainian • ur—Urdu • vi—Vietnamese • xh—Xhosa • zh_CN—Chinese (Simplified) • zh_HK—Chinese (Hong Kong) • zh_SG—Chinese (Singapore) • zh_TW—Chinese (Traditional) • zu—Zulu
MasterLabel	Type string Properties Filter, Group, Sort Description The label of the batch job.
ProcessGroup	Type string Properties Filter, Group, Sort

Field	Details
	Description The group or team that's using the batch job. This field is only applicable to Batch Management jobs.
Status	Type picklist Properties Defaulted on create, Filter, Group, Restricted picklist, Sort Description The status of the batch job. Possible values are: • Active • Inactive
Type	Type picklist Properties Defaulted on create, Filter, Group, Restricted picklist, Sort Description Specifies the type of batch job. Possible values are: • Calc—Data Processing Engine • Flow • BulkUpdate • ConsumptionOveragesCalculation • DecisionTableRefresh • DeepCloneSalesAgreement • EntitlementCreationBatchJob • HighScaleBreProcess • IndustriesLSCommercial • LoyaltyProgramProcess • ManagerProvisioning • NetUnitRateCalculation • PbbToOptyConversion • ProductCatalogCacheRefresh • RatableSummaryCreation • SummaryCreation

Field	Details
	When Data Processing Engine or Batch Management is available in a Salesforce org, the default values are <i>Calc</i> or <i>Flow</i> respectively. Other types may be available to you depending on the licenses available in your org.

Common Platform Event

Batch Management jobs and Data Processing Engine definitions are run using invocable actions in Flows. Use the `BatchJobStatusChanged` event to notify subscribers after a Batch Management job or a Data Processing Engine definition is processed in a flow.

[BatchJobStatusChangedEvent](#)

Notifies subscribers of when a batch job is completed in a flow. This object is available in API version 51.0 and later.

BatchJobStatusChangedEvent

Notifies subscribers of when a batch job is completed in a flow. This object is available in API version 51.0 and later.

Supported Calls

`describeSObjects()`

Supported Subscribers

Subscriber	Supported?
Flows	✓

Fields

Field	Details
<code>BatchJob</code>	Type string Description The unique identifier of the batch job.
<code>BatchJobDefinition</code>	Type string

EDITIONS

Available in: Lightning Experience

Available in: Data Processing Engine is available with Enterprise, Unlimited, and Performance Editions with the Financial Services Cloud, Loyalty Management, Manufacturing Cloud, Rebate Management, Accounting Subledger, or Provider Search in Health Cloud.

Available in: Batch Management is available with Enterprise, Unlimited, and Performance Editions with the Loyalty Management, Manufacturing Cloud, Rebate Management, or Accounting Subledger

Field	Details
	Properties Nillable Description The unique identifier of the batch job's definition.
EndDateTime	Type dateTime Properties Nillable Description The timestamp for when the batch job execution is complete.
EventUuid	Type string Properties Nillable Description A universally unique identifier (UUID) that identifies a platform event message. This field is available in API version 52.0 and later.
ReplayId	Type string Properties Nillable Description Represents an ID value that is populated by the system and refers to the position of the event in the event stream. Replay ID values aren't guaranteed to be contiguous for consecutive events. A subscriber can store a replay ID value and use it on resubscription to retrieve missed events that are within the retention window.
StartDateTime	Type dateTime Properties Nillable Description The timestamp for when the batch job execution is started.
Status	Type picklist Properties Restricted picklist

Field	Details
	Description The status of the batch job. Possible values are: • Canceled—Canceled • Failure • Success

Common Business APIs

Common Business APIs are RESTful APIs that are sometimes available as Apex classes and methods.

[REST Reference](#)

You can access Common Business APIs using REST endpoints. These REST APIs follow similar conventions as Connect REST APIs.

REST Reference

You can access Common Business APIs using REST endpoints. These REST APIs follow similar conventions as Connect REST APIs.

To understand the architecture, authentication, rate limits, and how the requests and responses work, see [Connect REST API Developer Guide](#).

[Resources](#)

Here's a list of Common Business API resources.

[Response Bodies](#)

Here's a list of Common Business API response bodies.

SEE ALSO:

[Connect REST API Developer Guide](#)

Resources

Here's a list of Common Business API resources.

[Batch Job Cancel](#)

Cancel a batch job of type data processing engine (calc job) and batch management. A batch job with only the status Submitted or In Progress can be canceled.

Batch Job Cancel

Cancel a batch job of type data processing engine (calc job) and batch management. A batch job with only the status Submitted or In Progress can be canceled.

Special Access Rules

- To use this resource, the following permissions are required:
- Your org must have the Batch Management and Data Processing Engine licenses
 - Users in your org require System Administration profile

Resource

```
/connect/batch-job/batchJobId/cancel-job
```

Resource example

```
/connect/batch-job/0mdxx00000000fxAAA/cancel-job
```

Available version

52.0

Requires Chatter

No

HTTP methods

POST

 **Note:** POST doesn't take request parameters or a request body.

Response body for POST

Returns HTTP 201 on success.

See [Batch Job Cancel Output](#) for HTTP code descriptions that are unique to this resource in case of failure of the batch job cancel request.

Response Bodies

Here's a list of Common Business API response bodies.

[Batch Job Cancel Output](#)
Output representation of the batch job cancel request.

Batch Job Cancel Output

Output representation of the batch job cancel request.

Property Name	Type	Description	Filter Group and Version	Available Version
message	String	Details about why the batch job cancel request failed.	Small, 52.0	52.0

This table lists HTTP response code descriptions that are unique to this resource.

HTTP Response Code	Error Code	Description
400	INVALID_STATUS	We can't cancel the batch job that doesn't have an active run. Specify the ID of a batch job with an active run and try again.
400	INVALID_STATUS	We can't cancel the batch job that is already canceled or completed. Specify the ID of a valid batch job with the status InProgress or Submitted and try again.
400	DELETE_FAILED	We can't cancel the batch job of which the results are already being written back.
400	DELETE_FAILED	We can't cancel the batch job because of an error in processing your org's data. Run the Data Processing Engine definition and try again.
403	FORBIDDEN	You don't have the permission to cancel a batch job. Ask your Salesforce admin for help.
404	RESOURCE_NOT_FOUND	Specify the ID of a valid batch job and try again.
500	INTERNAL_SERVER_ERROR	Something went wrong when we tried to cancel the batch job. Try again or ask your Salesforce admin for help.

Data Processing Engine

Transform data that's available in your Salesforce org and write back the transformation results as new or updated records. You can transform the data for standard and custom objects. Data Processing Engine consists of a Tooling API object, a standard object, a Metadata API, and an invocable action. You can use these to view, create, edit, and run Data Processing Engine definitions.

[Data Processing Engine Tooling API Objects](#)

Data Processing Engine consists of one Tooling API object, BatchCalcJobDefinition. Use this object to create and edit a Data Processing Engine definition.

[Data Processing Engine Standard Object](#)

Data Processing Engine contains one standard object, BatchCalcJobDefinitionView. Use this object to view all the Data Processing Engine definitions available in your Salesforce org, including file-based definitions.

[Data Processing Engine Metadata API](#)

Use a Metadata API to create, update, and activate Data Processing Engine definitions.

[Data Processing Engine Invocable Actions](#)

Run an active Data Processing Engine definition. For more information on custom invocable actions, see [REST API Developer Guide](#) and [Actions Developer Guide](#).

EDITIONS

Available in: Lightning Experience

Available in: Data Processing Engine is available with Enterprise, Unlimited, and Performance Editions with the Financial Services Cloud, Loyalty Management, Manufacturing Cloud, Rebate Management, Accounting Subledger, or Provider Search in Health Cloud.

Data Processing Engine Tooling API Objects

Data Processing Engine consists of one Tooling API object, BatchCalcJobDefinition. Use this object to create and edit a Data Processing Engine definition.

Tooling API objects let you interact with metadata for declarative development. For example, you can create your own version of Setup.

BatchCalcJobDefinition

Represents a Data Processing Engine (DPE) definition. This object is available in API version 51.0 and later.

BatchCalcJobDefinition

Represents a Data Processing Engine (DPE) definition. This object is available in API version 51.0 and later.

 **Note:** 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.

Supported SOAP API Calls

`create()`, `delete()`, `describeObjects()`, `query()`, `retrieve()`, `update()`, `upsert()`

Supported REST API Methods

`DELETE`, `GET`, `HEAD`, `PATCH`, `POST`, `Query`

Fields

Field	Details
BatchJobDefinitionId	Type reference Properties Filter, Group, Nillable, Sort Description The unique identifier of the associated batch job definition. This is a relationship field. Relationship Name BatchJobDefinition Relationship Type Lookup Refers To BatchJobDefinition

EDITIONS

Available in: Lightning Experience

Available in: Data Processing Engine is available with Enterprise, Unlimited, and Performance Editions with the Financial Services Cloud, Loyalty Management, Manufacturing Cloud, Rebate Management, Accounting Subledger, or Provider Search in Health Cloud.

Field	Details
<code>DataSpaceApiName</code>	Type string Properties Filter, Group, Nillable, Sort Description The Data Space API name from Data Cloud.
<code>DefinitionRunMode</code>	Type picklist Properties Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The execution mode of the Data Processing Engine definition. Possible values are: • <code>Batch</code> • <code>OnDemand</code>—This value is reserved for internal use. The default value is <code>Batch</code>.
<code>DeveloperName</code>	Type string Properties Filter, Group, Sort Description The developer name of the Data Processing Engine definition.
<code>DoesGenAllFailedRecords</code>	Type boolean Properties Defaulted on create, Filter, Group, Sort Description Indicates whether the error file includes a complete list of all failed writeback records (<code>true</code>) or not (<code>false</code>). The default value is <code>false</code>, and only the first instance of a failure is recorded in the error file. If set to <code>true</code>, all failed records are recorded in the error file for the writeback node. Available in API version 65.0 and later.
<code>ExecutionPlatformObjectType</code>	Type Picklist Properties Filter, Group, Nillable, Restricted picklist, Sort

Field	Details
	Description Required. The execution platform object type that's used during the read, transform, and writeback process for the Data Processing Engine definition. Possible values are: • <code>CalculatedInsightsObject</code> • <code>DataLakeObject</code> • <code>DataModelObject</code> • <code>None</code> Available in API version 65.0 and later.
<code>ExecutionPlatformType</code>	Type Picklist Properties Filter, Group, Restricted picklist, Sort, Not-Null Description Specifies the platform that's used to run the Data Processing Engine definition. Possible values: • <code>CDP</code>—Data Cloud • <code>CORE</code>—This value is reserved for internal use. • <code>CRMA</code>—CRM Analytics
<code>FullName</code>	Type string Properties Create, Group, Null Description The name of the Data Processing Engine definition. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
<code>IsTemplate</code>	Type boolean Properties Defaulted on create, Filter, Group, Sort Description Indicates whether this is a template Data Processing Engine definition.
<code>Language</code>	Type picklist Properties Defaulted on create, Filter, Group, Null, Restricted picklist, Sort

Field	Details
	Description The language in which this Data Processing Engine definition is created.
ManageableState	Type ManageableState enumerated list Properties Filter, Group, Nillable, Restricted picklist, Sort Description Indicates the manageable state of the specified component that is contained in a package: • beta • deleted • deprecated • deprecatedEditable • installed • installedEditable • released • unmanaged
MasterLabel	Type string Properties Filter, Group, Sort Description The label of the Data Processing Engine definition.
Metadata	Type complexvalue Properties Create, Nillable, Update Description Data Processing Engine definition's metadata. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
NamespacePrefix	Type string Properties Filter, Group, Nillable, Sort

Field	Details
	Description The namespace prefix associated with this object. Each Developer Edition organization that creates a managed package has a unique namespace prefix. Limit: 15 characters. You can refer to a component in a managed package by using the <code>namespacePrefix__componentName</code> notation. The namespace prefix can have one of the following values: • In Developer Edition organizations, the namespace prefix is set to the namespace prefix of the organization for all objects that support it. There's an exception if an object is in an installed managed package. In that case, the object has the namespace prefix of the installed managed package. This field's value is the namespace prefix of the Developer Edition organization of the package developer. • In organizations that aren't Developer Edition organizations, <code>NamespacePrefix</code> is only set for objects that are part of an installed managed package. There's no namespace prefix for all other objects.
ProcessType	Type picklist Properties Filter, Group, Restricted picklist, Sort Description Required. The process type for which the Data Processing Engine definition is created. Possible values are: • <code>AccountingPeriodClosure</code>—Legal Entity Accounting Period Closure—Available in API version 62.0 and later. • <code>ActionableList</code> • <code>AdvancedAccountForecast</code> • <code>BenefitManagement</code>—Available in API version 61.0 and later. • <code>BillingSchedulesforInvoiceGeneration</code>—Billing Schedules for Invoice Generation—Available in API version 62.0 and later. • <code>CDPEnrichment</code> • <code>ChannelInventoryManagement</code>—Available in API version 63.0 and later. • <code>CollectionsAndRecovery</code> • <code>CriteriaBsdSearchAndFilter</code> - Criteria-Based Search And Filter • <code>DataProcessingEngine</code> - Standard • <code>DecisionMatrixDataUpload</code>-This value is available only if you have Business Rules Engine enabled. • <code>Decisiontable</code>—Decision table activation—Available in API version 62.0 and later. • <code>Education</code> • <code>EmployeeService</code>—Available in API version 63.0 and later. • <code>FinancialSummaryRollup</code>—Available in API version 63.0 and later.

Field	Details
	• ForeignExchangeGainLossCalculations • FSHierarchyRollUp • Fundraising • FundraisingRollups—Available in API version 63.0 and later. • GeneralLedgerAccountBalancesSummary • InventoryBatchSearch • InventorySearch • InvoiceGeneration—Available in API version 62.0 and later. • LegalEntityAccountingPeriodClosureAdvanced—Available in API version 63.0 and later. • LifeSciencesCommercialTerritoryAlignment—Available in API version 63.0 and later. • LifeSciencesCustomerEngagement • Loyalty • LoyaltyPartnerManagement • LoyaltyPointsAggregation • NetZero • NextGenForecasting—Available in API version 62.0 and later. • PatientServicesProgram • PnmRosterFileUpload—Available in API version 62.0 and later. • PriceProtection—Available in API version 62.0 and later. • ProductCatalogManagement—Available in API version 63.0 and later. • ProgramBasedBusiness • ProgramManagementRollups • Rebates • RecordAggregation • RevenueTransactionManagement—Available in API version 63.0 and later. • AccountingSubledger—This value is reserved for internal use. • ProviderSearch—This value is reserved for internal use. • Recruitment—Available in API version 62.0 and later. • SalesAgreement—Available in API version 63.0 and later. • StockRotation • UsageManagement—Available in API version 62.0 and later. When Data Processing Engine is enabled for a Salesforce org, the default value is 'Standard'. Other process types may be available to you depending on the licenses available in your org.

Data Processing Engine Standard Object

Data Processing Engine contains one standard object, BatchCalcJobDefinitionView. Use this object to view all the Data Processing Engine definitions available in your Salesforce org, including file-based definitions.

BatchCalcJobDefinitionView

Represents the details of a Data Processing Engine definition. The definition can also be a file-based definition that is available in your Salesforce org. This object is available in API version 51.0 and later.

BatchCalcJobDefinitionView

Represents the details of a Data Processing Engine definition. The definition can also be a file-based definition that is available in your Salesforce org. This object is available in API version 51.0 and later.

 **Note:** 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.

EDITIONS

Available in: Lightning Experience

Available in: Data Processing Engine is available with Enterprise, Unlimited, and Performance Editions with the Financial Services Cloud, Loyalty Management, Manufacturing Cloud, Rebate Management, Accounting Subledger, or Provider Search in Health Cloud.

Supported Calls

`describeSObjects()`, `query()`

Fields

Field	Details
DataSpaceApiName	Type string Properties Filter, Group, Nillable, Sort Description Stores the data space API name from Data Cloud.
DefinitionRunMode	Type picklist Properties Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The execution mode of the Data Processing Engine definition. Possible values are: - Batch - OnDemand—On Demand The default value is Batch.

Field	Details
Description	Type string Properties Filter, Group, Nillable, Sort Description The description of a Data Processing Engine definition.
DurableId	Type string Properties Filter, Group, Nillable, Sort Description The unique identifier for the field. Always retrieve this value before using it, as the value isn't guaranteed to stay the same from one release to the next. Simplify queries by using this field instead of making multiple queries.
ExecutionPlatformObjectType	Type picklist Properties Filter, Group, Nillable, Restricted picklist, Sort Description Specifies the execution platform object type used to read, transform, and write back processes during the execution of a Data Processing Engine definition. This field is available in API version 65.0 and later. Possible values are: • <code>CalculatedInsightsObject</code>—Calculated Insights Object • <code>DataLakeObject</code>—Data Lake Object • <code>DataModelObject</code>—Data Model Object • None
ExecutionPlatformType	Type Picklist Properties Filter, Group, Restricted picklist, Sort, Not-Nillable Description Specifies the platform that's used to run the Data Processing Engine definition. Possible values: • <code>CDP</code>—Data Cloud • <code>CRMA</code>—CRM Analytics

Field	Details
InstalledPackageName	Type string Properties Filter, Group, Nillable, Sort Description The name of the package used to add the definition to the org.
IsActive	Type boolean Properties Defaulted on create, Filter, Group, Sort Description Indicates whether the definition is active.
IsTemplate	Type boolean Properties Defaulted on create, Filter, Group, Sort Description Indicates whether the definition is a template. You can make a copy of a template definition and update it based on your requirements.
LastModifiedBy	Type string Properties Filter, Group, Nillable, Sort Description The unique identifier of the user who modified the definition last.
ManageableState	Type picklist Properties Filter, Group, Nillable, Restricted picklist, Sort Description Indicates the manageable state of the definition if it's contained in a package. Possible values are: • beta—Managed-Beta • deleted—Managed-Proposed-Deleted • deprecated—Managed-Proposed-Deprecated • deprecatedEditable—SecondGen-Installed-Deprecated • installed—Managed-Installed

Field	Details
	• <code>installedEditable</code>—SecondGen-Installed-Editable • <code>released</code>—Managed-Released • <code>unmanaged</code>—Unmanaged
MasterLabel	Type string Properties Filter, Group, Nillable, Sort Description The label of the Data Processing Engine definition.
Name	Type string Properties Filter, Group, Nillable, Sort Description The name of the Data Processing Engine definition.
NamespacePrefix	Type string Properties Filter, Group, Nillable, Sort Description The namespace prefix associated with this object. Each Developer Edition organization that creates a managed package has a unique namespace prefix. Limit: 15 characters. You can refer to a component in a managed package by using the <code>namespacePrefix__componentName</code> notation. The namespace prefix can have one of the following values: • In Developer Edition organizations, the namespace prefix is set to the namespace prefix of the organization for all objects that support it. There's an exception if an object is in an installed managed package. In that case, the object has the namespace prefix of the installed managed package. This field's value is the namespace prefix of the Developer Edition organization of the package developer. • In organizations that aren't Developer Edition organizations, NamespacePrefix is only set for objects that are part of an installed managed package. There's no namespace prefix for all other objects.
ProcessType	Type picklist Properties Filter, Group, Nillable, Restricted picklist, Sort

Field**Details****Description**

The process type of the definition.

Possible values are:

- AccountingPeriodClosure—Legal Entity Accounting Period Closure—Available in API version 62.0 and later.
- ActionableList
- AdvancedAccountForecast
- BenefitManagement—Available in API version 61.0 and later.
- BillingSchedulesforInvoiceGeneration—Billing Schedules for Invoice Generation—Available in API version 62.0 and later.
- CDPEenrichment
- ChannelInventoryManagement—Available in API version 63.0 and later.
- CollectionsAndRecovery
- CriteriaBsdSearchAndFilter - Criteria-Based Search And Filter
- DataProcessingEngine - Standard
- DecisionMatrixDataUpload-This value is available only if you have Business Rules Engine enabled.
- Decisiontable—Decision table activation—Available in API version 62.0 and later.
- Education
- EmployeeService—Available in API version 63.0 and later.
- FinancialSummaryRollup—Available in API version 63.0 and later.
- ForeignExchangeGainLossCalculations
- FSHierarchyRollUp
- Fundraising
- FundraisingRollups—Available in API version 63.0 and later.
- GeneralLedgerAccountBalancesSummary
- InventoryBatchSearch
- InventorySearch
- InvoiceGeneration—Available in API version 62.0 and later.
- LegalEntityAccountingPeriodClosureAdvanced—Available in API version 63.0 and later.
- LifeSciencesCommercialTerritoryAlignment—Available in API version 63.0 and later.
- LifeSciencesCustomerEngagement
- Loyalty
- LoyaltyPartnerManagement
- LoyaltyPointsAggregation
- NetZero

Field	Details
	• <code>NextGenForecasting</code>—Available in API version 62.0 and later. • <code>PatientServicesProgram</code> • <code>PnmRosterFileUpload</code>—Available in API version 62.0 and later. • <code>PriceProtection</code>—Available in API version 62.0 and later. • <code>ProductCatalogManagement</code>—Available in API version 63.0 and later. • <code>ProgramBasedBusiness</code> • <code>ProgramManagementRollups</code> • <code>Rebates</code> • <code>RecordAggregation</code> • <code>RevenueTransactionManagement</code>—Available in API version 63.0 and later. • <code>AccountingSubledger</code>—This value is reserved for internal use. • <code>ProviderSearch</code>—This value is reserved for internal use. • <code>Recruitment</code>—Available in API version 62.0 and later. • <code>SalesAgreement</code>—Available in API version 63.0 and later. • <code>StockRotation</code> • <code>UsageManagement</code>—Available in API version 62.0 and later. When Data Processing Engine is enabled for a Salesforce org, the default value is 'Standard'. Other process types may be available to you depending on your industry solution and permission sets.

Data Processing Engine Metadata API

Use a Metadata API to create, update, and activate Data Processing Engine definitions.

[BatchCalcJobDefinition](#)

Represents a Data Processing Engine definition.

BatchCalcJobDefinition

Represents a Data Processing Engine definition.

Parent Type

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

`BatchCalcJobDefinition` components have the suffix `.batchCalcJobDefinition` and are stored in the `batchCalcJobDefinitions` folder.

EDITIONS

Available in: Lightning Experience

Available in: Data Processing Engine is available with Enterprise, Unlimited, and Performance Editions with the Financial Services Cloud, Loyalty Management, Manufacturing Cloud, Rebate Management, Accounting Subledger, or Provider Search in Health Cloud.

Version

BatchCalcJobDefinition components are available in API version 51.0 and later.

Special Access Rules

To use this metadata type, one of these licenses is required:

- Loyalty Management
- Financial Services Cloud
- Rebate Management
- Manufacturing Cloud
- Net Zero Cloud

Fields

Field Name	Field Type	Description
aggregates	BatchCalcJobAggregate[]	Collection of aggregate nodes in a data processing engine.
appends	BatchCalcJobUnion[]	Collection of append nodes in a data processing engine.
atomicWritebacks	BatchCalcJobAtomicWriteback[]	Collection of composite writeback nodes in a data processing engine definition. Available in API version 62.0 and later.
customNodes	BatchCalcJobCustomNode[]	Collection of custom nodes in a data processing engine. Available in API version 57.0 and later.
dataSpaceApiName	string	Stores the Data Space API Name from . Available in API version 60.0 and later.
datasources	BatchCalcJobDatasource[]	Collection of data source nodes in a data processing engine.
definitionRunMode	BatchCalcJobRunMode (enum of type string)	Specifies the execution mode in a data processing engine. Valid values are: • Batch • OnDemand
description	string	Description of a data processing engine definition.
doesGenAllFailedRecords	boolean	Indicates whether the error file includes a complete list of all failed writeback records (<code>true</code>) or not (<code>false</code>). The default value is <code>false</code> , and only the first instance of a failure is recorded in the error file. If set to <code>true</code> , all failed records are recorded in the error file for the writeback node. Available in API version 65.0 and later.

Field Name	Field Type	Description
executionPlatformObjectType	ExecutionPlatformType (enumeration of type string)	The execution platform object type that's used during the read, transform, and writeback process for the Data Processing Engine definition. Possible values are: - CalculatedInsightsObject - DataLakeObject - DataModelObject - None Available in API version 65.0 and later.
executionPlatformType	ExecutionPlatformType (enumeration of type string)	The platform that's used to run the Data Processing Engine definition. Valid values are: - CRMA - CDP - CORE Available in API version 59.0 and later.
filters	BatchCalcJobFilter[]	Collection of filter nodes in a data processing engine. definition.
forecasts	BatchCalcJobForecast[]	Collection of forecast nodes in a data processing engine. definition. Available in API version 58.0 and later.
hierarchyPaths	BatchCalcJobHierarchyPath[]	Collection of hierarchy path nodes in a data processing engine definition.
isTemplate	boolean	Indicates whether it's a template data processing engine definition.
joins	BatchCalcJobSourceJoin[]	Collection of join nodes in a data processing engine.
label	string	The label of a data processing engine definition.
parameters	BatchCalcJobParameter[]	Collection of input variables in a data processing engine.
processType	BatchCalcProcessType (enumeration of type string)	The process type of a data processing engine. These process types may be available to you depending on your industry solution and permission sets. Valid values are: - AccountingPeriodClosure - AccountingSubledger—This value is reserved for internal use. - ActionableList - AdvancedAccountForecast - BenefitManagement - BillingSchedulesforInvoiceGeneration - CDPEnrichment - ChannelInventoryManagement—Available in API version 63.0 and later.

Field Name	Field Type	Description
		- CollectionPlan—Available in API version 65.0 and later. - CriteriaBsdSearchAndFilter - DataProcessingEngine - DecisionMatrixDataUpload - Decisiontable - Education - EmployeeService—Available in API version 63.0 and later. - FinancialSummaryRollup - ForeignExchangeGainLossCalculations—Available in API version 65.0 and later. - FSCHierarchyRollUp - Fundraising—Available in API version 64.0 and later. - FundraisingRollups—Available in API version 63.0 and later. - GeneralLedgerAccountBalancesSummary—Available in API version 65.0 and later. - InventoryBatchSearch—Available in API version 65.0 and later. - InventorySearch—Available in API version 65.0 and later. - InvoiceGeneration - Loyalty - LegalEntityAccountingPeriodClosureAdvanced—Available in API version 63.0 and later. - LifeSciencesBatchCalesCommercialTerritoryAlignment—Available in API version 63.0 and later. - LifeSciencesCustomerEngagement—Available in API version 64.0 and later. - LoyaltyPartnerManagement - LoyaltyPointsAggregation - NextGenForecasting—Available in API version 64.0 and earlier. - NetZero - PatientServicesProgram—Available in API version 64.0 and later. - PnmRosterFileUpload - PriceProtection - ProductCatalogManagement - ProgramBasedBusiness - ProviderSearch—This value is reserved for internal use. - Rebates - Recruitment

Field Name	Field Type	Description
		• <code>RevenueTransactionManagement</code>—Available in API version 63.0 and later. • <code>SalesAgreement</code>—Available in API version 63.0 and later. • <code>TestAtomicWritebackScale</code>—Available in API version 64.0 and later. • <code>TestProcessType</code> • <code>UsageManagement</code>
<code>status</code>	<code>BatchJobDefinitionStatus</code> (enumeration of type string)	Status of a data processing engine definition. Valid values are: • <code>Active</code> • <code>Inactive</code>
<code>transforms</code>	BatchCalcJobTransform[]	Collection of data transformation nodes in a data processing engine.
<code>writebacks</code>	BatchCalcJobWritebackObject[]	Collection of writeback objects in which the results of the data processing engine are written back.

BatchCalcJobAggregate

Represents a collection of fields relating to an aggregate node in a data processing engine.

Fields

Field Name	Field Type	Description
<code>description</code>	string	Description of an aggregate node.
<code>fields</code>	BatchCalcJobAggregateField[]	Required. Collection of aggregation fields.
<code>groupBy</code>	string[]	Required. Collections of fields used to group data in an aggregate node.
<code>label</code>	string	Required. Label of an aggregate node.
<code>name</code>	string	Required. Name of an aggregate node.
<code>sourceName</code>	string	Required. Name of the source node.

BatchCalcJobAggregateField

Represents a collection of fields relating to an aggregation field in an aggregate node of a data processing engine.

Fields

Field Name	Field Type	Description
aggregateFunction	BatchCalcAggregateFunction (enumeration of type string)	Required. Function used for aggregation. Valid values are: • Unique—A count of unique values. • Sum—The sum of all values. • Max—The largest value. • Min—The smallest value. • Avg—The average value, calculated as the mean. • Std—The standard deviation. • Stdp—A standard deviation with population variance. • Var—The variance. • VarP—The variance with population. • Count—The total count of values.
alias	string	Required. Name that subsequent nodes within the data processing engine use to refer to the aggregate field.
sourceFieldName	string	Required. Source node field on which the aggregate is calculated.

BatchCalcJobAtomicWriteback

Represents a node in a DPE definition that stores the details about the relationship between the writeback nodes and the composite writeback operations between the nodes.

Field Name	Field Type	Description
description	string	Description of the composite writeback object.
label	string	Required. Name of the composite writeback object.
name	string	Required. API name of the composite writeback object.
writebackObjectRelationships	BatchCalcAtomicWritebackRelationship	Specifies the relationship between the writeback objects that are involved in the writeback operation.
writebackSequence	int	Sequence in which the data processing engine executes the composite writeback node.

BatchCalcJobAtomicWritebackRelationship

Represents the relationships between the writeback objects that are involved in a composite writeback operation. It captures the relationships between these objects and the sequence in which they should be processed.

Field Name	Field Type	Description
childWriteback ObjectField	string	Field name that's associated with the child writeback object in a composite writeback relationship. Available in API version 63.0 and later.
childWriteback ObjectName	string	Name of the child writeback object that's associated with the writeback relationship.
parentWriteback ObjectField	string	Field name that's associated with the parent writeback object in a composite writeback relationship. Available in API version 63.0 and later.
parentWriteback ObjectName	string	Required. Name of the parent writeback object that's associated with the writeback relationship.
relationshipName	string	Describes the relationship between the child and parent writeback objects in a composite writeback node. Available in API version 64.0 and later.
sequenceNumber	int	Sequence number of the writeback node that's associated with its parent node in the relationship.

BatchCalcJobCustomNode

Represents a collection of custom nodes in a data processing engine. Use a custom node to add a custom action.

Fields

Field Name	Field Type	Description
description	string	Description of a custom node.
extensionName	string	Required. Name of an extension node.
extensionNamespace	string	Required. Namespace of an extension node.
label	string	Required. Label of a custom node.
name	string	Required. Name of a custom node.
parameters	BatchCalcJobCustomNodeParameter[]	The field mappings of an extension node.
sources	string[]	Sources of an extension node.

BatchCalcJobCustomNodeParameter

Represents the field mappings of an extension node.

Fields

Field Name	Field Type	Description
name	string	Required. Name of a parameter.

Field Name	Field Type	Description
value	string	Required. Value of a parameter.

BatchCalcJobDatasource

Represents a collection of fields relating to a data source node in a data processing engine.

Fields

Field Name	Field Type	Description
CSVDelimiter	BatchCalcJobCSVDelimiter (enumeration of type string)	Specifies the field separator to read fields from a CSV file record. Possible values are: • COMMA • BACKQUOTE • CARET • PIPE • SEMICOLON • TAB The default value is <code>COMMA</code>. The same delimiter value used for the CSV file can't be used within any of the column values in the file. If you mistakenly use the same delimiter value in column values, it can cause data parsing issues.
description	string	Description of a data source node.
fields	BatchCalcJobDatasourceField[]	Required. Collection of data source fields.
fileIdentifier	string	Specifies the source of the file or file storage system.
filePath	string	The file path for the specified file.
fileSource	BatchCalcJobFileSource (enumeration of type string)	Specifies the source of the file or file storage system. Possible value is: • ContentManagement
label	string	Required. Label of a data source node.
name	string	Required. Name of a data source node.
sourceName	string	Required. Name of a standard or custom object from which the data source node extracts data.
type	BatchCalcJobDataSource Type (enumeration of type string)	Required. Type of object for the source object field. Supported values are: • Analytics • CalculatedInsightsObject

Field Name	Field Type	Description
		• CRMObject • CSV • DataModelObject • StandardObject

BatchCalcJobDatasourceField

Represents a collection of fields relating to a source object field that are selected in the data source node of a data processing engine.

Fields

Field Name	Field Type	Description
alias	string	Name that subsequent nodes within the data processing engine use to refer to the data source field. Required when the field name is lookup.
dataType	BatchCalcJobDataType (enumeration of type string)	Specifies the data type of the input field when using a CSV file as a data source. Possible values are: • Boolean—Available in API version 65.0 and later. • Date • DateTime • MultiValue • Numeric • Text
isPrimaryKey	boolean	Indicates whether a column name is the primary key (<code>true</code>) or not (<code>false</code>) for the Data Cloud CSV file.
name	string	Required. Name of the field. Can be either of the following: • Name of the source field selected in the associated data source object. • Name from a nested lookup object with three child levels.

BatchCalcJobFilter

Represents a collection of fields relating to a filter node in a data processing engine.

Fields

Field Name	Field Type	Description
criteria	BatchCalcJobFilterCriteria[]	Collection of filter criteria in a filter node. The field is required when <code>isDynamicFilter</code> is set to <code>False</code> .
description	string	Description of the batch calculation job filter.
filterCondition	string	Logic that is specified to apply the filter conditions. The field is required when <code>isDynamicFilter</code> is set to <code>False</code> .
filterParameterName	string	Name of the parameter of type filter.
isDynamicFilter	boolean	Indicates whether the filter criteria is dynamic. If value is set to <code>True</code> , filter criteria is passed in runtime with <code>filterParameterName</code> .
label	string	Required. Label of the filter node.
name	string	Required. Name of the filter node.
sourceName	string	Required. Name of the source node.

BatchCalcJobForecast

Represents a collection of fields relating to a forecast node in a data processing engine. Available in API version 58.0 and later.

Fields

Field Name	Field Type	Description
~~accuracy~~ BatchCalcJobFrstAccuracy (enumeration of type string)		The interval percentage to account for errors in forecasts. Possible values are: • <code>Eighty</code> • <code>NinetyFive</code> • <code>None</code> The default value is <code>None</code> .
~~aggrFields~~ BtchCalcJobFrstAggrFld[]		The list of fields to forecast.
~~dateField~~ string		Required. The date field from the source node used to forecast values for the specified forecast length.

Field Name	Field Type	Description
~~description~~	string	The description of the forecast node.
~~forecastModel~~	BatchCalcJobFrcstModel (enumeration of type string)	The model used to forecast data. Possible values are: • Additive • Auto • Multiplicative The default value is Auto.
~~forecastPeriod~~	int	The number of time periods to generate forecast data. For example, if you select Year-Month as the forecast period type, and 4 as the forecast period count, the forecast results are generated for the next 4 months. The minimum and the default count is 1, and the maximum is 100.
~~forecastPeriodType~~	BatchCalcJobFrcstPeriodType (enumeration of type string)	Required. The type of forecast period to group date field values in the forecast results. Possible values are: • FiscalYear • FiscalYearMonth • FiscalYearQuarter • FiscalYearWeek • Year • YearMonth • YearMonthDay • YearQuarter • YearWeek
~~groupFields~~	BatchCalcJobFrcstGrpFld[]	The source fields for grouping the data to be

Field Name	Field Type	Description
		processed by the forecast node.
label	string	Required. The name of the forecast node in the UI.
name	string	Required. A unique name for the forecast node.
startDate	string	Required. The start date of the forecast period.
seasonality	BatchCalcJobFrcstSeasonality (enumeration of type string)	Represents the periodic fluctuations that occur around the same time every year. Possible values are: - Two - Three - Four - Five - Six - Seven - Eight - Nine - Ten - Eleven - Twelve - Thirteen - Fourteen - Fifteen - Sixteen - Seventeen - Eighteen - Nineteen - Twenty - TwentyOne

Field Name	Field Type	Description
		• TwentyTwo • TwentyThree • TwentyFour • Auto • None The default value is None.
ignoreLastPeriod	boolean	Indicates whether to ignore the last period in the source node when it has incomplete data (<code>true</code>) or not (<code>false</code>). The default value is <code>false</code> .
sourceName	string	Required. The name of the source node. A source can be any node other than the datasink and register node.

BatchCalcJobForecastAggregateField

Represents a list of fields to forecast in a forecast node.

Field Name	Field Type	Description
aggregateFunction	BatchCalcJobAggregateFunction (enumeration of type string)	Required. The function of the aggregate field. Possible values are: • Avg • Count • Max • Min • Std • StdP • Sum • Unique • Var

Field Name	Field Type	Description
		VarP
aggregationResultLabel	string	Required. The name of the aggregation result generated from the aggregation function that's applied to the source node field.
fieldName	string	Required. The name of the source field.

BatchCalcJobFrstGrpFld

Represents source fields for grouping the data to be processed by the forecast node.

Field Name	Field Type	Description
fieldName	string	Required. The name of the source field to group the data to be processed by the forecast node.
groupBy	string	A comma-separated list of values to group data by. Required when the source field type is Date or DateTime. Possible values are: - Second - Second Epoch - Minute - Hour - Day - Day Epoch - Week - Month - Quarter - Year

BatchCalcJobHierarchyPath

Represents a collection of hierarchy path nodes in a data processing engine definition.

Fields

Field Name	Field Type	Description
description	string	Description of the hierarchy path node.
hierarchyFieldName	string	Required. Field name that contains the hierarchy path.
isSelfFieldValueIncluded	boolean	Indicates whether the self value is included in the calculated hierarchy path (True) or not (False).
label	string	Required. Label of the hierarchy path node.
name	string	Required. Name of the hierarchy path node.
parentFieldName	string	Required. Parent field name to calculate hierarchy path.
selfFieldName	string	Required. Self field name to calculate hierarchy path.
sourceName	string	Required. Name of the source node.

BatchCalcJobFilterCriteria

Represents a collection of fields relating to a filter condition in a filter node in a data processing engine.

Fields

Field Name	Field Type	Description
inputVariable	string	Name of the input variable used as a filter.
operator	BatchCalcJobFilter Operator (enumeration of type string)	Required. Operator that is specified in the filter condition. Valid values are: • Equals • NotEquals • GreaterThan • GreaterThanOrEqual • LessThan • LessThanOrEqual • StartsWith • EndsWith • Contains • DoesNotContain • IsNull • IsNotNull • In • NotIn

Field Name	Field Type	Description
sequence	integer	Required. Sequence number used to refer the criteria in a filter node.
sourceFieldName	string	Required. Name of the field from the source node to apply the filter.
value	string	Value used to filter data from the source node.

BatchCalcJobParameter

Represents a collection of fields relating to an input variable in a data processing engine.

Fields

Field Name	Field Type	Description
dataType	BatchCalcJobParameter DataType (enumeration of type string)	Required. Data type of the parameter. Valid values are: • Date • DateTime • Expression • FileIdentifier • Filter • Numeric • Text
defaultValue	string	Default value of the parameter.
description	string	Description of the batch calculation job parameter.
isMultiValue	boolean	Indicates whether the parameter has different values (<code>True</code>) or not (<code>False</code>). This field is supported only for the <code>Text</code> data type.
label	string	Required. Label of the batch calculation job parameter.
name	string	Required. Name of the batch calculation job parameter.

BatchCalcJobSourceJoin

Represents a collection of fields relating to a join node in a data processing engine.

Fields

Field Name	Field Type	Description
description	string	Description of the join node.
fields	BatchCalcJobJoin ResultField[]	Collection of fields in a join node.

Field Name	Field Type	Description
joinKeys	BatchCalcJobJoinKey[]	Collection of mapping of fields from the primary source node and the second source node in a join node.
label	string	Required. Label of the join node.
name	string	Required. Name of the join node.
primarySourceName	string	Required. Name associated with the node as the primary source node.
secondarySourceName	string	Required. Name associated with the node as the secondary source node.
type	BatchCalcJobSourceJoinType (enumeration of type string)	Required. Type of join specified between the primary source node and secondary source node. Valid values are: • LeftOuter • RightOuter • Inner • Outer • Lookup

BatchCalcJobJoinKey

Represents a collection of fields relating to a mapping of fields from the first source node and second source node in a join node of a data processing engine.

Fields

Field Name	Field Type	Description
primarySourceFieldName	string	Required. Mapped field name of the primary source node.
secondarySourceFieldName	string	Required. Mapped field name of the secondary source node.

BatchCalcJobJoinResultField

Represents a collection of fields relating to a set of resultant fields in a join node of a data processing engine.

Fields

Field Name	Field Type	Description
alias	string	Required. Name that subsequent nodes within the data processing engine definition use to refer to the resultant field.
sourceFieldName	string	Required. Name of field from the primary or secondary data source.
sourceName	string	Required. Source node of the primary or secondary data source.

BatchCalcJobTransform

Represents a collection of fields relating to a data transformation in a data processing engine.

Fields

Field Name	Field Type	Description
description	string	The description of the batch calculation job transform.
droppedFields	BatchCalcJobTransformDroppedField[]	The collection of dropped fields in a data transformation. Available when the transformation type is <code>Slice</code> .
expressionFields	BatchCalcJobTransformAddedField[]	The collection of formula fields in a data transformation. Available when the transformation type is <code>Expression</code> .
label	string	Required. The label of the batch calculation job transform.
name	string	Required. The name of the batch calculation job transform.
orderBy	BatchCalcJobOrderByField on page 859[]	A collection of fields that's used to sort the records within each partition group.
partitionBy	string[]	A group of fields that's used to partition the source data into partition groups.
sourceName	string	Required. Name of the source node.
transformType	BatchCalcJobTransformType (enumeration of type string)	Required. The type of transformation. Valid values are: • <code>ComputeRelative</code>—This transformation calculates values based on values of the same partition group. • <code>Expression</code>—This transformation calculates values based on existing values of fields in the same record. • <code>Slice</code>—This transformation removes fields from the source node.

BatchCalcJobTransformDroppedField

Represents a collection of fields relating to a dropped field in a data transformation of a data processing engine.

Fields

Field Name	Field Type	Description
sourceFieldName	string	Required. Name of the field that is dropped.

BatchCalcJobTransformAddedField

Represents a collection of fields relating to a formula in a data transformation of a data processing engine.

Fields

Field Name	Field Type	Description
alias	string	Required. Name that subsequent nodes within the data processing engine use to the transform node.
dataType	BatchCalcJobDataType (enumeration of type string)	Required. Data type of the formula. Valid values are: • Boolean—Available in API version 65.0 and later. • Date • DateTime • MultiValue • Numeric • Text
decimalPlaces	integer	Number of digits to the right of a decimal point in the value. Required for the Numeric data type.
expression	string	Required. Formula defined by the user.
length	integer	Total length of the value including the decimal places. Required for data types: Text and Numeric.

BatchCalcJobOrderByField

Represents a collection of fields that are used to sort the partitioned data.

Fields

Field Name	Field Type	Description
name	string	Required. Name of the field that is used to sort data.
orderType	BatchCalcJobOrderType (enumeration of type string)	Order in which the data is sorted. Valid values are: • Ascending • Descending

BatchCalcJobUnion

Represents a collection of fields relating to the union of data from two nodes in a data processing engine.

Fields

Field Name	Field Type	Description
description	string	Description of the batch calculation job union.
isDisjointedSchema	boolean	Indicates whether the union is of two disjointed datasets (<code>true</code>) or not (<code>false</code>). Set to <code>True</code> to allow joining of two datasets having no common fields.
label	string	Required. Label of the batch calculation job union.
name	string	Required. Name of the batch calculation job union.
sources	string[]	Names of the source nodes.

BatchCalcJobWritebackObject

Represents a collection of fields relating to the object in which the results of the data processing engine are written back.

Fields

Field Name	Field Type	Description
~~canWritebackToNonEditableFields~~	boolean	Indicates whether the non-editable fields are included in field mapping when the action type is upsert. The default value is <code>false</code> . Available in API version 64.0 and later.
description	string	Descriptions of the batch calculation job writeback object.
externalIdFieldName	string	Unique external field ID for the target object name. Available in API version 60.0 and later.
fields	BatchCalcJobWriteback Mapping []	Collection of the writeback fields.
filterCondition	string	The condition that filters the records from a writeback dataset for a user. Examples of a filter condition include a user ID, stage name, and a security policy that returns only the records that a user owns. Available in API version 57.0 and later.
folderName	string	The folder where the writeback dataset is saved. Available in API version 57.0 and later.
groupBy	string	Reserved for future use.
isChangedRow	boolean	Indicates whether a row in the write back object is changed. Set to <code>True</code> to write back the changed rows.
isExistingDataset	boolean	Indicates whether a CRM Application (CRMA) dataset or a Data Lake object is present (<code>true</code>) or will be created (<code>false</code>). Available in API version 62.0 and later.

Field Name	Field Type	Description
label	string	Required. Name of the write back object.
name	string	Required. Name of the batch calculation job write back object.
operationType	BatchCalcJobWriteback Opn (enumeration of type string)	Type of operation specified. Valid values are: • Delete—This value is available in API version 56.0 and later. • Insert • Overwrite—Available only when <code>storageType</code> is <code>DataLakeObject</code>. This value is available in API version 60.0 and later. • Update • Upsert
sharingInheritanceObjectName	string	The name of the source object from which the row-level sharing inheritance settings are applied. Available in API version 57.0 and later.
shouldCreateTargetObject	boolean	Indicates whether target Data Lake Object or Salesforce Object is created in Salesforce (<code>true</code>) or not (<code>false</code>). Available in API version 65.0 and later.
shouldMngRowLockFor GroupedRec	boolean	Reserved for future use.
sourceName	string	Required. Name of the source node associated with the write back object.
storageType	BatchCalcJobWriteback Type (enumeration of type string)	Specifies where you want to use the data stored in the source node. Available in API version 57.0 and later. Valid values are: • Analytics • DataLakeObject • sObject The default value is <code>sObject</code> .
targetObjectName	string	Required. Object that is inserted or upserted by the data processing engine.
writebackSequence	integer	Sequence in which the target object is updated by the data processing engine.
writebackUser	string	ID of the user whose permissions decide which objects and fields of the target object can be updated.

BatchCalcJobWritebackMapping

Represents a collection of fields relating to the mapping between results and the fields in the target object.

Fields

Field Name	Field Type	Description
fieldType	string	Target field type on the writeback object. Valid values are: • Primary Key • Qualifier Key Available in API version 64.0 and later.
isAutogenerated	boolean	Indicates whether the target field value on the writeback object is autogenerated (<code>true</code>) or not (<code>false</code>). Available in API version 64.0 and later.
parentName	string	Name of the lookup object. Required only when the <code>relationshipName</code> field is defined.
relationshipName	string	Name of the lookup relationship.
runtimeParameter	boolean	Indicates whether the source field from runtime parameter is <code>true</code> or <code>false</code> . The default value is <code>false</code> . Available in API version 59.0 and later.
sourceFieldName	string	Required. Name of the field in the source node that is written back.
targetFieldName	string	Name of the sObject field to which the results are written back.

Declarative Metadata Sample Definition

The following is an example of a BatchCalcJobDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<BatchCalcJobDefinition xmlns="http://soap.sforce.com/2006/04/metadata">
  <aggregates>
    <description>Aggregate Description</description>
    <fields>
      <aggregateFunction>Count</aggregateFunction>
      <alias>NameCount</alias>
      <sourceFieldName>Name</sourceFieldName>
    </fields>
    <groupBy>ContactId</groupBy>
    <groupBy>Name</groupBy>
    <label>AggregateOpportunities</label>
    <name>AggregateOpportunities</name>
    <sourceName>Opportunity</sourceName>
  </aggregates>
  <forecasts>
    <description>ForecastNode Description</description>
    <label>ContactForecast</label>
    <name>ContactForecast</name>
    <sourceName>Contact</sourceName>
    <dateFieldName>CreatedDate</dateFieldName>
  </forecasts>
</BatchCalcJobDefinition>
```

```
<forecastPeriodType>YearMonth</forecastPeriodType>
<shouldExcludeLastPeriod>>false</shouldExcludeLastPeriod>
<forecastPeriodCount>12</forecastPeriodCount>
<periodStartDateName>CreatedDateYM</periodStartDateName>
<forecastModelType>Auto</forecastModelType>
<seasonality>None</seasonality>
<accuracyPercent>None</accuracyPercent>
<aggregationFields>
  <aggregateFunction>Count</aggregateFunction>
  <aggregationResultLabel>CountOfLastName</aggregationResultLabel>
  <fieldName>LastName</fieldName>
</aggregationFields>
<groupFields>
  <fieldName>LastModifiedDate</fieldName>
  <groupBy>Week</groupBy>
</groupFields>
</forecasts>
<appends>
  <description>Append desc</description>
  <isDisjointedSchema>>true</isDisjointedSchema>
  <label>AppendAllAccounts</label>
  <name>AppendAllAccounts</name>
  <sources>AccountsOfManufacturingIndustry</sources>
  <sources>ComputeRelativeManufacturingIndustry</sources>
</appends>
<datasources>
  <description>Desc Contact</description>
  <fields>
    <alias>Id</alias>
    <name>Id</name>
    <isPrimaryKey>>false</isPrimaryKey>
    <dataType>Text</dataType>
  </fields>
  <fields>
    <alias>LastName</alias>
    <name>LastName</name>
    <isPrimaryKey>>false</isPrimaryKey>
    <dataType>Text</dataType>
  </fields>
  <fields>
    <alias>CreatedDate</alias>
    <name>CreatedDate</name>
    <isPrimaryKey>>false</isPrimaryKey>
    <dataType>Date</dataType>
  </fields>
  <fields>
    <alias>LastModifiedDate</alias>
    <name>LastModifiedDate</name>
    <isPrimaryKey>>false</isPrimaryKey>
    <dataType>Date</dataType>
  </fields>
  <label>Contact</label>
  <name>Contact</name>
  <sourceName>Contact</sourceName>
```

```

    <type>StandardObject</type>
    <fileSource>ContentManagement</fileSource>
    <fileIdentifier>069xx0000004CAeAAM</fileIdentifier>
    <CSVDelimiter>COMMA</CSVDelimiter>
    <filePath>parentFolder/childFolder</filePath>
  </datasources>
  <datasources>
    <fields>
      <alias>Name</alias>
      <name>Name</name>
      <isPrimaryKey>false</isPrimaryKey>
      <dataType>Text</dataType>
    </fields>
    <fields>
      <alias>ContactId</alias>
      <name>ContactId</name>
      <isPrimaryKey>false</isPrimaryKey>
      <dataType>Text</dataType>
    </fields>
    <label>Opportunity</label>
    <name>Opportunity</name>
    <sourceName>Opportunity</sourceName>
    <type>StandardObject</type>
    <fileSource>ContentManagement</fileSource>
    <fileIdentifier>069xx0000004CAeAAM</fileIdentifier>
    <CSVDelimiter>COMMA</CSVDelimiter>
    <filePath>parentFolder/childFolder</filePath>
  </datasources>
  <description>Calculates and creates transaction journal records based on the orders
placed by the loyalty program members. The transaction journals are used to accrue points
to the member.</description>
  <filters>
    <criteria>
      <operator>Equals</operator>
      <sequence>1</sequence>
      <sourceFieldName>LastName</sourceFieldName>
      <value>Salesforce</value>
    </criteria>
    <description>Filter Desc</description>
    <filterCondition>1</filterCondition>
    <isDynamicFilter>false</isDynamicFilter>
    <label>AccountsOfManufacturingIndustry</label>
    <name>AccountsOfManufacturingIndustry</name>
    <sourceName>AccountOpportunities</sourceName>
  </filters>
  <hierarchyPaths>
    <description>Hierarchy Path Node</description>
    <hierarchyFieldName>Hierarchy_Path</hierarchyFieldName>
    <isAggregationRequired>true</isAggregationRequired>
    <isSelfFieldValueIncluded>true</isSelfFieldValueIncluded>
    <label>Get Hierarchy</label>
    <name>Get_Hierarchy</name>
    <parentFieldName>ContactId</parentFieldName>
    <selfFieldName>LastName</selfFieldName>
  </hierarchyPaths>

```



```

    <sourceName>AppendAllAccounts</sourceName>
    <aggregateFields>
      <aggregateFunction>Count</aggregateFunction>
      <aggregationFieldName>*</aggregationFieldName>
      <aggregateFieldAliasName>CountOfLastName</aggregateFieldAliasName>
    </aggregateFields>
  </hierarchyPaths>
  <isTemplate>>false</isTemplate>
  <executionPlatformObjectType>None</executionPlatformObjectType>
  <joins>
    <description>Left Outer Join</description>
    <fields>
      <alias>ContactId</alias>
      <sourceFieldName>Id</sourceFieldName>
      <sourceName>Contact</sourceName>
    </fields>
    <fields>
      <alias>LastName</alias>
      <sourceFieldName>LastName</sourceFieldName>
      <sourceName>Contact</sourceName>
    </fields>
    <fields>
      <alias>NameCount</alias>
      <sourceFieldName>NameCount</sourceFieldName>
      <sourceName>AggregateOpportunities</sourceName>
    </fields>
    <fields>
      <alias>OpportunityName</alias>
      <sourceFieldName>Name</sourceFieldName>
      <sourceName>AggregateOpportunities</sourceName>
    </fields>
    <joinKeys>
      <primarySourceFieldName>Id</primarySourceFieldName>
      <secondarySourceFieldName>ContactId</secondarySourceFieldName>
    </joinKeys>
    <label>AccountOpportunities</label>
    <name>AccountOpportunities</name>
    <primarySourceName>Contact</primarySourceName>
    <secondarySourceName>AggregateOpportunities</secondarySourceName>
    <type>LeftOuter</type>
  </joins>
  <label>Create Transaction Journals Based on Orders</label>
  <parameters>
    <dataType>Date</dataType>
    <defaultValue>2020-01-01</defaultValue>
    <description>Desc TextParameter</description>
    <isMultiValue>>false</isMultiValue>
    <label>DateParameter</label>
    <name>DateParameter</name>
  </parameters>
  <parameters>
    <dataType>Filter</dataType>
    <defaultValue>{&quot;filterCondition&quot;;: &quot;1 AND 2&quot;;,
&quot;criteria&quot;;: [{&quot;sourceFieldName&quot;;:

```

```

    &quot;NameCount&quot;;,&quot;operator&quot;;: &quot;GreaterThan&quot;;,&quot;value&quot;;:
    &quot;20&quot;;,&quot;sequence&quot;;: &quot;1&quot;;}, {&quot;sourceFieldName&quot;;:
    &quot;Name&quot;;,&quot;operator&quot;;: &quot;Equals&quot;;,&quot;value&quot;;:
    &quot;Salesforce&quot;;,&quot;sequence&quot;;: &quot;2&quot;;}}</defaultValue>
    <isMultiValue>false</isMultiValue>
    <label>FilterParameter</label>
    <name>FilterParameter</name>
  </parameters>
  <parameters>
    <dataType>Numeric</dataType>
    <defaultValue>5000</defaultValue>
    <description>Desc TextParameter</description>
    <isMultiValue>false</isMultiValue>
    <label>NumericParameter</label>
    <name>NumericParameter</name>
  </parameters>
  <parameters>
    <dataType>Text</dataType>
    <defaultValue>@salesforce.com</defaultValue>
    <description>Desc TextParameter</description>
    <isMultiValue>false</isMultiValue>
    <label>TextParameter</label>
    <name>TextParameter</name>
  </parameters>
  <processType>Rebates</processType>
  <definitionRunMode>Batch</definitionRunMode>
  <status>Inactive</status>
  <transforms>
    <description>transforms Desc</description>
    <expressionFields>
      <alias>NewLastName</alias>
      <dataType>Text</dataType>
      <expression>TODAY ()</expression>
      <length>80</length>
    </expressionFields>
    <label>ManufacturingIndustry</label>
    <name>ManufacturingIndustry</name>
    <sourceName>AccountsOfManufacturingIndustry</sourceName>
    <transformationType>Expression</transformationType>
  </transforms>
  <transforms>
    <droppedFields>
      <sourceFieldName>NewLastName</sourceFieldName>
    </droppedFields>
    <label>MediaIndustry</label>
    <name>MediaIndustry</name>
    <sourceName>ManufacturingIndustry</sourceName>
    <transformationType>Slice</transformationType>
  </transforms>
  <transforms>
    <description>compute relative transforms Desc</description>
    <expressionFields>
      <alias>NewLastName</alias>
      <dataType>Text</dataType>

```

```
<expression>rank()</expression>
<length>80</length>
</expressionFields>
<label>ComputeRelativeManufacturingIndustry</label>
<name>ComputeRelativeManufacturingIndustry</name>
<orderBy>
  <name>LastName</name>
  <orderType>Ascending</orderType>
</orderBy>
<partitionBy>LastName</partitionBy>
<sourceName>MediaIndustry</sourceName>
<transformationType>ComputeRelative</transformationType>
</transforms>
<customNodes>
  <name>RebatesCustomNode</name>
  <label>Rebates Custom Node</label>
  <description>customNodes Desc</description>
  <sources>Get_Hierarchy</sources>
  <extensionName>RebatesExpression</extensionName>
  <extensionNamespace>industries_mfg</extensionNamespace>
  <parameters>
    <name>inputColumn</name>
    <value>LastName</value>
  </parameters>
  <parameters>
    <name>isFilterCriteria</name>
    <value>true</value>
  </parameters>
  <parameters>
    <name>outputColumn</name>
    <value>GenName</value>
  </parameters>
</customNodes>
<writebacks>
  <fields>
    <sourceFieldName>GenName</sourceFieldName>
    <targetFieldName>LastName</targetFieldName>
  </fields>
  <isChangedRow>false</isChangedRow>
  <label>exportToContact</label>
  <name>exportToContact</name>
  <description>Export To Contact</description>
  <operationType>Insert</operationType>
  <sourceName>RebatesCustomNode</sourceName>
  <targetObjectName>Contact</targetObjectName>
  <writebackSequence>1</writebackSequence>
  <canWrtbckToNonEditableFields>false</canWrtbckToNonEditableFields>
</writebacks>
<writebacks>
  <fields>
    <sourceFieldName>CreatedDateYM</sourceFieldName>
    <targetFieldName>CreatedDate</targetFieldName>
  </fields>
  <isChangedRow>false</isChangedRow>
```

```
<isExistingDataset>false</isExistingDataset>
<label>exportToContactFC</label>
<name>exportToContactFC</name>
<description>Export To Contact</description>
<operationType>Insert</operationType>
<sourceName>ContactForecast</sourceName>
<targetObjectName>Contact</targetObjectName>
<writebackSequence>2</writebackSequence>
<canWrtbckToNonEditableFields>false</canWrtbckToNonEditableFields>
</writebacks>
</BatchCalcJobDefinition>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  ~ Copyright 2020 Salesforce, Inc.
  ~ All Rights Reserved
  ~ Company Confidential
-->
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
  <types>
    <members>*</members>
    <name>BatchCalcJobDefinition</name>
  </types>
  <version>60.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character * (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see [Deploying and Retrieving Metadata with the Zip File](#).

Data Processing Engine Invocable Actions

Run an active Data Processing Engine definition. For more information on custom invocable actions, see [REST API Developer Guide](#) and [Actions Developer Guide](#).

Data Processing Engine Actions

Run an active Data Processing Engine definition. This action executes a Data Processing Engine definition asynchronously.

Data Processing Engine Actions

Run an active Data Processing Engine definition. This action executes a Data Processing Engine definition asynchronously.

A Data Processing Engine definition transforms data from your Salesforce org and writes back the results to your org. For more information about running Data Processing Engine definitions, see [Run a Data Processing Engine Definition](#) in Salesforce Help.

This object is available in API version 51.0 and later.

Supported REST HTTP Methods

URI

/services/data/v~~XX.X~~/actions/custom/dataProcessingEngineAction

Formats

JSON

HTTP Methods

GET, POST

Authentication

Authorization: Bearer *token*

Inputs

Use the GET method to retrieve input variables of a Data Processing Engine definition. The input variables of each Data Processing Engine definition are unique. The Data Processing Engine action uses the input variables to execute the Data Processing Engine definition and generate a batch job ID.



Note: You can't use this action to start two simultaneous runs of a definition with the same input variables.

Outputs

Output	Details
batchJobId	Type string Description ID of the batch job generated after the request is successful. This ID is used to track the progress of the batch job in Monitor Workflow Services.

Usage

JSON Sample Request to execute PointsAccrual DPE Definition

```
{
  "PointsAccrual" : {
    "memberTier" : "Gold",
    "minimumPointBalanceRequired" : "50000",
    "pointType" : "non-qualifying"
  }
}
```

JSON Sample Response

```
{
  "actionName":"PointsAccrual",
  "errors":null,
  "isSuccess":true,
  "outputValues":{
    "batchJobId":"01Mxx0000A000001EAA"
```

```
}  
}
```

Example

GET

This example shows how to retrieve input variables of a Data Processing Engine action type.

```
curl --include --request GET \  
--header "Authorization: Authorization: Bearer 00DR...xyz" \  
--header "Content-Type: application/json" \  
"https://instance.salesforce.com/services/data/v60.0/actions/custom/dataProcessingEngineAction/newinputvardefn"
```

POST

Here's a request to retrieve DPE definition

```
{  
  "inputs": [  
    {  
      "start_date": "26-09-2023",  
      "end_date": "12-12-2023",  
      "randomkey": "069SM0000001SgbYAE"  
    }  
  ]  
}
```

Here's a response for this action.

```
[  
  {  
    "actionName": "newinputvardefn",  
    "errors": null,  
    "invocationID": null,  
    "isSuccess": true,  
    "outputValues": {  
      "batchJobId": "0mdSM0000006EJdYAM",  
      "accepted": true  
    },  
    "version": 1  
  }  
]
```

Batch Management

Automate the processing of records in scheduled flows. You can process a high volume of standard and custom object records. Batch Management consists of three Tooling API objects, a standard object, a Metadata API, and an invocable action. You can use these resources to view, create, edit, and run Batch Management jobs.

[Batch Management Tooling API Objects](#)

You can use the Batch Management Tooling API object to view the settings of Batch Management jobs.

[Batch Management Standard Objects](#)

Batch Management contains a standard BatchProcessJobDefinitionView object, which you can use to view all the batch jobs available in your Salesforce org, including file-based jobs.

[Batch Management Metadata API](#)

Use a Metadata API to create, update, and activate Batch Management jobs.

[Batch Management Invocable Actions](#)

Run an active Batch Management job definition. For more information on custom invocable actions, see [REST API Developer Guide](#) and [Actions Developer Guide](#).

EDITIONS

Available in: Lightning Experience

Available in: Batch Management is available with Enterprise, Unlimited, and Performance Editions with the Loyalty Management, Manufacturing Cloud, Rebate Management, or Accounting Subledger

Batch Management Tooling API Objects

You can use the Batch Management Tooling API object to view the settings of Batch Management jobs.

Tooling API objects let you interact with metadata for declarative development. For example, you can create your own version of Setup.

[BatchDataSource](#)

Represents the source of information from which a batch job retrieves records for processing. This object is available in API version 66.0 and later.

[BatchDataSrcFilterCriteria](#)

Represents the details of a condition in the filter criteria used to retrieve records from the data source of a batch job. This object is available in API version 66.0 and later.

[BatchProcessJobDefinition](#)

Represents the details of a Batch Management job. This object is available in API version 51.0 and later.

EDITIONS

Available in: Lightning Experience

Available in: Batch Management is available with Enterprise, Unlimited, and Performance Editions with the Loyalty Management, Manufacturing Cloud, Rebate Management, or Accounting Subledger

BatchDataSource

Represents the source of information from which a batch job retrieves records for processing. This object is available in API version 66.0 and later.

Supported Calls

`describeSObjects()`, `query()`, `retrieve()`

Fields

Field	Details
BatchJobDefinitionId	Type reference Properties Filter, Group, Sort Description The ID of the batch job definition associated with batch data source. This field is a relationship field. Relationship Name BatchJobDefinition Relationship Type Master-detail Refers To BatchJobDefinition (the master object)
CriteriaJoinCondition	Type string Properties Filter, Group, Nillable, Sort Description The logic that's used to decide how data source records are filtered.
CriteriaJoinType	Type picklist Properties Defaulted on create, Filter, Group, Restricted picklist, Sort Description Specifies the criteria type used to filter data source records. Possible values are: • <code>all</code>—All conditions are met (AND) • <code>any</code>—Any condition is met (OR) • <code>custom</code>—Customize the logic • <code>none</code>—No conditions are met The default value is <code>all</code>.
DataSourceType	Type picklist Properties Filter, Group, Nillable, Restricted picklist, Sort

Field	Details
	Description Specifies the type of data source. Possible values are: • <code>MultipleSubjects</code> • <code>SingleSubject</code>
<code>RelatedSubjects</code>	Type string Properties Filter, Group, Nillable, Sort Description The list of objects that are used as data sources for the batch job definition.
<code>SourceFieldName</code>	Type string Properties Filter, Group, Nillable, Sort Description The field from the source object that's used to run the batch job.
<code>SourceTableName</code>	Type string Properties Filter, Group, Sort Description The name of the object from which records are processed by the batch job.

BatchDataSrcFilterCriteria

Represents the details of a condition in the filter criteria used to retrieve records from the data source of a batch job. This object is available in API version 66.0 and later.

Supported Calls

`describeSObjects()`, `query()`, `retrieve()`

Fields

Field	Details
<code>BatchDataSourceId</code>	Type reference

Field	Details
	Properties Filter, Group, Sort Description The ID of the batch data source associated with the batch data source filter criteria. This field is a relationship field. Relationship Name BatchDataSource Relationship Type Master-detail Refers To BatchDataSource (the master object)
DomainObjectName	Type string Properties Filter, Group, Nillable, Sort Description The name of the object that contains the field that's used in the filter criteria condition.
DynamicValueType	Type picklist Properties Filter, Group, Nillable, Restricted picklist, Sort Description Specifies the data type of the input variable that's used in the filter criteria condition. Possible values are: • boolean • currency • date • datetime • integer • picklist • reference • string
FieldName	Type string Properties Filter, Group, Sort

Field	Details
	Description The name of the field that's used in the filter criteria condition.
FieldPath	Type string Properties Filter, Group, Nillable, Sort Description The patch of the related object field that's used in the filter criteria condition.
FieldValue	Type string Properties Filter, Group, Sort Description The value of the specified field used to filter records of the data source.
FilterCriteriaSequence	Type int Properties Filter, Group, Sort Description The sequence number of the condition in the filter criteria.
IsDynamicValue	Type boolean Properties Defaulted on create, Filter, Group, Sort Description Indicates whether the value of the filter criteria condition is provided by an input variable. The default value is <code>false</code> .
Operator	Type picklist Properties Filter, Group, Restricted picklist, Sort Description Specifies the operator used in the filter criteria condition. Possible values are: • <code>equals</code>—Equals • <code>excludes</code>—Excludes

Field	Details
	• <code>greaterThan</code>—Greater Than • <code>greaterThanOrEqualTo</code>—Greater Than Or Equal To • <code>in</code>—In • <code>includes</code>—Includes • <code>isNotNull</code>—Is Not Null • <code>isNull</code>—Is Null • <code>lessThan</code>—Less Than • <code>lessThanOrEqualTo</code>—Less Than Or Equal To • <code>like</code>—Like • <code>notEquals</code>—Not Equals • <code>notIn</code>—Not In

BatchProcessJobDefinition

Represents the details of a Batch Management job. This object is available in API version 51.0 and later.



Note: 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.

Data and processes in your org are impacted if you update or delete a BatchProcessJobDefinition record. Update or delete a Batch Management job using the [Metadata API](#) on page 884.

Supported SOAP API Calls

`describeSObjects()`, `query()`, `retrieve()`

Supported REST API Methods

DELETE, GET, HEAD, PATCH, POST, Query

Fields

Field	Details
<code>BatchJobDefinitionId</code>	Type reference Properties Filter, Group, Nillable, Sort Description The unique identifier of the associated batch job definition. This is a relationship field. Relationship Name BatchJobDefinition

Field	Details
	Relationship Type Lookup Refers To BatchJobDefinition
BatchJobDefinitionName	Type string Properties Filter, Group, Nillable, Sort Description The name of the associated batch job definition.
BatchSize	Type int Properties Filter, Group, Sort Description Required. The number of records that each Batch Management job part can process. The maximum number of transaction journal records that a batch management job can process for flow or loyalty program process is 2000.
Description	Type textarea Properties Filter, Group, Nillable, Sort Description The description of the Batch Management job.
DeveloperName	Type string Properties Filter, Group, Sort Description The developer name of the Batch Management job.
FlowDefinitionId	Type reference Properties Filter, Group, Nillable, Sort Description The input variable of the associated flow that uniquely identifies each record that the Batch Management job processes.

Field	Details
FullName	Type string Properties Create, Group, Nillable Description The name of the Batch Management job. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
Language	Type picklist Properties Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort Description The language in which the batch job is created. Possible values are: • da—Danish • de—German • en_US—English • es—Spanish • es_MX—Spanish (Mexico) • fi—Finnish • fr—French • it—Italian • ja—Japanese • ko—Korean • nl_NL—Dutch • no—Norwegian • pt_BR—Portuguese (Brazil) • ru—Russian • sv—Swedish • th—Thai • zh_CN—Chinese (Simplified) • zh_TW—Chinese (Traditional)
ManageableState	Type ManageableState enumerated list

Field	Details
	Properties Filter, Group, Nillable, Restricted picklist, Sort Description Indicates the manageable state of the specified component that is contained in a package: • beta • deleted • deprecated • deprecatedEditable • installed • installedEditable • released • unmanaged
MasterLabel	Type string Properties Filter, Group, Sort Description The label of the Batch Management job.
Metadata	Type complexvalue Properties Create, Nillable, Update Description The Batch Management job's metadata. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
NamespacePrefix	Type string Properties Filter, Group, Nillable, Sort Description The namespace prefix that is associated with this object. Each Developer Edition org that creates a managed package has a unique namespace prefix. Limit: 15 characters. You can refer to a component in a managed package by using the <i>namespacePrefix__componentName</i> notation. The namespace prefix can have one of the following values.

Field	Details
	• In Developer Edition orgs, <code>NamespacePrefix</code> is set to the namespace prefix of the org for all objects that support it, unless an object is in an installed managed package. In that case, the object has the namespace prefix of the installed managed package. This field's value is the namespace prefix of the Developer Edition org of the package developer. • In orgs that are not Developer Edition orgs, <code>NamespacePrefix</code> is set only for objects that are part of an installed managed package. All other objects have no namespace prefix.
ProcessGroup	Type string Properties Filter, Group, Nillable, Sort Description The group or team for which the Batch Management job processes records.
RecordIdVariable	Type string Properties Filter, Group, Nillable, Sort Description The unique identifier that identifies each record that must be processed by the Batch Management job.
RetryCount	Type int Properties Defaulted on create, Filter, Group, Sort Description Required. The number of times this Batch Management job must be rerun in case it fails.
RetryInterval	Type int Properties Defaulted on create, Filter, Group, Sort Description Required. The number of milliseconds after which the Batch Management job must be rerun in case it fails. A retry interval can be 1,000–10,000 milliseconds.
Status	Type picklist Properties Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort

Field	Details
	Description The status of the Batch Management job. Possible values are: • Active • Inactive
Type	Type picklist Properties Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort Description The type of process for which the Batch Management job processes records. Possible values are: • Flow • BulkUpdate • ConsumptionOveragesCalculation • DecisionTableRefresh • DeepCloneSalesAgreement • EntitlementCreationBatchJob • HighScaleBreProcess • IndustriesLSCommercial • LoyaltyProgramProcess • ManagerProvisioning • NetUnitRateCalculation • PbbToOptyConversion • ProductCatalogCacheRefresh • RatableSummaryCreation • SummaryCreation The default value is <code>Flow</code>. Other types may be available to you depending on the licenses available in your org. This field is available in API version 55.0 and later.
TypeInstance	Type string Properties Filter, Group, Sort Description Required. The API name of the process that the Batch Management job must execute.

Batch Management Standard Objects

Batch Management contains a standard BatchProcessJobDefinitionView object, which you can use to view all the batch jobs available in your Salesforce org, including file-based jobs.

[BatchProcessJobDefView](#)

Represents the details of a Batch Job definition. The definition can also be file-based definitions that are available in your Salesforce org. This object is available in API version 51.0 and later.

BatchProcessJobDefView

Represents the details of a Batch Job definition. The definition can also be file-based definitions that are available in your Salesforce org. This object is available in API version 51.0 and later.

Supported Calls

`describeSObjects(), query()`

Fields

Field	Details
DurableId	Type string Properties Filter, Group, Nillable, Sort Description The unique identifier for the field. Always retrieve this value before using it, as the value isn't guaranteed to stay the same from one release to the next. Simplify queries by using this field instead of making multiple queries.
IsActive	Type boolean Properties Defaulted on create, Filter, Group, Sort Description Indicates whether the definition is active.
Label	Type string Properties Filter, Group, Nillable, Sort Description The label of the Batch Job definition.

EDITIONS

Available in: Lightning Experience

Available in: Batch Management is available with Enterprise, Unlimited, and Performance Editions with the Loyalty Management, Manufacturing Cloud, Rebate Management, or Accounting Subledger

Field	Details
Name	Type string Properties Filter, Group, Nillable, Sort Description The name of the Batch Job definition.
NamespacePrefix	Type string Properties Filter, Group, Nillable, Sort Description The namespace prefix associated with this object. Each Developer Edition organization that creates a managed package has a unique namespace prefix. Limit: 15 characters. You can refer to a component in a managed package by using the namespacePrefix__componentName notation. The namespace prefix can have one of the following values: • In Developer Edition organizations, the namespace prefix is set to the namespace prefix of the organization for all objects that support it. There is an exception if an object is in an installed managed package. In that case, the object has the namespace prefix of the installed managed package. This field's value is the namespace prefix of the Developer Edition organization of the package developer. • In organizations that are not Developer Edition organizations, NamespacePrefix is only set for objects that are part of an installed managed package. There is no namespace prefix for all other objects.
ProcessDefinition	Type textarea Properties Nillable Description The name of the process group for the batch process job definition.
ProcessGroup	Type string Properties Filter, Group, Nillable, Sort Description The team or group for which the definition processes records.
SourceObjectName	Type string

Field	Details
	Properties Filter, Group, Nillable, Sort Description The API name of the object whose records are processed.
Type	Type string Properties Filter, Group, Nillable, Sort Description The type of process for which the Batch Management job processes records. Possible values are: - Flow - LoyaltyProgramProcess This field is available in API version 55.0 and later.
TypeInstance	Type string Properties Filter, Group, Nillable, Sort Description The API name of the process that's processed by the Batch Job definition.

Batch Management Metadata API

Use a Metadata API to create, update, and activate Batch Management jobs.

[BatchProcessJobDefinition](#)

Represents the details of a Batch Management job definition.

BatchProcessJobDefinition

Represents the details of a Batch Management job definition.

This type extends the Metadata metadata type and inherits its `fullName` field.

 **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.

EDITIONS

Available in: Lightning Experience

Available in: Batch Management is available with Enterprise, Unlimited, and Performance Editions with the Loyalty Management, Manufacturing Cloud, Rebate Management, or Accounting Subledger

File Suffix and Directory Location

BatchProcessJobDefinition components have the suffix `.batchProcessJobDefinition` and are stored in the `batchProcessJobDefinitions` folder.

Version

BatchProcessJobDefinition components are available in API version 51.0 and later.

Special Access Rules

To use this metadata type, your Salesforce org must have the Loyalty Management or the Rebate Management license. The Loyalty Program Process type is only available in orgs that have Loyalty Management enabled.

Fields

Field Name	Field Type	Description
<code>batchSize</code>	integer	Required. Number of records that each Batch Management job can process. Flow type Batch Management jobs can process up to 2000 records and Loyalty Program Process type Batch Management jobs can process up to 250 records.
<code>dataSource</code>	BatchDataSource on page 886[]	Required. Source of information whose records must be processed by the Batch Management job.
<code>description</code>	string	Description of the Batch Management job, up to 255 characters.
<code>executionProcessApiName</code>	string	API name of process that must be executed by the Batch Management job. This field is available in API version 55.0 and later. • If the batch job's type is Flow, enter the API name of an active flow that the batch job must execute. • If the batch job's type is Loyalty Program Process, enter: – Transaction_Journals if you want the batch job to process Transaction Journal records by applying the applicable active loyalty program process of the type TransactionJournal. – API name of an active loyalty process of the type TierProcessing if you want the batch job to run the loyalty program process to assess the tier of eligible members. The API name consists of the name of the process, the process type, and the name of the loyalty program separated by two consecutive underscores. For example, the process API name is <code>Update_Member_Tier__TierProcessing__Inner_Circle</code> if the process name is Update Member Tier, the process type is TierProcessing, and the loyalty program name is Inner Circle.

You can use database-based APEX classes that let you use flex queues in the Batch Management job, allowing to place more than 5 jobs in a queue. This functionality is applicable to all Industry Clouds that use managed packages. See [Apex Flex Queue](#).

Field Name	Field Type	Description
<code>flowApiName</code>	string	API name of an active flow process that must be executed by the Batch Management job. You can either specify the flow API name in the <code>executionProcessApiName</code> field or in the <code>flowApiName</code> field.
<code>flowInputVariable</code>	string	Input variable of associated flow that is used by the batch job to uniquely identify records.
<code>masterLabel</code>	string	Required. Name of the Batch Management job, up to 80 characters.
<code>processGroup</code>	string	Required. Name of the group for which the Batch Management job processes records.
<code>retryCount</code>	integer	Required. Number of times this Batch Management job must be rerun in case it fails. The maximum retry count is 3. Valid values are 1–3.
<code>retryInterval</code>	integer	Required. Number of milliseconds after which the Batch Management job must be rerun in case it fails. Valid values are 1,000–10,000.
<code>status</code>	string	Indicates the status of the Batch Management job. Valid values are <code>Active</code> and <code>Inactive</code> .
<code>type</code>	string (enumeration of type string)	The type of process that the Batch Management job must execute. This field is available in API version 55.0 and later. Valid values are: • <code>Flow</code> • <code>Loyalty Program Process</code>

BatchDataSource

Represents the source of information whose records must be processed by the Batch Management job.

Fields

Field Name	Field Type	Description
<code>condition</code>	string	Required. Criteria defined to filter the records.
<code>criteria</code>	string	Type of filter criteria that's used to filter records for processing.
<code>dataSourceType</code>	string	Type of data source that's used to create the batch job definition. Valid values are: • <code>SingleSubject</code> • <code>MultiSubject</code> Available in API version 64.0 and later.
<code>filters</code>	BatchDataSrcFilterCriteria on page 887[]	Filter criterion that decides which records must be processed by the Batch Management job.

Field Name	Field Type	Description
orderFields	BatchDataSourceOrderField on page 888	Fields that are used to order the records before the records are added to a batch in a job.
sourceObject	string	Required. API name of an object whose records must be processed by the batch job. If the batch job type is Loyalty Program Process, the source object must be: • TransactionJournal if the batch job is used to process transaction journals by applying the applicable loyalty program process. • An object that stores the details of loyalty program members whose tier must be assessed by the loyalty program process specified in the executionProcessApiName field.
sourceObjectField	string	API name of the source object field that uniquely identifies records for which the batch job is executed. This field is available in API version 57.0 and later. This field is only applicable when the batch job's type is Loyalty Program Process and a TierProcess type active loyalty program process is specified in the executionProcessApiName field. Specify the API name of a field that is a lookup to the LoyaltyProgramMember object and uniquely identifies the members whose tier must be assessed.

BatchDataSrcFilterCriteria

Represents the filter conditions that decide which records must be processed by the Batch Management job.

Fields

Field Name	Field Type	Description
domainObjectName	string	Name of the object the field is associated with. Available in API version 64.0 and later.
dynamicValueType	string	Data type of the input variable used as a filter.
fieldName	string	Required. Name of the field that must be used to filter records.
fieldPath	string	Stores the path to a field in the object. Available in API version 64.0 and later.
fieldValue	string	Required. Value of the field that must be filtered. Specify the field if isDynamicValue is set to False.
isDynamicValue	boolean	Required. Indicates whether the filter criteria is dynamic.
operator	string (enumeration of type string)	Required. Operator that is specified in the filter criteria. Valid values are: • equals • excludes • greaterThan • greaterThanOrEqualTo

Field Name	Field Type	Description
		• in • includes • lessThan • LessThanOrEqualTo • GreaterOrEqual • like • notEquals • notIn
sequenceNo	integer	Required. Sequence number used to refer the criteria in a filter.

BatchDataSourceOrderField

Represents the fields that are used to group data.

Fields

Field Name	Field Type	Description
domainObjectName	string	Required. Name of the object the field is associated with. Available in API version 64.0 and later.
fieldName	string	Required. Name of the field that must be used to filter records. Available in API version 64.0 and later.
fieldPath	string	Required. Stores the path to a field in the object. Available in API version 64.0 and later.

Declarative Metadata Sample Definition

The following is an example of a BatchProcessJobDefinition component.

```
<?xml version="1.0" encoding="UTF-8"?>
<BatchProcessJobDefinition xmlns="http://soap.sforce.com/2006/04/metadata">
  <batchSize>10</batchSize>
  <dataSource>
    <condition>1</condition>
    <criteria>all</criteria>
    <filters>
      <dynamicValue>>false</dynamicValue>
      <dynamicValueType>string</dynamicValueType>
      <fieldName>Name</fieldName>
      <fieldValue>abcd</fieldValue>
      <operator>equals</operator>
      <sequenceNo>1</sequenceNo>
    </filters>
  </dataSource>
</BatchProcessJobDefinition>
```



```

    <sourceObject>Account</sourceObject>
  </dataSource>
  <flowApiName>Flow1</flowApiName>
  <flowInputVariable>recordId</flowInputVariable>
  <masterLabel>BatchJob1</masterLabel>
  <processGroup>Loyalty</processGroup>
  <retryCount>2</retryCount>
  <retryInterval>1000</retryInterval>
  <status>Inactive</status>
  <description>test</description>
  <type>Flow</type>
  <executionProcessApiName>testFlow</executionProcessApiName>
</BatchProcessJobDefinition>

```

The following is an example of a Flow object used in Metadata API.

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
  ~ Copyright 2020 Salesforce, Inc.
  ~ All Rights Reserved
  ~ Company Confidential
-->
<Flow xmlns="http://soap.sforce.com/2006/04/metadata">
  <apiVersion>51.0</apiVersion>
  <interviewLabel>Flow1 {!$Flow.CurrentDateTime}</interviewLabel>
  <label>Flow1</label>
  <processMetadataValues>
    <name>BuilderType</name>
    <value>
      <stringValue>LightningFlowBuilder</stringValue>
    </value>
  </processMetadataValues>
  <processMetadataValues>
    <name>OriginBuilderType</name>
    <value>
      <stringValue>LightningFlowBuilder</stringValue>
    </value>
  </processMetadataValues>
  <processType>AutoLaunchedFlow</processType>
  <recordLookups>
    <name>getAcc</name>
    <label>getAcc</label>
    <locationX>614</locationX>
    <locationY>465</locationY>
    <assignNullValuesIfNoRecordsFound>false</assignNullValuesIfNoRecordsFound>
    <filterLogic>and</filterLogic>
    <filters>
      <field>Id</field>
      <operator>EqualTo</operator>
      <value>
        <elementReference>recordId</elementReference>
      </value>
    </filters>
    <getFirstRecordOnly>true</getFirstRecordOnly>
  </recordLookups>

```

```
<object>Account</object>
<storeOutputAutomatically>true</storeOutputAutomatically>
</recordLookups>
<start>
  <locationX>73</locationX>
  <locationY>213</locationY>
  <connector>
    <targetReference>getAcc</targetReference>
  </connector>
</start>
<status>Draft</status>
<variables>
  <name>recordId</name>
  <dataType>String</dataType>
  <isCollection>false</isCollection>
  <isInput>true</isInput>
  <isOutput>false</isOutput>
</variables>
</Flow>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
  <types>
    <members>*</members>
    <name>BatchProcessJobDefinition</name>
  </types>
  <types>
    <members>Flow1</members>
    <name>Flow</name>
  </types>
  <version>51.0</version>
</Package>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character * (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see [Deploying and Retrieving Metadata with the Zip File](#).

Batch Management Invocable Actions

Run an active Batch Management job definition. For more information on custom invocable actions, see [REST API Developer Guide](#) and [Actions Developer Guide](#).

Batch Job Actions

Run an active Batch Management job definition. This action executes a defined Batch Management job asynchronously.

Submit Failed Records Batch Job

Run to resubmit an existing batch job with failed records for processing. This action executes the batch job asynchronously.

Batch Job Actions

Run an active Batch Management job definition. This action executes a defined Batch Management job asynchronously.

A Batch Management job processes a flow in manageable parts. For more information about running an active Batch Management jobs, see [Schedule a Batch Job](#) in Salesforce Help.

This object is available in API version 51.0 and later.

Supported REST HTTP Methods

- URI
- /services/data/v~~XX.X~~/actions/custom/batchJobAction
- Formats
- JSON
- HTTP Methods
- GET, POST
- Authentication
- Authorization: Bearer *token*

Inputs

The batch job action uses the batch job definition ID and input variables to execute the job and generate a batch job ID. The input values vary according to the input variables in that flow.

Outputs

Output	Details
batchJobId	Type string Description ID of the batch job generated after the request is successful. This batch job ID is used to track the progress of the batch job in Monitor Workflow Services.

Usage

A request body is always required. The input values vary according to the input variables in that flow. If your batch job doesn't require any input variables, you still must send an empty JSON body.

```
{
  "inputs": [{}]
}
```

JSON Sample Request

```
{
  "noOfEmployees" : 900,
  "accountIndustry" : "Technology"
}
```

JSON Sample Response

```
{
  "batchJobId": "01Mxx0000A000001EAA"
  "accepted": "true"
}
```

Submit Failed Records Batch Job

Run to resubmit an existing batch job with failed records for processing. This action executes the batch job asynchronously. This object is available in API version 52.0 and later.

Supported REST HTTP Methods

- URI
 - /services/data/v`xx.x`/actions/standard/submitFailedRecordsBatchJob
- Formats
 - JSON
- HTTP Methods
 - GET, POST
- Authentication
 - Authorization: Bearer *token*

Inputs

Input	Details
failedRecordIds	Type - array Description - The IDs of failed records in a batch job.
parentBatchJobId	Type - string Description - Required. The ID of a batch job with failed records.

Outputs

Output	Details
batchJobId	Type string

Output	Details
	Description The ID of a batch job generated after the request is successful. This batch job ID is used to track the progress of the batch job in Monitor Workflow Services in the org.
status	Type string Description Indicates whether a batch job succeeded or failed.

Usage

JSON Sample Request

```
{
  "inputs": [{
    "parentBatchJobId": "0mdRM0000004DXrYAM",
    "failedRecordIds": [
      "001RM0000005AG0bYAG", "001RM0000005AERZYA4", "001RM0000005AG0WYAW"
    ]
  }]
}
```

JSON Sample Response

```
[ {
  "actionName" : "submitFailedRecordsBatchJob",
  "errors" : null,
  "isSuccess" : true,
  "outputValues" : {
    "batchJobId" : "0mdRM0000004DZ9YAM"
  }
} ]
```

Monitor Workflow Services

The Monitor Workflow Services standard objects can be used to track the run of Data Processing Engine definitions and Batch Management jobs. During a run, you can view details about each part that the run is broken down into. After the run is complete, you can view its status and the records which weren't processed during the run.

The objects of Monitor Workflow Services aren't available in Object Manager of your Salesforce org.

[BatchJob](#)

Represents an instance of a batch job that is either running and has been run. This object is available in API version 51.0 and later.

[BatchJobPart](#)

Represents one part of a batch job. This object is available in API version 51.0 and later.

EDITIONS

Available in: Lightning Experience

Available in: Monitor Workflow Services is available with Enterprise, Unlimited, and Performance Editions where Data Processing Engine or Batch Management is available

[BatchJobPartFailedRecord](#)

Represents records that a batch job part couldn't successfully process. This object is available in API version 51.0 and later.

BatchJob

Represents an instance of a batch job that is either running and has been run. This object is available in API version 51.0 and later.

Supported Calls

`delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`

Fields

Field	Details
<code>AdditionalInformation</code>	Type textarea Properties Create, Nillable, Update Description A JSON that contains additional context about the batch jon.
<code>BatchJobDefinitionId</code>	Type reference Properties Filter, Group, Sort Description The unique identifier of the associated batch job definition. This is a relationship field. Relationship Name BatchJobDefinition Relationship Type Lookup Refers To BatchJobDefinition
<code>BatchJobDefinitionName</code>	Type string Properties Filter, Group, Sort Description The developer name of the associated batch job definition.

Field	Details
EndTime	Type dateTime Properties Filter, Nillable, Sort Description The timestamp when the batch job run was completed.
ErrorDescription	Type string Properties Filter, Nillable, Sort Description The error message in case the batch job run failed.
ExecutionStage	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description Specifies the stage at which the batch job's run failed. This field is available in API version 66.0 and later. Possible values are: • Datasync • Execution • Preprocessing • Writeback
ExternalReference	Type string Properties Filter, Group, Nillable, Sort Description The unique identifier of the process that's running or has run the batch job.
IsDebugOn	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether debug mode was turned on (true) or not (false) when a definition was run.

Field	Details
	The default value is <code>false</code> .
IsDebugRecipeDeleted	Type boolean Properties Create, Defaulted on create, Filter, Group, Sort, Update Description Indicates whether debug recipes and datasets were deleted (true) or not (false). When the IsDebugEnabled is set to True, and the definition is run, after 7 days IsDebugRecipeDeleted is automatically set to True, and debug recipes and datasets are deleted. The default value is <code>false</code> .
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The timestamp for when the current user last viewed the batch job.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The timestamp for when the current user last viewed this item.
Name	Type string Properties Filter, Group, idLookup, Sort Description The name of the batch job.
OwnerId	Type reference Properties Filter, Group, Sort Description Unique identifier of the user who initiated the batch job run. This is a polymorphic relationship field.

Field	Details
	Relationship Name Owner
	Relationship Type Lookup
	Refers To Group, User
ProcessGroup	Type string
	Properties Filter, Group, Sort
	Description The group or team for which the batch job is run.
RetryCount	Type int
	Properties Filter, Group, Nillable, Sort
	Description The number of times the batch job run is automatically rerun in case it fails.
RuntimeParameter	Type textarea
	Properties Nillable
	Description The values of the input variables that are used as filter criteria in a Batch Management job.
StartTime	Type dateTime
	Properties Filter, Nillable, Sort
	Description The timestamp when the batch job run was started.
Status	Type picklist
	Properties Filter, Group, Restricted picklist, Sort
	Description The status of the batch job run.

Field	Details
	Possible values are: • Canceled • Completed • CompletedWithFailures • Failed • InProgress • Queued • QueueingInProgress • Submitted
TotalInputRecordCount	Type int Properties Create, Filter, Group, Nillable, Sort, Update Description The total number of records that were provided as input to the batch job. This field is available in API version 66.0 and later.
TotalProcessedRecordCount	Type int Properties Filter, Group, Nillable, Sort Description The total number of records that were processed by all the batch job parts associated with the batch job. This field is available in API version 66.0 and later.
Type	Type picklist Properties Defaulted on create, Filter, Group, Restricted picklist, Sort Description The type of batch job that is run. Possible values are: • Calc—Data Processing Engine • DecisionTableRefresh • Flow • DeepCloneSalesAgreement • ManagerProvisioning The process types available to you vary depending on the licenses available in your org.

Field	Details
UtilisedExecutionLimit	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The CRM Analytics or Data 360 execution capacity utilized by Data Processing Engine batch jobs before the current run started. This field is available in API version 66.0 and later.
UtilisedWritebackLimit	Type string Properties Create, Filter, Group, Nillable, Sort, Update Description The CRM Analytics or Data 360 writeback capacity utilized by Data Processing Engine batch jobs before the current run started. This field is available in API version 66.0 and later.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

BatchJobFeed

Feed tracking is available for the object.

BatchJobHistory

History is available for tracked fields of the object.

BatchJobPart

Represents one part of a batch job. This object is available in API version 51.0 and later.

When a batch job is run, it is divided in to multiple parts. Each part is used to process a specific number of records.

Supported Calls

`describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`

Fields

Field	Details
BatchJobId	Type reference Properties Filter, Group, Sort

Field	Details
	Description The unique identifier of the associated batch job. This is a relationship field. Relationship Name BatchJob Relationship Type Lookup Refers To BatchJob
EndTime	Type dateTime Properties Filter, Nillable, Sort Description The timestamp when the batch job part was processed.
ErrorDescription	Type string Properties Filter, Nillable, Sort Description The error message in case the batch job part failed.
FailedRecFileBody	Type base64 Properties Nillable Description Contains the details of the records that the batch job part failed to process.
FailedRecFileContentType	Type picklist Properties Filter, Group, Nillable, Restricted picklist, Sort Description Shows the type of data that the batch job part failed to process. For example, application/html or text/csv.
FailedRecFileLength	Type int

Field	Details
	Properties Filter, Group, Nillable, Sort Description The character length of the failed record file.
FailedRecFileName	Type string Properties Filter, Group, Nillable, Sort Description The name of the file that contains the details of the failed records.
FailedRecordCount	Type int Properties Filter, Group, Nillable, Sort Description The number of records that the batch job part couldn't process.
FailedRowCount	Type long Properties Filter, Group, Nillable, Sort Description The number of records that were processed but the batch job part failed to write back for a Data Processing Engine definition run. This field is available in API version 66.0 and later.
InputRecordCount	Type int Properties Filter, Group, Nillable, Sort Description The number of records that the batch job part must process.
InputRowCount	Type long Properties Filter, Group, Nillable, Sort Description The number of records that were submitted to the batch job part for a Data Processing Engine definition run. This field is available in API version 66.0 and later.

Field	Details
Name	Type string Properties Filter, Group, idLookup, Sort Description The name of the batch job part.
OutputRecordCount	Type int Properties Filter, Group, Nillable, Sort Description The number of records the batch job part has processed.
OutputRowCount	Type long Properties Filter, Group, Nillable, Sort Description The number of records that were processed by the batch job part for a Data Processing Engine definition run. This field is available in API version 66.0 and later.
ParentBatchJobPartId	Type reference Properties Filter, Group, Nillable, Sort Description The unique identifier of the part batch job part associated with the batch job part. This is a relationship field. Relationship Name ParentBatchJobPart Relationship Type Lookup Refers To BatchJobPart
RecordFileBody	Type base64 Properties Nillable

Field	Details
	Description Contains the details of the records that the batch job part processed.
RecordFileContentType	Type picklist Properties Filter, Group, Nillable, Restricted picklist, Sort Description Shows the type of data that the batch job part processed. For example, application/html or text/csv.
RecordFileLength	Type int Properties Filter, Group, Nillable, Sort Description The character length of the file that contains the records that the batch job part processed.
RecordFileName	Type string Properties Filter, Group, Nillable, Sort Description The name of the file that contains the details of the records that the batch job part processed.
RetryCount	Type int Properties Filter, Group, Nillable, Sort Description The number of times the batch job part is automatically rerun in case it fails.
StartTime	Type dateTime Properties Filter, Nillable, Sort Description The timestamp when the batch job part's run was started.
Status	Type picklist

Field	Details
	Properties Defaulted on create, Filter, Group, Restricted picklist, Sort Description The status of the batch job part. Possible values are: • Canceled • Completed • Failed • InProgress • New • Waiting
Type	Type picklist Properties Filter, Group, Nillable, Restricted picklist, Sort Description The type of node in case the associated batch job is of the type Calc (Data Processing Engine). Possible values are: • Aggregate • Analysis • Append • AtomicWriteback • Compute • CsvIngestion • Custom • Datasync • Execution • Filter • Forecast • Hierarchy • Join • OutputRecordsNode • Register • Slice • Source • Summary • Transform

Field	Details
	Writeback
UserReference	Type string Properties Filter, Group, Nillable, Sort Description The ID of the user who is assigned as the writeback user in the Writeback Object node of the Data Processign Engine definition for which the batch job part has written back results. This field is available in API version 66.0 and later.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

BatchJobPartFeed

Feed tracking is available for the object.

BatchJobPartHistory

History is available for tracked fields of the object.

BatchJobPartFailedRecord

Represents records that a batch job part couldn't successfully process. This object is available in API version 51.0 and later.

Supported Calls

`describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`

Fields

Field	Details
BatchJobId	Type reference Properties Filter, Group, Nillable, Sort Description The unique identifier of the associated batch job. This is a relationship field. Relationship Name BatchJob

Field	Details
	Relationship Type Lookup Refers To BatchJob
BatchJobPartId	Type reference Properties Filter, Group, Sort Description The unique identifier of the associated batch job part. This is a relationship field. Relationship Name BatchJobPart Relationship Type Lookup Refers To BatchJobPart
ErrorDescription	Type string Properties Filter, Nillable, Sort Description The error message that indicates why the batch job part couldn't process the records.
Name	Type string Properties Filter, Group, idLookup, Sort Description The name of the batch job part failed record.
Record	Type string Properties Filter, Group, Nillable, Sort Description The unique identifier of the batch record that processed the failed records.

Field	Details
RecordName	Type string Properties Filter, Group, Nillable, Sort Description The name of the record that's associated with the batch job part failed record.
ResubmittedBatchJobId	Type reference Properties Filter, Group, Nillable, Sort Description The batch job used to submit failed records. This field is available in API version 52.0 and later. This is a relationship field. Relationship Name ResubmittedBatchJob Relationship Type Lookup Refers To BatchJob
Status	Type picklist Properties Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort Description Specifies the status of the failed records. This field is available in API version 52.0 and later. Possible values are: - Failed - Resubmitted The default value is 'Failed'.

Associated Objects

This object has the following associated objects. If the API version isn't specified, they're available in the same API versions as this object. Otherwise, they're available in the specified API version and later.

BatchJobPartFailedRecordFeed

Feed tracking is available for the object.

BatchJobPartFailedRecordHistory

History is available for tracked fields of the object.

CHAPTER 14 StandardValueSet Names and Standard Picklist Fields for Loyalty Management

This table lists the names of standard picklists as standard value sets and their corresponding field names.



Note: The names of standard value sets and picklist fields are case-sensitive.

Standard Value Set Name	Field Name
LoyaltyTierChangeType	LoyaltyMemberTier.ChangeReasonType
BookingChannel	TransactionJournal.BookingFareClass
RoomType	TransactionJournal.BookingRoomType
LPPartnerPartnerIndustry	TransactionJournal.Industry
JournalReason	TransactionJournal.JournalReason
PaymentMethod	TransactionJournal.PaymentMethod
PriceType	TransactionJournal.PriceType