Chatter REST API Developer's Guide

Version 34.0, Summer '15

@salesforcedocs

Last updated: July 13, 2015

© Copyright 2000–2015 salesforce.com, inc. All rights reserved. Salesforce is a registered trademark of salesforce.com, inc., as are other names and marks. Other marks appearing herein may be trademarks of their respective owners.

CONTENTS

Chapter 1: Introducing Chatter REST API 1

Deciding When to Use Chatter REST API 2
Chatter REST API Architecture 2
Chatter REST API Rate Limits 3
Building the Chatter REST API Resource URL 4
Sending HTTP Requests 5
Understanding a Chatter REST API Request and Response 6
Using Chatter REST API Inputs 10
Wildcards 19
Specifying Response Sizes 20
- filterGroup 20
- exclude 22
- include 24

Response Body Encoding 26
Status Codes and Error Responses 26
Understanding OAuth and Chatter REST API 28
- Understanding the Web Server OAuth Authentication Flow 30
- Understanding the User-Agent OAuth Authentication Flow 35
- Using Tokens 37
- Requesting Static Assets and Posting Forms Directly from HTML 38

Use CORS to Access Chatter REST API and REST API 40

Chapter 2: Chatter REST API Quick Start 41

Prerequisites 42
Step One: Obtain a Salesforce Developer Edition Organization 42
Step Two: Set Up Authorization 42
Step Three: Connect to Chatter REST API Using OAuth 43
Connecting to Salesforce Communities 44

Chapter 3: Chatter REST API Features 46

Working with Action Links 47
- Action Links Use Cases 49
- Action Links Overview, Authentication, and Security 51
- Action Link Templates 55

Working with Communities 68
Working with Feeds and Feed Elements 70
Working with Salesforce Files 76

Chapter 4: Chatter REST API Examples 79

Filter a Response Body . . . 80
Make a Request in a Community . . . 81
Post a Feed Item . . . 81
Post a Feed Element and Include a File from Salesforce . . . 82
Post a Comment . . . 83
Post a Comment and Include a File from Salesforce . . . 83
Post a Binary File . . . 84
Post an @Mention . . . 84
Post Multiple @Mentions . . . 85
Get a List of @Mention Proposals . . . 87
Define Action Links and Post with a Feed Element . . . 87
Define Action Links in a Template and Post with a Feed Element . . . 90
Update the Status of an Action Link or Invoke an Action Link . . . 93
Edit a Feed Post . . . 94
Edit a Comment . . . 95
Edit a Question Title and Post . . . 95
Like a Feed Item . . . 96
Like a Feed Item and Return the Total Number of Likes . . . 96
Share a Feed Item . . . 97
Get Feed Elements Updated Since Last Request . . . 97
Get My News Feed . . . 98
Get a Group Feed . . . 98
Get a Feed with Internal Feed Items Only . . . 99
Search a Feed . . . 99
Get a List of Records a User Follows . . . 99
Get Chatter Activity Statistics for a User . . . 100
Get Recommendations of Users to Follow . . . 100
Create a Group . . . 100
Update and Crop a Group Photo . . . 101
Join a Group . . . 101
Request to Join a Private Group . . . 102
Approve or Reject a Request to Join a Private Group . . . 102
Post to a Group . . . 103
Follow a Record . . . 103
Stop Following a Record . . . 104
Get a Record Feed . . . 104
Update a User Profile . . . 104
Update and Crop a User Photo . . . 105
Send a Private Message . . . 106
Get User Information for Multiple Users . . . 106

Chapter 5: Chatter REST API Resources . . . 107

Action Links Resources . . . 109
Action Link Group Definitions . . . 110

Action Link Group Definition . . . 113
Action Link Group . . . 113
Action Link . . . 114
Action Link Diagnostic Information . . . 117
Announcements Resource . . . 117
Batch Resource . . . 119
Chatter Resource . . . 124
Comments Resources . . . 125
Communities Resources . . . 128
Communities Moderation Resources . . . 129
Connect Resources . . . 136
Dashboard Component Snapshot Resource . . . 137
Datacloud Resources . . . 139
Datacloud Company Information . . . 139
Datacloud Contact Information . . . 139
Datacloud OrderId Information . . . 140
Datacloud Company Order Information . . . 140
Datacloud Contact Order Information . . . 141
Datacloud Orders . . . 142
Datacloud Usage . . . 143
Feed Elements Resources . . . 143
Feed Elements, Post and Search . . . 145
Feed Elements, Batch Post . . . 148
Feed Elements, Batch Get . . . 151
Feed Element . . . 151
Feed Elements Capabilities . . . 152
Feed Elements Capability, Approval . . . 153
Feed Elements Capability, Associated Actions . . . 153
Feed Elements Capability, Banner . . . 154
Feed Elements Capability, Bookmarks . . . 154
Feed Elements Capability, Bundle . . . 155
Feed Elements Capability, Bundle Elements . . . 156
Feed Elements Capability, Canvas . . . 157
Feed Elements Capability, Case Comment . . . 157
Feed Elements Capability, Chatter Likes . . . 158
Feed Elements Capability, Chatter Likes Items . . . 158
Feed Elements Capability, Comments . . . 159
Feed Elements Capability, Comments Items . . . 159
Feed Elements Capability, Content . . . 161
Feed Elements Capability, Dashboard Component Snapshot . . . 161
Feed Elements Capability, Editability . . . 162
Feed Elements Capability, Email Message . . . 162
Feed Elements Capability, Enhanced Link . . . 163
Feed Elements Capability, Link . . . 163

Feed Elements Capability, Moderation 163
Feed Elements Capability, Origin 165
Feed Elements Capability, Poll 165
Feed Elements Capability, Question and Answers 166
Feed Elements Capability, Recommendations 167
Feed Elements Capability, Record Snapshot 168
Feed Elements Capability, Topics 168
Feed Elements Capability, Tracked Changes 170
Feed Items Resources 170
Feed-Items Batch 170
Feed-Items Query 171
Feed-Items, Specific Feed Item 172
Feed Items Comments 173
Feed-Items Likes 175
Feed-Items Polls 176
Feed-Items Topics 177
Feeds Resources 178
Bookmarks Feed Resources 179
Communities Home Feed Resource 185
Communities Moderation Feed Resources 186
Company Feed Resources 189
Favorites Feed Resources 194
Feeds Resource 200
Files Feed Resources 200
Filter Feed Resources 206
Groups Feed Resources 213
News Feed Resources 219
People Feed Resources 236
Record Feed Resource 242
To Feed Resources 258
Topics Feed Resources 264
User Profile Feed Resources 267
Files Connect Repository Resources 282
Files Connect Repository List 282
Files Connect Repository 283
Files Connect Repository File 283
Files Connect Repository File Content 284
Files Connect Repository Folder Items 284
Files Resources 285
File Information 285
File Content 288
File Shares 288
Files Shares Link 290
File Rendition 291

File Information, Batch . . . 292
Folders Resources . . . 292
Groups Resources . . . 296
List of Groups . . . 297
Group Information . . . 300
Batch Group Information . . . 303
Group Announcements . . . 304
Group Files . . . 306
Group Members . . . 307
Group Members—Private . . . 309
Group Membership Requests—Private . . . 312
Group Memberships Resources . . . 313
Group Photo . . . 316
Group Records . . . 319
Group Record . . . 320
Group Settings . . . 320
Group Topics . . . 322
Influence Resource . . . 322
Likes Resource . . . 323
Managed Topics Resources . . . 323
Managed Topics . . . 324
Managed Topic . . . 328
Mentions Resources . . . 328
Mentions Completions . . . 329
Mentions Validations . . . 331
Motifs Resources . . . 332
Push Notifications Resource . . . 334
Question and Answers Suggestions Resource . . . 335
Recommendation Definition Resources . . . 336
Recommendation Definitions . . . 336
Recommendation Definition . . . 338
Recommendation Definition Photo . . . 339
Records Resources . . . 342
Followers on Records Resource . . . 342
Topics on Records Resource . . . 343
Scheduled Recommendations Resources . . . 344
Scheduled Recommendations . . . 345
Scheduled Recommendation . . . 347
Subscriptions Resource . . . 350
Topics Resources . . . 351
List of Topics . . . 352
Topics Suggestions . . . 353
Trending Topics . . . 354
Topic . . . 355

Topics, Endorse People . . . 357
Topic Endorsements . . . 358
Topic Files . . . 359
Topic Groups . . . 361
Topics, Knowledgeable People List . . . 361
Related Topics . . . 362
Topics, Opt Outs from Knowledgeable People List . . . 362
Topics, Opt Out from Knowledgeable People List . . . 363
User Profiles Resource . . . 364
Users Resources . . . 364
User Information . . . 366
User Profile Information . . . 367
Batch User Information . . . 369
User Conversations, General . . . 369
User Conversations, Specific . . . 371
Change Status of Conversations . . . 372
Get Unread Count for Conversations . . . 373
Users Files, General . . . 373
User Files, Filtered by Group . . . 376
User Files, Filtered by Sharing . . . 377
User Followers . . . 378
Following . . . 378
User Groups . . . 380
Topics a User is Knowledgeable About . . . 381
User Messages, General . . . 382
User Messages, Specific . . . 389
User Photo . . . 389
User Recommendations, General . . . 391
User Recommendations for a Specific Action . . . 393
User Recommendations for a Specific Action and Object Category . . . 395
User Recommendations for a Specific Action and Key Prefix . . . 398
User Recommendations for a Specific Action and Object ID . . . 400
User Recommendations for a Specific Action and Object Type . . . 400
User Reputation . . . 401
User Settings . . . 401
User Status . . . 403
Topics Recently Used . . . 404
Zones Resources . . . 404
List of Zones . . . 404
Zone Information . . . 405
Zone Search Results . . . 405

Chapter 6: Chatter REST API Request Bodies . . . 407

Action Link Definition Input . . . 407

Action Link Group Definition Input 412
Action Link Input 415
Action Link Template Binding Input 415
Announcement Input 416
Associated Actions Capability Input 417
Attachment Input: Canvas App 417
Attachment Input: Existing Content 418
Attachment Input: Feed Poll 419
Attachment Input: Link 419
Attachment Input: New File Upload 420
Batch Collection Input 421
Batch Input 422
Bookmarks Capability Input 423
Canvas Capability Input 424
Comment Input 425
Comment Capabilities Input 426
Content Capability Input 426
Datacloud Order Input 427
Dashboard Component Snapshot Input 428
Feed Element Capabilities Input 429
Feed Favorite Input 430
Feed Item Input 431
Feed Item Topic Input 436
Feed Poll Input 436
File Input 436
File Shares Input 437
Folder Input 438
Folder Item Input 438
Following Input 439
Generic Batch Collection Input 440
Generic Batch Input 441
Group Chatter Settings Input 442
Group Information Input 443
Group Input 443
Group Member Input 445
Group Membership Request Update 445
Group Record Input 446
Link Capability Input 446
Managed Topic Input 447
Managed Topic Position Collection Input 448
Managed Topic Position Input 450
Message Body Input 450
Message Segment Input: Hash Tag 451
Message Segment Input: Link 451

Message Segment Input: Mention . . . 451
Message Segment Input: Text . . . 452
Moderation Flag Input . . . 452
Photo Input . . . 452
Poll Capability Input . . . 453
Push Notification Input . . . 454
Question and Answers Capability Input . . . 455
Recommendation Definition Input . . . 455
Request Header Input . . . 456
Scheduled Recommendation Input . . . 457
Share Input . . . 459
Topic Assignment Input . . . 459
Topic Endorsement Input . . . 460
Topic Input . . . 460
User Chatter Settings Input . . . 461
User Input . . . 461
User Message . . . 462
User Message Recipient Input . . . 463
User Message Recipient List Input . . . 463
User Status Input . . . 463
User Stream (Conversation) Input . . . 464

Chapter 7: Chatter REST API Response Bodies . . . 465

Action Link Definition . . . 465
Action Link Diagnostic . . . 468
Action Link Group Definition . . . 468
Address . . . 470
Announcement . . . 470
Announcement Page . . . 470
Approval Capability . . . 471
Approval Post Template Field . . . 472
Article Item . . . 472
Associated Actions Capability . . . 472
Banner Capability . . . 473
Batch Result Item . . . 473
Batch Results . . . 474
Bookmarks Capability . . . 474
Case Comment Capability . . . 475
Canvas Capability . . . 476
Chatter Activity . . . 476
Chatter Directory . . . 477
Chatter Influence . . . 477
Chatter Likes Capability . . . 477
ClientInfo . . . 478

Comment 478
Comment Capabilities 479
Comment Page 480
Comments Capability 480
Community 481
Community Page 482
Connect Directory 482
Content Capability 482
Conversation Detail 484
Conversation Summary 484
Conversation Summary Collection 484
Conversation Unread Count 485
Dashboard Component Snapshot 485
Dashboard Component Snapshot Capability 486
Datacloud Company 486
Datacloud Company Collection 488
Datacloud Contact 489
Datacloud Contact Collection 489
Datacloud Order 490
Datacloud Purchase Usage 490
Edit Capability 491
Email Address 492
Email Message Capability 492
Enhanced Link Capability 492
Error Message 493
Error with Output 493
Explanation Details 493
Explanation Summary 495
Favorite 496
Favorites Page 497
Features 497
Feed 499
Feed Directory 499
Feed Directory Item 499
Feed Element Capabilities 501
Feed Element Page 503
Feed Entity Is Editable 504
Feed Item 504
Feed Item Attachment: Approval 511
Feed Item Attachment: Basic Template 512
Feed Item Attachment: Canvas Template 512
Feed Item Attachment: Case Comment 513
Feed Item Attachment: Content 514
Feed Item Attachment: Dashboard 516

Feed Item Attachment: Email Message . . . 517
Feed Item Attachment: Link . . . 517
Feed Item Attachment: Poll . . . 517
Feed Item Attachment: Record Snapshot . . . 518
Feed Item Attachment: Tracked Change . . . 518
Feed Item Body . . . 518
Feed Item Page . . . 519
Feed Item Preamble Message Body . . . 519
Feed Item Topics . . . 520
Feed Poll Choice . . . 520
Feed Tracked Change . . . 520
File Detail . . . 521
File Share Link . . . 521
File Shares Page . . . 522
File Summary . . . 522
File Summary Page . . . 525
Files Connect Provider Type . . . 525
Files Connect Repository . . . 526
Files Connect Repository Collection . . . 527
Files Connect Repository Features . . . 527
Flat Feed . . . 527
Flat Feed Item . . . 528
Folder . . . 528
Folder Item . . . 529
Folder Item Page . . . 529
Follower Page . . . 529
Following Counts . . . 530
Following Page . . . 530
Generic Bundle Capability . . . 531
Generic Feed Element . . . 531
Group . . . 532
Group Chatter Settings . . . 534
Group Detail . . . 535
Group Information . . . 536
Group Member . . . 537
Group Member Page . . . 537
Group Membership Request . . . 538
Group Membership Request Collection . . . 538
Group Page . . . 539
Group Record . . . 539
Group Record Page . . . 539
Icon . . . 540
Influence Thresholds . . . 540
Knowledgeable People . . . 540

Knowledgeable People Collection 541
Like 541
Like Message Body 542
Like Page 542
Link Capability 542
List View Column 543
List View Definition 543
List View Object 544
List View Record 544
List View Sort 545
Maintenance Information 545
Managed Topic 546
Managed Topic Collection 546
Mention Completion 547
Mention Completion Page 548
Mention Validation 548
Mention Validations 549
Message 549
Message Collection 550
Message Segment: Entity Link 550
Message Segment: Field Change 550
Message Segment: Field Change Name 551
Message Segment: Field Change Value 552
Message Segment: Hashtag 552
Message Segment: Link 553
Message Segment: Mention 553
Message Segment: More Changes 554
Message Segment: Resource Link 554
Message Segment: Text 554
Moderation Capability 555
Moderation Flags 555
Motif 555
Non-Entity Recommendation 557
Organization 557
Origin Capability 558
Phone Number 558
Photo 559
Platform Action 559
Platform Action Group 562
Poll Capability 563
Question and Answers Capability 563
Question and Answers Suggestion Collection 564
Recommendation 564
Recommendation Definition 565

Recommendation Definition Collection 566
Recommendations 566
Recommendations Capability 566
Recommended Object 567
Record Field 567
Record Field: Blank 568
Record Field: Compound 568
Record Field: Currency 569
Record Field: Date 569
Record Field: Percent 570
Record Field: Picklist 570
Record Field: Reference 571
Record Field: Reference with Date 571
Record Snapshot Capability 571
Record Summary 572
Record Summary List 572
Record View 573
Record View Section 573
Reference 574
Repository File Detail 574
Repository File Summary 574
Repository Folder Item 575
Repository Folder Items Collection 576
Repository Folder Summary 576
Reputation 577
Reputation Level 577
Request Header 578
Scheduled Recommendation 578
Scheduled Recommendation Collection 579
Shares 579
Subscription 579
Subscription Page 580
Time Zone 580
Topic 581
Topic Collection 581
Topic Endorsement 581
Topic Endorsement Collection 582
Topic Images 582
Topic Opt Out 583
Topic Opt Out Collection 583
Topic Suggestion 583
Topic Suggestion Collection 583
Topics Capability 584
Topics People Are Knowledgeable About 584

Topics People Are Knowledgeable About Collection 584
Tracked Change Bundle Capability 585
Tracked Changes Capability 585
Unauthenticated User 586
User Capabilities 586
User Chatter Settings 586
User Detail 587
User Group Page 589
User Page 590
User Profile 590
User Profile Tab 590
User Settings 591
User Status 592
User Summary 592
Zone 594
Zone Page 595
Zone Search Page 595
Zone Search Result 596

Chapter 8: Chatter REST API Appendices 597

Action Links Labels 597

Index 603

CHAPTER 1 Introducing Chatter REST API

In this chapter ...

- Deciding When to Use Chatter REST API
- Chatter REST API Architecture
- Chatter REST API Rate Limits
- Building the Chatter REST API Resource URL
- Sending HTTP Requests
- Understanding a Chatter REST API Request and Response
- Using Chatter REST API Inputs
- Wildcards
- Specifying Response Sizes
- Response Body Encoding
- Status Codes and Error Responses
- Understanding OAuth and Chatter REST API
- Use CORS to Access Chatter REST API and REST API

Use Chatter REST API to display Salesforce data, especially in mobile applications. Chatter REST API responses are localized and structured for presentation. You can filter responses so they contain only what the app needs. In addition to Chatter feeds, users, groups, and followers, Chatter REST API provides programmatic access to files, recommendations, topics, notifications, Data.com purchasing, and more. Chatter REST API is similar to APIs offered by other companies with feeds, such as Facebook and Twitter, but it also exposes Salesforce features beyond Chatter.

EDITIONS

Available in: All editions except **Personal** Edition. In addition, some features require Chatter to be enabled for the organization.

Deciding When to Use Chatter REST API

Use Chatter REST API to:

- Build a mobile app.
- Integrate a third-party Web application with Salesforce so it can notify groups of users about events.
- Display a feed on an external system, such as an intranet site, after users are authenticated.
- Make feeds actionable and integrated with third-party sites. For example, an app that posts a Chatter item to Twitter whenever the post includes #tweet hashtag.
- Create simple games that interact with the feed for notifications.
- Creating a custom, branded skin for Chatter for your organization.

How Is Chatter REST API Different from Other Salesforce APIs?

The following are some of the differences between Chatter REST API and REST API:

- Data is structured for rendering on Web sites and mobile devices.
- Returned information is localized to the user's time zone and language.
- Changed values that are tracked in a feed are returned as value-pair representations.
- Rate limiting for Chatter REST API is per user, per application, per hour. The rate limiting for SOAP API and REST API is by organization.

If needed, extract an ID from the REST API and use it in requests to Chatter REST API.

Chatter in Apex

Many Chatter REST API resource actions are exposed as static methods on Apex classes in the `ConnectApi` namespace. This namespace is also referred to as *Chatter in Apex*. Use Chatter in Apex to develop Chatter applications on the Force.com platform without using HTTP callouts from Apex.

The documentation for Chatter in Apex is available in the *Apex Code Developer's Guide*.

SEE ALSO:

Chatter REST API Rate Limits

Chatter REST API Architecture

To use Chatter REST API, understand a few characteristics of its architecture, like how to authenticate, and which data formats it supports.

Authentication

Chatter REST API uses OAuth 2.0.

Note: Salesforce is accessed securely, that is, using HTTPS.

CORS (Cross-Origin Resource Scripting)

Chatter REST API and REST API support CORS (cross-origin resource sharing). To access these APIs from JavaScript in a Web browser, add the origin serving the script to the CORS whitelist. See Use CORS to Access Chatter REST API and REST API.

Default HTML entity encoding

Chatter REST API responses are minimally HTML entity-encoded by default.

For non-HTML contexts, such as native mobile applications, Chatter REST API clients can request raw (unencoded) output. Set the `X-Chatter-Entity-Encoding` HTTP header in a request to `false`.

For more information, see Response Body Encoding on page 26.

JSON and XML support

The JavaScript Object Notation (JSON) format is supported with UTF-8. Date-time information is in ISO8601 format.

XML serialization is similar to SOAP API. XML requests are supported in UTF-8 and UTF-16, and XML responses are provided in UTF-8.

Responses are in JSON format by default. To specify XML, use the HTTP header `Accept: application/xml`. (To specify JSON, use `Accept: application/json`.)

You can append either `.json` or `.xml` to the URI of any resource to specify how the return is formatted. For example, `/chatter/feeds/filter/me/001/feed-elements.xml`.

Note: Not all features support XML.

Localized names and times

Chatter REST API localizes both the names and datetimes to the locale setting of the user. Users set their locale in their personal information. If the user hasn't set their locale, Chatter REST API uses the default locale for the organization.

Clients can use any locale available in the organization to override the default locale. To override the locale for any API request, use the Accept-Language HTTP header.

Note: ISO-8601 dates are always in GMT.

Salesforce ID length

Salesforce IDs in response bodies are always 18 character IDs. In request bodies, you can use either 15 character IDs or 18 character IDs.

UI Themes

Clients can use the `X-Connect-Theme` HTTP header to specify a UI theme for any API request. There are 2 themes, each with their own set of motif icons: `Classic` and `Salesforce1`. Theme names are case-sensitive.

Method overriding

To override an HTTP method if you use an HTTP library that doesn't allow overriding or setting an arbitrary HTTP method name, use the request parameter `_HttpMethod`.

```
POST https://instance_name/services/data/v34.0/chatter/
/chatter/users/me/conversations/03MD0000000008KMAQ
?_HttpMethod=PATCH&read=true
```

Note: The `_HttpMethod` parameter is case-sensitive. Use the correct case for all values.

Chatter REST API Rate Limits

Chatter REST API requests are subject to rate limiting. Chatter REST API has a different rate limit than other Salesforce APIs. Chatter REST API has a per user, per application, per hour rate limit. When you exceed the rate limit, all Chatter REST API resources return a 503 Service Unavailable error code.

Note: Load, performance, and other system issues can prevent some limits from being reached. Limits can change without notice. Applications should make efficient use of available requests and gracefully handle the 503 error code.

For applications using a session ID from Force.com, the rate limit is per user, per hour—there isn't a separate bucket for applications. All applications the user accesses with a session ID use this general quota. To take advantage of the per user, per application, per hour limit, use OAuth tokens.

To build apps that don't hit the rate limit, use these guidelines:

- If you hit limits when running tests, use multiple users to simulate a real-world scenario.
- When polling for feed updates, do not exceed one poll per minute (60 polls per hour). To return more results in 1 request, increase the page size.
- When polling for private messages, do not exceed 60 polls per hour.
- To avoid making multiple requests, cache metered static assets such as file and dashboard renditions (group and user profile pictures are not metered).
- Each developer on a team should set up 2 connected apps: one for automated testing and one for manual testing and development. Don't share connected apps with other developers.
- Use a unique connected app for the production environment.
- Don't share connected apps between applications.

Important: To extract data, use REST API or SOAP API to operate on sObjects. Do not use Chatter REST API to extract data.

REST API's rate limiting and query language work better for these use cases:

- Migrate or synchronize user profiles between systems.
- Create analytic applications.
- Write queries that return records and fields.

SEE ALSO:

Deciding When to Use Chatter REST API

Building the Chatter REST API Resource URL

Access all Chatter REST API resources by using the URI for your company's instance, combined with version and community information, and the URI for the resource.

For example, building a URL without communities, if your instance is:

```
https://na1.salesforce.com
```

And the version information is:

```
/services/data/v34.0
```

And the resource is :

```
/chatter/feeds/news/me/feed-elements
```

Put together, the full URL is:

```
https://na1.salesforce.com/services/data/v34.0/chatter/
feeds/news/me/feed-elements
```

To access resources that are part of a community, use the following after the version information and before the resource:

```
/connect/communities/communityId
```

For example, this is a full URL that accesses a news feed resource in a community:

```
https://na1.salesforce.com/services/data/v34.0/connect
/communities/communityId/chatter/feeds/news/me/feed-elements
```

To access resources in the default community, use the `internal` shortcut as the community Id:

```
/connect/communities/internal
```

You can use the `internal` shortcut even if communities are not enabled. You may want do this if you're developing an application for multiple organizations and not all of them use communities.

The response bodies include URLs with `/connect/communities/internal`. This format enables you to perform string interpolation when referencing communities.

To access resources in the default community, you can also use the `/chatter` resources directly (without `/connect/communities/internal`). The response bodies don't include URLs with `/connect/communities/internal`.

In addition, you can also use your organization's custom URL for a community to access resources in it. For example:

```
https://communitydomain.force.com/communitypath/services/data/v34.0/connect
/communities/communityId/chatter/feeds/news/me/feed-elements
```

Sending HTTP Requests

Chatter REST API uses HTTP methods to send and receive JSON and XML content, so it is very simple to build client applications using the tool or the language of your choice.

HTTP requests to a Chatter REST API resource contain the following information:

- An HTTP method (such as GET, POST, PATCH, or DELETE).
- An OAuth 2.0 access token used to authenticate the request. For information on how to retrieve the token, see Chatter REST API Quick Start.
- The Chatter REST API resource URL.
- Request parameters or a request body containing information needed for requests, such as information to update a record.

  The request body can contain JSON or XML. If you pass resource-specific request parameters and a request body, the request parameters are ignored. Request parameters that aren't specific to a resource, such as parameters in a bearer token URL and the `_HttpMethod` parameter, are processed along with the request body. Use bearer token parameters as they are. Don't add or remove parameters.

The HTTP methods are used to indicate the desired action, such as retrieving information, as well as creating, updating, and deleting records.

- GET is used to retrieve information, such as basic resource summary information.
- POST is used to create a new item, such as a feed item or a comment or like, or subscribe to a group.
- PATCH is used for partial updates to an item, for example, renaming a file.
- PUT is used for whole updates to an item, for example, marking a conversation as read.
- DELETE is used to delete an item, such as a feed element, or to unsubscribe from a group.
- HEAD is used to retrieve resource metadata. It's similar to using GET but doesn't return a response body. You can use HEAD to test the availability of a resource.

Understanding a Chatter REST API Request and Response

The general process flow for using Chatter REST API is as follows:

1. Send an HTTP request to Salesforce.
2. The system authenticates the user and client app.
3. Request Chatter REST API resources using an HTTP request.
4. The system returns an HTTP return code. Additional information (such as a Chatter REST API response body) is optionally included in the response.

Each resource in Chatter REST API is a URI used with an HTTP method (such as GET, POST, DELETE, or HEAD). Each resource represents something in Chatter, such as a user's news feed, or the comments on a post.

Many of the resources use either a specified user ID or the keyword `me` to indicate the current, context user. For example, to return the user-profile information for the current, context user, use the following resource:

```
/chatter/users/me
```

You could also specify a particular user by user Id:

```
/chatter/users/005D0000001GLoh
```

When you successfully access a resource, a response body is returned in JSON or XML. By default, JSON is returned. Each response body may contain one or more response bodies. For example, the following is the response body of the above resource (users). Note that it includes other response bodies as well, such as address, current status, and phone numbers:

Property	Type	Description	Filter Group and Version	Available Version
`aboutMe`	String	Text from user's profile	Big, 29.0	23.0
`additionalLabel`	String	An additional label for the user, for example, "Customer," "Partner," or "Acme Corporation." If the user doesn't have an additional label, the value is `null`.	Small, 30.0	30.0
`address`	Address	Address	Big, 29.0	23.0
`chatterActivity`	Chatter Activity	Chatter activity statistics	Big, 29.0	23.0
`chatterInfluence`	Chatter Influence	User's influence rank	Big, 29.0	23.0
`communityNickname`	String	User's nickname in the community	Medium, 32.0	32.0
`companyName`	String	Company name	Small, 29.0	23.0
`currentStatus`	User Status	Note: This property is no longer available as of version 25.0. User's current status		23.0–25.0
`displayName`	String	User's name that is displayed in the community. If nicknames are enabled, the nickname is displayed. If nicknames aren't enabled, the full name is displayed.	Small, 32.0	32.0

Property	Type	Description	Filter Group and Version	Available Version
`email`	String	User's email address	Medium, 29.0	23.0
`firstName`	String	User's first name	Small, 29.0	23.0
`followersCount`	Integer	Number of users following this user	Big, 29.0	23.0
`followingCounts`	Following Counts	Information about items the user is following	Big, 29.0	23.0
`groupCount`	Integer	Number of groups user is following	Big, 29.0	23.0
`hasChatter`	Boolean	`true` if user has access to Chatter; `false` otherwise	Small, 31.0	31.0
`id`	String	18-character ID of the user	Small, 29.0	23.0
`isActive`	Boolean	`true` if user is active, `false` otherwise	Small, 29.0	23.0
`isChatterGuest`	Boolean	Note: This property is no longer available as of version 26.0. Use the `userType` property instead. `true` if user is a Chatter customer, `false` otherwise		23.0–25.0
`isInThis Community`	Boolean	`true` if user is in the same community as the context user; `false` otherwise. Can be `false` for record feed @mentions on posts with the scope "All Users."	Small, 29.0	28.0
`lastName`	String	User's last name	Small, 29.0	23.0
`managerId`	String	18-character ID of the user's manager	Big, 29.0	23.0
`managerName`	String	Locale-based concatenation of manager's first and last names	Big, 29.0	23.0
`motif`	Motif	Small, medium, and large icons indicating that the object is a user. The motif can also contain the object's base color. The motif icon is not the user's profile picture.	Small, 29.0	27.0
`mySubscription`	Reference	If the context user is following this user, this contains information about the subscription, else returns `null`	Medium, 29.0	23.0
`name`	String	Locale-based concatenation of user's first and last names	Small, 29.0	23.0
`phoneNumbers`	Phone Number[]	Collection of user's phone numbers	Big, 29.0	23.0
`photo`	Photo	Information about the user's photos	Small, 29.0	23.0
`reputation`	Reputation	Reputation of the user in the community, which is determined by the number of points earned from events and actions performed in the community	Big, 32.0	32.0
`thanksReceived`	Integer	Number of thanks the user has received	Big, 29.0	29.0

Property	Type	Description	Filter Group and Version	Available Version
`title`	String	Title of the user	Small, 29.0	23.0
`type`	String	`User`	Small, 29.0	23.0
`url`	String	URL to the user's Chatter profile	Small, 29.0	23.0
`username`	String	User name of the user, such as *`Admin@mycompany.com`*	Small, 29.0	24.0
`userType`	String	Specifies the type of user. Valid values are: • `ChatterGuest`—User is an external user in a private group. • `ChatterOnly`—User is a Chatter Free customer. • `Guest`—User is unauthenticated. • `Internal`—User is a standard organization member. • `Portal`—User is an external user in a customer portal, partner portal, or community. • `System`—User is Chatter Expert or a system user. • `Undefined`—User is a user type that is a custom object.	Medium, 29.0	26.0

In JSON format, the output is as follows:

```
{
   "aboutMe": "I&#39;m excited to be part of the team focused on building out our apps
business
 and showing our customers their future.",
   "additionalLabel" : null,
   "address": {
      "city": "Seattle",
      "country": "US",
      "state": "WA",
      "street": "2001 8th Ave",
      "zip": "98121"
      "formattedAdress": "2001 8th Ave\nSeattle, WA 98121\nUS"
   },
   "chatterActivity": {
      "commentCount": 0,
      "commentReceivedCount": 1,
      "likeReceivedCount": 0,
      "postCount": 2
   },
   "chatterInfluence" : {
     "percentile" : "0.6666666666666667",
     "rank" : 1
   },
   "communityNickname": "marRave",
```

```
   "companyName": "Seattle Apps",
   "displayName": "Marion Raven",
   "email": "mraven@seattleapps.com",
   "firstName": "Marion",
   "followersCount": 1,
   "followingCounts": {
      "people": 5,
      "records": 0,
      "total": 5
   },
   "groupCount": 3,
   "hasChatter": true,
   "id": "005D0000001Kl6xIAC",
   "isActive": true,
   "isInThisCommunity" : true,
   "lastName": "Raven",
   "managerId": null,
   "managerName": null,
   "motif": {
      "color" : "20aeb8",
      "largeIconUrl": "/img/icon/profile64.png",
      "mediumIconUrl": "/img/icon/profile32.png",
      "smallIconUrl": "/img/icon/profile16.png"
   },
   "mySubscription": null,
   "name": "Marion Raven",
   "phoneNumbers": [{
      "label": "Work",
      "phoneNumber": "(206) 999-2846",
      "phoneType": "Work"
   }],
   "photo": {
      "fullEmailPhotoUrl":
"https://instance_name/ncsphoto/SqRfG_-PUTY6KtfOp_f0cGW2pGjfqoWUUgTENHjVyS1i9x6Z16L8wXEGigpcWjAd",

      "largePhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/F",
      "photoVersionId": "729D00000000BDV",
      "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/T",
      "standardEmailPhotoUrl":
"https://instance_name/ncsphoto/SqRfG_-PUTY6KtfOp_f0cGW2pGjfqoWUUgTENHjVyS1q43x_rW2ciXBHst0Eh3Y6",

      "url":  "/services/data/v34.0/chatter/users/005D0000001LL8OIAW/photo"
   },
   "reputation" : null,
   "thanksReceived" : null,
   "title": null,
   "type": "User",
   "url": "/services/data/v34.0/chatter/users/005D0000001Kl6xIAC",
   "userType": "Internal";
   "username": "mraven@seattleapps.com"
}
```

Note: Salesforce runs on multiple server instances. The examples in this guide use *instance_name* to indicate the instance. The instance for your organization may be similar to `na1.salesforce.com` or `na6.mycompany.com`.

Some of the items in a response body have a fully qualified URL, however, most of the returned URLs are relative. If the URL is relative, prepend it with the instance given to the client app when the item is authenticated.

Using Chatter REST API Inputs

Using POST, PATCH, or PUT for Input

When you make a request using the HTTP POST, PATCH, and PUT methods, you can use request parameters or a request body. The request body can contain JSON or XML. If you pass resource-specific request parameters and a request body, the request parameters are ignored. Request parameters that aren't specific to a resource, such as parameters in a bearer token URL and the `_HttpMethod` parameter, are processed along with the request body. Use bearer token parameters as they are. Don't add or remove parameters.

To submit a request using request parameters, use a `Content-Type` header field with the value `application/x-www-form-urlencoded`.

To submit a request using a request body, use one of the following values in a `Content-Type` header field and in an `Accept` header field:

- `application/json`
- `application/xml`

Uploading Binary Files

To upload a binary file, you must send it as a body part in a multipart/form-data request. You can send information such as the text of a post or comment as a JSON or XML rich input body part in the same multipart/form-data request. Alternately, you can choose to send that information in request parameters. If you pass both a rich input request body and request parameters, the request parameters are ignored.

To create a multipart/form-data request, in the head of the request, set the `Content-Type` HTTP header to `multipart/form-data`.

For information about additional HTTP headers, see W3C Form content types, and RFC 2388, which defines the multipart/form-data internet media type.

This table describes the HTTP headers and parameters required in the rich input body part of a multipart/form-data request:

HTTP Headers for Rich Input Body Part	Header Value and Parameters	Description
`Content-Disposition`	`form-data; name="json"` `form-data; name="xml"`	The request body for a post or comment. For JSON, the value of `name` use `"json"`. For XML, the value of `name` use `"xml"`.
`Content-Type`	`application/json; charset=UTF-8` `application/xml; charset=UTF-8`	The data format and character set of the request body. For JSON, the value must be `application/json`.

HTTP Headers for Rich Input Body Part	Header Value and Parameters	Description
		For XML, the value must be `application/xml`.

Tip: Web browsers are typically incapable of making multipart requests when the non-binary parts, such as rich input bodies, have their own `Content-Type`. To work around this issue, specify a certain `Content-Disposition` `name` and Salesforce can read the `Content-Type` of the rich input part. (You don't need to specify a `Content-Type` for the rich input body.)

This table describes the HTTP headers and parameters required in the rich input body part of a multipart/form-data request from a Web browser:

HTTP Headers for Rich Input Body Part from Web browser	Header Value and Parameters	Description

`Content-Disposition`	`form-data; name="feedElement"`
	`form-data; name="comment"`
	`form-data; name="photo"`
	`form-data; name="file"`
		To post a feed element with a binary file, the value of `name` must be `"feedElement"`.
		To post a comment with a binary file, the value of `name` must be `"comment"`.
		To post a user or group photo, the value of `name` must be `"photo"`.
		To post a file to the Files list, the value of `name` must be `"file"`.

This table describes the HTTP headers and parameters required in the binary upload body part of a multipart/form-data request:

HTTP Headers for Binary Upload Body Part	Header Value and Parameters	Description
`Content-Disposition`	`form-data; name="feedElementFileUpload"; filename=`***`string`*** `form-data; name=name="feedElementFileUpload"; filename=`***`string`*** `form-data; name="fileUpload" filename=`***`string`*** `form-data; name="fileData" filename=`***`string`***	To post a feed element with a binary attachment, the value of `name` must be `"feedElementFileUpload"`. To post a comment with a binary file, the value of `name` must be `"feedElementFileUpload"`. To post a user or group photo, the value of `name` must be `"fileUpload"`. To post a file to the Files list, the value of `name` must be `"fileData"`. Note: You must specify a `filename` parameter and value. However, please note that Chatter uses the value of the `title` property of the Content Capability Input as the file name, not the value of the `filename` parameter.

HTTP Headers for Binary Upload Body Part	Header Value and Parameters	Description
`Content-Type`	`application/octet-stream; charset=ISO-8859-1`	The media type and character set of the binary file.

Note: Spacing and carriage returns and line feeds (CRLF) are important. For example, this line requires spaces: `Content-Disposition: form-data; name="feedItemFileUpload"; title="2012_q1_review.ppt"`. If you had used CRLF instead of spaces, you would have received an error.

Post a feed element with a binary file

This example uses the `/chatter/feed-elements` resource to post a new feed item (which is a type of feed element), upload a binary file, and attach it to the new feed item. This example uses cURL.

```
curl -H "X-PrettyPrint: 1" -F 'json={ "body":{ "messageSegments":[ { "type":"Text",
"text":"Please accept this receipt." } ] },
"capabilities":{ "content":{ "description":"Receipt for expenses", "title":"receipt.pdf"
 } }, "feedElementType":"FeedItem", "subjectId":"005RR000000DmOb" };
type=application/json' -F
"feedElementFileUpload=@receipt.pdf;type=application/octet-stream"
 -X POST https://instance_name/services/data/v34.0/chatter/feed-elements
-H 'Authorization: OAuth 00DRR0000000N0g!...' --insecure
```

This is the same example in a different format so you can see the request body parts:

```
POST /services/data/v34.0/chatter/feed-elements HTTP/1.1
Authorization: OAuth 00DRR0000000N0g!...
User-Agent: Jakarta Commons-HttpClient/3.0.1
Host: instance_name
Content-Length: 845
Content-Type: multipart/form-data; boundary=a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq
Accept: application/json

--a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq
Content-Disposition: form-data; name="json"
Content-Type: application/json; charset=UTF-8

{
   "body":{
      "messageSegments":[
         {
            "type":"Text",
            "text":"Please accept this receipt."
         }
      ]
   },
   "capabilities":{
      "content":{
         "description":"Receipt for expenses",
         "title":"receipt.pdf"
      }
   },
   "feedElementType":"FeedItem",
```

```
   "subjectId":"005RR000000DmOb"
}

--a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq
Content-Disposition: form-data; name="feedElementFileUpload"; filename="receipt.pdf"
Content-Type: application/octet-stream; charset=ISO-8859-1

...contents of receipt.pdf...

--a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq--
```

Post a batch of feed elements with binary files

This example uses the `/chatter/feed-elements/batch` resource to post a batch of feed elements, uploads two binary files, and attach them to the feed elements in the batch. You must include the information about the binary parts in the request body. This example uses cURL.

```
curl -H "X-PrettyPrint: 1" -F 'json={"inputs":[{"binaryPartNames":["bin1"],
"richInput":{"subjectId":"me","body":{
"messageSegments":[{"type":"Text","text":"Please accept this receipt"}]},"capabilities":
 {"content" : {"description": "Receipt for expenses",
"title":"receipt.pdf"}},"feedElementType":"FeedItem"}},
{"binaryPartNames":["bin2"],"richInput":{"subjectId":"me","body":{"messageSegments":[{"type":"Text","text":"Post
 Number 2"}]}},
"capabilities": {"content" : {"description": "Receipt for expenses",
"title":"receipt2.pdf"}},"feedElementType":"FeedItem"}}]};type=application/json'
-F "bin1=@/Users/jsmith/Desktop/receipt.pdf;type=application/octet-stream" -F
"bin2=@/Users/jsmith/Desktop/receipt2.pdf;type=application/octet-stream"
-X POST https://instance_name/services/data/v34.0/chatter/feed-elements/batch
-H 'Authorization: OAuth 00DD00000007HoR44QATPNzhzYEJBfU' --insecure
```

This is the same example formatted to show the request body parts:

```
POST /services/data/v34.0/chatter/feed-elements/batch HTTP/1.1
Authorization: OAuth 00DRR0000000N0g!...
User-Agent: Jakarta Commons-HttpClient/3.0.1
Host: instance_name
Content-Length: 845
Content-Type: multipart/form-data; boundary=a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq
Accept: application/json

--a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq
Content-Disposition: form-data; name="json"
Content-Type: application/json; charset=UTF-8

{
  "inputs": [
    {
      "binaryPartNames": [
        "bin1"
      ],
      "richInput": {
        "subjectId": "me",
        "body": {
          "messageSegments": [
            {
```

```
              "type": "Text",
              "text": "Please accept this receipt"
            }
          ]
        },
        "capabilities": {
          "content": {
            "description": "Receipt for expenses",
            "title": "receipt.pdf"
          }
        },
        "feedElementType": "FeedItem"
      }
    },
    {
      "binaryPartNames": [
        "bin2"
      ],
      "richInput": {
        "subjectId": "me",
        "body": {
          "messageSegments": [
            {
              "type": "Text",
              "text": "Post Number 2"
            }
          ]
        },
        "capabilities": {
          "content": {
            "description": "Receipt for expenses",
            "title": "receipt2.pdf"
          }
        },
        "feedElementType": "FeedItem"
      }
    }
  ]
}

--a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq
Content-Disposition: form-data; name="bin1"; filename="receipt.pdf"
Content-Type: application/octet-stream; charset=ISO-8859-1

...contents of receipt.pdf...


--a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq
Content-Disposition: form-data; name="bin2"; filename="receipt2.pdf"
Content-Type: application/octet-stream; charset=ISO-8859-1

...contents of receipt2.pdf...

--a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq--
```

Note: The binary part names in the `binaryPartNames` properties must match the binary part names in the `name` properties of the binary request parts.

Post a comment with a binary file

This example posts a comment to a feed item and uploads a binary attachment using cURL.

```
curl -H "X-PrettyPrint: 1" -F 'json={ "body":{ "messageSegments":[ { "type":"Text",
"text":"Here's another receipt." } ] }, "capabilities":{ "content":{ "title":"receipt2"
 } } };
type=application/json' -F
"feedElementFileUpload=@receipt2.txt;type=application/octet-stream"
-X POST
https://instance_name/services/data/v34.0/chatter/feed-elements/0D5RR0000004Grx/capabilities/comments/items

-H 'Authorization: OAuth 00DRR0000000N0g!ARoAQB...' --insecure
```

This is the sample example formatted to see the request body parts.

```
POST
/services/data/v34.0/chatter/feed-elements/0D5RR0000004Grx/capabilities/comments/items
  HTTP/1.1
Authorization: OAuth
00DD0000000Jhd2!AQIAQC.lh4qTQcBhOPm4TZom5IaOOZLVPVK4wI_rPYJvmE8r2VW8XA.
OZ7S29JEM_7Ctq1lst2dzoV.owisJc0KacUbDxyae
Accept: application/json
User-Agent: Jakarta Commons-HttpClient/3.0.1
Host: instance_name
Content-Length: 978
Content-Type: multipart/form-data; boundary=F9jBDELnfBLAVmLNbnLIYibT5Icp0h3VJ7mkI

--F9jBDELnfBLAVmLNbnLIYibT5Icp0h3VJ7mkI
Content-Disposition: form-data; name="json"
Content-Type: application/json; charset=UTF-8

{
   "body":{
      "messageSegments":[
         {
            "type":"Text",
            "text":"Here's another receipt."
         }
      ]
   },
   "capabilities":{
      "content":{
         "title":"receipt2"
      }
   }
}

--F9jBDELnfBLAVmLNbnLIYibT5Icp0h3VJ7mkI
Content-Disposition: form-data; name="feedElementFileUpload"; filename="receipt2.txt"
Content-Type: application/octet-stream; charset=ISO-8859-1

...contents of receipt2.txt...
```

```
--F9jBDELnfBLAVmLNbnLIYibT5Icp0h3VJ7mkI--
```

Post multiple comments with binary files

This example is a generic batch request to `https://`***`instance_name`***`/services/data/v34.0/connect/batch` that contains two subrequests. Each subrequest comments on a feed item and uploads a binary file to the comment.

You must include the request body of each subrequest in the `richInput` properties.

You must also include information about the binary parts in the `binaryPartName` and `binaryPartNameAlias` properties.

This is the cURL example:

```
curl -H "X-PrettyPrint: 1" -F 'json={ "haltOnError":true, "batchRequests":[ {
"url":"/v33.0/chatter/feed-elements/0D5D0000000YG0N/capabilities/comments/items",
"method":"Post", "binaryPartName":"binaryPart1",
"binaryPartNameAlias":"feedElementFileUpload", "richInput":{ "body":{ "messageSegments":[
 { "type":"Text", "text":"Check out this file, it may help." } ] },
 "capabilities":{ "content":{ "title":"Presentation1.txt" } } } }, {
"url":"/v33.0/chatter/feed-elements/0D5D0000000YG0M/capabilities/comments/items",
"method":"Post", "binaryPartName":"binaryPart2",
"binaryPartNameAlias":"feedElementFileUpload", "richInput":{ "body":{ "messageSegments":[
 { "type":"Text", "text":"Check out this file, it may help." } ] },
 "capabilities":{ "content":{ "title":"Presentation2.txt" } } } } ]
};type=application/json' -F
"binaryPart1=@/Users/jbleyle/Desktop/Presentation1.txt;type=application/octet-stream"
-F "binaryPart2=@/Users/jbleyle/Desktop/Presentation2.txt;type=application/octet-stream"

-X POST https://instance_name/services/data/v34.0/connect/batch
-H 'Authorization: OAuth 00DD00000007HoR!...' --insecure
```

This is the multipart/form-data request body:

```
https://instance_name/services/data/v34.0/connect/batch

Authorization: OAuth 00DD0000000Jhd2!...
Accept: application/json
Host: instance_name
Content-Type: multipart/form-data; boundary=123123

--123123
Content-Disposition: form-data; name="json"
Content-Type: application/json

{
   "haltOnError":true,
   "batchRequests":[
      {
        "url":"/v34.0/chatter/feed-elements/0D5D0000000YG0D/capabilities/comments/items",

          "method":"Post",
          "binaryPartName":"binaryPart1",
          "binaryPartNameAlias":"feedElementFileUpload",
          "richInput":{
             "body":{
```

```
               "messageSegments":[
                  {
                     "type":"Text",
                     "text":"Check out this file, it may help."
                  }
               ]
            },
            "capabilities":{
               "content":{
                  "title":"Presentation1.txt"
               }
            }
         }
      },
      {
        "url":"/v34.0/chatter/feed-elements/0D5D0000000YG0H/capabilities/comments/items",

         "method":"Post",
         "binaryPartName":"binaryPart2",
         "binaryPartNameAlias":"feedElementFileUpload",
         "richInput":{
            "body":{
               "messageSegments":[
                  {
                     "type":"Text",
                     "text":"Check out this file, it may help."
                  }
               ]
            },
            "capabilities":{
               "content":{
                  "title":"Presentation2.txt"
               }
            }
         }
      }
   ]
}


--123123
Content-Disposition: form-data; name="binaryPart1"; filename="Presentation1.txt"
Content-Type: application/octet-stream; charset=ISO-8859-1
Content-Transfer-Encoding: binary

This is the content of file 1
--123123
Content-Disposition: form-data; name="binaryPart2"; filename="Presentation2.txt"
Content-Type: application/octet-stream; charset=ISO-8859-1
Content-Transfer-Encoding: binary

This is the content of file 2
--123123--
```

Post and crop a user photo

This example uploads a user profile photo and crops it using cURL.

```
curl -H "X-PrettyPrint: 1" -F 'json={"cropX": "0", "cropY": "0", "cropSize":
"200"};type=application/json'
 -F "fileUpload=@myPhoto.jpg;type=application/octet-stream"
-X POST https://instance_name/services/data/v34.0/chatter/users/me/photo
 -H 'Authorization: OAuth 00DRR0000000N0g!ARoAQFRi_gBqZhajAX22MNuLrrE2Xk...'
 --insecure
```

This is the same example formatted to see the request body parts.

```
POST /services/data/v34.0/chatter/users/me/photo HTTP/1.1
Authorization: OAuth
00DD0000000Jhd2!AQIAQC.lh4qTQcBhOPm4TZom5IaOOZLVPVK4wI_rPYJvmE8r2VW8XA.
OZ7S29JEM_7Ctq1lst2dzoV.owisJc0KacUbDxyae
User-Agent: Jakarta Commons-HttpClient/3.0.1
Host: instance_name
Content-Length: 543
Content-Type: multipart/form-data; boundary=a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq
Accept: application/json

--a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq
Content-Type: application/json; charset=UTF-8
Content-Disposition: form-data; name="json"

{
   "cropX" : "0",
   "cropY" : "0",
   "cropSize" : "200"
}

--a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq
Content-Disposition: form-data; name="fileUpload"; filename="myPhoto.jpg"
Content-Type: application/octet-stream; charset=ISO-8859-1

...contents of myPhoto.jpg...

--a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq--
```

Post a file to the Files list

This example uploads a file to the Files list using cURL.

```
curl -H "X-PrettyPrint: 1" -F 'json={"title":"BoatPrices"};type=application/json'
-F "fileData=@BoatPrices.pdf;type=application/octet-stream"
-X POST https://instance_name/services/data/v34.0/chatter/users/me/files
-H 'Authorization: OAuth 00DRR0000000N0g!...' --insecure
```

This is the same example formatted to see the request body parts.

```
POST /services/data/v34.0/chatter/users/me/files

Authorization: OAuth
00DD0000000Jhd2!AQIAQC.lh4qTQcBhOPm4TZom5IaOOZLVPVK4wI_rPYJvmE8r2VW8XA.
OZ7S29JEM_7Ctq1lst2dzoV.owisJc0KacUbDxyae
```

```
User-Agent: Jakarta Commons-HttpClient/3.0.1
Host: instance_name
Content-Length: 489
Content-Type: multipart/form-data; boundary=a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq
Accept: application/json

--a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq
Content-Type: application/json; charset=UTF-8
Content-Disposition: form-data; name="json"

{
   "title":"BoatPrices"
}

--a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq
Content-Disposition: form-data; name="fileData"; filename="BoatPrices.txt"
Content-Type: application/octet-stream; charset=ISO-8859-1

...contents of BoatPrices.txt...

--a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq--
```

Wildcards

Use wildcard characters to match text patterns in Chatter REST API and Chatter in Apex searches.

A common use for wildcards is searching a feed. Pass a search string and wildcards in the `q` parameter. This example is a Chatter REST API request:

```
/chatter/feed-elements?q=chat*
```

This example is a Chatter in Apex method call:

```
ConnectApi.ChatterFeeds.searchFeedElements(null, 'chat*');
```

You can specify the following wildcard characters to match text patterns in your search:

Wildcard	Description
*	Asterisks match zero or more characters at the middle or end (not the beginning) of your search term. For example, a search for `john*` finds items that start with *john*, such as, *john*, *johnson*, or *johnny*. A search for `mi* meyers` finds items with *mike meyers* or *michael meyers*. If you are searching for a literal asterisk in a word or phrase, then escape the asterisk (precede it with the `\` character).
?	Question marks match only one character in the middle or end (not the beginning) of your search term. For example, a search for `jo?n` finds items with the term *john* or *joan* but not *jon* or *johan*.

When using wildcards, consider the following notes:

- Wildcards take on the type of the preceding character. For example, `aa*a` matches *aaaa* and *aabcda*, but not *aa2a* or *aa.!//a*, and `p?n` matches *pin* and *pan*, but not *p1n* or *p!n*. Likewise, `1?3` matches *123* and *143*, but not *1a3* or *1b3*.

- A wildcard (*) is appended at the end of single characters in Chinese, Japanese, Korean, and Thai (CJKT) searches, except in exact phrase searches.
- The more focused your wildcard search, the faster the search results are returned, and the more likely the results will reflect your intention. For example, to search for all occurrences of the word `prospect` (or `prospects`, the plural form), it is more efficient to specify `prospect*` in the search string than to specify a less restrictive wildcard search (such as `prosp*`) that could return extraneous matches (such as `prosperity`).
- Tailor your searches to find all variations of a word. For example, to find `property` and `properties`, you would specify `propert*`.
- Punctuation is indexed. To find `*` or `?` inside a phrase, you must enclose your search string in quotation marks and you must escape the special character. For example, `"where are you\?"` finds the phrase `where are you?`. The escape character (`\`) is required in order for this search to work correctly.

Specifying Response Sizes

To specify the response size, use `filterGroup` as a request parameter or in a request body to specify a group of properties by size. In addition, use either `exclude` or `include` as a request parameter or in a request body to exclude or include specific properties. Use filters to return only the data the application needs.

IN THIS SECTION:

filterGroup

Use the `filterGroup` parameter in a request URL or in a request body to specify whether to return a small, medium, or big group of properties. The `filterGroup` property is available in API version 29.0 and later.

exclude

To exclude specific properties from a response body, use the `exclude` request parameter. The `exclude` parameter is available in API version 27.0 and later.

include

To limit the response body to specific properties, use the `include` query string parameter.

`filterGroup`

Use the `filterGroup` parameter in a request URL or in a request body to specify whether to return a small, medium, or big group of properties. The `filterGroup` property is available in API version 29.0 and later.

Note: The `filterGroup` parameter is supported for JSON only and for REST only.

Every property in every response body is assigned a group size:

- `Big`—The default size. Includes every property in a response body, including those marked `Big`, `Medium`, and `Small`.
- `Medium`—Includes properties marked `Medium` and `Small`.
- `Small`—The smallest response size. Includes only properties marked `Small`.

Note: The size assignments are visible only in the documentation. They aren't visible in the response bodies.

When a request includes a `filterGroup`, the response body only includes properties that are part of the specified group.

Things to know about using filter groups:

- The default `filterGroup` size is `Big`.

- Each filter group size assigned to a property has an API version. In future API versions properties may move to different filter groups.
- Filter groups affect all HTTP verbs except DELETE and HEAD.
- You can use a `filterGroup` parameter on any request URL, including subrequests in a batch request body, unless otherwise stated.
- You can use a `filterGroup` parameter in conjunction with either an `include` or an `exclude` filter parameter to change the set of properties returned. The resulting set is a union of the two sets.

Using a `filterGroup` parameter with an `include` or `exclude` parameter

Requests can contain a `filterGroup` parameter and group size value, and either an `include` or an `exclude` filter parameter and value on the URL. This is also true for subrequests in a generic batch request.

When a request (or a subrequest) contains both a `filterGroup` parameter and an `include` parameter, the response always includes the properties specified by the `include` parameter.

When a request (or a subrequest) contains both a `filterGroup` parameter and an `exclude` parameter, the response always excludes the properties specified by the `exclude` parameter.

In this table, read a row from left to right to see whether the property is included in the response.

`include` **Filter Provided in Request**	**Property Provided in** `include` **Filter in Request**	`exclude` **Filter Provided in Request**	**Property Provided in** `exclude` **Filter in Request**	`filterGroup` **Provided in Request**	**Property Provided in** `filterGroup` **in Request**	**Property Included In Response**
Yes	No	No	No	No	No	**No**
Yes	Yes	No	No	Yes or No	Yes or No	**Yes**
No	No	Yes	No	No	No	**Yes**
No	No	Yes	No	Yes	No	**No**
No	No	Yes	Yes	Yes or No	Yes or No	**No**
No	No	No	No	Yes	No	**No**
No	No	No	No	Yes	Yes	**Yes**
Yes	Yes or No	Yes	Yes or No	Yes or No	Yes or No	**Error (400)**

Example

This example gets the Small response body for the User Profile resource:

```
GET /chatter/users/me?filterGroup=Small
```

This is the response:

```
{
   "companyName": "Universal Containers",
   "firstName": "Deanna",
   "hasChatter": true,
   "id": "005D0000001LL8OIAW",
```

```
   "isActive": true,
   "isInThisCommunity": true,
   "lastName": "Li",
   "motif": {
      "color": "1797C0",
      "largeIconUrl": "/img/icon/profile64.png",
      "mediumIconUrl": "/img/icon/profile32.png",
      "smallIconUrl": "/img/icon/profile16.png"
   },
   "name": "Deanna Li",
   "photo": {
      "photoVersionId": "729D00000000LJFIA2",
      "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000LJF/T",
      "standardEmailPhotoUrl": "https://instance_name/ncsphoto
       /SqRfG_-PUTY6KtfOp_f0cEw-MQI3xmkJ21oNEOQ-5sYVo6fntPnCQ_-xIH-GU3vI",
      "url": "/services/data/v34.0/chatter/users/005D0000001LL8OIAW/photo"
   },
   "title": null,
   "type": "User",
   "url": "/services/data/v34.0/chatter/users/005D0000001LL8OIAW",
   "username": "dli@uc.com"
}
```

exclude

To exclude specific properties from a response body, use the `exclude` request parameter. The `exclude` parameter is available in API version 27.0 and later.

The value for the `exclude` query is a list of properties separated by bars (`|`). You must URL encode the bars as `%7C`. You must include a forward slash before each property name.

The following request excludes both the `aboutMe` and `address` properties from a response body:

```
/chatter/users/me?exclude=/aboutMe%7C/address
```

The following table lists rules for using the `exclude` query parameter, as well as additional examples.

Description	Original Output	Exclude Value	Filtered Output
Include a forward slash (`/`) before property names or the request returns an error.	`{ "bar" : "bar value", "baz" : "baz value" }`	exclude=bar	400 error with error code INVALID_FILTER_VALUE
Filter properties by name.	`{ "bar" : "bar value", "baz" : "baz value" }`	exclude=/bar	`{ "baz" : "baz value" }`
Top-level properties cannot be filtered. Note	`<foo> <bar>bar value</bar>`	exclude=/foo	`<foo> <bar>bar value`

Description	Original Output	Exclude Value	Filtered Output
that this query string doesn't return an error.	<baz>baz value</baz> </foo>		</bar> <baz>baz value </baz> </foo>

Use a bar (|) delimiter to filter multiple properties. URL encode the delimiter as %7C.	{
	"foo" : "foo value",
	"bar" : "bar value",
	"baz" : "baz value"
	}
		exclude=/foo%7C/bar	{
			"baz" : "baz value"
			}

Filtering a response body filters everything in the response body.	<foo> <bar>bar value</bar> <baz>baz value</baz> <someObject> _{sub 1} </someObject> </foo>	exclude=/someObject	<foo> <bar>bar value</bar> <baz>baz value </baz> </foo>
To filter a property nested in a response body, include the parent response body name as a filter segment.	<foo> <bar>bar value</bar> <baz>baz value</baz> <someObject> _{sub value} </someObject> </foo>	exclude=/someObject/sub	<foo> <bar>bar value </bar> <baz>baz value </baz> <someObject> </someObject> </foo>
Identify an item in a list by its property name. This example uses XML.	<foo> <bar>bar valu</bar> <baz>baz value</baz> <someList> <item> <id>1</id> </item> <item> <id>2</id> </item> <item> <id>3</id> </item> </someList> </foo>	exclude=/someList/item/id	<foo> <bar>bar value </bar> <baz>baz value </baz> <someList> <item/> <item/> <item/> </someList> </foo>
Identify an item in a list by its property name. This example uses JSON.	{ "bar" : "bar value",	exclude=/someList/item/id	{ "bar" : "bar value", "baz" : "baz value",

Description	Original Output	Exclude Value	Filtered Output

	<pre>"baz" : "baz value",
	"someList" : [
	{
	"id" : "1"
	},
	{
	"id" : "2"
	} ,
	{
	"id" : "3"
	}
	]
	}</pre>
			<pre>"someList" : [
			{},
			{},
			{}
			]
			}</pre>

include

To limit the response body to specific properties, use the `include` query string parameter.

The value for the `include` query string parameter is a list of properties separated by bars (`|`). You must URL encode the bars as `%7C`. Include a forward slash before every property name.

The following request includes only the `aboutMe` property and the `address` property in a response body:

```
/chatter/users/me?include=/aboutMe%7C/address
```

The following table lists rules for using the `include` query parameter.

Description	Original Output	Include Value	Filtered Output

Include a forward slash (`/`) before property names or the request returns an error.	<pre>{
	"bar" : "bar value",
	"baz" : "baz value"
	}</pre>
		include=bar	400 error with error code `INVALID_FILTER_VALUE`

Filter properties by name.	<pre>{
	"bar" : "bar value",
	"baz" : "baz value"
	}</pre>
		include=/bar	<pre>{
			"bar" : "bar value",
			}</pre>

Top-level properties are ignored. Note there is no error when you try this.	<pre><foo>
	<bar>bar value</bar>
	<baz>baz value</baz>
	</foo></pre>
		include=/foo	<pre><foo>
			<bar>bar value
			</bar>
			<baz>baz value
			</baz>
			</foo></pre>

Use a bar (`|`) delimiter to filter multiple	<pre>{
	"foo" : "foo value",</pre>
		include=/foo%7C/bar	<pre>{
			"foo" : "foo</pre>

Description	Original Output	Include Value	Filtered Output

properties. URL encode the delimiter as `%7C`.	<pre>"bar" : "bar value",
	"baz" : "baz value"
	}</pre>
			<pre>value",
			"bar" : "bar
			value",
			}</pre>

Filtering a response body filters everything below it.	<pre><foo>
	<bar>bar value</bar>
	<baz>baz value</baz>
	<someObject>
	_{sub 1}
	</someObject>
	</foo></pre>
		include=/someObject	<pre><foo>
			<someObject>
			_{sub 1
			}
			</someObject>
			</foo></pre>

To filter a property nested in a response body, include the parent response body name as a filter segment.	<pre><foo>
	<bar>bar value</bar>
	<baz>baz value</baz>
	<someObject>
	_{sub value}
	</someObject>
	</foo></pre>
		include=/someObject/sub	<pre><foo>
			<someObject>
			_{sub 1
			}
			</someObject>
			</foo></pre>

Identify an item in a list by its property name. This example uses XML.	<pre><foo>
	<bar>bar valu</bar>
	<baz>baz value</baz>
	<someList>
	<item>
	<id>1</id>
	</item>
	<item>
	<id>2</id>
	</item>
	<item>
	<id>3</id>
	</item>
	</someList>
	</foo></pre>
		include=/someList/item/id	<pre><foo>
			<someList>
			<item>
			<id>1</id>
			</item>
			<item>
			<id>2</id>
			</item>
			<item>
			<id>3</id>
			</item>
			</someList>
			</foo></pre>

Identify an item in a list by its property name. This example uses JSON.	<pre>{
	"bar" : "bar value",
	"baz" : "baz value",
	"someList" : [
	{
	"id" : "1"
	},
	{
	"id" : "2"
	} ,
	{</pre>
		include=/someList/item/id	<pre>{
			"someList" : [
			{
			"id" : "1"
			},
			{
			"id" : "2"
			} ,
			{
			"id" : "3"
			}</pre>

Description	Original Output	Include Value	Filtered Output

	<pre>"id" : "3"
	}
	]
	}</pre>
			<pre>]
			}</pre>

Response Body Encoding

Chatter REST API responses are minimally HTML entity-encoded by default.

These characters are escaped in all strings:

Character	Escaped as
<	`<`
>	`>`
"	`"`
'	`'`
\	`\`
&	`&`

Chatter REST API does special encoding of any URL values included in response payloads. The main part of the URL is URL-encoded as per RFC2396, and the query string is HTML-form encoded. This encoding cannot be turned off.

Warning: Chatter REST API serves user-submitted content that may not be filtered at input and can come from many sources including third-party mobile and web applications. Process Chatter REST API output for the context in which your app uses the output. Failing to process output can expose applications and users to Cross Site Scripting (XSS) exploits and other problems that can result in data loss, application failure, and exposure of company information.

Chatter REST API output can be used in many contexts. Don't assume that the default entity encoding is appropriate for all contexts. Using Chatter REST API output inside HTML attribute values, inside URLs, with JavaScript, inside `<script>` tags, and inside CSS all require different encoding and whitelisting. See the Open Web Application Security Project for information on how to handle API output in different contexts.

For non-HTML contexts, such as native mobile applications, Chatter REST API clients can request raw (unencoded) output. Set the `X-Chatter-Entity-Encoding` HTTP header in a request to `false`.

Status Codes and Error Responses

The response header contains an HTTP status code. If the response isn't successful, the response body contains an error message accompanying that error, and if appropriate, information about the field or object where the error occurred.

HTTP Status Code	Description
200	Success with GET, PATCH, or HEAD request
201	Success with POST request
204	Success with DELETE
400	The request could not be understood, usually because the ID is not valid for the particular resource. For example, if you use a `userId` where a `groupId` is required, the request returns 400.
401	The session ID or OAuth token has expired or is invalid, or, if the request is made by a guest user, the resource isn't accessible to guest users. The response body contains the `message` and `errorCode`.
403	The request has been refused. Verify that the context user has the appropriate permissions to access the requested data, or that the context user is not an external user.
404	Either the specified resource was not found, or the resource has been deleted.
409	A conflict has occurred. For example, an attempt was made to update a request to join a group, but that request had already been approved or rejected.
412	A precondition has failed. For example, in a batch request, if `haltOnError` is `true` and a subrequest fails, subsequent subrequests return 412.
500	An error has occurred within Force.com, so the request could not be completed. Contact Salesforce Customer Support.
503	Too many requests in an hour.

Example

This request requires a comment ID, but it uses a feed item ID instead:

```
GET /chatter/comments/0D5D0000000XZoHKAW
```

The response body is:

```
HTTP/1.1 400 Bad Request
Date: Fri, 22 May 2015 17:32:10 GMT
Set-Cookie: BrowserId=D_3feNw7RYmP4DA;Path=/;Domain=.salesforce.com;Expires=Tue, 21-Jul-2015
 17:32:10 GMT
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked


[ {
  "errorCode" : "INVALID_ID_FIELD",
  "message" : "Invalid identifier: 0D5D0000000XZoHKAW"
} ]
```

Understanding OAuth and Chatter REST API

OAuth is an open protocol that allows secure authentication for access to a user's data, without handing out the user's username and password. It is often described as the valet key of software access: a valet key only allows access to certain features of your car: for example, you can't open the trunk or glove compartment using a valet key.

Chatter REST API uses OAuth to securely identify your application before connecting to Salesforce.

When you use OAuth, you can avoid storing login credentials in your application. Instead, your application can prompt the user to login using a standard Salesforce page, which returns an access token to your application. Your application can use this access token to access Chatter REST API Web services. This approach to authentication can be used in both mobile apps and from a Web page.

Chatter REST API uses OAuth 2.0.

Understanding Connected Apps

To use Chatter REST API, you must create a REST entry point in your organization. Salesforce defines this entry point as a connected app definition.

For an example in creating a connected app definition, see Step Two: Set Up Authorization on page 42.

Considerations when creating and using a connected app:

- The following maps the labels used in the connected app definition to OAuth terms:

Connected App Label	OAuth Term	Description
`Consumer Key`	`client_id`	A unique identifier that identifies your application to Salesforce.
`Consumer Secret`	`client_secret`	A secret key associated with your application.
`Callback URL`	`redirect_uri`	A URL associated with your client application. In some contexts, this must be a real URL that the client's Web browser is redirected to. In others, the URL isn't used; however, between your client application and the server (the connected app definition) the value must be the same. For example, you may want to use a value that identifies the application, such as http://mycomponent.myapp.

- If you are developing a mobile app, you can use the following value for the `Callback URL` (`redirect_uri`) to avoid setting up an application server of your own:

  ```
  https://login.instance_name/services/oauth2/success
  ```

  This value should NOT be used when developing a Web application.
- Your connected app does not have to reside in the same organization as your users. The connected app you create can be used to sign into any organization.

- We recommend creating more than one connected app definition: one for testing and one for production. If you create only one, you must change the value of the `Callback URL` to reflect the location of your application code. In addition, you should create more than one connected app if you are developing for different platforms, such as iOS and Android.

OAuth Basics

- OAuth grants access by client applications to resources owned by resource owners. In terms of OAuth, a resource is anything that must be kept secured. For Chatter REST API, the resource we are concerned about protecting is all the data accessible through the API.
- It is up to a resource owner to grant access to resources. In our context, the resource owner is a combination of the administrator (who administers users and the connected app) and the users (who log in and grant access to third party applications).
- When an application wants to access a resource, it requests authorization from the resource owner. OAuth outlines various ways that an application can be granted access to a resource. These are referred to as grant types or flows. Different flows are suitable for different contexts.
- Once a client application has been authorized to access to a resource, the client application is given an access token and a refresh token. The access token must be included by the authorized client application in any subsequent Web service requests to identify itself. Access tokens have a limited lifetime. When an access token expires, the authorized client application can make a special request using the refresh token to obtain a new access token.

OAuth Flows

A user must be authenticated before accessing Salesforce. OAuth has multiple authentication flows. There are several steps in each authentication flow, as dictated by the OAuth standard and the type of application trying to access Salesforce. On successful authorization, the client application is provided with access and refresh tokens.

Salesforce supports the following flows for use with the Chatter REST API:

- Web server flow
- User-agent flow

In addition, you can also use the refresh token to get a new access token after your application has been authorized for access.

For detailed information about using OAuth with Salesforce Communities, see *Getting Started with Communities*.

Revoking Access

After users have been granted access to a client application, they can revoke access by clicking **Revoke** in the Connected Apps section of their Personal Information page.

Additional Resources

To find out the latest information about Chatter REST API, as well as additional code examples, see https://developer.salesforce.com/page/Chatter_API.

Chatter REST API shares some infrastructure with the Force.com REST API. The information on authentication in these Force.com tutorials is also applicable to Chatter REST API:

- Getting Started with the Force.com REST API
- Digging Deeper into OAuth on Force.com
- Using OAuth to Authorize External Applications

The authentication part of the Force.com REST API is shared with Chatter REST API. You may find the following library supporting the Force.com REST API useful:

- Force.com JavaScript REST Toolkit

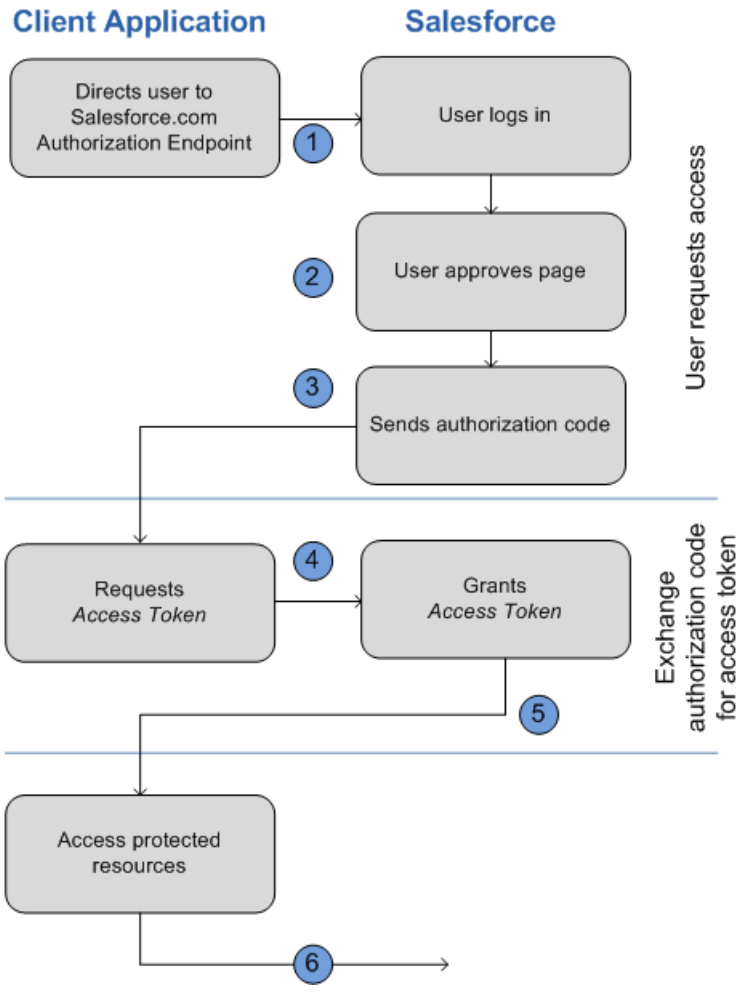
OAuth is a popular authentication standard. Client libraries for OAuth help smooth the development process for client applications. Here are some client programming libraries you may find useful:

- Ruby on Rails: OmniAuth
- Java Apache Amber

Understanding the Web Server OAuth Authentication Flow

Typically this flow is used by web applications that can confidentially store the client secret. A critical aspect of the Web server flow is that the application must be able to protect the consumer secret.

The following is the general flow. The individual step descriptions follow.



The following is a general description of the OAuth web-server flow:

1. To request authorization for a resource, the client application redirects the end-user's browser to a web page hosted on the resource owner's authorization server. In this case, it is the Salesforce log in page.

2. The end-user logs into Salesforce to authenticate themselves. Since this web page is hosted by the resource owner (Salesforce) and interacted with directly by the end user, the client web application never finds out the user's login credentials. The end-user also grants authorization to the client application.
3. Salesforce sends the authorization code back to the client application using the specified callback URL.
4. After obtaining the authorization code, the client application passes back the authorization code to obtain an access token.
5. After validating the authorization code, Salesforce passes back a response token. If there was no error, the response token includes an access code, a refresh token, and additional information.
6. The protected resources are Chatter REST API endpoints.

Using the Web Server Flow with Chatter REST API and Salesforce

The following provides specific details for the OAuth Web-server flow when used with Salesforce and Chatter REST API:

1. Direct the client's web browser to the page `https://login.`*`instance_name`*`/services/oauth2/authorize`, with the following request parameters:

Parameter	Description
`response_type`	Must be `code` for this authentication flow
`client_id`	The `Consumer Key` value from the connected app defined for this application
`redirect_uri`	The `Callback URL` value from the connected app defined for this application

You can also include the following optional request parameters:

Parameter	Description
`state`	Specifies URL-encoded state data to be returned in the callback URL after approval.
`immediate`	Determines whether the user should be prompted for login and approval. Values are either `true` or `false`. Default is `false`. • If set to `true`, and if the user is currently logged in and has previously approved the application, the approval step is skipped. • If set to `true` and the user is not logged in or has not previously approved the application, the session is immediately terminated with the `immediate_unsuccessful` error code.
`display`	Indicates the type of web pages that is provided. Valid values are: • `page`—Full-page authorization screen. This is the default value if none is specified.

Parameter	Description
	• `popup`—Compact dialog optimized for modern web browser popup windows. • `touch`—mobile-optimized dialog designed for modern smartphones such as Android and iPhone. • `mobile`—mobile optimized dialog designed for less capable smartphones such as BlackBerry OS 5.

2. After successfully being logged in, the user is asked to authorize the application.

Note that if the user has already authorized the application, this step is skipped.

3. Once Salesforce has confirmed that the client application is authorized, the end-user's web browser is redirected to the callback URL specified by the `redirect_uri` parameter, appended with the following values in its query string:

Parameter	Description
`code`	The authorization code that is passed to get the access and refresh tokens
`state`	The state value that was passed in as part of the initial request, if applicable.

It is expected that the `redirect_uri` web page is hosted by the client application server.

4. The client application server must extract the authorization code and pass it in a request to Salesforce for an access token. This request should be made as a POST against this URL: `https://login.`*`instance_name`*`/services/oauth2/token` with the following query parameters:

Parameter	Description
`grant_type`	Value must be `authorization_code` for this flow.
`client_id`	Consumer key from the connected app definition.
`client_secret`	Consumer secret from the connected app definition.
`redirect_uri`	URI to redirect the user to after approval. This must match the value in the `Callback URL` field in the connected app definition exactly, and is the same value sent by the initial redirect.
`code`	Authorization code obtained from the callback after approval.
`format`	Expected return format. This parameter is optional. The default is `json`. Values are: • `urlencoded` • `json` • `xml`

5. If this request is successful, the server returns a response body holding the following:

Parameters	Description
`access_token`	Session ID that you can use for making Chatter REST API requests. This session ID cannot be used in the user interface. Treat this like a user's session and diligently protect it.
`refresh_token`	Token that can be used in the future to obtain new access tokens (sessions). Warning: This value is a secret. You should treat it like the user's password and use appropriate measures to protect it.
`instance_url`	URL indicating the instance of the user's organization. In this example, the instance is `na1`: `https://na1.salesforce.com`
`id`	Identity URL that can be used to both identify the user as well as query for more information about the user. Can be used in an HTTP request to get more information about the end user.
`signature`	Base64-encoded HMAC-SHA256 signature signed with the consumer's private key containing the concatenated ID and

Parameters	Description
	`issued_at`. This can be used to verify the identity URL was not modified since it was sent by the server.
`issued_at`	When the signature was created.

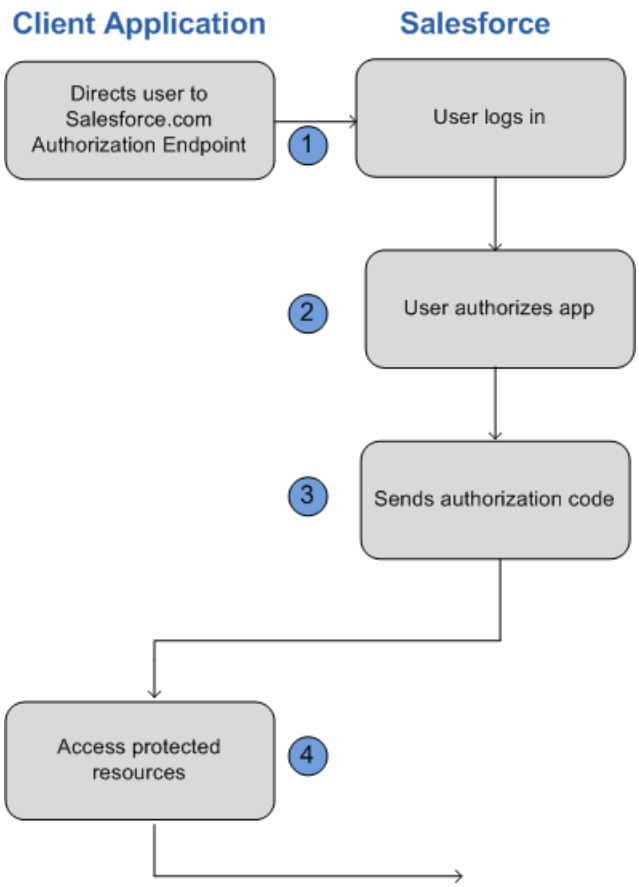
Understanding the User-Agent OAuth Authentication Flow

Typically this flow is used by mobile apps or applications that directly access the Chatter REST API from JavaScript. In this flow, it is assumed that the client application can't be trusted to store client credentials, nor the user login credentials.

Warning: Because the access token is encoded into the redirection URI, it might be exposed to the end-user and other applications residing on the computer or device.

If you are authenticating using JavaScript, we recommend calling `window.location.replace();` to remove the callback from the browser's history.

The following is the general flow. The individual step descriptions follow.



The following is a general description of the OAuth user-agent flow:

1. To request authorization for a resource, the client application redirects the end-user's browser to a Web page hosted on the resource owner's authorization server. In this case, it's the Salesforce login page.
2. The end-user logs into Salesforce to authenticate themselves. Since this Web page is hosted by the resource owner (Salesforce) and interacted with directly by the end user, the client Web application never finds out the user's login credentials.

   If the client application is running in the browser already, the browser can just be redirected. If the client application is a mobile app, a browser must be opened (such as an `UIWebView` in iOS). The end-user uses this web page to authenticate themselves and then grant authorization to the client application.
3. Once authorization has been granted, the authorization server redirects the end-user's Web browser to a redirection URL. This URL can be used by the client application to retrieve the access and refresh tokens.
4. The protected resources are Chatter REST API resources.

Using the User-Agent Flow with Chatter REST API and Salesforce

The following provides specific details for the OAuth user-agent flow when used with Salesforce and Chatter REST API:

1. Direct the client's Web browser to the page `https://login.`*`instance_name`*`/services/oauth2/authorize`, with the following request parameters:

Parameter	Description
`response_type`	Must be `token` for this authentication flow
`client_id`	The `Consumer Key` value from the connected app defined for this application
`redirect_uri`	The `Callback URL` value from the connected app defined for this application

You can also include the following optional request parameters:

Parameter	Description
`display`	Indicates the type of Web page that's provided. Valid values are: • `page`—Full-page authorization screen. This is the default value if none is specified. • `popup`—Compact dialog optimized for modern Web browser popup windows. • `touch`—mobile-optimized dialog designed for modern smartphones such as Android and iPhone. • `mobile`—mobile optimized dialog designed for older smartphones such as BlackBerry OS 5.
`scope`	Specifies what data your app can access. You do not have to specify any value to access Chatter REST API If you specify any values, you must include the `chatter_api` value.
`state`	Specifies URL-encoded state data to be returned in the callback URL after approval.

2. After successfully logging in, the user is asked to authorize the application. Note that if the user has already authorized the application, this step is skipped.
3. Once Salesforce has confirmed that the client application is authorized, the end-user's Web browser is redirected to the callback URL specified by the `redirect_uri` parameter, appended with the following values after the hash sign (#). This is not a query string.

Parameters	Description
`access_token`	Session ID that you can use for making Chatter REST API requests. This session ID cannot be used in the user interface. Treat this like a user's session and diligently protect it.
`expires_in`	Amount of time the access token is valid, in seconds.
`refresh_token`	Token that can be used in the future to obtain new access tokens (sessions). Warning: This value is a secret. You should treat it like the user's password and use appropriate measures to protect it. The refresh token is only returned if the redirect URI is `https://login.`*`instance_name`*`/services/oauth2/success` or used with a custom protocol that is not HTTPS.

The following additional parameters are returned; however, they aren't used with this flow and Chatter REST API:

- `instance_url`
- `id`
- `signature`
- `issued_at`

4. If the client application is a Web app, the redirection Web page that it uses may include JavaScript that the client application can manipulate in order to retrieve the access and refresh tokens.

   Alternatively, a client application (native or web) may use the page `https://login.`*`instance_name`*`/services/oauth2/success` as their callback URL. Instead of actually loading the page on redirect, they should intercept the redirection URI and extract the parameters values. Note that the `refresh_token` parameter values are only provided if the redirection URI is this URI or if the redirection URI uses a custom protocol.

   Once the client application has retrieved the access and refresh tokens, it can use them to make requests to Chatter REST API.

Using Tokens

As part of both authentication flows, you will be working with access tokens and refresh tokens.

Using Access Tokens

To make a successful Chatter REST API request, you must include a valid access token in the request. This can be done by using the HTTP Authorization header:

```
Authorization: OAuth <access token>
```

If the request yields an error response due to an expired token, the refresh token can be used to get a new access token.

If the request yields an error because an access token has been revoked, the client application must be re-authorized by the user in order to gain access.

Using Refresh Tokens

If the client application has a refresh token, it can use it to send a request for a new access token.

To ask for a new access token, the client application should send a POST request to `https://login.instance_name/services/oauth2/token` with the following query parameters:

Parameters	Description
`grant_type`	Value must be `refresh_token` for this flow.
`refresh_token`	The refresh token the client application already received.
`client_id`	Consumer key from the connected app definition.

You can also include the following optional parameters:

Parameters	Description
`client_secret`	Consumer secret from the connected app definition.
`format`	Expected return format. The default is `json`. Values are: • `urlencoded` • `json` • `xml`

If this request is successful, the server returns a payload holding the `access_token`.

The following additional parameters are also returned, however, they aren't used with this flow and Chatter REST API:

- `instance_url`
- `id`
- `signature`
- `issued_at`

Requesting Static Assets and Posting Forms Directly from HTML

HTML pages that aren't hosted on Salesforce have had difficulty displaying user and group images and posting file attachments in forms. The reason is that URLs for these assets require authentication, which is provided with a session cookie when pages are hosted on

Salesforce. Pages not hosted on Salesforce don't have access to the session cookie. Also, it isn't possible to pass an OAuth token from the HTML contexts `<img>`, `<a>`, and `<form>`.

To solve this problem, request self-authenticating *bearer token URLs*. A bearer token URL is a Chatter REST API resource with an appended bearer token that provides access to the resource. Use bearer token URLs as they are. Don't add or remove parameters or you'll receive an error. Use bearer token URLs in `<img>`, `<a>`, and `<form>` tags to do the following:

- Request static Chatter REST API assets such as Chatter user profile images, files, and file and dashboard previews.
- Post form requests to Chatter REST API when files are attached, for example, feed items or comments with a file attachment, or user profile photo updates.

To request static assets or post form requests, request a resource that returns a bearer token URL. Then use the bearer token URL to request the resource.

1. Request a resource with a response body property that contains a bearer token URL:
   - Comment Page—`currentPageUrl`*
   - Content Capability—`downloadUrl`, `externalDocumentUrl`, `renditionUrl`, `repositoryFileUrl`
   - Feed—`feedElementPostUrl`*, `feedElementsUrl`*, `feedItemsUrl`*
   - Feed Item—`photoUrl`
   - Feed Item Attachment: Content—`downloadUrl`, `externalDocumentUrl`, `renditionUrl`, `repositoryFileUrl`
   - Feed Item Attachment: Dashboard—`fullSizeImageUrl`, `thumbnailUrl`
   - File Detail—`downloadUrl`, `externalDocumentUrl`, `renditionUrl`, `renditionUrl240By180`, `renditionUrl720By480`, `repositoryFileUrl`
   - File Summary—`downloadUrl`, `externalDocumentUrl`, `renditionUrl`, `renditionUrl240By180`, `renditionUrl720By480`, `repositoryFileUrl`
   - File Summary Page—`currentPageUrl`*
   - Icon—`url`
   - Mention Completion—`photoUrl`
   - Photo—`largePhotoUrl`, `smallPhotoUrl`, `url`*
   - Repository File Detail—`downloadUrl`
   - Repository File Summary—`downloadUrl`

   *Use to POST or PUT new items that include binary files: for example, a feed item or comment with a file attachment.

2. Pass the `X-Connect-Bearer-Urls` request header with a value of `true`:

   ```
   X-Connect-Bearer-Urls: true
   ```

3. Parse the bearer token URL from the response body and use it in an HTML tag to request a static asset or to post a form request.

   Bearer token URLs have the following characteristics:
   - The bearer token is good for the given URL only. Adding or reordering query parameters invalidates the URL.
   - The bearer token has a time-to-live (TTL) of 20 minutes. If the token expires you must request it again.
   - The request to the bearer token URL authenticates as the user who requested the resource that returned the bearer token URL.

Use CORS to Access Chatter REST API and REST API

Chatter REST API and REST API support CORS (cross-origin resource sharing). To access these APIs from JavaScript in a Web browser, add the origin serving the script to the CORS whitelist.

CORS is a W3C recommendation that enables Web browsers to request resources from origins other than their own (cross-origin request). For example, using CORS, a JavaScript script at `https://www.example.com` could request a resource from `https://www.salesforce.com`.

If a browser that supports CORS makes a request to an origin in the Salesforce CORS whitelist, Salesforce returns the origin in the `Access-Control-Allow-Origin` HTTP header, along with any additional CORS HTTP headers. If the origin is not included in the whitelist, Salesforce returns HTTP status code 403.

EDITIONS

Available in: **Developer**, **Enterprise**, **Performance**, and **Unlimited**

USER PERMISSIONS

To create, read, update, and delete:

- "Modify All Data"

1. From Setup, choose **Security Controls** > **CORS**.
2. Choose **New**.
3. Enter an origin URL pattern.

   The origin URL pattern must include the HTTPS protocol and a domain name, and may include a port. The wildcard character (*) is supported and must be in front of a second-level domain name. For example, `https://*.example.com` adds all subdomains of `example.com` to the whitelist.

   The origin URL pattern can be an IP address. However, an IP address and a domain that resolve to the same address are not the same origin, and you must add them to the CORS whitelist as separate entries.

Important: You must still pass an OAuth token with requests that require it.

CHAPTER 2 Chatter REST API Quick Start

In this chapter ...

- Prerequisites
- Step One: Obtain a Salesforce Developer Edition Organization
- Step Two: Set Up Authorization
- Step Three: Connect to Chatter REST API Using OAuth
- Connecting to Salesforce Communities

Connect to Salesforce and authenticate, then make a request to Chatter REST API and look at the response.

Prerequisites

Complete these prerequisites before you begin the quick start.

Become familiar with:

- cURL, which is a command-line tool the quick start uses as the client application that makes HTTP requests to Salesforce. cURL is pre-installed on many Linux and Mac systems. Windows users can download a version at `curl.haxx.se/`. When using HTTPS on Windows, ensure that your system meets the cURL requirements for SSL.

  Note: cURL is an open source tool and is not supported by Salesforce.

- JavaScript Object Notation (JSON), which is the data format returned in this quick start.
- OAuth 2.0, which is the framework Salesforce uses for authentication. The quick start provides the steps, but it would be helpful to familiarize yourself with OAuth terms and concepts.

Step One: Obtain a Salesforce Developer Edition Organization

If you are not already a member of the Force.com developer community, go to `developer.salesforce.com/signup` and follow the instructions for signing up for a Developer Edition organization. Even if you already have Enterprise Edition, Unlimited Edition, or Performance Edition, use Developer Edition for developing, staging, and testing your solutions against sample data to protect your organization's live data. This is especially true for applications that insert, update, or delete data (as opposed to simply reading data).

If you already have a Developer Edition organization, verify that you have the "API Enabled" permission. This permission is enabled by default, but may have been changed by an administrator. For more information, see the help in the Salesforce user interface.

Step Two: Set Up Authorization

Create a connected app in a Salesforce organization and enable OAuth. The client application uses the connected app to connect to Salesforce.

1. In your Developer Edition organization, from Setup, click **Create** > **Apps**, and in the Connected Apps section, click **New** to create a new connected app.

   Clients can use a connected app to sign in to any organization, even if the connected app isn't defined in that organization.

2. Enter a connected app name.
3. Enter the contact email, as well as any other required information.
4. Select `Enable OAuth Settings`.
5. Enter a `Callback URL`. It must be secure: `http://` does not work, only `https://` works.
   For this quick start, you can simply enter *`https://`*.
6. Enter an OAuth scope. Select `Access and manage your Chatter feed` in addition to any other scope you want your connected app to allow access to.
7. Click Save.
   The `Consumer Key` is created and displayed, and a `Consumer Secret` is created (click the link to reveal it).

Step Three: Connect to Chatter REST API Using OAuth

Use OAuth to connect to Salesforce and get an access token. Pass the access token in requests to Chatter REST API.

Complete Step Two: Set Up Authorization and create a connected app before starting this task.

This table maps the terms used in the connected app you created to the OAuth properties used in the examples. The OAuth 2.0 specification uses the word "client" instead of "consumer."

Connected App Application Label	Value in Example
`Consumer Key`	`client_id`
`Consumer Secret`	`client_secret`

Note: This quick start uses the username-password OAuth authentication flow. The username-password authentication flow passes the user's credentials back and forth. Use this authentication flow only when necessary, such as in this quick start. No refresh token will be issued. In addition, Salesforce Communities doesn't support the username-password authentication flow. In this quick start, don't make a request to a Communities URL.

To make a request to Salesforce, substitute values from your organization into these examples:

1. Generate the access token.

   This cURL command generates an access token:

   ```
   curl --form client_id=3MVG9PhR6g6B7ps4xDycwGrI4PvjVZvK9
        --form client_secret=8870355475032095511
        --form grant_type=password
        --form username=admin@seattleapps.com
        --form password=1Lsfdc!
         https://login.salesforce.com/services/oauth2/token
   ```

   Tip: To paste a multi-line command into a Mac or Linux command line interface, escape each line with a backslash ("\") to indicate that the command continues on the next line. An escaped line looks like this:

   ```
   curl --form client_id=3MVG9PhR6g6B7ps4xDycwGrI4PvjVZvK9 \
   ```

   To paste a multi-line command into the Windows Command Prompt, escape each line with a caret ("^"). An escaped line looks like this:

   ```
   curl --form client_id=3MVG9PhR6g6B7ps4xDycwGrI4PvjVZvK9 ^
   ```

   The response includes the server instance and the access token:

   ```
   {
   "id":"https://login.salesforce.com/id/00Di0000000hT9uEAE/005i00000022uIbAAI",
   "issued_at":"1302907727777",
   "instance_url":"https://na1.salesforce.com",
   "signature":"5jcevY5fUai0lWntuSxkwBzWcvRjd01RCOkIBZpyGv0=",
   "access_token":"00DD0000000FJ6T!AQkAQPde_DMF2vGzddfZmBRS95GojDbtA
        rKkgukAgZP0OVFYY5KkAqhLw9ejeKIlpJ3FgwGAWeRlBiWRt8mfXEuAZGbZNosk"
   }
   ```

2. To request a Chatter REST API resource, use the returned `instance_url` as the server instance. Pass the returned `access_token` as a `Bearer` token in the `Authorization` request header.

```
curl -X GET https://na1.salesforce.com/services/data/v34.0/chatter/users/me
     -H 'Authorization: Bearer 00DD0000000FJ6T!AQkAQPde_DMF2vGzddfZmBRS95Goj
     DbtArKkgukAgZP0OVFYY5KkAqhLw9ejeKIlpJ3FgwGAWeRlBiWRt8mfXEuAZGbZNosk'
```

This example uses these values:

Property	Value
Server instance	na1.salesforce.com
`client_id`	3MVG9PhR6g6B7ps4xDycwGrI4PvjVZvK9
`client_secret`	8870355475032095511
`grant_type`	password The value of `grant_type` depends on the OAuth authentication flow you are using.
`username`	admin@seattleapps.com
`password`	1Lsfdc!

Connecting to Salesforce Communities

To use OAuth to connect to a Salesforce community, replace the server instance name with the full path to the community URL.

To connect to a Salesforce community, you can use the OAuth Web server and user-agent workflows.

To authenticate a user using the authorize URL, replace the `login.salesforce.com` host name with the full path to the community URL. This is the non-Communities URL:

```
https://login.salesforce.com/services/oauth2/authorize?
response_type=token&client_id=your_app_id&redirect_uri=your_redirect_uris
```

This is the Communities URL:

```
https://acme.force.com/customers/services/oauth2/authorize?
response_type=token&client_id=your_app_id&redirect_uri=your_redirect_uri
```

When implemented successfully, this URL directs users to your app's branded login page. After they authorize the app, set a user access token and a refresh token for future authentication. In requests for the token endpoint, replace the host with the community, like this:

```
https://acme.force.com/customers/services/oauth2/token
```

To request a Chatter REST API resource, use the Salesforce host name and specify the community ID:

```
https://na1.salesforce.com/services/data/v29.0/connect
/communities/communityId/chatter/feeds/news/me/feed-elements
```

Alternately, you can replace the host name with the full path to the community URL:

```
https://communitydomain.force.com/communitypath/services/data/v29.0/connect
/communities/communityId/chatter/feeds/news/me/feed-elements
```

CHAPTER 3 Chatter REST API Features

In this chapter ...

- Working with Action Links
- Working with Communities
- Working with Feeds and Feed Elements
- Working with Salesforce Files

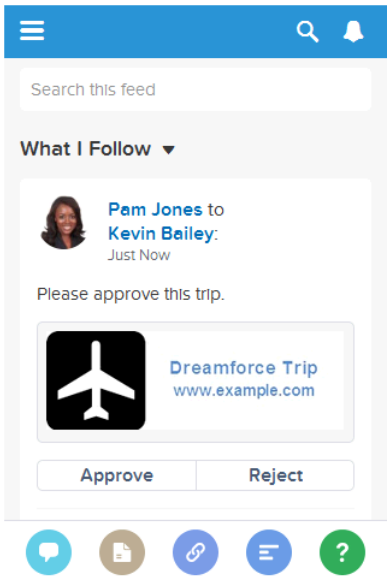
This chapter of the *Chatter REST API Developer's Guide* describes which resources to use to work with Chatter REST API features.

Working with Action Links

An action link is a button on a feed element. Clicking an action link can take a user to a Web page, initiate a file download, or invoke an API call to Salesforce or to an external server. An action link includes a URL and an HTTP method, and can include a request body and header information, such as an OAuth token for authentication. Use action links to integrate Salesforce and third-party services into the feed so that users can take action to drive productivity and accelerate innovation.

Workflow

This feed item contains one action link group with two action links: **Approve** and **Reject**.



The workflow to create and post action links with a feed element:

1. (Optional) Create an action link template.
2. Make a request to /connect/action-link-group-definitions to define an action link group or to instantiate an action link group from a template. An action link group contains at least one action link.
3. Make a request to /chatter/feed-elements to post a feed element and associate the action link with it.

Action Links Resources

Use these resources to work with action links:

Resource	HTTP Method	Task
`/connect/action-link-group-definitions`	POST	Create an action link group definition. To associate an action link with a feed element, first create an action link group definition. Then post a feed element with an associated actions capability.

Resource	HTTP Method	Task
`/chatter/feed-elements`	POST	Post a feed element with an associated actions capability. Associate up to 10 action link groups with a feed element.
`/connect/action-links/actionLinkId`	GET, PATCH	Get information about or update the `status` of an action link. The `status` of the action link in combination with the `labelKey` property of the Action Link Definition Input determines the text that displays in the UI. To trigger the API request for `Api` and `ApiAsync` action links, set the `status` to `PendingStatus`.
`/connect/action-link-group-definitions/actionLinkGroupId`	GET, DELETE	Get information about or delete an action link group definition. Deleting an action link group definition removes all references to it from feed elements. To get or delete the action link group definition, the client must be the same connected app that created the definition and the user must either be the user who created the definition or have "View all Data" permission.
`/connect/action-link-groups/actionLinkGroupId`	GET	Get information about an action link group including state for the context user.
`/connect/action-links/actionLinkId/diagnostic-info`	GET	Get diagnostic information returned when an action link executes. Diagnostic information is given only for users who can access the action link.
`/chatter/feed-elements/feedElementId/capabilities/associated-actions`	GET	Get all the platform actions associated with a feed element. Action links are a type of platform action.

Feeds Resources
`/chatter/feeds/feedType/userId/feed-elements`
`/chatter/feeds/feedType/feed-elements`
	GET	Get the feed elements of a specified feed type. If a feed element has action links associated with it, the action links data is returned in the associated actions capability.

IN THIS SECTION:

Action Links Use Cases

Use action links to integrate Salesforce and third-party services with a feed. An action link can make an HTTP request to a Salesforce or third-party API. An action link can also download a file or open a Web page. This topic contains two use case examples.

Action Links Overview, Authentication, and Security

Learn about Chatter REST API action links security, authentication, labels, and errors.

Action Link Templates

Create action link templates in Setup so that you can instantiate action link groups with common properties from Chatter REST API or Apex. You can package templates and distribute them to other Salesforce organizations.

SEE ALSO:

Define Action Links and Post with a Feed Element

Define Action Links in a Template and Post with a Feed Element

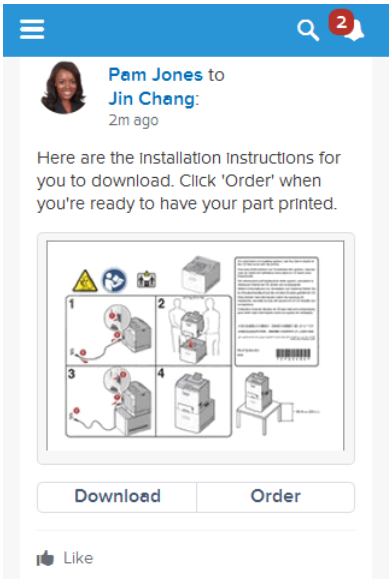
Define Action Links in a Template and Post with a Feed Element

Action Links Use Cases

Use action links to integrate Salesforce and third-party services with a feed. An action link can make an HTTP request to a Salesforce or third-party API. An action link can also download a file or open a Web page. This topic contains two use case examples.

Download Instructions and Order a Part from the Feed

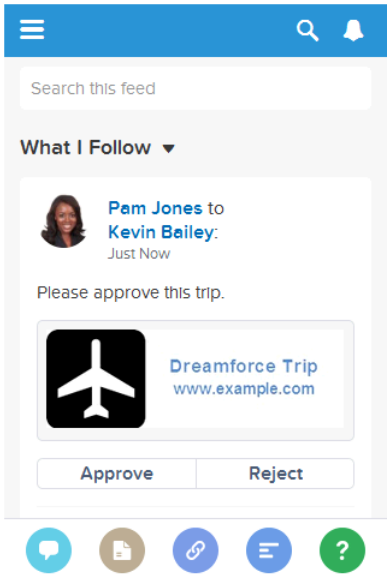
Let's imagine that a fictional 3D print shop called "BuildIt" creates an AppExchange app that uses action links to integrate with Salesforce. If a BuildIt customer has the app installed, they can go to the BuildIt website, create an account, and send posts containing action links to Salesforce that invite certain users to order parts from BuildIt. In this sample feed element posted from BuildIt customer Pam Jones to her report Jin Chang, clicking the **Download** action link downloads a file with parts information from the BuildIt website. Clicking the **Order** action link takes Jin Chang to a page on the BuildIt website to order a part.



Approve or Reject an Itinerary from the Feed

Suppose you work as a developer for a fictional company called "Travel Example" and you create an app for the Salesforce AppExchange called "Travel Example for Salesforce." Company A has "Travel Example for Salesforce" installed. Let's imagine that someone from Company A goes to the Travel Example website and books an itinerary. Your Travel Example server-side code makes a Chatter REST API request to post a feed item in Company A's Salesforce organization with the body text **Please approve this trip.**, and action links with the

labels **Approve** and **Reject**. The action links make HTTP requests to a REST resource on the Travel Example server to approve or reject the trip.



To create the "Travel Example for Salesforce" app, you created action link templates in Setup that define these action links and the action link group they're associated with. Your server-side code makes a Chatter REST API request to instantiate the action link group from the template. It makes another Chatter REST API request to associate the action link group with a feed item and post it.

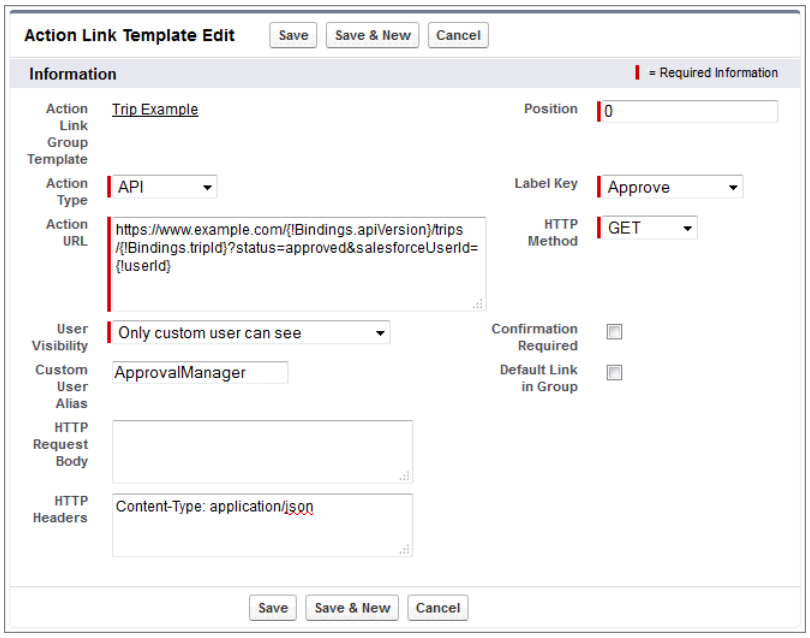
The `Action URL`, `HTTP Headers`, and `HTTP Request Body` template fields can contain *context variables* and *binding variables*.

You include a `{!userId}` context variable in the `Action URL` so that when a user clicks the action link in the feed, Salesforce sends the user's ID to your server-side code to tell it who clicked the action link. Action links support a predefined set of context variables.

You can create any binding variables you need in the action link template to make your code future-proof and dynamic. Your server-side code sets the binding variable values when it instantiates the action link group. You decide to use `{!Bindings.apiVersion}` so that you can release a new API version without having to update your package. You also use `{!Bindings.tripId}` so that clicking the action link approves or rejects the trip the user created.

You set the `User Visibility` field so that only one user can see the action links. You enter a value in the `Custom User Alias` field which is also a binding variable whose value your server-side code sets to the ID of the user's manager when it instantiates the action link group.

This is the action link template for the **Approve** action link:

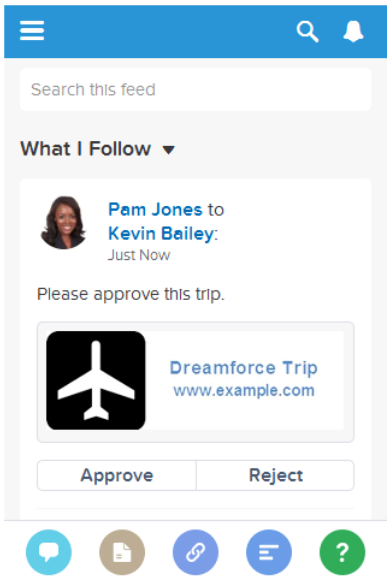


Action Links Overview, Authentication, and Security

Learn about Chatter REST API action links security, authentication, labels, and errors.

Workflow

This feed item contains one action link group with two action links: **Approve** and **Reject**.



The workflow to create and post action links with a feed element:

1. (Optional) Create an action link template.

2. Make a request to /connect/action-link-group-definitions to define an action link group or to instantiate an action link group from a template. An action link group contains at least one action link.
3. Make a request to /chatter/feed-elements to post a feed element and associate the action link with it.

Action Link Templates

Create action link templates in Setup to instantiate action link groups with common properties. You can package templates and distribute them to other Salesforce organizations.

Specify binding variables in the template and set the values of the variables when you instantiate the action link group. For example, use a binding variable for the API version number, a user ID, or an OAuth token.

You can also specify context variables in the templates. When a user executes the action link, Salesforce provides values for these variables, such as who executed the link and in which organization.

To instantiate the action link group, make a request to the `/connect/action-link-group-definitions` resource. Specify the template ID and the values for any binding variables defined in the template.

See Designing Action Link Templates.

Type of Action Links

Specify the action link type in the `actionType` property when you define an action link.

There are four types of action links:

- `Api`—The action link calls a synchronous API at the action URL. Salesforce sets the status to `SuccessfulStatus` or `FailedStatus` based on the HTTP status code returned by your server.
- `ApiAsync`—The action link calls an asynchronous API at the action URL. The action remains in a `PendingStatus` state until a third party makes a request to `/connect/action-links/`***`actionLinkId`*** to set the status to `SuccessfulStatus` or `FailedStatus` when the asynchronous operation is complete.
- `Download`—The action link downloads a file from the action URL.
- `Ui`—The action link takes the user to a web page at the action URL.

Different types of action links have different workflows to set the status of an action link (to `Pending`, `Successful`, or `Failed`). See `/connect/action-links/`***`actionLinkId`*** for more information.

Authentication

When you define an action link, specify a URL (`actionUrl`) and the HTTP headers (`headers`) required to make a request to that URL.

If an external resource requires authentication, include the information wherever the resource requires.

If a Salesforce resource requires authentication, you can include OAuth information in the HTTP headers or you can include a bearer token in the URL.

Salesforce automatically authenticates these resources:

- Relative URLs in templates
- Relative URLs beginning with `/services/apexrest` when the action link group is instantiated from Apex

Don't use these resources for sensitive operations.

Security

HTTPS

The action URL in an action link must begin with `https://` or be a relative URL that matches one of the rules in the Authentication section.

Encryption

API details are stored with encryption, and obfuscated for clients.

The `actionURL`, `headers`, and `requestBody` data for action links that are not instantiated from a template are encrypted with the organization's encryption key. The `Action URL`, `HTTP Headers`, and `HTTP Request Body` for an action link template are not encrypted. The binding values used when instantiating an action link group from a template are encrypted with the organization's encryption key.

Action Link Templates

Only users with "Customize Application" user permission can create, edit, delete, and package action link templates in Setup.

Don't store sensitive information in templates. Use binding variables to add sensitive information when you instantiate the action link group. After the action link group is instantiated, the values are stored in an encrypted format. See Define Binding Variables.

Connected Apps

When creating action links via a connected app, it's a good idea to use a connected app with a consumer key that never leaves your control. The connected app is used for server-to-server communication and is not compiled into mobile apps that could be decompiled.

Expiration Date

When you define an action link group, specify an expiration date (`expirationDate`). After that date, the action links in the group can't be executed and disappear from the feed. If your action link group definition includes an OAuth token, set the group's expiration date to the same value as the expiration date of the OAuth token.

Action link templates use a slightly different mechanism for excluding a user. See Set the Action Link Group Expiration Time.

Exclude a User or Specify a User

Use the `excludeUserId` property of the action link definition input to exclude a single user from executing an action.

Use the `userId` property of the action link definition input to specify the ID of a user who alone can execute the action. If you don't specify a `userId` property or if you pass `null`, any user can execute the action. You can't specify both `excludeUserId` and `userId` for an action link

Action link templates use a slightly different mechanism for excluding a user. See Set Who Can See the Action Link.

Read, Modify, or Delete an Action Link Group Definition

There are two views of an action link and an action link group: the definition, and the context user's view. The definition includes potentially sensitive information, such as authentication information. The context user's view is filtered by visibility options and the values reflect the state of the context user.

Action link group definitions can contain sensitive information (such as OAuth tokens). For this reason, to read, modify, or delete a definition, the user must have created the definition or have "View All Data" permission. In addition, in Chatter REST API, the request must be made via the same connected app that created the definition. In Apex, the call must be made from the same namespace that created the definition.

Context Variables

Use context variables to pass information about the user who executed the action link and the context in which it was invoked into the HTTP request made by invoking an action link. You can use context variables in the `actionUrl`, `headers`, and `requestBody` properties of the Action Link Definition Input request body or `ConnectApi.ActionLinkDefinitionInput` object. You can also use context variables in the `Action URL`, `HTTP Request Body`, and `HTTP Headers` fields of action link templates. You can edit these fields, including adding and removing context variables, after a template is published.

The context variables are:

Context Variable	Description
`{!actionLinkId}`	The ID of the action link the user executed.
`{!actionLinkGroupId}`	The ID of the action link group containing the action link the user executed.
`{!communityId}`	The ID of the community in which the user executed the action link. The value for your internal organization is the empty key `"000000000000000000"`.
`{!communityUrl}`	The URL of the community in which the user executed the action link. The value for your internal organization is empty string `""`.
`{!orgId}`	The ID of the organization in which the user executed the action link.
`{!userId}`	The ID of the user that executed the action link.

Versioning

To avoid issues due to upgrades or changing functionality in your API, we recommend using versioning when defining action links. For example, the `actionUrl` property in the Action Link Definition Input should look like `https://www.example.com/api/v1/exampleResource`.

You can use templates to change the values of the `actionUrl`, `headers`, or `requestBody` properties, even after a template is distributed in a package. For example, if you release a new API version that requires new inputs, an admin can change the inputs in the action link template. Even action links already associated with a feed element use the new inputs. However, you can't add new binding variables to a published action link template.

If your API isn't versioned, you can use the `expirationDate` property of the Action Link Group Definition Input to avoid issues due to upgrades or changing functionality in your API. See Set the Action Link Group Expiration Time.

Errors

Use the Action Link Diagnostic Information resource (`/connect/action-links/actionLinkId/diagnostic-info`) to return status codes and errors from executing `Api` and `AsyncApi` action links. Diagnostic info is given only for users who can access the action link.

Localized Labels

Action links use a predefined set of localized labels specified in the `labelKey` property of the Action Link Definition Input request body and the `Label Key` field of an action link template.

For a list of labels, see Action Links Labels.

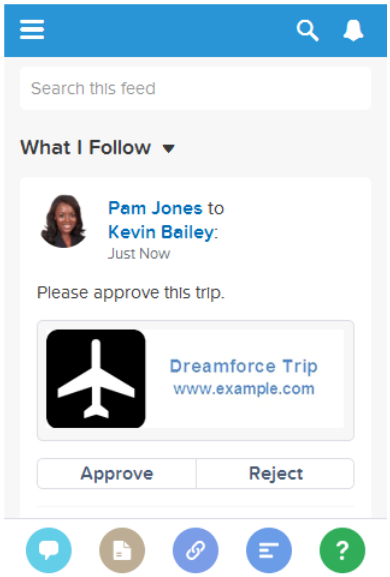
Note: If none of the label key values make sense for your action link, specify a custom label in the `Label` field of an action link template and set `Label Key` to None. However, custom labels aren't localized.

Action Link Templates

Create action link templates in Setup so that you can instantiate action link groups with common properties from Chatter REST API or Apex. You can package templates and distribute them to other Salesforce organizations.

An action link is a button on a feed element. Clicking an action link can take a user to a Web page, initiate a file download, or invoke an API call to Salesforce or to an external server. An action link includes a URL and an HTTP method, and can include a request body and header information, such as an OAuth token for authentication. Use action links to integrate Salesforce and third-party services into the feed so that users can take action to drive productivity and accelerate innovation.

In this example, **Approve** and **Reject** are action links that make API calls to the REST API of a fictional travel website to approve or reject an itinerary. When Pam created the itinerary on the travel website, the travel website made a Chatter REST API request to post the feed item with the action links to Pam's manager Kevin so that he can approve or reject the itinerary.



Important: Action links are a developer feature. Although you create action link templates in Setup, you must use Apex or Chatter REST API to generate action links from templates and add them to feed elements.

IN THIS SECTION:

Designing Action Link Templates

Before you create a template, consider which values you want to set in the template and which values you want to set with binding variables when you instantiate action link groups from the template.

Create Action Link Templates

Create action link templates in Setup so that you can instantiate action link groups with common properties from Chatter REST API or Apex. You can package templates and distribute them to other Salesforce organizations.

Edit Action Link Templates

You can edit all fields on an unpublished action link group template and on its associated action link templates.

Delete Action Link Group Templates

When you delete an action link group template, you delete its associated action link templates and all action link groups that have been instantiated from the templates. Deleted action link groups disappear from any feed elements they've been associated with.

Package Action Link Templates

Package action link templates to distribute them to other Salesforce organizations.

SEE ALSO:

Define Action Links in a Template and Post with a Feed Element

Define Action Links in a Template and Post with a Feed Element

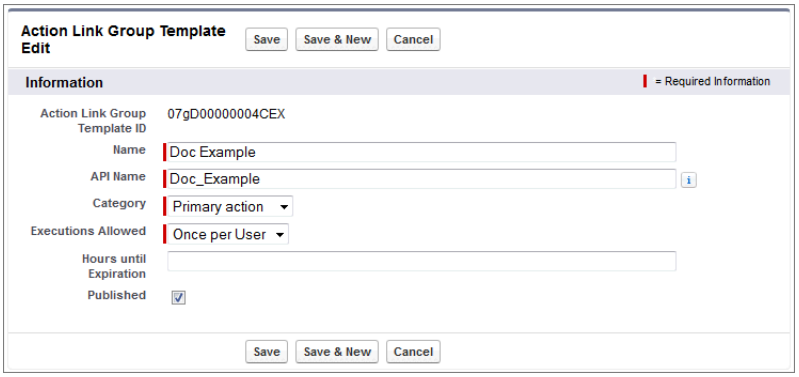
Designing Action Link Templates

Before you create a template, consider which values you want to set in the template and which values you want to set with binding variables when you instantiate action link groups from the template.

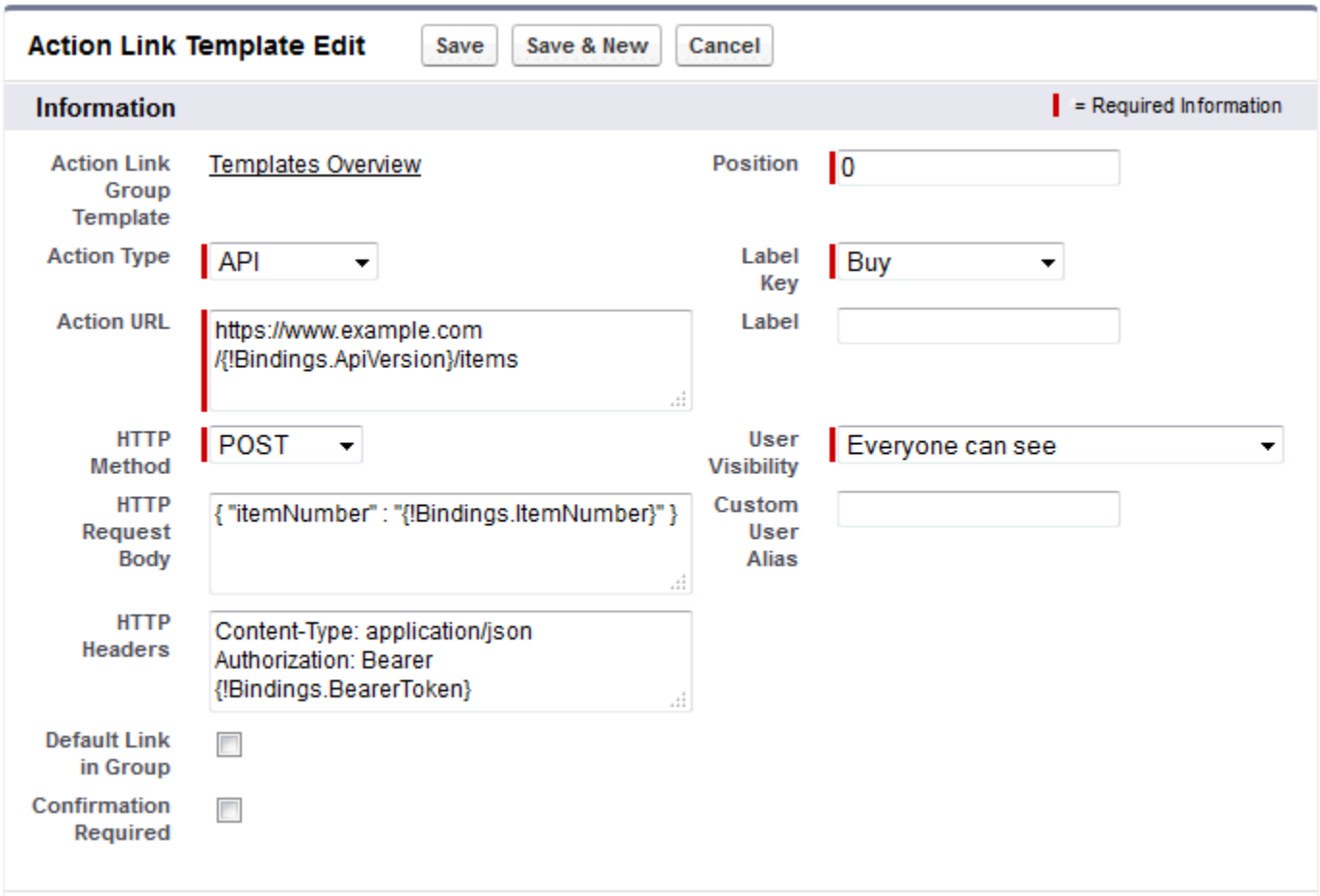
- Action Link Templates Overview
- Template Design Considerations
- Set the Action Link Group Expiration Time
- Define Binding Variables
- Set Who Can See the Action Link
- Use Context Variables

Action Link Templates Overview

Here's an action link group template in Setup:



Each action link group should contain at least one action link. This example action link template has three binding variables: the API version number in the `Action URL`, the Item Number in the `HTTP Request Body`, and the OAuth token value in the `HTTP Header` field.



The Chatter REST API request to instantiate the action link group and set the values of the binding variables:

```
POST /connect/action-link-group-definitions

{
   "templateId":"07gD00000004C9r",
   "templateBindings":[
      {
         "key":"ApiVersion",
         "value":"v1.0"
      },
      {
         "key":"ItemNumber",
         "value":"8675309"
      },
      {
         "key":"BearerToken",

"value":"00DRR0000000N0g!ARoAQMZyQtsP1Gs27EZ8hl7vdpYXH5O5rv1VNprqTeD12xYnvygD3JgPnNR"
      }
   ]
}
```

This is the Apex code that instantiates the action link group from the template and sets the values of the binding variables:

```
// Get the action link group template Id.
ActionLinkGroupTemplate template = [SELECT Id FROM ActionLinkGroupTemplate WHERE
DeveloperName='Doc_Example'];

// Add binding name-value pairs to a map.
Map<String, String> bindingMap = new Map<String, String>();
bindingMap.put('ApiVersion', '1.0');
bindingMap.put('ItemNumber', '8675309');
bindingMap.put('BearerToken',
'00DRR0000000N0g!ARoAQMZyQtsP1Gs27EZ8hl7vdpYXH5O5rv1VNprqTeD12xYnvygD3JgPnNR');

// Create ActionLinkTemplateBindingInput objects from the map elements.
List<ConnectApi.ActionLinkTemplateBindingInput> bindingInputs = new
List<ConnectApi.ActionLinkTemplateBindingInput>();
for (String key : bindingMap.keySet()) {
    ConnectApi.ActionLinkTemplateBindingInput bindingInput = new
ConnectApi.ActionLinkTemplateBindingInput();
    bindingInput.key = key;
    bindingInput.value = bindingMap.get(key);
    bindingInputs.add(bindingInput);
}

// Set the template Id and template binding values in the action link group definition.
ConnectApi.ActionLinkGroupDefinitionInput actionLinkGroupDefinitionInput = new
ConnectApi.ActionLinkGroupDefinitionInput();
actionLinkGroupDefinitionInput.templateId = template.id;
actionLinkGroupDefinitionInput.templateBindings = bindingInputs;

// Instantiate the action link group definition.
ConnectApi.ActionLinkGroupDefinition actionLinkGroupDefinition =
ConnectApi.ActionLinks.createActionLinkGroupDefinition(Network.getNetworkId(),
actionLinkGroupDefinitionInput);
```

Template Design Considerations

Considerations for designing a template:

- Determine the expiration time of the action link group.

  See Set the Action Link Group Expiration Time.

- Define *binding variables* in the template and set their values when you instantiate the group. Don't store sensitive information in templates. Use binding variables to add sensitive information at run time.

  See Define Binding Variables.

- Determine who can see the action link when it's associated with a feed element.

  Set Who Can See the Action Link.

- Use *context variables* in the template to get information about the execution context of the action link.

  When the action link executes, Salesforce fills in the values and sends them in the HTTP request. See Use Context Variables.

Set the Action Link Group Expiration Time

When creating an action link group from a template, the expiration date can be calculated based on a period provided in the template, or the action link group can be set not to expire at all.

To set the hours until expiration in a template, enter a value in the `Hours until Expiration` field of the action link group template. This value is the number of hours from when the action link group is instantiated until it's removed from associated feed elements and can no longer be executed. The maximum value is 8760, which is 365 days.

To set the action link group expiration date when you instantiate it, set the `expirationDate` property of either the Action Link Group Definition request body (Chatter REST API) or the `ConnectApi.ActionLinkGroupDefinition` input class (Apex).

To create an action link group that doesn't expire, don't enter a value in the `Hours until Expiration` field of the template and don't enter a value for the `expirationDate` property when you instantiate the action link group.

Here's how `expirationDate` and `Hours until Expiration` work together when creating an action link group from a template:

- If you specify `expirationDate`, its value is used in the new action link group.
- If you don't specify `expirationDate` and you specify `Hours until Expiration` in the template, the value of `Hours until Expiration` is used in the new action link group.
- If you don't specify `expirationDate` or `Hours until Expiration`, the action link groups instantiated from the template don't expire.

Define Binding Variables

Define binding variables in templates and set their values when you instantiate an action link group.

Important: Don't store sensitive information in templates. Use binding variables to add sensitive information at run time. When the value of a binding is set, it is stored in encrypted form in Salesforce.

You can define binding variables in the `Action URL`, `HTTP Request Body`, and `HTTP Headers` fields of an action link template. After a template is published, you can edit these fields, you can move binding variables between these fields, and you can delete binding variables. However, you can't add new binding variables.

Define a binding variable's key in the template. When you instantiate the action link group, specify the key and its value.

Binding variable keys have the form `{!Bindings.`***`key`***`}`.

The *`key`* supports Unicode characters in the predefined `\w` character class: `[\p{Alpha}\p{gc=Mn}\p{gc=Me}\p{gc=Mc}\p{Digit}\p{gc=Pc}]`.

This `Action URL` field has two binding variables:

```
https://www.example.com/{!Bindings.ApiVersion}/items/{!Bindings.ItemId}
```

This `HTTP Headers` field has two binding variables:

```
Authorization: OAuth {!Bindings.OAuthToken}
Content-Type: {!Bindings.ContentType}
```

Specify the keys and their values when you instantiate the action link group in Chatter REST API:

```
POST /connect/action-link-group-definitions

{
     "templateId":"07gD00000004C9r",
     "templateBindings" : [
```

```
        {
           "key":"ApiVersion",
           "value":"1.0"
        },
        {
           "key":"ItemId",
           "value":"8675309"
        },
        {
           "key":"OAuthToken",
           "value":"00DRR0000000N0g_!..."
        },
        {
           "key":"ContentType",
           "value":"application/json"
        }
     ]
}
```

Specify the binding variable keys and set their values in Apex:

```
Map<String, String> bindingMap = new Map<String, String>();
bindingMap.put('ApiVersion', '1.0');
bindingMap.put('ItemId', '8675309');
bindingMap.put('OAuthToken', '00DRR0000000N0g_!...');
bindingMap.put('ContentType', 'application/json');

List<ConnectApi.ActionLinkTemplateBindingInput> bindingInputs =
 new List<ConnectApi.ActionLinkTemplateBindingInput>();

for (String key : bindingMap.keySet()) {
    ConnectApi.ActionLinkTemplateBindingInput bindingInput = new
ConnectApi.ActionLinkTemplateBindingInput();
    bindingInput.key = key;
    bindingInput.value = bindingMap.get(key);
    bindingInputs.add(bindingInput);
}

// Define the action link group definition.
ConnectApi.ActionLinkGroupDefinitionInput actionLinkGroupDefinitionInput =
 new ConnectApi.ActionLinkGroupDefinitionInput();
actionLinkGroupDefinitionInput.templateId = '07gD00000004C9r';
actionLinkGroupDefinitionInput.templateBindings = bindingInputs;

// Instantiate the action link group definition.
ConnectApi.ActionLinkGroupDefinition actionLinkGroupDefinition =
ConnectApi.ActionLinks.createActionLinkGroupDefinition(Network.getNetworkId(),
actionLinkGroupDefinitionInput);
```

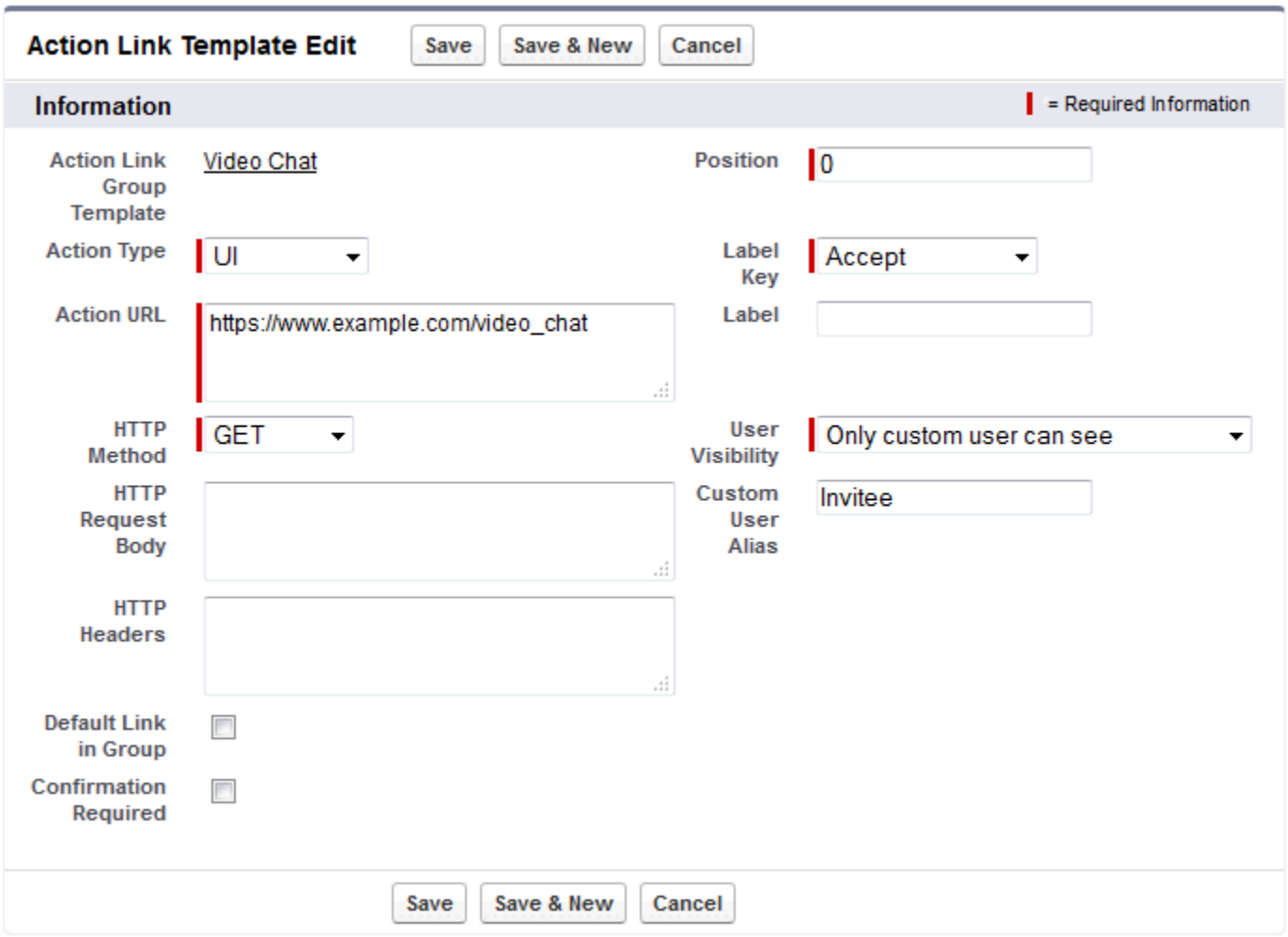
Tip: You can use the same binding variable multiple times in action link templates, and only provide the value once during instantiation. For example, you could use `{!Bindings.MyBinding}` twice in the `HTTP Request Body` field of one action link template, and again in the `HTTP Headers` of another action link template within the same action link group template, and when you instantiate an action link group from the template, you would need to provide only one value for that shared variable.

Set Who Can See the Action Link

Choose a value from the User Visibility drop-down list to determine who can see the action link after it's associated with a feed element.

Among the available options are Only Custom User Can See and Everyone Except Custom User Can See. Choose one of these values to allow only a specific user to see the action link or to prevent a specific user from seeing it. Then enter a value in the `Custom User Alias` field. This value is a binding variable key. In the code that instantiates the action link group, use the key and specify the value as you would for any binding variable.

This template uses the `Custom User Alias` value **Invitee**:



When you instantiate the action link group, set the value just like you would set a binding variable:

```
POST /connect/action-link-group-definitions

{
     "templateId":"07gD00000004C9r",
     "templateBindings" : [
        {
           "key":"Invitee",
```

```
            "value":"005D00000017u6x"
        }
    ]
}
```

If the template uses **Only creator's manager can see**, a user that doesn't have a manager receives an error when instantiating an action link group from the template. In addition, the manager is the manager at the time of instantiation. If the user's manager changes after instantiation, that change isn't reflected.

Use Context Variables

Use context variables to pass information about the user who executed the action link and the context in which it was invoked into the HTTP request made by invoking an action link. You can use context variables in the `actionUrl`, `headers`, and `requestBody` properties of the Action Link Definition Input request body or `ConnectApi.ActionLinkDefinitionInput` object. You can also use context variables in the `Action URL`, `HTTP Request Body`, and `HTTP Headers` fields of action link templates. You can edit these fields, including adding and removing context variables, after a template is published.

These are the available context variables:

Context Variable	Description
`{!actionLinkId}`	The ID of the action link the user executed.
`{!actionLinkGroupId}`	The ID of the action link group containing the action link the user executed.
`{!communityId}`	The ID of the community in which the user executed the action link. The value for your internal organization is the empty key `"000000000000000000"`.
`{!communityUrl}`	The URL of the community in which the user executed the action link. The value for your internal organization is empty string `""`.
`{!orgId}`	The ID of the organization in which the user executed the action link.
`{!userId}`	The ID of the user that executed the action link.

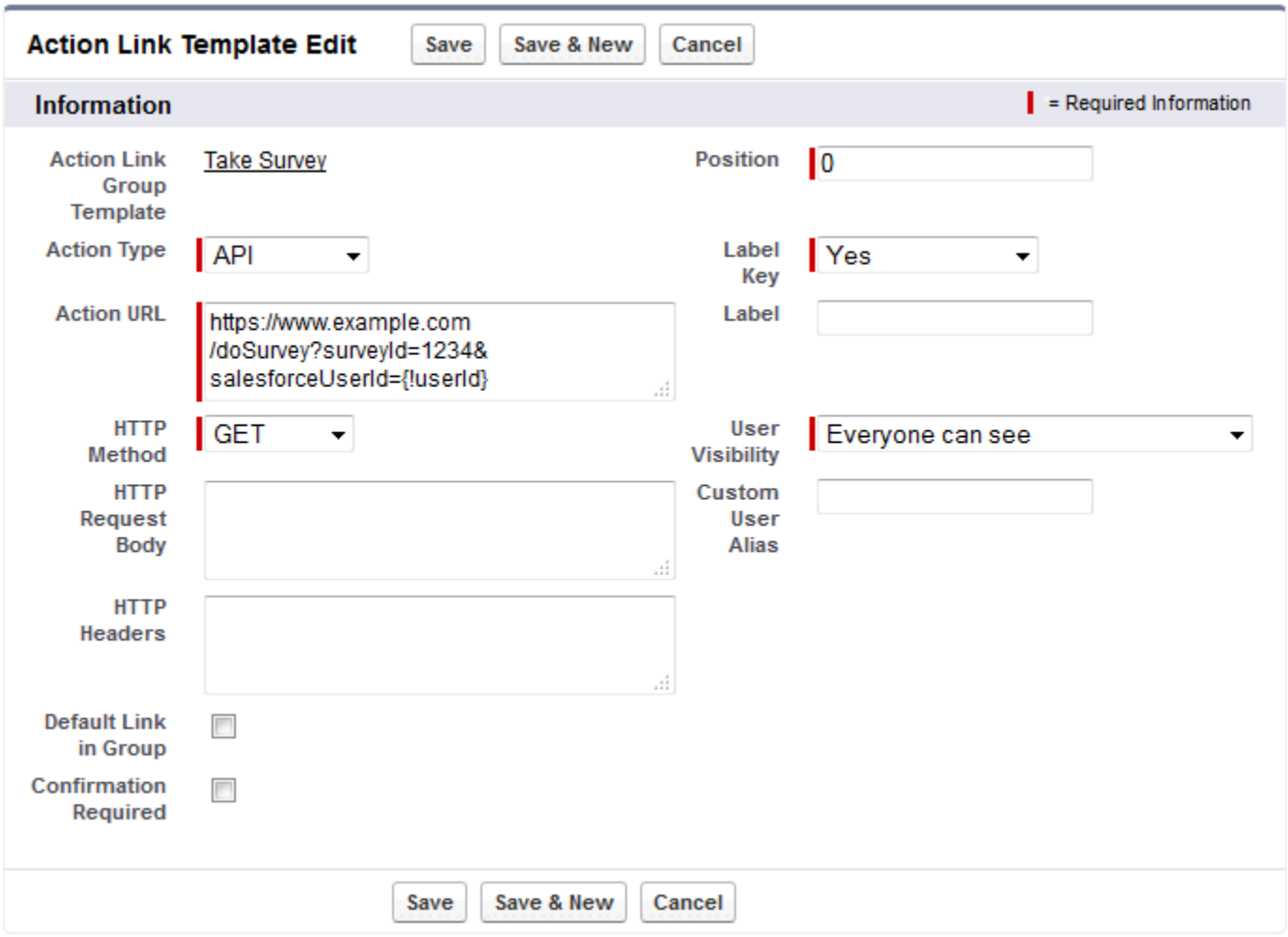
For example, suppose you work for a company called Survey Example and you create an app for the Salesforce AppExchange called **Survey Example for Salesforce**. Company A has **Survey Example for Salesforce** installed. Let's imagine that someone from company A goes to `surveyexample.com` and makes a survey. Your Survey Example code uses Chatter REST API to create a feed item in Company A's Salesforce organization with the body text **Take a survey**, and an action link with the label **OK**.

This `UI` action link takes the user from Salesforce to a web page on `surveyexample.com` to take a survey.

If you include a `{!userId}` context variable in either the `HTTP Request Body` or the `Action URL` for that action link, when a user clicks the action link in the feed, Salesforce sends the ID of the user who clicked in the HTTP request it makes to your server.

If you include an `{!actionLinkId}` context variable in the Survey Example server-side code that creates the action link, Salesforce sends an HTTP request with the ID of the action link and you can save that to your database.

This example includes the `{!userId}` context variable in the `Action URL` in the action link template:



Tip: Binding variables and context variables can be used in the same field. For example, this action URL contains a binding variable and a context variable:

```
https://www.example.com/{!Bindings.apiVersion}/doSurvey?salesforceUserId={!userId}
```

Create Action Link Templates

Create action link templates in Setup so that you can instantiate action link groups with common properties from Chatter REST API or Apex. You can package templates and distribute them to other Salesforce organizations.

Note: In addition to creating action link templates in Setup, you can also use Metadata API, SOAP API, and REST API to create action link templates.

The `Action URL`, `HTTP Request Body`, and `HTTP Headers` fields support *binding variables* and *context variables*. Define binding variables in a template and set their values when you instantiate the action link group. Use context variables in a template and when an action link executes, Salesforce fills in the value and returns it in the request. For information about how to use these variables in a template, see Designing Action Link Templates.

EDITIONS

Available in: All editions except **Personal** edition.

USER PERMISSIONS

To create action link group templates:
- "Customize Application"

To create action link templates:
- "Customize Application"

1. From Setup, click **Create** > **Action Link Templates**.
2. Click **New**.
3. Enter the `Name` of the template. This name is displayed in the list of action link group templates.

   This is the only action link group template value you can edit after the action link group template has been published.
4. Enter the `Developer Name`. Use the Developer Name to refer to this template from code. It defaults to a version of the `Developer Name` without spaces. Only letters, numbers, and underscores are allowed.
5. Select the `Category`, which indicates where to display the instantiated action link groups on feed elements. Primary displays action link groups in the body of feed elements. Overflow displays action link groups in the overflow menu of feed elements.

   If an action link group template is Primary, it can contain up to three action link templates. If an action link group template is Overflow, it can contain up to four action link templates.
6. Select the number of `Executions Allowed`, which indicates how many times the action link groups instantiated from this template can be executed. (Action links within a group are mutually exclusive.) If you choose Unlimited, the action links in the group cannot be of type `Api` or `ApiAsync`.
7. (Optional) Enter the `Hours until Expiration`, which is the number of hours from when the action link group is created until it's removed from associated feed elements and can no longer be executed. The maximum value is 8760.

   See Set the Action Link Group Expiration Time.
8. Click **Save**.
9. Click **New** to create an action link template.

   The action link template is automatically associated with an action link group template in a master-detail relationship.
10. Select the `Action Type`.

    Values are:

    - `Api`—The action link calls a synchronous API at the action URL. Salesforce sets the status to `SuccessfulStatus` or `FailedStatus` based on the HTTP status code returned by your server.
    - `ApiAsync`—The action link calls an asynchronous API at the action URL. The action remains in a `PendingStatus` state until a third party makes a request to `/connect/action-links/`***`actionLinkId`*** to set the status to `SuccessfulStatus` or `FailedStatus` when the asynchronous operation is complete.
    - `Download`—The action link downloads a file from the action URL.
    - `Ui`—The action link takes the user to a web page at the action URL.
11. Enter an `Action URL`, which is the URL for the action link.

For a `UI` action link, the URL is a Web page. For a `Download` action link, the URL is a link to a file to download. For an `Api` action link or an `ApiAsync` action link, the URL is a REST resource.

Links to resources hosted on Salesforce servers can be relative, starting with a `/`. All other links must be absolute and start with `https://`. This field can contain binding variables in the form `{!Bindings.key}`, for example, `https://www.example.com/{!Bindings.itemId}`. Set the binding variable's value when you instantiate the action link group from the template, as in this Chatter REST API example, which sets the value of `itemId` to `8675309`.

```
POST /connect/action-link-group-definitions

{
     "templateId" : "07gD00000004C9r",
     "templateBindings" : [
        {
           "key":"itemId",
           "value": "8675309"
        }
     ]
}
```

This field can also contain context variables. Use context variables to pass information about the user who executed the action link to your server-side code. For example, this action link passes the user ID of the user who clicked on the action link to take a survey to the server hosting the survey.

```
actionUrl=https://example.com/doSurvey?surveyId=1234&salesforceUserId={!userId}
```

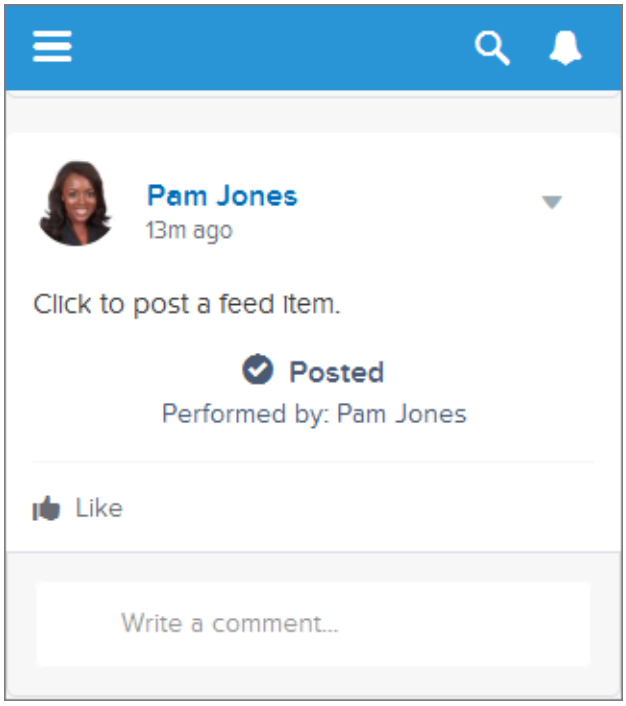
12. Enter the `HTTP Method` to use to make the HTTP request.
13. (Optional) If the `Action Type` is `Api` or `ApiAsync`, enter an `HTTP Request Body`.

    This field can contain binding variables and context variables.
14. (Optional) If the `Action Type` is `Api` or `ApiAsync`, enter `HTTP Headers`.

    This field can contain binding variables and context variables.

    If an action link instantiated from the template makes a request to a Salesforce resource, the template must have a Content-Type header.
15. (Optional) To make this action link the default link in the group (which has special formatting in the UI), select `Default Link in Group`. There can be only one default link in a group.
16. (Optional) To display a confirmation dialog to the user before the action link executes, select `Confirmation Required`.
17. Enter the relative `Position` of the action link within action link groups instantiated from this template. The first position is 0.
18. Enter the `Label Key`. This value is the key for a set of UI labels to display for these statuses: NewStatus, PendingStatus, SuccessfulStatus, FailedStatus.

    For example, the **Post** set contains these labels: **Post**, **Post Pending**, **Posted**, **Post Failed**. This image shows an action link with the **Post** label key when the value of status is SuccessfulStatus:



19. (Optional) If none of the `Label Key` values make sense for the action link, set `Label Key` to **None** and enter a value in the `Label` field.

    Action links have four statuses: NewStatus, PendingStatus, SuccessStatus, and FailedStatus. These strings are appended to the label for each status:

    - *label*
    - *label* Pending
    - *label* Success
    - *label* Failed

    For example, if the value of `label` is "See Example," the values of the four action link states are: See Example, See Example Pending, See Example Success, and See Example Failed.

    An action link can use either a `LabelKey` or `Label` to generate label names, it can't use both.

20. Select `User Visibility`, which indicates who can see the action link group.

    If you select **Only creator's manager can see**, the manager is the creator's manager when the action link group is instantiated. If the creator's manager changes after the action link group is instantiated, that change is not reflected.

21. (Optional) If you selected Only Custom User Can See or Everyone Except Custom User Can See, enter a `Custom User Alias`.

    Enter a string and set its value when you instantiate an action link group, just like you would set the value for a binding variable. However don't use the binding variable syntax in the template, just enter a value. For example, you could enter `ExpenseApprover`. This Chatter REST API example sets the value of `ExpenseApprover` to `005B0000000Ge16`:

```
POST /connect/action-link-group-definitions

{
     "templateId" : "07gD00000004C9r",
```

```
    "templateBindings" : [
       {
          "key":"ExpenseApprover",
          "value": "005B0000000Ge16"
       }
    ]
}
```

22. To create another action link template for this action link group template, click **Save & New**.
23. If you're done adding action link templates to this action link group template, click **Save**.
24. To publish the action link group template, click **Back to List** to return to the Action Link Group Template list view.

    Important: You must publish a template before you can instantiate an action link group from it in Apex or Chatter REST API.

25. Click **Edit** for the action link group template you want to publish.
26. Select `Published` and click **Save**.

Edit Action Link Templates

You can edit all fields on an unpublished action link group template and on its associated action link templates.

EDITIONS

Available in: All editions except **Personal** edition.

USER PERMISSIONS

To edit action link group templates:
- "Customize Application"

To edit action link templates:
- "Customize Application"

1. From Setup, click **Create** > **Action Link Templates**.
2. To edit an action link group template, click **Edit** next to its name.

   If the group template isn't published, edit any field. If it is published, edit the `Name` field only.
3. To edit an action link template:
   a. Click the name of its master action link group template.
   b. Click the Action Link Template ID to open the detail page for the action link template.
   c. Click **Edit**.

      If the associated action link group template isn't published, edit any field. If it's published, edit any of these fields:

      - `Action URL`
      - `HTTP Request Body`
      - `HTTP Headers`

      These fields support context variables and binding variables.

      You can add and delete context variables in any of these fields.

      You cannot add a new binding variable. You can:

      - Move a binding variable to another editable field in an action link template.
      - Use a binding variable more than once in an action link template.
      - Use a binding variable more than once in any action link templates associated with the same action link group template.
      - Remove binding variables.

Delete Action Link Group Templates

When you delete an action link group template, you delete its associated action link templates and all action link groups that have been instantiated from the templates. Deleted action link groups disappear from any feed elements they've been associated with.

1. From Setup, click **Create** > **Action Link Templates**.
2. To delete an action link group template, click **Del** next to its name.

   Important: When you delete an action link group template, you delete its associated action link templates and all action link groups that have been instantiated from the template. The action link group is deleted from any feed elements it has been associated with, which means that action links disappear from those posts in the feed.

3. To delete an action link template:
   a. Click the name of its master action link group template.
   b. Click the Action Link Template ID to open the detail page for the action link template.
   c. Click **Delete**.

   Important: You can't delete an action link template that's associated with a published action link group template.

EDITIONS

Available in: All editions except **Personal** edition.

USER PERMISSIONS

To delete action link group templates:
- "Customize Application"

To delete action link templates:
- "Customize Application"

Package Action Link Templates

Package action link templates to distribute them to other Salesforce organizations.

When you add an action link group template, any associated action link templates are also added to the package. You can add an action link group template to a managed or unmanaged package. As a packageable component, action link group templates can also take advantage of all the features of managed packages, such as listing on the AppExchange, push upgrades, post-install Apex scripts, license management, and enhanced subscriber support. To create a managed package, you must use a Developer Edition organization.

- See *Creating and Editing a Package* at https://help.salesforce.com.

EDITIONS

Available in: All editions except **Personal** edition.

USER PERMISSIONS

To package action link templates:
- "Create AppExchange Package"

Working with Communities

Communities are branded spaces for your employees, customers, and partners to connect. You can customize and create communities to meet your business needs, then transition seamlessly between them.

You can request most Chatter REST API resources within a specific community. For example, you can make requests to feeds, files, groups, topics, and users resources within a specific community. If a resource is available in communities, its documentation includes a resource that starts with `/connect/communities/communityId`.

Some Chatter REST API resources are available only within a community. These include resources for flagging and moderating content, using managed topics, and getting information about a user's reputation.

Resources that are available in communities are available whether you allow access without logging in or not. However, if your community allows access without logging in, guest users don't have access to any Chatter REST API resources.

The resources that are available only within a community are listed in this table.

Resource	HTTP Methods	Tasks
/connect/communities	GET	Returns a list of all communities available to the context user.
/connect/communities/***communityId***	GET	Returns information about the specified community.
/connect/communities/***communityId***/chatter/comments/***commentId***/moderation-flags	GET, POST, DELETE	If the context user is a community moderator, returns the number of times a comment has been flagged by community members. Also flags a comment for moderation and deletes all flags on a comment. If the context user isn't a community moderator, returns whether or not the user has flagged a comment. Also flags a comment for moderation and removes a flag the user previously added.
/connect/communities/***communityId***/chatter/feed-elements/***feedElementId***/capabilities/moderation	GET, POST, DELETE	Get information about a feed element's moderation capability, add a moderation flag to a feed element, or delete all moderation flags from a feed element.
/connect/communities/***communityId***/chatter/files/moderation	GET	If the context user is a community moderator, returns the files that have been flagged by community members.
/connect/communities/***communityId***/chatter/files/***fileId***/moderation-flags	GET, POST, DELETE	If the context user is a community moderator, returns the number of times a file has been flagged by community members. Also flags a file for moderation and deletes all flags on a file. If the context user isn't a community moderator, returns whether or not the user has flagged a file. Also flags a file for moderation and removes a flag the user previously added.
/connect/communities/***communityId***/chatter/feeds/moderation/feed-elements	GET	If the context user is a community moderator with the "Moderate Communities Feeds" permission, returns a list of all feed elements and comments in the community that have been flagged by community members. By default, returns 25 feed elements per page of results.
/connect/communities/***communityId***/chatter/feeds/home/feed-elements	GET	Returns a list of all feed elements associated with any managed topic in the community.
/connect/communities/***communityId***/managed-topics	GET, POST, PATCH	Get all the managed topics for a community. Create a managed topic. Reorder managed topics.
/connect/communities/***communityId***/managed-topics/***managedTopicId***	GET, DELETE	Get information about or delete a managed topic in a community.

Resource	HTTP Methods	Tasks
`/connect/communities/communityId/chatter/users/userId/reputation`	GET	Returns the specified community user's reputation.

SEE ALSO:

Building the Chatter REST API Resource URL

Make a Request in a Community

Working with Feeds and Feed Elements

In API versions 30.0 and earlier, a Chatter feed was a container of feed items. In API version 31.0, the definition of a feed expanded to include new objects that didn't entirely fit the feed item model. The feed became a container of *feed elements*. In API version 32.0, feed elements can be either feed items or generic feed elements. Because feeds and feed elements are the core of Chatter, understanding them is crucial to developing applications with Chatter REST API.

Note: Salesforce Help refers to feed items as *posts*.

Capabilities

As part of the effort to diversify the feed, pieces of functionality found in feed elements have been broken out into *capabilities*. Capabilities provide a consistent way to interact with objects in the feed. Don't inspect the feed element type to determine which functionality is available for a feed element. Inspect the capability object, which tells you explicitly what's available. Check for the presence of a capability to determine what a client can do to a feed element.

All feed elements have a `capabilities` property, which holds a Feed Element Capabilities response body. The Feed Element Capabilities response body can have one or more properties, each holding a capability of the feed element.

A capability includes both an indication that a feature is possible and data associated with that feature. If a capability exists on a feed element, that capability is available, even if there isn't any data associated with the capability yet. For example, if the `chatterLikes` capability property exists on a feed element (with or without any likes included in the list of likes found in the `chatterLikes.page.items` property), the context user can like that feed element. If the capability property doesn't exist on a feed element, it isn't possible to like that feed element.

How the Salesforce UI Displays Feed Items

As we learned in Capabilities, a client should use the `capabilities` property to determine what it can do with a feed element and how it should render a feed element. For all feed elements other than feed item, the client doesn't need to know what type feed element it is, the client can simply look at the capabilities. Feed items do have capabilities, but they also have a few properties, such as `actor`, that aren't exposed as capabilities. For this reason, feed items must be handled a bit differently than other feed elements.

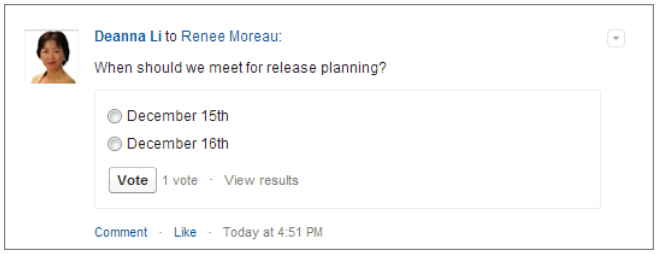
To give customers a consistent view of feed items and to give developers an easy way to create UI, the Salesforce UI uses one layout to display every feed item. The layout always contains the same pieces and the pieces are always in the same position; only the content of the layout pieces changes.

The feed item layout elements are:

1. Actor (`actor` property of Feed Item response body)—A photo or icon of the creator of the feed item.
2. Header (`header` property of Feed Item response body)—Provides context. The same feed item can have a different header depending on who posted it and where. For example, Gordon posted this feed item to his profile. If he shared it to a group, the header of the feed item in the group feed would be "Gordon Johnson (originally posted by Gordon Johnson)". The "originally posted" text would link to the feed item on Gordon's profile.
3. Body (`body` property of Feed Item response body)—All feed items have a body, but the body can be `null`, which is the case when the user doesn't provide text for the feed item. Because the body can be `null`, you can't use it as the default case for rendering text. Instead, use the `text` property of the feed item's header, which always contains a value.
4. Auxiliary Body (`capabilities` property of Feed Item response body)—The visualization of the capabilities of a feed item. See Capabilities.

   Important: The `attachment` property is not supported in API versions 32.0 and later. Instead, use the `capabilities` property, which holds a Feed Element Capabilities response body, to discover what to render for a feed element.
5. Created By Timestamp (`relativeCreatedDate` property of Feed Item response body)—The date and time when the feed item was posted. If the feed item is less than two days old, the date and time are formatted as a relative, localized string, for example, "17m ago" or "Yesterday". Otherwise, the date and time are formatted as an absolute, localized string.

Here's another example of a feed item in the Salesforce UI. This feed item's auxiliary body contains a poll capability:

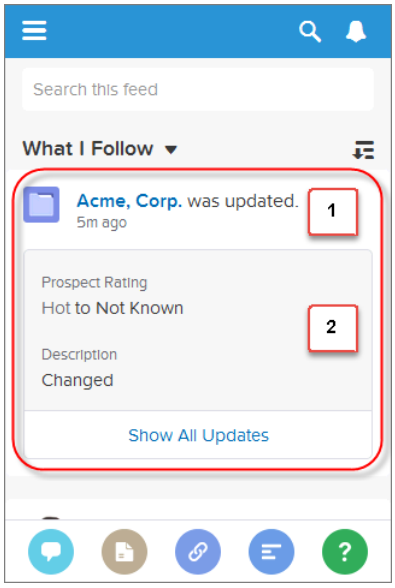


How the Salesforce UI Displays Bundles

Note: Bundled posts contain feed-tracked changes. In Salesforce1 downloadable apps, bundled posts are in record feeds only.

As we learned in the Capabilities section, a client should use the `ConnectApi.FeedElement.capabilities` property to determine what it can do with a feed element and how to render a feed element. This section uses bundles as an example of how to render a feed element, but these properties are available for every feed element.

To give customers a clean, organized feed, feed-tracked change feed items can be aggregated into a bundle. To see individual feed items, click the bundle.



A bundle is a generic feed element with a bundle capability. These are the bundle layout elements:

1. Header (`header` property of the Generic Feed Element response body)—For feed-tracked change bundles, this text is "This record was updated."

   The time below the header is the `relativeCreatedDate` property of the Generic Feed Element response body.

2. Auxiliary Body (`capabilities.bundle.changes` property of the Generic Feed Element response body)—The bundle displays the `fieldName` and the `oldValue` and `newValue` properties for the first two feed-tracked changes in the bundle. If there are more than two feed-tracked changes, the bundle displays a "Show All Updates" link.

Feed Element Visibility

The feed elements a user sees depend on how the administrator has configured feed tracking, sharing rules, and field-level security. For example, if a user doesn't have access to a record, they don't see updates for that record. If a user can see the parent of the feed element, the user can see the feed element. Typically, a user sees feed updates for:

- Feed elements that @mention the user (if the user can access the feed element's parent)
- Feed elements that @mention groups the user is a member of
- Record field changes on records whose parent is a record the user can see, including User, Group, and File records
- Feed elements posted to the user
- Feed elements posted to groups the user owns or is a member of
- Feed elements for standard and custom records, for example, tasks, events, leads, accounts, files

Feed Types

There are many types of feeds. Each feed type is an algorithm that defines a collection of feed elements.

Important: The algorithms, and therefore the collection of feed elements, can change between releases.

The feed types are exposed in the resources:

```
/chatter/feeds/feedType
/chatter/feeds/feedType/subjectId
```

For example, the resources for a news feed and a topics feed are:

```
/chatter/feeds/news/subjectId
/chatter/feeds/topics/subjectId
```

The feed types and their descriptions:

- `Bookmarks`—Contains all feed items saved as bookmarks by the context user.
- `Company`—Contains all feed items except feed items of type `TrackedChange`. To see the feed item, the user must have sharing access to its parent.
- `Files`—Contains all feed items that contain files posted by people or groups that the context user follows.
- `Filter`—Contains the news feed filtered to contain feed items whose parent is a specified object type.
- `Groups`—Contains all feed items from all groups the context user either owns or is a member of.
- `Home`—Contains all feed items associated with any managed topic in a community.
- `Moderation`—Contains all feed items that have been flagged for moderation. The Communities Moderation feed is available only to users with “Moderate Community Feeds” permissions.
- `News`—Contains all updates for people the context user follows, groups the user is a member of, and files and records the user is following. Also contains all updates for records whose parent is the context user and every feed item and comment that mentions the context user or that mentions a group the context user is a member of.
- `People`—Contains all feed items posted by all people the context user follows.
- `Record`—Contains all feed items whose parent is a specified record, which could be a group, user, object, file, or any other standard or custom object. When the record is a group, the feed also contains feed items that mention the group. When the record is a user, the feed contains only feed items on that user.
- `To`—Contains all feed items with mentions of the context user, feed items the context user commented on, and feed items created by the context user that are commented on.
- `Topics`—Contains all feed items that include the specified topic.
- `UserProfile`—Contains feed items created when a user changes records that can be tracked in a feed, feed items whose parent is the user, and feed items that @mention the user. This feed is different than the news feed, which returns more feed items, including group updates.
- `Favorites`—Contains favorites saved by the context user. Favorites are feed searches, list views, and topics.
- `Filter`—Contains the news feed filtered to contain items whose parent is a specified entity type.

Post a Feed Item Using `/chatter/feed-elements`

Tip: The `/chatter/feed-elements` resource is the simplest, most efficient way to post feed items because it doesn't require you to pass a feed type. As of API version 31.0, feed items are the only feed element type you can post. To future-proof your application, use this resource because there may be other feed element types in the future.

Use this resource to post feed items:

```
POST /chatter/feed-elements
```

When you post a feed item, you create a child of a standard or custom object. Specify the parent object in the `subjectId` request parameter or in the `subjectId` property of the Feed Item Input request body.

The `parent` property of the posted feed item contains information about the parent object.

Post to your user profile

This example uses request parameters to post a text post to the context user's user profile:

```
POST /chatter/feed-elements?feedElementType=FeedItem&subjectId=me&text=Test
```

The `subjectId`, specifies the alias `me`. You can also specify the user ID of the context user.

The `parent` property of the newly posted feed item contains the User Summary response body of the context user.

Post to another user

This example uses a request body to post to another user. The `subjectId` property specifies that user's ID.

```
POST /chatter/feed-elements

{
   "feedElementType":"FeedItem",
   "subjectId":"005RR000000DmOg",
   "body" : {
      "messageSegments" : [
      {
         "type" : "Text",
         "text" : "When should we meet for release planning?"
      }
      ]
   }
}
```

The `parent` property of the newly posted feed item contains the User Summary response body of the target user.

Post to a group

This example uses a request body to post to a group. The `subjectId` property specifies the group ID.

```
POST /chatter/feed-elements

{
   "feedElementType":"FeedItem",
   "subjectId":"0F9RR0000004CPw",
   "body" : {
      "messageSegments" : [
      {
         "type" : "Text",
         "text" : "When should we meet for release planning?"
      }
```

```
        ]
    }
}
```

The `parent` property of the newly posted feed item contains the Group response body of the specified group.

Post to a record (such as a file or an account)

This example uses a request body to mention a user on a record feed. The `subjectId` property specifies the record ID.

```
POST /chatter/feed-elements

{
   "feedElementType" : "FeedItem",
   "subjectId" : "001RR000002G4Y0",
   "body" : {
      "messageSegments" : [
      {
         "type" : "Mention",
         "id" : "005RR000000DmOg"
      },
      {
         "type" : "Text",
         "text" : " Have you checked on this account recently?"
      }
      ]
   }
}
```

The `parent` property of the new feed item depends on the record type specified. If the record type is File, the parent is the File Summary response body. If the record type is Group, the parent is a Group response body. If the record type is User, the parent is a User Summary response body. For all other record types, the parent is a Record Summary response body.

Get Feed Elements from a Feed

Tip: To return a feed that includes feed elements, call these methods. As of API version 31.0, the only feed element types are feed item and bundle, but that could change in the future.

Getting feed elements from a feed is similar, but not identical, for each feed type.

To get the feed elements from the company feed, the home feed, or the moderation feed, don't specify a subject ID:

Feed Type	Resource
Company	`/chatter/feeds/company/feed-elements`
Home	`/connect/communities/communityId/chatter/feeds/home/feed-elements`
Moderation	`/connect/communities/communityId` `/chatter/feeds/moderation/feed-elements`

To get the feed elements from the favorites and filter feeds, specify a *favoriteId* or a *keyPrefix*. The *keyPrefix* indicates the object type and is the first three characters of the object ID. For these feeds, the *subjectId* must be the ID of the context user or the alias `me`.

Feed Type	Resource
Favorites	`/chatter/feeds/favorites/subjectId/favoriteId/feed-elements`
Filter	`/chatter/feeds/filter/subjectId/keyPrefix/feed-elements`

To get the feed elements from a record feed, specify a record ID.

Feed Type	Resource
Record	`/chatter/feeds/record/recordId/feed-elements`

Tip: The `recordId` can be a record of any type that supports feeds, including group. The feed on the group page in the Salesforce UI is a record feed.

To get the feed elements from all other feed types, specify a subject ID. Replace the `feedType` to specify a different feed. For all the feed types in this table except the user profile feed and the topics feed, the `subjectId` must be the ID of the context user or the alias `me`.

Feed Type	Resource
Bookmarks, Files, Groups, News, People, To, Topics, User Profile	`/chatter/feeds/feedType/subjectId/feed-elements` For example: `/chatter/feeds/news/me/feed-elements`

SEE ALSO:

Feeds Resource

Working with Salesforce Files

Use Chatter REST API to work with Salesforce Files, which includes Salesforce Files Sync.

Use Salesforce Files to upload, store, find, follow, share, and collaborate on files in the cloud.

Use Salesforce Files Sync to securely store, sync, and share files between Chatter, the desktop, and mobile devices. Access files and folders from anywhere, and keep them instantly updated across your organization.

Resource	HTTP Methods	Tasks
`/chatter/feed-elements`	GET, POST	Post feed elements and search all feed elements in an organization.
`/chatter/feeds/files/me/feed-elements`	GET	Returns all feed elements posted with files by people or groups that the context user follows.
`/chatter/feeds/news/me/feed-elements`	GET	All feed elements from all groups the context user either owns or is a member of, as well as all files, records, and users the context user follows. Use this resource to get

Resource	HTTP Methods	Tasks
		information about feed elements and to post feed elements.
/chatter/feeds/records/***recordId***/feed-elements	GET	Returns the feed elements for all the records the context user is following, or all the feed elements of the specified *recordId*. Use this resource to search a feed or to get the feed elements for a specific feed. To use this resource to get the feed elements for a group, specify the group ID as the *recordId*.
/chatter/feeds/user-profile/***userId***/feed-elements	GET	Returns the feed elements for all actions on all records that can be tracked in a feed.
/chatter/files/***fileId***	GET, POST, PATCH, DELETE	Get information about a specified file, including references to external files. Upload a new version of an existing file, including references to external files. Rename a file, move a file to a different folder, and delete a file.
/chatter/files/***fileId***/content	GET	Returns the content of the file, including references to external files. The content is streamed as the body of the response.
/chatter/files/***fileId***/file-shares	GET, POST	Returns information about the objects with which the specified file has been shared. Objects can be users, groups, or records. Can also be used to share the specified file with users.
/chatter/files/***fileId***/file-shares/link	GET, PUT, DELETE	A description of a file shared as a link. Create, access, and delete a file's share link.
/chatter/files/***fileId***/rendition	GET	A rendition of the file. A rendition is a binary preview of the file. The format of the binary depends on the format specified in the request.
/chatter/files/batch/***fileIds***	GET, DELETE	Information about a list of files, specified by file IDs. Get information about a list of files and delete a list of files.
/chatter/folders/***folderId***	GET, PATCH, DELETE	Get information about the specified folder. Move, rename, and delete a folder. Sync the root folder.
/chatter/folders/***folderId***/items	GET, POST	Get information about the contents of a folder. Add a file to a folder and create a folder. Sync the root folder.
/chatter/groups/***groupId***/files	GET	Returns information about the files posted to the specified group.
/connect/topics/***topicId***/files	GET	Returns information about the five files most recently posted to the specified topic.
/chatter/users/***userId***/files	GET, POST	Post a file, including references to external files, to Salesforce Files Sync or get information about files a user owns. These files don't include files shared with a user, files a user follows, or general organization files.

Resource	HTTP Methods	Tasks
`/chatter/users/me/files/filter/groups`	GET	Returns information about files posted to groups that the specified user is a member of.
`/chatter/users/me/files/filter/shared-with-me`	GET	Returns information about files that have been shared with the specified user.

Note: You can replace the alias `me` with the user ID of the context user.

CHAPTER 4 Chatter REST API Examples

Use Chatter REST API examples to perform tasks.

While using the Chatter REST API, keep this in mind:

- Request parameters may be included as part of the Chatter REST API resource URL, for example, `/chatter/users?q=`***`searchtext`***. A request body is a rich input which may be included as part of the request. When accessing a resource, you can use either a request body or request parameters. You cannot use both.
- With a request body, use `Content-Type: application/json` or `Content-Type: application/xml`.
- With request parameters, use `Content-Type: application/x-www-form-urlencoded`.

For more information on request bodies, see Using POST, PATCH, or PUT for Input.

For examples of how to upload a binary file, see Uploading Binary Files.

IN THIS SECTION:

Filter a Response Body

Make a Request in a Community

Post a Feed Item

Post a Feed Element and Include a File from Salesforce

Post a Comment

Post a Comment and Include a File from Salesforce

Post a Binary File

Post an @Mention

Post Multiple @Mentions

Get a List of @Mention Proposals

Define Action Links and Post with a Feed Element

Define Action Links in a Template and Post with a Feed Element

Update the Status of an Action Link or Invoke an Action Link

Edit a Feed Post

Edit a Comment

Edit a Question Title and Post

Like a Feed Item

Like a Feed Item and Return the Total Number of Likes

Share a Feed Item

Get Feed Elements Updated Since Last Request

Get My News Feed

Get a Group Feed

Get a Feed with Internal Feed Items Only

Search a Feed
Get a List of Records a User Follows
Get Chatter Activity Statistics for a User
Get Recommendations of Users to Follow
Create a Group
Update and Crop a Group Photo
Join a Group
Request to Join a Private Group
Approve or Reject a Request to Join a Private Group
Post to a Group
Follow a Record
Stop Following a Record
Get a Record Feed
Update a User Profile
Update and Crop a User Photo
Send a Private Message
Get User Information for Multiple Users

Filter a Response Body

Resource

All

HTTP method

GET

Request parameters

`exclude`, `include`, and `filterGroup`

Use the `exclude` and `include` query parameters to filter a response body when making a request. The query value is a bar (`|`) delimited list of properties. URL encode the bar as `%7C`. Use a forward slash (`/`) before each property name.

Use the `filterGroup` query parameter to specify whether to return a `Big`, `Medium`, or `Small` response body. Every property in every response body is assigned one of these group sizes.

Request parameter example

This News Feed URL resource:

```
/chatter/users/me?include=/aboutMe%7C/address
```

For a detailed list of rules and examples, see Specifying Response Sizes.

Returns

A filtered response body

The request parameter example above returns this for the Feed response body:

```
{
   "aboutMe": "Salesforce Administrator and Citizen Developer",
   "address": {
      "city": "San Francisco",
      "country": "US",
      "formattedAddress": "1 Market St\nSan Francisco, CA 94105\nUS",
      "state": "CA",
      "street": "1 Market St",
      "zip": "94105"
   }
}
```

Make a Request in a Community

Resource

A resource supports Communities unless the documentation states that it doesn't.

HTTP method

Various

Example

To make a request to a specific community, prefix the resource with `/connect/communities/`***`communityId`***. For example, this resource gets the feed items from a news feed in a specific community:

`/connect/communities/`***`communityId`***`/chatter/feeds/news/me/feed-elements`

For more information, see Building the Chatter REST API Resource URL and Connecting to Salesforce Communities.

Returns

Various

Post a Feed Item

Resource

`/chatter/feed-elements`

HTTP method

POST

Request body

Feed Item Input

Request parameters

`feedElementType`, `subjectId`, `text`

Request body example

In the `subjectId`, specify the user, group, or record that will parent the feed item.

```
{
   "body" : {
      "messageSegments" : [
         {
            "type" : "Text",
```

```
            "text" : "When should we meet for release planning? "
         },
         {
            "type" : "Mention",
            "id" : "005D00000016Qxp"
         }]
      },
   "feedElementType" : "FeedItem",
   "subjectId" : "0F9B000000000W2"
}
```

Request parameter example

```
/services/data/v34.0/chatter/feed-elements?feedElementType=FeedItem&subjectId=0F9B000000000W2&text=New+post
```

Returns

Feed Item

Post a Feed Element and Include a File from Salesforce

This example posts a feed element with a file that has been uploaded to Salesforce. To upload a new file, see Uploading Binary Files.

Resource

`/chatter/feed-elements`

HTTP method

POST

Request body

Feed Item Input

Request body example

```
{
   "body":{
      "messageSegments":[
         {
            "type":"Text",
            "text":"Please take a look at this file."
         }
      ]
   },
   "capabilities":{
      "content":{
         "contentDocumentId":"069D00000001pyS"
      }
   },
   "subjectId":"me",
   "feedElementType":"FeedItem"
}
```

Returns

Feed Item

Post a Comment

Resource

`/chatter/feed-elements/feedElementId/capabilities/comments/items`

HTTP method

POST

Request body

Comment Input

Request parameter

`text`

Request body example

```
{
   "body":{
      "messageSegments":[
         {
            "type":"Text",
            "text":"New comment"
         }
      ]
   }
}
```

Request parameter example

```
/services/data/v34.0/chatter/feed-elements/feedElementId/capabilities/comments/items?text=New+comment
```

Returns

Comment

Post a Comment and Include a File from Salesforce

This example posts a comment with a file that has already been uploaded to Salesforce. To upload a new file, see Uploading Binary Files.

Resource

`/chatter/feed-elements/feedElementId/capabilities/comments/items`

HTTP method

POST

Request body

Comment Input

Request parameter

`text`

Request body example

```
{
   "body":{
      "messageSegments":[
         {
            "type":"Text",
```

```
            "text":"Please take a look at this file."
         }
      ]
   },
   "capabilities":{
      "content":{
         "contentDocumentId":"069D00000001pyS"
      }
   }
}
```

Returns

Comment

Post a Binary File

For detailed information about posting binary files, see Using Chatter REST API Inputs, which includes examples for:

- Post a feed element with a binary file
- Post a batch of feed elements with binary files
- Post a comment with a binary file
- Post a batch of comments with binary files
- Post and crop a user photo
- Post a binary file to the Files list

Post an @Mention

Resource

To post a feed item with a mention, use `/chatter/feed-elements`

To post a comment with a mention, use `/chatter/feed-elements/feedElementId/capabilities/comments/items`.

HTTP method

POST

Request body

For a feed item, use Feed Item Input

For a comment, use Comment Input

For either input, use the Message Segment Input: Mention

Request body example

This example is for a feed item:

```
{
   "body" : {
      "messageSegments" : [
         {
            "type" : "Text",
            "text" : "When should we meet for release planning? "
         },
```

```
            {
               "type" : "Mention",
               "id" : "005T0000000mzCy"
            }
         ]
   },
   "feedElementType" : "FeedItem",
   "subjectId" : "005D00000016Qxp"
}
```

This example is for a comment:

```
{ "body" :
   {
      "messageSegments" : [
         {
           "type": "mention",
           "id" : "005D0000001GpHp"
         },
         {
           "type": "text",
           "text" : " Did you see the new plans?"
         }
      ]
   }
}
```

Note: Include a space in the text message segment because spaces aren't added automatically before or after a mention.

Returns

Feed Item

Comment

Post Multiple @Mentions

Resource

To post a feed item with up to 25 mentions, use `/chatter/feed-elements`

To post a comment with up to 25 mentions, use `/chatter/feed-elements/`***`feedElementId`***`/capabilities/comments/items`.

HTTP method

POST

Request body

For a feed item, use Feed Item Input

For a comment, use Comment Input

For either input, use the Message Segment Input: Mention

Request body example

This example is for a feed item:

```
{
   "body" : {
```

```
      "messageSegments" : [
         {
            "type" : "Text",
            "text" : "When should we meet for release planning? "
         },
         {
            "type" : "Mention",
            "id" : "005T0000000mzCy"
         },
         {
            "type" : "Text",
            "text" : " "
         },
         {
            "type" : "Mention",
            "id" : "005B0000000Ek1S"
         }
      ]
   },
   "feedElementType" : "FeedItem",
   "subjectId" : "005D00000016Qxp"
}
```

This example is for a comment:

```
{ "body" :
   {
      "messageSegments" : [
          {
            "type": "mention",
            "id":"005D0000001LL8YIAW"
          },
          {
            "type" : "Text",
            "text" : " "
         },
          {
            "type": "mention",
            "id":"005D0000001LL8sIAG"
          },
          {
             "type": "text",
             "text":" Did you see the new plans?"
          }
      ]
   }
}
```

Note: Include spaces where you need them because spaces aren't added automatically before or after a mention.

Returns

Feed Item

Comment

Get a List of @Mention Proposals

Resource

Use the Mentions Completions resource to get a list of mention proposals: `/chatter/mentions/completions`. Use the Mentions Validations resource to determine which of those proposals are valid for the context user: `/chatter/mentions/validations`.

HTTP method

GET

Request parameters

`contextId`, `page`, `pageSize`, `q`, `type`

Returns

Mention Completion Page

Mention Validations

Define Action Links and Post with a Feed Element

Step 1: Define the Action Link Group

Just like radio buttons, action links must be nested in a group. Action links within a group share the properties of the group and are mutually exclusive (you can click on only one action link within a group). Even if you define only one action link, it must be part of an action link group.

Think of an action link as a button on a feed item. Like a button, an action link includes a label (`labelKey`). An action link definition also includes other properties like a URL (`actionUrl`), an HTTP method (`method`), and an optional request body (`requestBody`) and HTTP headers (`headers`).

When a user clicks this action link, an HTTP POST request is made to a Chatter REST API resource, which posts a feed item to Chatter. The `requestBody` property holds the request body for the `actionUrl` resource, including the text of the new feed item. In this example, the new feed item includes only text, but it could include other capabilities such as a file attachment, a poll, or even action links.

Resource

`/connect/action-link-group-definitions`

HTTP method

POST

Request body

Action Link Group Definition Input

This request body includes an `actionLinks` property that holds a collection of Action Link Definition Input request bodies. This example has one action link.

Request body example

```
POST /services/data/v34.0/connect/action-link-group-definitions
{
   "executionsAllowed":"OncePerUser",
   "expirationDate":"2015-05-20T19:32:10+0000",
   "category":"Primary",
   "actionLinks":[
```

```
      {
         "actionType":"Api",
         "actionUrl":"/services/data/v34.0/chatter/feed-elements",
         "labelKey":"Post",
         "method":"HttpPost",
         "requiresConfirmation":"false",
        "requestBody":"{\"subjectId\": \"me\",\"feedElementType\": \"FeedItem\",\"body\":

{\"messageSegments\": [{\"type\": \"Text\",\"text\": \"This post was created via an API
 action link.\"}]}}",
         "headers":[
            {
               "name":"Content-Type",
               "value":"application/json"
            },
            {
               "name":"Authorization",
               "value":"OAuth 00DD00000007WNP!ARsAQHhIdgwbihG7e8ga"
            }
         ]
      }
   ]
}
```

Request example using cURL

To use cURL to make the request, enter the following and substitute your Developer Edition instance name, and OAuth information.

```
curl -H "X-PrettyPrint: 1" -H "Content-Type: application/json"
-d '{ "executionsAllowed":"OncePerUser", "expirationDate":"2015-09-20T19:32:10+0000",
"category":"Primary",
"actionLinks":[ { "actionType":"Api",
"actionUrl":"/services/data/v34.0/chatter/feed-elements",
"groupDefault":"true", "labelKey":"Post", "method":"HttpPost",
"requiresConfirmation":"true",
"requestBody":"{\"subjectId\": \"me\",\"feedElementType\": \"FeedItem\",\"body\":
{\"messageSegments\": [{\"type\": \"Text\",\"text\": \"This is a test post created via
 an API action link.\"}]}}",
"headers":[ { "name":"Content-Type", "value":"application/json" }, {
"name":"Authorization", "value":"OAuth 00DD00000007WNP!ARsAQHhIdgwbihG7e8ga" }] }] }'
-X POST "https://instance_name/services/data/v34.0/connect/action-link-group-definitions"

-H 'Authorization: OAuth 00DRR0000000N0g!RWaPj94O6yOD.lfjB9LqMk' --insecure
```

Response body

Action Link Group Definition

Step 2: Associate the Action Link Group with a Feed Item and Post It

You can associate up to 10 action link groups with a feed element.

Resource

`/chatter/feed-elements`

HTTP method

POST

Request body

Feed Item Input

Request body example

Grab the action link group ID from the `id` property of the Action Link Group Definition response from step 1.

```
POST /services/data/v34.0/chatter/feed-elements

{
  "body": {
    "messageSegments": [
      {
        "type": "Text",
        "text": "Click to post a feed item."
       }
    ]
    },
  "subjectId": "me",
  "feedElementType": "feedItem",
  "capabilities": {
    "associatedActions": {
      "actionLinkGroupIds": ["0AgRR0000004CTr0AM"]
    }
  }
}
```

Request example using cURL

To use cURL to make the request, enter the following and substitute the action link group ID returned in step 1, your Developer Edition instance name, and your OAuth information.

```
curl -H "X-PrettyPrint: 1" -H "Content-Type: application/json" -d '{ "body": {
"messageSegments":
[ { "type": "Text", "text": "Click to post a feed item." } ] }, "subjectId": "me",
"feedElementType": "feedItem", "capabilities":
{ "associatedActions": { "actionLinkGroupIds": ["0AgRR0000004CTr0AM"] } } }'
-X POST "https://instance_name/services/data/v34.0/chatter/feed-elements"
-H 'Authorization: OAuth OOARoAQETAKEIddnMWXh462dprYAgRWaPj94O6yOD' --insecure
```

Response body

Feed Item

SEE ALSO:

Working with Action Links

Action Link Templates

Action Link Group Definitions

Feed Elements, Post and Search

Define Action Links in a Template and Post with a Feed Element

Define Action Links in a Template and Post with a Feed Element

Define Action Links in a Template and Post with a Feed Element

Step 1: Create the Action Link Templates

Just like radio buttons, action links must be nested in a group. Action links within a group share the properties of the group and are mutually exclusive (you can click on only one action link within a group). Even if you define only one action link, it must be part of an action link group. The same is true for action link templates and action link group templates.

This example uses one action link in an action link group. When a user clicks the action link, an HTTP POST request is made to a Chatter REST API resource, which posts a feed item to Chatter. The `HTTP Request Body` field holds the request body for the `Action URL` resource, including the text of the new feed item. In this example, the new feed item includes only text, but it could include other capabilities such as a file attachment, a poll, or even action links.

1. From Setup, click **Create** > **Action Link Templates**.
2. Use these values in a new Action Link Group Template:

Field	Value
Name	Doc Example
Developer Name	Doc_Example
Category	Primary action
Executions Allowed	Once per User

3. Use these values in a new Action Link Template:

Field	Value
Action Link Group Template	Doc Example
Action Type	Api
Action URL	/services/data/{!Bindings.ApiVersion}/chatter/feed-elements
User Visibility	Everyone can see
HTTP Request Body	{"subjectId": "{!Bindings.SubjectId}","feedElementType": "FeedItem","body": {"messageSegments": [{"type": "Text","text": "{!Bindings.Text}"}]}}
HTTP Headers	Content-Type: application/json
Position	0
Label Key	Post
HTTP Method	POST

4. Go back to the Action Link Group Template and select `Published`. Click **Save**.

Make a note of the Action Link Group Template ID and skip to Step 3.

Step 2: Get the Action Link Group Template ID

If you have access to the action link group template in Setup, you can get the Action Link Group Template ID and skip to Step 3.

Resource

```
/services/data/v33.0/query
```

HTTP method

GET

Request parameters

q—A SOQL query string.

Request parameters example

```
GET
/services/data/v34.0/query?q=SELECT+id+FROM+ActionLinkGroupTemplate+WHERE+DeveloperName='Doc_Example'
```

cURL example

```
curl
https://na1.salesforce.com/services/data/v34.0/query?q=SELECT+id+FROM+ActionLinkGroupTemplate+
WHERE+DeveloperName='Doc_Example' -H "Authorization: Bearer token"
```

Response body

An array of query result records.

```
{
   "totalSize":1,
   "done":true,
   "records":[
      {
         "attributes":{
            "type":"ActionLinkGroupTemplate",

"url":"/services/data/v34.0/sobjects/ActionLinkGroupTemplate/07gD00000004CEhIAM"
         },
         "Id":"07gD00000004CEhIAM"
      }
   ]
}
```

Step 3: Instantiate the Action Link Group

Resource

/connect/action-link-group-definitions

HTTP method

POST

Request body

Action Link Group Definition Input

Request body example

```
POST /services/data/v34.0/connect/action-link-group-definitions
{
     "templateId":"07gD00000004CEhIAM",
     "templateBindings" : [
        {
           "key":"ApiVersion",
           "value":"v33.0"
        },
        {
           "key":"Text",
           "value":"This post was created by an API Action Link!"
        },
        {
           "key":"SubjectId",
           "value":"me"
        }
     ]
}
```

Request example using cURL

To use cURL to make the request, enter the following and substitute your Developer Edition instance name, and OAuth information.

```
curl -H "X-PrettyPrint: 1" -H "Content-Type: application/json" -d '{
"templateId":"07gD00000004CEhIAM",
"templateBindings" : [ { "key":"ApiVersion", "value":"v34.0" },
{ "key":"Text", "value":"This post was created by an API Action Link!" }, {
"key":"SubjectId", "value":"me" } ] }'
 -X POST
"https://instance_name/services/data/v34.0/connect/action-link-group-definitions"
 -H 'Authorization: OAuth 00DRR0000000N0g!RWaPj94O6yOD.lfjB9LqMk' --insecure
```

Response body

Action Link Group Definition

Step 4: Associate the Action Link Group with a Feed Element and Post It

You can associate up to 10 action link groups with a feed element.

Resource

`/chatter/feed-elements`

HTTP method

POST

Request body

Feed Item Input

Request body example

Grab the action link group ID from the `id` property of the Action Link Group Definition response from step 3.

```
POST /services/data/v34.0/chatter/feed-elements

{
```

```
  "body": {
    "messageSegments": [
      {
        "type": "Text",
        "text": "Click to post a feed item."
       }
    ]
    },
  "subjectId": "me",
  "feedElementType": "feedItem",
  "capabilities": {
    "associatedActions": {
      "actionLinkGroupIds": ["0AgRR0000004CTr0AM"]
    }
  }
}
```

Request example using cURL

To use cURL to make the request, enter the following and substitute the action link group ID returned in step 2, your Developer Edition instance name, and your OAuth information.

```
curl -H "X-PrettyPrint: 1" -H "Content-Type: application/json" -d '{ "body": {
"messageSegments": [ { "type": "Text", "text": "Click to post a feed item." } ] },
"subjectId": "me", "feedElementType": "feedItem", "capabilities": { "associatedActions":
 { "actionLinkGroupIds": ["0AgRR0000004CTr0AM"] } } }' -X POST
"https://instance_name/services/data/v34.0/chatter/feed-elements" -H 'Authorization:
OAuth OOARoAQETAKEIddnMWXh462dprYAgRWaPj94O6yOD' --insecure
```

Response body

Feed Item

SEE ALSO:

Action Link Group Definitions

Feed Elements, Post and Search

Action Link Templates

Working with Action Links

Define Action Links and Post with a Feed Element

Update the Status of an Action Link or Invoke an Action Link

Usage

Use this resource to update the `status` property of an action link. To invoke an `Api` or an `ApiAsync` action link, set the status to `PendingStatus`. The `status` of the action link in combination with the `labelKey` property of the Action Link Definition Input determines the text that displays in the UI.

Different types of action links have different `status` workflows:

`Api`

To invoke an `Api` action link from an app, make a PATCH request to mark the link `PendingStatus`. Salesforce handles the call out to the `actionUrl` that was provided in the Action Link Definition Input. For an `Api` action link, Salesforce sets the `status` to `SuccessfulStatus` or `FailedStatus` based on the HTTP status code returned by your server.

`ApiAsync`
To invoke this action from an app, make a PATCH request to mark the link `PendingStatus`. Salesforce handles the call out to the `actionUrl` that was provided in the Action Link Definition Input. For an `ApiAsync` action link, the callout starts an asynchronous operation on your server. Salesforce sets `status` to `PendingStatus` unless the call returned an error status code, in which case it sets `status` to `FailedStatus`. When your server has finished the asynchronous processing it should call back and PATCH the action link to `SuccessfulStatus` or `FailedStatus`.

`Download` **and** `Ui`
Direct the user to download or visit the UI from the link in the `actionUrl` of the Platform Action response body. Your application must decide when to make a PATCH request to set `status` to `SuccessfulStatus` or `FailedStatus`.

Note: You can't PATCH the `status` of an action link:

- Back to `NewStatus` from a different `status`.
- From a terminal `status` (`FailedStatus` or `SuccessfulStatus`) to `NewStatus` or `PendingStatus`.
- When the action link belongs to an action link group whose `executionsAllowed` property is `Unlimited`.
- For an `Api` or `ApiAsync` action link whose `executionsAllowed` property is `Once`, when a user PATCHes the `status` to `PendingStatus` for the first time, the callout is made to the action link. If the same user PATCHes it to `PendingStatus` again, Salesforce returns the current action link information. However, if another user PATCHes the status to `PendingStatus`, Salesforce throws an error. In both cases, the callout to the action link is not resent.

Resource
`/connect/action-links/`***`actionLinkId`***

HTTP method
PATCH

Request body
Action Link Input

Request parameter
`status`

Request body example

```
{ "status":"SuccessfulStatus" }
```

Request parameter example

```
/services/data/v34.0/connect/action-links/actionLinkId?status=SuccessfulStatus
```

Returns
Platform Action

Edit a Feed Post

Resource
`/chatter/feed-elements/`***`feedElementId`***

HTTP method
PATCH

Request body
Feed Item Input

The `body` property of the Feed Item Input request body can be edited.

Request body example

```
{
   "body":{
      "messageSegments":[
         {
            "type":"Text",
           "text":"Check out the latest API documentation on developer.salesforce.com."

         }
      ]
   }
}
```

Returns

Feed Item

Edit a Comment

Resource

`/chatter/comments/`***`commentId`***

HTTP method

PATCH

Request body

Comment Input

The `body` property of the Comment Input can be edited.

Request body example

```
{   "body":{
      "messageSegments":[
         {
            "type":"Text",
            "text":"I am definitely going to check that out."
         }
      ]
   }
}
```

Returns

Comment

Edit a Question Title and Post

Resource

`/chatter/feed-elements/`***`feedElementId`***

HTTP method

PATCH

Request body

Feed Item Input

The `body` property and the `questionTitle` property can be edited.

Request body example

```
{
   "body":{
      "messageSegments":[
         {
            "type":"Text",
            "text":"The latest public versions are on developer.salesforce.com."
         }
      ]
   },
   "capabilities":{
      "questionAndAnswers" : {
          "questionTitle" : "Where can I find API documentation"
      }
   }
}
```

Returns

Feed Item

Like a Feed Item

Resource

`/chatter/feed-elements/feedElementId/capabilities/chatter-likes/items`

HTTP method

POST

Example

```
/services/data/v34.0/chatter/feed-elements/feedElementId/capabilities/chatter-likes/items
```

Returns

Like

Like a Feed Item and Return the Total Number of Likes

Resource

`/connect/batch/`

HTTP method

POST

Request body

Generic Batch Collection Input

Request body example

```
{
   "batchRequests" : [
    {
       "method" : "Post",
       "url" :
"/v34.0/chatter/feed-elements/feedElementId/capabilities/chatter-likes/items"
    },
    {
       "method" : "Get",
       "url" :
"/v34.0/chatter/feed-elements/feedElementId/capabilities/chatter-likes/items"
    }
   ],
   "haltOnError" : "false"
}
```

Returns

Batch Results

Share a Feed Item

Resource

`/chatter/feed-elements`

HTTP method

POST

Request body

Feed Item Input

Request parameters

`originalFeedElementId`, `subjectId`

Request body example

```
{
   "originalFeedElementId": "0D5D0000000DaZBKA0",
   "subjectId": "me"}
```

Request parameter example

```
/services/data/v34.0/chatter/feed-elements?originalFeedElementId=0D5D0000000DaZBKA0&subjectId=me
```

Returns

Feed Item

Get Feed Elements Updated Since Last Request

Resource

To get feed elements that were updated since the last call to a feeds resource, make a request to the value of the `updatesUrl` property of the Feed Element Page response body.

These feeds return an `updatesUrl` property:

- Company—`/chatter/feeds/company/feed-elements`
- Files—`/chatter/feeds/files/me/feed-elements`
- Filter—`/chatter/feeds/filter/me/filterId/feed-elements`
- Groups—`/chatter/feeds/groups/me/feed-elements`
- News—`/chatter/feeds/news/me/feed-elements`
- People—`/chatter/feeds/people/me/feed-elements`
- Record—`/chatter/feeds/record/recordId/feed-elements`

This excerpt from the Feed Elements Page response body shows a sample value for the `updatesUrl` property:

```
{
...
..."updatesUrl" : "/services/data/v34.0/chatter/feeds
   /news/005D0000001LL8OIAW/feed-elements?updatedSince=2:1384811809000"
...
}
```

A feed element is considered to be updated if it was created since the last feed request, or if `sort=LastModifiedDateDesc` and a comment was added to the feed element since the last feed request. Adding likes and topics doesn't update a feed element.

HTTP method

GET

Returns

Feed Element Page

Get My News Feed

Resource

`/chatter/feeds/news/me/feed-elements`

HTTP method

GET

Example

```
/services/data/v34.0/chatter/feeds/news/me/feed-elements
```

Returns

Feed Element Page

Get a Group Feed

Resource

`/chatter/feeds/record/recordId/feed-elements`

Substitute ***recordId*** with the group ID.

HTTP method

GET

Example

```
/services/data/v34.0/chatter/feeds/record/0F9D00000000I4O/feed-elements
```

Returns

Feed Element Page

Get a Feed with Internal Feed Items Only

Resource

`/chatter/feeds/record/`***`recordId`***`/feed-elements`

HTTP method

GET

Request parameter

`showInternalOnly`

Example

```
/services/data/v34.0/chatter/feeds/record/0F9D00000000I4O/feed-elements?showInternalOnly=true
```

Returns

Feed Element Page

Search a Feed

Resource

`/chatter/feed-elements?q=`***`searchString`***

The search string can contain wildcards and must contain at least two characters that aren't wildcards.

HTTP method

GET

Request parameter

`q`

Request parameter example

```
/services/data/v34.0/chatter/feed-elements?q=track
```

Returns

Feed Element Page

Get a List of Records a User Follows

Resource

`/chatter/users/`***`userId`***`/following`

HTTP method

GET

Request parameter

`page` (optional)

Request parameter example

```
/services/data/v34.0/chatter/users/005D0000001GLowIAB/following?page=1
```

Returns

Subscription Page

Get Chatter Activity Statistics for a User

Resource

`/chatter/users/`***`userId`***

HTTP method

GET

Example

```
/services/data/v34.0/chatter/users/005D0000001GLowIAB?include=/chatterActivity
```

Returns

User Detail

Get Recommendations of Users to Follow

Resource

`/chatter/users/me/recommendations/follow/users`

HTTP method

GET

Example

```
/services/data/v34.0/chatter/users/me/recommendations/follow/users
```

Returns

Recommendations

Create a Group

Resource

`/chatter/groups`

HTTP method

POST

Request body

Group Input

Request parameters (required)

`name`, `visibility`

Request body example

```
{
   "name" : "Chatter API",
   "visibility" : "PublicAccess"
}
```

Request parameter example

```
/services/data/v34.0/chatter/groups?name=Chatter+API&visibility=PublicAccess
```

Returns

Group Detail

Update and Crop a Group Photo

This example crops an existing photo (a photo that has already been uploaded). You can also upload and crop a photo. To upload a new photo you must use a multipart/form-data request. See Uploading Binary Files.

Resource

`/chatter/groups/groupId/photo`

HTTP method

POST

Request body

Photo Input

Request parameters

`cropSize`, `cropX`, `cropY`, `fileId`

Request body example

```
{    "cropSize" : "240",
     "cropX" : "20",
     "cropY" : "20",
     "fileId" : "069D00000001IOh"
}
```

Request parameter example

```
/services/data/v34.0/chatter/groups/0F9300000008ORJ
/photo?cropSize=240&cropX=20&cropY=20&fileId=069D00000001IOh
```

Returns

Photo

Join a Group

Resource

`/chatter/groups/groupId/members`

HTTP method

POST

Request body
Group Member Input

Request parameter
`userId`

Request body example

```
{"userId": "005D0000001GpHp" }
```

Request parameter example

```
/services/data/v34.0/chatter/groups/0F9D000000006bOKAQ
/members?userId=005D0000001GpHp
```

Returns
Group Member

Request to Join a Private Group

Resource
`/chatter/groups/groupId/members/requests`

HTTP method
POST

Example

```
/services/data/v34.0/chatter/groups/0F9D000000006bOKAQ/members/requests
```

Returns
Group Membership Request

Approve or Reject a Request to Join a Private Group

Resource
`/chatter/group-memberships-requests/requestId`

HTTP method
PATCH

Request body
Group Membership Request Update

Request parameters
`status` and `responseMessage`

The `responseMessage` parameter is used only if the value of `status` is `Declined`.

Request body example

```
{ "status " : "Accepted" }
```

Request parameter example

```
/services/data/v34.0/chatter/group-membership-requests
/0I5D000000002fwKAA?status=Accepted
```

Returns

Group Membership Request

Post to a Group

Resource

`/chatter/feed-elements`

HTTP method

POST

Request body

Feed Item Input

Request parameters

`feedElementType`, `subjectId`, `text`

Request body example

Specify the group in the `subjectId`.

```
{
   "body" : {
      "messageSegments" : [
         {
            "type" : "Text",
            "text" : "When should we meet for release planning?"
         },
         {
            "type" : "Mention",
            "id" : "005D00000016Qxp"
         }]
       },
   "feedElementType" : "FeedItem",
   "subjectId" : "0F9B000000000W2"
}
```

Request parameter example

```
/services/data/v34.0/chatter
   /feed-elements?feedElementType=FeedItem&subjectId=0F9B000000000W2&text=New+post
```

Returns

Feed Item

Follow a Record

Resource

`/chatter/users/me/following`

HTTP method

POST

Request body

Following Input

Request parameter

`subjectId`

The `subjectId` is the ID of the item you want to follow. In versions 29.0 and above, the `subjectId` can be a topic ID.

Request body example

```
{ "subjectId" : "001D000000Iyu2p" }
```

Request parameter example

```
/services/data/v34.0/chatter/users/me/following?subjectId=001D000000Iyu2p
```

Returns

Subscription

Stop Following a Record

Resource

`/chatter/subscriptions/`***`subscriptionId`***

HTTP method

DELETE

Example

```
/services/data/v34.0/chatter/subscriptions/0E8D00000001JkFKAU
```

Returns

204: Successful Delete

A complete structured response body is not returned, just a return code.

Get a Record Feed

Resource

`/chatter/feeds/record/`***`recordId`***`/feed-elements`

HTTP method

GET

Example

```
/services/data/v34.0/chatter/feeds/record/005x0000001oXx9AAE
/feed-elements
```

Returns

Feed Element Page

Update a User Profile

Resource

`/chatter/users/`***`userId`***

HTTP method
PATCH

Request body
User Input

Request parameter
`aboutMe`

Request body example

```
{   "aboutMe": "Writer at Salesforce"}
```

Request parameter example

```
/services/data/v34.0/chatter/users/me?aboutMe=Writer+at+Salesforce
```

Returns
User Detail

Update and Crop a User Photo

This example crops an existing photo (a photo that has already been uploaded). You can also upload and crop a photo. To upload a new photo you must use a multipart/form-data request. See Uploading Binary Files. To use this example to update and crop a group photo, substitute `/chatter/groups/groupId/photo`.

Resource
`/chatter/users/userId/photo`

HTTP method
POST

Request body
Photo Input

Request parameters
`cropSize`, `cropX`, `cropY`, `fileId`

Request body example

```
{   "cropSize" : "240",
    "cropX" : "20",
    "cropY" : "20",
    "fileId" : "069D00000001IOh"
}
```

Request parameter example

```
/services/data/v34.0/chatter/users
/me/photo?cropSize=240&cropX=20&cropY=20&fileId=069D00000001IOh
```

Returns
Photo

Send a Private Message

Resource

`/chatter/users/me/messages/`

HTTP method

POST

Request body

User Message

Request parameter

`recipients` OR `inReplyTo`

`text`

Request body example

```
{ "body" : "Are we ready for next week's customer meeting?",
  "recipients" : ["005D0000001GOYpIAO", "005D0000001GbdHIAS"]
}
```

Request parameter example

```
/services/data/v34.0/chatter/users/me/messages?recipients=
005D0000001GOYpIAO,005D0000001GbdHIAS&text=Are+we+ready+for+next+week's+customer+meeting?
```

Returns

Message

Get User Information for Multiple Users

Resource

`/chatter/users/batch/`***`user_list`*** where *`user_list`* is a list of user Ids. Your app may need to call Chatter REST API twice: once to get the feed items and user Ids, the second time to call this resource with the list of user Ids.

HTTP method

GET

Example

```
/services/data/v34.0/chatter/users
/batch/005D0000001Kl6xIAC,005D0000005J3D4HVB, . . .
```

Returns

Batch Results

CHAPTER 5 Chatter REST API Resources

This chapter contains the Chatter REST API resources. REST resources are sometimes called *endpoints*.

IN THIS SECTION:

Action Links Resources

An action link is a button on a feed element. Clicking an action link can take a user to a Web page, initiate a file download, or invoke an API call to Salesforce or to an external server. An action link includes a URL and an HTTP method, and can include a request body and header information, such as an OAuth token for authentication. Use action links to integrate Salesforce and third-party services into the feed so that users can take action to drive productivity and accelerate innovation.

Announcements Resource

Information about an announcement. Get information or update the expiration date of an existing announcement. An announcement displays in a designated location in the Salesforce UI until 11:59 p.m. on its expiration date, unless it's deleted or replaced by another announcement.

Batch Resource

Create a request body with a batch of up to 25 subrequests and send them to the server in a single request to improve your application's performance. The response bodies and HTTP statuses for each subrequest in the batch are returned within a single response body. Each subrequest counts against rate limits.

Chatter Resource

A directory of Chatter feeds, groups, and users resources.

Comments Resources

Get information about, edit, like, or delete a comment. Determine whether the context user can edit a comment.

Communities Resources

Information about the communities available to the context user, or information about a specific community.

Communities Moderation Resources

Information about whether a user has flagged a feed element, comment, or file as inappropriate. Also used to delete flags.

Connect Resources

Information about the context user's organization and settings, and information about the communities the context user is a member of.

Dashboard Component Snapshot Resource

A snapshot of a dashboard component.

Datacloud Resources

Information about Datacloud resources used to purchase Data.com contacts or companies.

Feed Elements Resources

Information about feed elements. Access, edit, post, search, and delete feed elements. Access a feed element's capabilities, including bundles and their feed elements.

Feed Items Resources

Information about feed items. Access, search, delete, bookmark, and like feed items. Access and post comments. Access and vote in polls. And access, add, and remove topics.

Feeds Resources

Feeds are made up of feed items. There are many feed types. Each feed type has an algorithm that determines which feed items it contains.

Files Connect Repository Resources

A files connect repository is an external content repository that's connected to Salesforce. Use these resources to get a list of repositories, to get information about a repository or a repository file, and to get the content of a repository file. Also use these resources to get information about the files and folders in a repository folder.

Files Resources

Information about files, including content, rendered version, and sharing.

Folders Resources

Information about a folder or the contents of a folder. Modify and delete folders, add items to folders, and sync root folders.

Groups Resources

Information about groups, such as the group's members, photo, and the groups in the organization. Create and delete a group, add members to a group, and change the group photo.

Influence Resource

List of the percentile thresholds used to categorize an organization's users by their influence in Chatter.

Likes Resource

Information about the specified like. Also used to delete a like.

Managed Topics Resources

Managed topics for a community. Get a list of managed topics. Create, delete, or reorder managed topics in a community.

Mentions Resources

Information about users and groups that a user can mention in a feed item body or comment body. When a user or group is mentioned, they receive notification.

Motifs Resources

Returns URLs for sets of small, medium, and large motif icons. Records have different sets of motif icons depending on the record type. The motif can also include the record type's base color.

Push Notifications Resource

Send a mobile push notification to connected apps on users' devices. This resource is only accessible when the session is established with a connected app that is developed in the same organization or installed in the same package as the recipient's connected app.

Question and Answers Suggestions Resource

Get question and answers suggestions.

Recommendation Definition Resources

Recommendation definitions allow you to create custom recommendations that appear in communities, encouraging users to watch videos, take training and more. Use these resources to get information about, create, modify, and delete recommendation definitions. Also get information about, upload, modify, and delete recommendation definition photos.

Records Resources

Get information about the followers of a record. Add and remove topics from records.

Scheduled Recommendations Resources

Get information about, create, update, or delete scheduled recommendations.

Subscriptions Resource

Information about the specified subscription. Also used to delete a subscription, for example, to unfollow a record or a topic.

Topics Resources

List of topics for the organization, list of suggested topics for a specified string of text, feed item, or record, list of trending topics for the organization, information about a specified topic, endorsements for a specific topic, information about the files posted recently to a specified topic, information about the groups recently contributing to a specified topic, knowledgeable people for a specified topic, and list of topics related to a specified topic.

User Profiles Resource

Returns details about a user's profile. The user profile data populates the profile page (also called the Chatter profile page). This data includes user information (such as address, manager, and phone number), some user capabilities (permissions), and a set of subtab apps, which are custom tabs on the profile page.

Users Resources

Information about the user, such as who is following the user, the user's files or recommendations. Also used for posting feed items and updating conversation status.

Zones Resources

Information about Chatter Answers zones.

Action Links Resources

An action link is a button on a feed element. Clicking an action link can take a user to a Web page, initiate a file download, or invoke an API call to Salesforce or to an external server. An action link includes a URL and an HTTP method, and can include a request body and header information, such as an OAuth token for authentication. Use action links to integrate Salesforce and third-party services into the feed so that users can take action to drive productivity and accelerate innovation.

There are two views of an action link and an action link group: the definition, and the context user's view. The definition includes potentially sensitive information, such as authentication information. The context user's view is filtered by visibility options and the values reflect the state of the context user.

Available resources:

Resource	Description
`/connect/action-link-group-definitions`	Create an action link group definition. To associate an action link with a feed element, first create an action link group definition. Then post a feed element with an associated actions capability.
`/connect/action-link-group-definitions/`***`actionLinkGroupId`***	Get information about or delete an action link group definition. Deleting an action link group definition removes all references to it from feed elements. To get or delete the action link group definition, the client must be the same connected app that created the definition and the user must either be the user who created the definition or have "View all Data" permission.
`/connect/action-link-groups/`***`actionLinkGroupId`***	Get information about an action link group including state for the context user.
`/connect/action-links/`***`actionLinkId`***	Get information about or update the `status` of an action link. The `status` of the action link in combination with the `labelKey` property of the Action Link Definition Input determines the text that displays in the UI. To trigger the API request for `Api` and `ApiAsync` action links, set the `status` to `PendingStatus`.

Resource	Description
`/connect/action-links/actionLinkId/diagnostic-info`	Get diagnostic information returned when an action link executes. Diagnostic information is given only for users who can access the action link.

SEE ALSO:

Working with Action Links

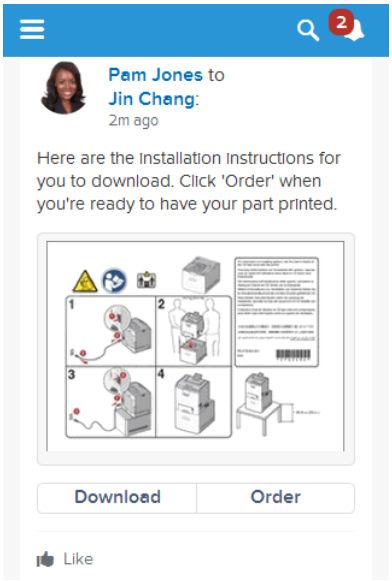
Define Action Links and Post with a Feed Element

Action Link Group Definitions

Create an action link group definition. To associate an action link with a feed element, first create an action link group definition. Then post a feed element with an associated actions capability.

All action links must belong to a group. Action links in a group are mutually exclusive and share some properties. Define stand-alone actions in their own action group.

This image shows a feed element with one action link group containing two action links, **Download** and **Order**:



Resource

```
/connect/action-link-group-definitions
```

```
/connect/communities/communityId/action-link-group-definitions
```

Available since release

33.0

Requires Chatter

No

HTTP methods

POST

The Action Link Group Definitions resource doesn't support query parameters for POST. You must use an Action Link Group Definition Input request body.

Request body for POST

Action Link Group Definition Input

Response body for POST

Action Link Group Definition

Example for POST

To associate an action link group with a feed item, first create an action link group definition:

Note: An `actionUrl` to a Salesforce resource can be relative. All others must be absolute and start with `https://`.

```
POST /connect/action-link-group-definitions

{
    "actionLinks": [
    {
       "actionType" : "Api",
       "actionUrl" : "https://www.example.com/api/v1/cofirm",
       "groupDefault" : "true",
       "headers" : [
       {
          "name" : "Authorization",
          "value" : "OAuth 00DRR0000000N0g!ARoAQMZyQtsP1Gs27EZ8hl7vdpYXH5O5rv1
                       VNprqTeD12xYnvygD3JgPnNR._nmE6thZakAbBVJUsjUInMUaKMlkNKaWXdHO"
       },
       {
          "name" : "Content-Type",
          "value" : "application/json"
       },
       {
          "name" : "Accept",
          "value" : "application/json"
       },
       {
          "name" : "X-PrettyPrint",
          "value" : "1"
       }
       ],
       "labelKey" : "Confirm",
       "method" : "HttpPost",
       "requiresConfirmation" : "true"
    },
    {
       "actionType" : "Api",
       "actionUrl" : "https://www.example.com/api/v1/deny",
       "groupDefault"  : "false",
       "headers" : [
       {
          "name" : "Authorization",
          "value" : "OAuth 00DRR0000000N0g!ARoAQMZyQtsP1Gs27EZ8hl7vdpYXH5O5rv1
                       VNprqTeD12xYnvygD3JgPnNR._nmE6thZakAbBVJUsjUInMUaKMlkNKaWXdHO"
       },
       {
```

```
            "name" : "Content-Type",
            "value" : "application/json"
        },
        {
            "name" : "Accept",
            "value" : "application/json"
        },
        {
            "name" : "X-PrettyPrint",
            "value" : "1"
        }
        ],
        "labelKey": "Deny",
        "method" : "HttpPost",
        "requiresConfirmation"  : "true"
    },
    {
        "actionType": "Ui",
        "actionUrl" : "https://www.example.com/reviewPage.html",
        "groupDefault" : "false",
        "labelKey" : "Review",
        "method" : "HttpGet",
        "requiresConfirmation" : "false"
    },
    "executionsAllowed" : "OncePerUser",
    "category" : "Primary",
    "expirationDate" :  "2014-05-09T18:45:39.000Z"
}
```

Next, associate the action link group with the feed item and post the feed item. Grab the action link group ID from the `/connect/action-link-group-definitions` response and pass it in a POST request to `/chatter/feed-elements`.

```
POST  /chatter/feed-elements

{
  "body": {
    "messageSegments": [
      {
        "type": "Text",
        "text": "Please confirm or deny this item. You can also click Review to provide
 more feedback."
       }
    ]
    },
  "subjectId": "me",
  "feedElementType": "FeedItem",
  "capabilities": {
    "associatedActions": {
      "actionLinkGroupIds": ["0AgRR0000004CTr0AM"]
    }
  }
}
```

Different types of action links have different workflows to set their `status` property, whose value displays in the UI. See `/connect/action-links/actionLinkId` for more information.

Note: This example doesn't include other capabilities, but feed items with action links do support other capabilities, such as content, polls, and links.

SEE ALSO:

Define Action Links and Post with a Feed Element

Action Link Templates

Define Action Links and Post with a Feed Element

Define Action Links in a Template and Post with a Feed Element

Action Link Group Definition

Get information about or delete an action link group definition. Deleting an action link group definition removes all references to it from feed elements. To get or delete the action link group definition, the client must be the same connected app that created the definition and the user must either be the user who created the definition or have "View all Data" permission.

Resource

```
/connect/action-link-group-definitions/actionLinkGroupId
```

```
/connect/communities/communityId/action-link-group-definitions/actionLinkGroupId
```

Available since release

33.0

Requires Chatter

No

HTTP methods

GET, DELETE, HEAD

Response body for GET

Action Link Group Definition

Response body for DELETE

204: Successful Delete

Action Link Group

Get information about an action link group including state for the context user.

All action links must belong to a group. Action links in a group are mutually exclusive and share some properties. Note that action link groups are accessible by clients, unlike action link group definitions.

Resource

```
/connect/action-link-groups/actionLinkGroupId
```

```
/connect/communities/communityId/action-link-groups/actionLinkGroupId
```

Available since release

33.0

Requires Chatter
No

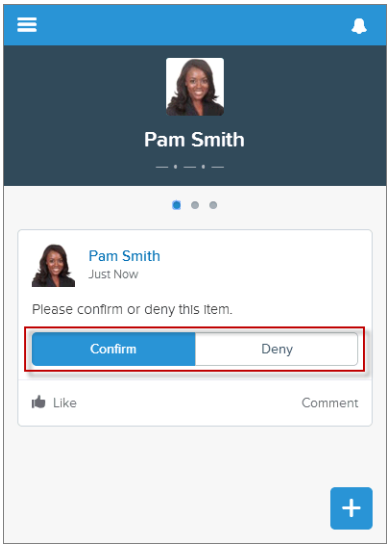
HTTP methods
GET, HEAD

Response body for GET
Platform Action Group

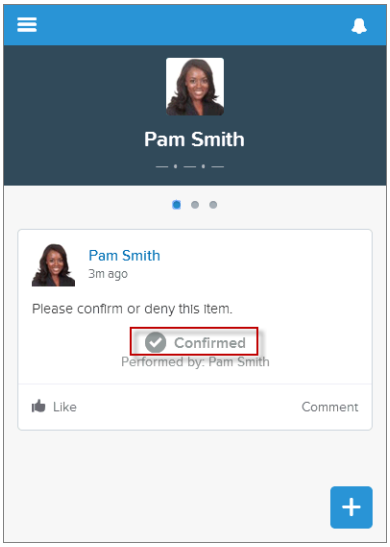
Action Link

Get information about or update the `status` of an action link. The `status` of the action link in combination with the `labelKey` property of the Action Link Definition Input determines the text that displays in the UI. To trigger the API request for `Api` and `ApiAsync` action links, set the `status` to `PendingStatus`.

This image shows two action links. In this image, the `status` of both action links is `NewStatus`.



If Pam Smith clicks **Confirm**, the action link `status` is set to `SuccessfulStatus` and results in the feed item shown in this image.



Different types of action links have different `status` workflows:

`Api`
: To invoke an `Api` action link from an app, make a PATCH request to mark the link `PendingStatus`. Salesforce handles the call out to the `actionUrl` that was provided in the Action Link Definition Input. For an `Api` action link, Salesforce sets the `status` to `SuccessfulStatus` or `FailedStatus` based on the HTTP status code returned by your server.

`ApiAsync`
: To invoke this action from an app, make a PATCH request to mark the link `PendingStatus`. Salesforce handles the call out to the `actionUrl` that was provided in the Action Link Definition Input. For an `ApiAsync` action link, the callout starts an asynchronous operation on your server. Salesforce sets `status` to `PendingStatus` unless the call returned an error status code, in which case it sets `status` to `FailedStatus`. When your server has finished the asynchronous processing it should call back and PATCH the action link to `SuccessfulStatus` or `FailedStatus`.

`Download` **and** `Ui`
: Direct the user to download or visit the UI from the link in the `actionUrl` of the Platform Action response body. Your application must decide when to make a PATCH request to set `status` to `SuccessfulStatus` or `FailedStatus`.

Note: You can't PATCH the `status` of an action link:

- Back to `NewStatus` from a different `status`.
- From a terminal `status` (`FailedStatus` or `SuccessfulStatus`) to `NewStatus` or `PendingStatus`.
- When the action link belongs to an action link group whose `executionsAllowed` property is `Unlimited`.
- For an `Api` or `ApiAsync` action link whose `executionsAllowed` property is `Once`, when a user PATCHes the `status` to `PendingStatus` for the first time, the callout is made to the action link. If the same user PATCHes it to `PendingStatus` again, Salesforce returns the current action link information. However, if another user PATCHes the status to `PendingStatus`, Salesforce throws an error. In both cases, the callout to the action link is not resent.

Resource

```
/connect/action-links/actionLinkId
```

```
/connect/communities/communityId/action-links/actionLinkId
```

Available since release
33.0

Requires Chatter
No

HTTP methods
GET, PATCH, HEAD

Request body for PATCH

Root XML tag
`<actionLink>`

JSON example

```
{  "status" : "SuccessfulStatus"  }
```

Properties

Name	Type	Description	Required or Optional	Available Version
`status`	String	The current status of the action. Values are: • `FailedStatus`—The action link execution failed. • `NewStatus`—The action link is ready to be executed. Available for `Download` and `Ui` action links only. • `PendingStatus`—The action link is executing. Choosing this value triggers the API call for `Api` and `ApiAsync` action links. • `SuccessfulStatus`—The action link executed successfully.	Required	33.0

Request parameters for PATCH

Name	Type	Description	Available Version
`status`	String	Values are: • `FailedStatus`—The action link execution failed. • `NewStatus`—The action link is ready to be executed. Available for `Download` and `Ui` action links only.	33.0

Name	Type	Description	Available Version
		• `PendingStatus`—The action link is executing. Choosing this value triggers the API call for `Api` and `ApiAsync` action links. • `SuccessfulStatus`—The action link executed successfully.	

Response body for GET or PATCH

Platform Action

Example for PATCH

```
PATCH /connect/action-links/0AnRR0000004CTS0A2?status=FailedStatus
```

Action Link Diagnostic Information

Get diagnostic information returned when an action link executes. Diagnostic information is given only for users who can access the action link.

Resource

```
/connect/action-links/actionLinkId/diagnostic-info
```

```
/connect/communities/communityId/action-links/actionLinkId/diagnostic-info
```

Available version

33.0

Requires Chatter

No

HTTP methods

GET, HEAD

Response body

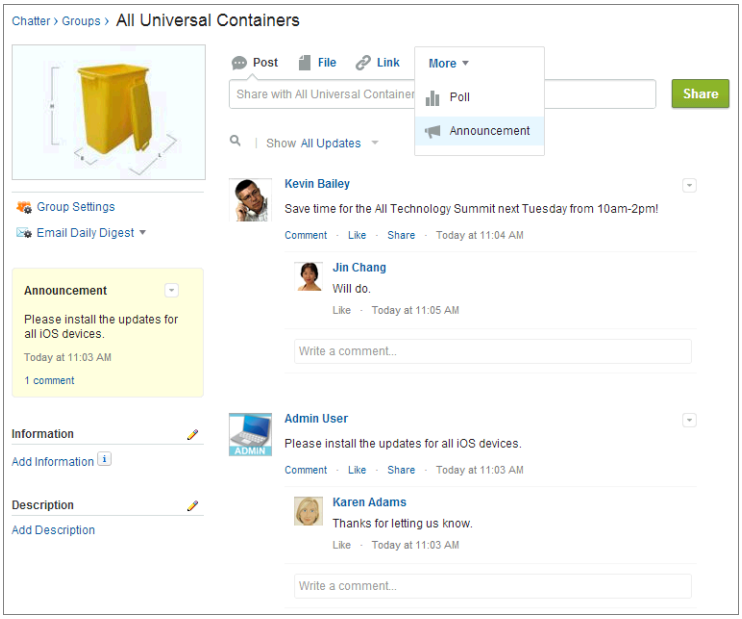
Action Link Diagnostic

Announcements Resource

Information about an announcement. Get information or update the expiration date of an existing announcement. An announcement displays in a designated location in the Salesforce UI until 11:59 p.m. on its expiration date, unless it's deleted or replaced by another announcement.

Use an announcement to highlight information. Users can discuss, like, and post comments on announcements in the group feed. Group members receive an email notification when you post an announcement, same as for other posts, depending on their selected group email notification frequency. Deleting the feed post deletes the announcement.

This image of Salesforce shows an announcement displayed in yellow. Creating an announcement also creates a feed item with the announcement text, which you can also see in the image.



To post an announcement to a group, or to get information about all the announcements in a group, use the Group Announcements resource (`/chatter/groups/`***`groupId`***`/announcements`).

Resource

`/chatter/announcements/`***`announcementId`***

`/connect/communities/`***`communityId`***`/chatter/announcements/`***`announcementId`***

Available since version

31.0

Requires Chatter

Yes

HTTP methods

GET, PATCH, DELETE, HEAD

Request parameters for PATCH

Parameter Name	Value	Description	Required or Optional	Available Version
`expirationDate`	Date	The Salesforce UI displays an announcement until 11:59 p.m. on this date unless another announcement is posted first. The Salesforce UI ignores the time value in the	Required	31.0

Parameter Name	Value	Description	Required or Optional	Available Version
		`expirationDate`. However, you can use the time value to create your own display logic in your own UI.		

Response for GET, PATCH

Announcement

Response for DELETE

HTTP Status Code 204

Batch Resource

Create a request body with a batch of up to 25 subrequests and send them to the server in a single request to improve your application's performance. The response bodies and HTTP statuses for each subrequest in the batch are returned within a single response body. Each subrequest counts against rate limits.

Use this resource to minimize the number of round-trips between the client and the server. For example, in a mobile app, when a user taps the button to like a feed item, the response body includes information about the like, but it doesn't include the total number of likes on the feed item. To like the feed item, POST a request to `/chatter/feed-elements/`***`feedElementId`***`/capabilities/chatter-likes/items`. To get the like total, GET the feed item information from `/chatter/feed-elements/`***`feedElementId`***`/capabilities/chatter-likes/items`.

Add both requests to a Generic Batch Collection Input request body and send them both in a single POST request to `/connect/batch`:

```
{
   "batchRequests" : [
    {
       "method" : "Post",
      "url" : "/v34.0/chatter/feed-elements/feedElementId/capabilities/chatter-likes/items"

    },
    {
       "method" : "Get",
      "url" : "/v34.0/chatter/feed-elements/feedElementId/capabilities/chatter-likes/items"

    }
   ],
   "haltOnError" : "false"
}
```

The requests in a batch are called *subrequests*. All subrequests are executed in the context of the same user. Subrequests are independent and information can't be passed between them. Subrequests execute serially in their order in the request body. When a subrequest executes successfully, it commits its data. Commits are reflected in the output of later subrequests (as in the previous example which liked a feed item then got the total number of likes). If a subrequest fails, commits made by previous subrequests are not rolled back.

Resource

```
/connect/batch
```

Available since version

28.0

Requires Chatter

No

HTTP methods

POST

Request body

Generic Batch Collection Input

Root XML Tag

```
<batch>
```

JSON

```
{
   "batchRequests" : [
    {
       "method" : "Get",
       "url" : "/v34.0/chatter/feeds/news/me"
    },
    {
       "method" : "Get",
       "url" : "/v34.0/chatter/feeds/user-profile/me"
    }
   ]
}
```

Properties

Name	Type	Description	Available Version
`batchRequests`	Generic Batch Input[]	Collection of Batch Request Input request bodies containing the subrequest URLs to execute.	28.0
`haltOnError`	Boolean	The default value is `false`. If the value is `false` and a subrequest in the batch does not complete, Salesforce attempts to execute subsequent subrequests in the batch. If the value is `true` and a subrequest in the batch does not complete due to an HTTP response in the 400 or 500 range, Salesforce halts execution. It does not attempt to execute subsequent subrequests in the batch, and it returns an HTTP 412 status code and a `BATCH_PROCESSING_HALTED` error	28.0

Name	Type	Description	Available Version
		message for each subsequent subrequest indicating that a previous request was unsuccessful. The top-level request to `/connect/batch` returns HTTP 200 and the `hasErrors` property in the response is set to `true`.	

Request parameters

None.

Response

Batch Results

Example

This simple example is a POST request made to `https://`***`instance_name`***`/services/data/v34.0/connect/batch` that includes the following request body containing two subrequests:

```
{
   "batchRequests" : [
    {
       "method" : "Get",
       "url" : "/v34.0/chatter/feeds/user-profile/me"
    },
    {
       "method" : "Get",
       "url" : "/v34.0/chatter/feeds/news/me"
    }
   ]
}
```

The Batch Results response body includes a Batch Result Item response body for each subrequest:

```
{
   "hasErrors": false,
   "results": [
      {
         "result": {
            "feedElementPostUrl": "/services/data/v34.0/chatter/feed-elements",
            "feedElementsUrl": "/services/data/v34.0/chatter/feeds
/user-profile/005D0000001LL8OIAW/feed-elements",
            "isModifiedUrl": null
         },
         "statusCode": 200
      },
      {
         "result": {
            "feedElementPostUrl": "/services/data/v34.0/chatter/feed-elements",
            "feedElementsUrl": "/services/data/v34.0/chatter/feeds
/news/005D0000001LL8OIAW/feed-elements",
            "isModifiedUrl": null
```

```
            },
            "statusCode": 200
        }
    ]
}
```

Example: `haltOnError` **is** `true`

This example is the same as the previous example but it includes an error in the first subrequest and `haltOnError` is set to `true`. Note that the first subrequest contains a typo.

```
{
   "batchRequests" : [
    {
       "method" : "Get",
       "url" : "/v34.0/chatter/feeds/user-profile/men"
    },
    {
       "method" : "Get",
       "url" : "/v34.0/chatter/feeds/news/me"
    }
   ],
   "haltOnError" : "true"
}
```

The Batch Results response body includes a Batch Result Item response body for each subrequest. The top-level request to `/connect/batch` returns an HTTP 200, but the first subrequest returns an HTTP 404 because the resource can't be found. Because `haltOnError` is `true`, the next subrequest returns an HTTP 412. Also note that `hasErrors` is `true`.

```
{
   "hasErrors": true,
   "results": [
      {
         "result": [{
            "errorCode": "NOT_FOUND",
            "message": "The requested resource does not exist"
         }],
         "statusCode": 404
      },
      {
         "result": [{
            "errorCode": "BATCH_PROCESSING_HALTED",
            "message": "Batch processing halted per request"
         }],
         "statusCode": 412
      }
   ]
}
```

Post multiple comments with binary files

This example is a generic batch request to `https://`***`instance_name`***`/services/data/v34.0/connect/batch` that contains two subrequests. Each subrequest comments on a feed item and uploads a binary file to the comment.

You must include the request body of each subrequest in the `richInput` properties.

You must also include information about the binary parts in the `binaryPartName` and `binaryPartNameAlias` properties.

This is the cURL example:

```
curl -H "X-PrettyPrint: 1" -F 'json={ "haltOnError":true, "batchRequests":[ {
"url":"/v33.0/chatter/feed-elements/0D5D0000000YG0N/capabilities/comments/items",
"method":"Post", "binaryPartName":"binaryPart1",
"binaryPartNameAlias":"feedElementFileUpload", "richInput":{ "body":{ "messageSegments":[
 { "type":"Text", "text":"Check out this file, it may help." } ] },
 "capabilities":{ "content":{ "title":"Presentation1.txt" } } } }, {
"url":"/v33.0/chatter/feed-elements/0D5D0000000YG0M/capabilities/comments/items",
"method":"Post", "binaryPartName":"binaryPart2",
"binaryPartNameAlias":"feedElementFileUpload", "richInput":{ "body":{ "messageSegments":[
 { "type":"Text", "text":"Check out this file, it may help." } ] },
 "capabilities":{ "content":{ "title":"Presentation2.txt" } } } } ]
};type=application/json' -F
"binaryPart1=@/Users/jbleyle/Desktop/Presentation1.txt;type=application/octet-stream"
-F "binaryPart2=@/Users/jbleyle/Desktop/Presentation2.txt;type=application/octet-stream"

-X POST https://instance_name/services/data/v34.0/connect/batch
-H 'Authorization: OAuth 00DD00000007HoR!...' --insecure
```

This is the multipart/form-data request body:

```
https://instance_name/services/data/v34.0/connect/batch

Authorization: OAuth 00DD0000000Jhd2!...
Accept: application/json
Host: instance_name
Content-Type: multipart/form-data; boundary=123123

--123123
Content-Disposition: form-data; name="json"
Content-Type: application/json

{
   "haltOnError":true,
   "batchRequests":[
      {
        "url":"/v34.0/chatter/feed-elements/0D5D0000000YG0D/capabilities/comments/items",

          "method":"Post",
          "binaryPartName":"binaryPart1",
          "binaryPartNameAlias":"feedElementFileUpload",
          "richInput":{
             "body":{
                "messageSegments":[
                   {
                      "type":"Text",
                      "text":"Check out this file, it may help."
                   }
                ]
             },
             "capabilities":{
                "content":{
                   "title":"Presentation1.txt"
```

```
                }
            }
         }
      },
      {
        "url":"/v34.0/chatter/feed-elements/0D5D0000000YG0H/capabilities/comments/items",

         "method":"Post",
         "binaryPartName":"binaryPart2",
         "binaryPartNameAlias":"feedElementFileUpload",
         "richInput":{
            "body":{
               "messageSegments":[
                  {
                     "type":"Text",
                     "text":"Check out this file, it may help."
                  }
               ]
            },
            "capabilities":{
               "content":{
                  "title":"Presentation2.txt"
               }
            }
         }
      }
   ]
}


--123123
Content-Disposition: form-data; name="binaryPart1"; filename="Presentation1.txt"
Content-Type: application/octet-stream; charset=ISO-8859-1
Content-Transfer-Encoding: binary

This is the content of file 1
--123123
Content-Disposition: form-data; name="binaryPart2"; filename="Presentation2.txt"
Content-Type: application/octet-stream; charset=ISO-8859-1
Content-Transfer-Encoding: binary

This is the content of file 2
--123123--
```

SEE ALSO:

Like a Feed Item and Return the Total Number of Likes

Feed Elements, Batch Post

Chatter Resource

A directory of Chatter feeds, groups, and users resources.

Resource

```
/chatter
```

```
/connect/communities/communityId/chatter
```

Available since version

23.0

Requires Chatter

No

HTTP methods

GET or HEAD

Response body

Chatter Directory

Example response body

```
{
   "feeds": "/services/data/v34.0/chatter/feeds",
   "groups": "/services/data/v34.0/chatter/groups",
   "users": "/services/data/v34.0/chatter/users"
}
```

Comments Resources

Get information about, edit, like, or delete a comment. Determine whether the context user can edit a comment.

Available resources:

Resource	Description
/chatter/comments/commentId	Get information about, edit, or delete a comment.
/chatter/comments/commentId/capabilities/edit/is-editable-by-me	Determines whether the context user can edit a comment.
/chatter/comments/commentId/likes	Get information about the likes for a comment. Also used to like a comment.

Comment

Get information about, edit, or delete a comment.

Resource

```
/chatter/comments/commentId
```

```
/connect/communities/communityId/chatter/comments/commentId
```

Available since version

23.0

Requires Chatter
Yes

HTTP methods
GET, DELETE, PATCH

PATCH is supported in version 34.0 and later.

Request body for PATCH

Root XML tag
`<comment>`

Properties

Name	Type	Description	Available Version
`attachment`	Depends on type.	Optional. Values are: • Attachment Input: Existing Content • Attachment Input: New File Upload Important: As of version 32.0, use the `capabilities` property.	24.0-31.0
`body`	Message Body Input	The body of the comment. To edit this property in a comment, make a PATCH request to `/chatter/comments/`***`commentId`***. Editing comments is supported in version 34.0 and later.	23.0
`capabilities`	Comment Capabilities Input	Capabilities of the comment, such as a file attachment	32.0

Response body for GET and PATCH
Comment

Example

You can find the ID for a specific comment from any feed, such as a news feed or a record feed.

Use the following resource and the HTTP method GET to return information about a comment:

```
/chatter/comments/0D7D00000000NtkKAE
```

Comment Capability, Editability

Determines whether the context user can edit a comment.

Resource

```
/chatter/comments/commentId/capabilities/edit/is-editable-by-me
```

```
/connect/communities/communityId/chatter/comments/commentId/capabilities/edit/is-editable-by-me
```

Available since version
34.0

Requires Chatter
Yes

HTTP methods
GET

Response body for GET
Feed Entity Is Editable

Comment Likes

Get information about the likes for a comment. Also used to like a comment.

Resource

```
/chatter/comments/commentId/likes
```

```
/connect/communities/communityId/chatter/comments/commentId/likes
```

Available since version
23.0

Requires Chatter
Yes

HTTP methods
GET, POST or HEAD

Request parameters for GET or HEAD

Note: POST takes no request parameters, nor a request body.

Parameter Name	Type	Description
`page`	Integer	Specifies the number of the page you want returned.
`pageSize`	Integer	Specifies the number of items per page. Valid values are from 1 through 100. If you don't specify a size, the default is 25.

Response body for GET
Like Page

Response body for POST
Like

SEE ALSO:
Edit a Comment

Communities Resources

Information about the communities available to the context user, or information about a specific community.

Available resources are:

Resource	Description
`/connect/communities`	Returns a list of all communities available to the context user.
`/connect/communities/communityId`	Returns information about the specified community.

List of Communities

Returns a list of all communities available to the context user. If the user is an administrator, returns a list of all communities that meet the filter criteria. If the user is not an administrator, returns a list of all active communities that meet the filter criteria.

Resource

```
/connect/communities
```

Available since release
28.0

Requires Chatter
No

HTTP methods
GET or HEAD

Request parameters

Name	Type	Description	Available Version
`status`	String	Returns only communities with the specified status. Valid values are: • `Live` • `Inactive` • `UnderConstruction` Only users with the "Manage Network" permission enabled can see communities with an `Inactive` status.	28.0

Name	Type	Description	Available Version
		This parameter is optional.	

Response body
Community Page

Community Information

Returns information about the specified community.

Resource

```
/connect/communities/communityId
```

To access resources that are part of a specified community, insert this segment into the URL after the version information and before the resource you want to access. For example, to access a news feed resource in a community, use this URL:

```
https://instance_name/services/data/v34.0/connect
/communities/communityId/chatter/feeds/news/me/feed-elements
```

See Building the Chatter REST API Resource URL on page 4 for more information.

Available since release
26.0

Requires Chatter
No

HTTP methods
GET or HEAD

Response body
Community

Communities Moderation Resources

Information about whether a user has flagged a feed element, comment, or file as inappropriate. Also used to delete flags.

Available resources are:

Resource	Description
`/connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities/moderation`	Get information about a feed element's moderation capability, add a moderation flag to a feed element, or delete all moderation flags from a feed element.
`/connect/communities/communityId/chatter/comments/commentId/moderation-flags`	If the context user is a community moderator, returns the number of times a comment has been flagged by community members. Also flags a comment for moderation and deletes all flags on a comment.

Resource	Description
	If the context user isn't a community moderator, returns whether or not the user has flagged a comment. Also flags a comment for moderation and removes a flag the user previously added.
`/connect/communities/communityId/chatter/files/moderation`	Files that are flagged in a community. If the context user is a community moderator, returns the files that have been flagged by community members.
`/connect/communities/communityId/chatter/files/fileId/moderation-flags`	If the context user is a community moderator, returns the number of times a file has been flagged by community members. Also flags a file for moderation and deletes all flags on a file. If the context user isn't a community moderator, returns whether or not the user has flagged a file. Also flags a file for moderation and removes a flag the user previously added.

Moderation Flags on Posts

If the context user is a community moderator with the "Moderate Communities Feeds" permission, returns the number of times a post has been flagged by community members. Also flags a post for moderation and deletes all flags on a post.

If the context user isn't a community moderator, returns whether or not the user has flagged a post. Also flags a post for moderation and removes a flag the user previously added.

Resource

```
/connect/communities/communityId/chatter/feed-items/feedItemId/moderation-flags
```

Available versions

29.0–31.0

Important: In version 32.0 and later, use Feed Elements Capability, Moderation.

Requires Chatter

Yes

HTTP methods

GET, DELETE, HEAD, POST

Request parameters for GET

Name	Type	Description	Required or Optional	Available Version
`visibility`	String	Specifies the visibility behavior of a flag for various user types. • `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item. • `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.	Optional	30.0–31.0

Request body for POST

Root XML tag

```
<moderationFlag>
```

JSON

```
{
   "visibility" : "ModeratorsOnly"
}
```

Properties

Name	Type	Description	Available Version
`visibility`	String	Specifies the visibility behavior of a flag for various user types. • `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item. • `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.	30.0

Request parameters for POST

Name	Type	Description	Required or Optional	Available Version
`visibility`	String	Specifies the visibility behavior of a flag for various user types. • `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item. • `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.	Optional	30.0–31.0

Request parameters for DELETE

Name	Type	Description	Required or Optional	Available Version
`userId`	String	The ID of the context user. If specified, removes the flag the user previously added to the post. If not specified and the context user is a moderator for the community, all flags on the post are removed.	Optional	29.0–31.0

Response body for GET and POST
Moderation Flags

Moderation Flags on Comments

If the context user is a community moderator, returns the number of times a comment has been flagged by community members. Also flags a comment for moderation and deletes all flags on a comment.

If the context user isn't a community moderator, returns whether or not the user has flagged a comment. Also flags a comment for moderation and removes a flag the user previously added.

Resource

```
/connect/communities/communityId/chatter/comments/commentId/moderation-flags
```

Available since release
29.0

Requires Chatter
Yes

HTTP methods
GET, DELETE, HEAD, POST

Request parameters for GET

Name	Type	Description	Required or Optional	Available Version
`visibility`	String	Specifies the visibility behavior of a flag for various user types. • `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item. • `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.	Optional	30.0

Request body for POST

Root XML tag
```
<moderationFlag>
```

JSON

```
{
   "visibility" : "ModeratorsOnly"
}
```

Properties

Name	Type	Description	Available Version
`visibility`	String	Specifies the visibility behavior of a flag for various user types. • `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item. • `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.	30.0

Request parameters for POST

Name	Type	Description	Required or Optional	Available Version
`visibility`	String	Specifies the visibility behavior of a flag for various user types. • `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item. • `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.	Optional	30.0

Request parameters for DELETE

Name	Type	Description	Required or Optional	Available Version
`userId`	String	The ID of the context user. If specified, removes the flag the user previously added to the comment. If not specified and the context user is a moderator for the community, all flags on the comment are removed.	Optional	29.0

Response body for GET and POST

Moderation Flags

Moderation Flags on Files

If the context user is a community moderator, returns the number of times a file has been flagged by community members. Also flags a file for moderation and deletes all flags on a file.

If the context user isn't a community moderator, returns whether or not the user has flagged a file. Also flags a file for moderation and removes a flag the user previously added.

Resource

```
/connect/communities/communityId/chatter/files/fileId/moderation-flags
```

Available since release

30.0

Requires Chatter

No

HTTP methods

GET, DELETE, HEAD, POST

Request parameters for GET

Name	Type	Description	Required or Optional	Available Version
`visibility`	String	Specifies the visibility behavior of a flag for various user types. • `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item. • `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.	Optional	30.0

Request body for POST

Root XML tag

```
<moderationFlag>
```

JSON

```
{
   "visibility" : "ModeratorsOnly"
}
```

Properties

Name	Type	Description	Available Version
`visibility`	String	Specifies the visibility behavior of a flag for various user types. • `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item. • `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.	30.0

Request parameters for POST

Name	Type	Description	Required or Optional	Available Version
`visibility`	String	Specifies the visibility behavior of a flag for various user types. • `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item. • `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.	Optional	30.0

Request parameters for DELETE

Name	Type	Description	Required or Optional	Available Version
`userId`	String	The ID of the context user. If specified, removes the flag the user previously added to the file. If not specified and the context user is a moderator for the community, all flags on the file are removed.	Optional	30.0

Response body for GET and POST

Moderation Flags

Flagged Files

Files that are flagged in a community.

If the context user is a community moderator, returns the files that have been flagged by community members.

Resource

```
/connect/communities/communityId/chatter/files/moderation
```

Available since release

30.0

Requires Chatter

No

HTTP methods

GET, HEAD

Request parameters for GET

Name	Type	Description	Required or Optional	Available Version
`page`	Integer	Specifies the number of the page you want returned.	Optional	30.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are from 1 through 100. If you don't specify a size, the default is 25.	Optional	30.0
`q`	String	A query string to filter the results. The string must contain at least two characters, not including wildcards. For information about wildcards, see Wildcards.	Optional	33.0

Response body

File Summary

Connect Resources

Information about the context user's organization and settings, and information about the communities the context user is a member of.

Note: These resources have the base URL of `/services/data/v34.0/connect`, not `/services/data/v34.0/chatter`.

Available resources are:

Resource	Description
`/connect`	A directory of the organization and community resources available to the context user.
`/connect/organization`	Information about the context user's organization and settings.
`/connect/communities`	Returns a list of all communities available to the context user.

Connect

A directory of the organization and community resources available to the context user.

Resource

```
/connect
```

Available since version

24.0

Requires Chatter

No

HTTP methods

GET and HEAD

Response body
Connect Directory

Organization

Information about the context user's organization and settings.

Resource

```
/connect/organization
```

Available since version
24.0

Requires Chatter
No

HTTP methods
GET or HEAD

Response body
Organization

SEE ALSO:
News Feed Resources

Dashboard Component Snapshot Resource

A snapshot of a dashboard component.

Use this resource to post a snapshot of a dashboard component to a user, group, or dashboard feed to help other users follow changes in the data.

Resource

```
/connect/dashboards/components/componentId/snapshots
```

Available since version
25.0

Requires Chatter
Yes

HTTP methods
POST

Request body

Root XML tag

```
<dashboardComponentSnapshot>
```

JSON

```
{
   "parentId" : "0F9D00000000I4O",
   "feedItemBody" : {
      "messageSegments" : [
```

```
      {
         "text" : "Sales Heat",
         "type" : "Text"
      }
      ]
   }
}
```

Properties

Name	Type	Description	Available Version
`parentId`	String	A user, group, or dashboard ID to which you are posting the snapshot.	25.0
`filterId1`	String	Optional. A dashboard filter item ID.	25.0
`filterId2`	String	Optional. A second dashboard filter item ID.	25.0
`filterId3`	String	Optional. A third dashboard filter item ID.	25.0
`feedItemBody`	Message Body Input	Contains the text that accompanies the feed item associated with the dashboard component snapshot.	25.0
`runningUserId`	String	Optional. A user ID from which the view of the dashboard will be created, if other than the creator.	25.0

Request parameters

Name	Type	Description
`parentId`	String	A user, group, or dashboard Id to which you are posting the snapshot.
`filterId1`	String.	Optional. A dashboard filter item Id.
`filterId2`	String	Optional. A second dashboard filter item Id.
`filterId3`	String	Optional. A third dashboard filter item Id.
`feedItemText`	String	Text that accompanies the dashboard component snapshot.
`runningUserId`	String	Optional. A user Id from which the view of the dashboard will be created, if other than the creator.

Response body

API versions 23.0-31.0 return: Feed Item Attachment: Dashboard

API versions 32.0 and later return: Dashboard Component Snapshot

Datacloud Resources

Information about Datacloud resources used to purchase Data.com contacts or companies.

Use the Datacloud resources to purchase Data.com contacts or companies and get information about your purchases, including how many credits are available for your use.

Resource	Description
/connect/datacloud/companies/*companyId*	Returns the company fields for the specified `companyId`.
/connect/datacloud/contacts/*contactId*	Returns the contact information for the specified contact.
/connect/datacloud/orders/*orderId*/companies	Returns all the information about all companies in the specified order.
/connect/datacloud/orders/*orderId*/contacts	Returns all the order information for a specific `orderId` for contacts.
/connect/datacloud/orders	Purchases the contacts or companies for the listed IDs. The POST request can have either contact IDs or Company IDs not both. `orderInput`.
/connect/datacloud/usage/*userId*	Returns usage information for a Monthly user or a Listpool user.

Datacloud Company Information

Returns the company fields for the specified `companyId`.

Resource

```
/connect/datacloud/companies/companyId
```

Available version

32.0

Requires Chatter

No

HTTP methods

GET

Response Body

DatacloudCompanyRepresentation

Datacloud Contact Information

Returns the contact information for the specified contact.

Resource

```
/connect/datacloud/contacts/contactId
```

Available version

32.0

Requires Chatter
No

HTTP methods
GET

Response body
DatacloudContactRepresentation

Datacloud OrderId Information

Returns information for a specific `orderId`.

Resource

Order information includes the `orderId`, the number of purchased records for the `orderId`, the URL for the order information, and the URL for a list of the purchased records.

```
/connect/datacloud/orders/orderId
```

Available version
32.0

Requires Chatter
No

HTTP methods
GET

Response body
DatacloudOrderRepresentation

Datacloud Company Order Information

Returns all the information about all companies in the specified order.

The *`orderId`* must be for purchased companies. A request for a contact *`orderId`* fails. Order information includes the *`orderId`*, the number of purchased records for the order, information for the purchased companies, and the URLs for next and previous pages.

Resource

```
/connect/datacloud/orders/orderId/companies
```

Available version
32.0

Requires Chatter
No

HTTP methods
GET

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
`page`	Integer	The page number to be set as the URL for the current page.	Required	32.0
`pageSize`	Integer	The number of the page that you want returned. The default page size is set to 25.	Optional	32.0

Response body for GET

DataCloudCompanyCollectionRepresentation

Datacloud Contact Order Information

Returns all the order information for a specific `orderId` for contacts.

Order information includes the `orderId`, the number of purchased records for the `orderId`, the URL for the order information, and details for the purchased contacts. The `orderId` must be for purchased contacts. A request with an `orderId` for purchased companies fails.

Resource

```
/connect/datacloud/orders/orderId/contacts
```

Available version

32.0

Requires Chatter

No

HTTP methods

GET

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
`page`	Integer	The number of the page that you want returned.	Optional	32.0
`pageSize`	Integer	The number of contacts you want returned on each page. The default page size is 25.	Optional	32.0

Response body for GET

DataCloudContactCollectionRepresentation

Datacloud Orders

Purchases the contacts or companies for the listed IDs. The POST request can have either contact IDs or Company IDs not both. `orderInput`.

Purchase contacts or companies in a single POST request, and specify .

Resource

```
/connect/datacloud/orders
```

Available version

32.0

Requires Chatter

No

HTTP methods

POST

Request body for POST

Root XML tag

```
<DatacloudOrder>
```

JSON example

```
{
   "companyIds":[
      "23456",
      "24548"
   ],
   "userType":"Monthly"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
companyIds	String	The list of companies to purchase. If `contactIds` are included with `companyIds`, your purchase fails.	Required	32.0
contactIds	String	The list of contacts to purchase. We recommend that you include only contacts in your list. If `companyIds` are included with `contactIds`, your purchase fails.	Required	32.0
userType	String	The Data.com user type. • `Monthy`—A user type that's assigned monthly point limits for purchasing Data.com records. Only the assigned user can use monthly points. Points expire at the end of the month. Monthly is the default setting for `DatacloudUserType`. • `Listpool`— A user type that allows users to draw from a pool of points to purchase Data.com records.	Optional—If a user type isn't specified, Monthly is the default.	32.0

Request parameters for POST

Parameter Name	Type	Description	Required or Optional	Available Version
`orderInput`	Object	A request body with a list of contact or company IDs.	Required	32.0

Response body for POST

DatacloudOrderRepresentation

Datacloud Usage

Returns usage information for a Monthly user or a Listpool user.

Resource

```
/connect/datacloud/usage/userId
```

Available version

32.0

Requires Chatter

No

HTTP methods

GET

Response body

DatacloudPurchaseUsageRepresentation

Feed Elements Resources

Information about feed elements. Access, edit, post, search, and delete feed elements. Access a feed element's capabilities, including bundles and their feed elements.

Available resources:

Resource	Description
`/chatter/feed-elements`	Post feed elements and search all feed elements in an organization.
`/chatter/feed-elements/batch`	Post a batch of up to 500 feed elements.
`/chatter/feed-elements/batch/feedElementIds`	Get information about a comma-separated list of up to 500 feed elements. Errors for feed elements that can't be loaded are returned in the results.
`/chatter/feed-elements/feedElementId`	Access, edit, or delete a feed element. Feed items are the only type of feed element that can be edited.

Resource	Description
/chatter/feed-elements/***feedElementId***/capabilities	Access a feed element's set of capabilities.
/chatter/feed-elements/***feedElementId***/capabilities/approval	Access the approval capability of a feed element.
/chatter/feed-elements/***feedElementId***/capabilities/associated-actions	Access the associated actions capability of a feed element.
/chatter/feed-elements/***feedElementId***/capabilities/banner	Access the banner capability of a feed element.
/chatter/feed-elements/***feedElementId***/capabilities/bookmarks	Get information about whether a feed element is bookmarked, or add a bookmark to a feed element.
/chatter/feed-elements/***feedElementId***/capabilities/bundle	Access the bundle capability of a feed element.
/chatter/feed-elements/***feedElementId***/capabilities/bundle/feed-elements	Access all feed elements that are grouped together in a bundle.
/chatter/feed-elements/***feedElementId***/capabilities/canvas	Access the canvas capability of a feed element.
/chatter/feed-elements/***feedElementId***/capabilities/case-comment	Access the case comment capability of a feed element.
/chatter/feed-elements/***feedElementId***/capabilities/chatter-likes	Access information about the chatter likes capability for a feed element.
/chatter/feed-elements/***feedElementId***/capabilities/chatter-likes/items	Access a page of likes for a feed element, or add a like to a feed element.
/chatter/feed-elements/***feedElementId***/capabilities/comments	Access the comments capability for a feed element
/chatter/feed-elements/***feedElementId***/capabilities/comments/items	Access comments for a feed element, or add a comment to a feed element.
/chatter/feed-elements/***feedElementId***/capabilities/content	Access the content capability of a feed element.
/chatter/feed-elements/***feedElementId***/capabilities/dashboard-component-snapshot	Access the dashboard component capability of a feed element.
/chatter/feed-elements/***feedElementId***/capabilities/edit/is-editable-by-me	Indicates whether the context user can edit the feed element. Feed items are the only type of feed element that can be edited.
/chatter/feed-elements/***feedElementId***/capabilities/email-message	Access the email message capability of a feed element.
/chatter/feed-elements/***feedElementId***/capabilities/enhanced-link	Access the enhanced link capability of a feed element.

Resource	Description
/chatter/feed-elements/***feedElementId***/capabilities/link	Access the link capability of a feed element.
/chatter/feed-elements/***feedElementId***/capabilities/moderation	Get information about a feed element's moderation capability, add a moderation flag to a feed element, or delete all moderation flags from a feed element.
/chatter/feed-elements/***feedElementId***/capabilities/origin	Access the origin capability of a feed element. If a feed element has this capability, it was created by a feed action.
/chatter/feed-elements/***feedElementId***/capabilities/poll	Access a poll for a feed element, create a poll, or vote on a poll.
/chatter/feed-elements/***feedElementId***/capabilities/question-and-answers	Access question and answers for a feed element, or update question and answers on a feed element.
/chatter/feed-elements/***feedElementId***/capabilities/recommendations	Access the recommendations capability of a feed element.
/chatter/feed-elements/***feedElementId***/capabilities/record-snapshot	Access the record snapshot capability of a feed element.
/chatter/feed-elements/***feedElementId***/capabilities/topics	Access question and answers for a feed element, or update question and answers on a feed element.
/chatter/feed-elements/***feedElementId***/capabilities/tracked-changes	Access the tracked changes capability of a feed element.

Feed Elements, Post and Search

Post feed elements and search all feed elements in an organization.

Resource

```
/chatter/feed-elements
```

```
/connect/communities/communityId/chatter/feed-elements
```

Available version

31.0

Requires Chatter

Yes

HTTP methods

GET, POST

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
`page`	String	Specifies the page token to be used to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.	Optional	31.0
`pageSize`	Integer	Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.	Optional	31.0
`q`	String	One or more keywords to search for in the feed elements visible to the context user. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.	Required	31.0
`recentCommentCount`	Integer	The maximum number of comments to return with each feed element. The default value is 3.	Optional	31.0
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity.	Optional	31.0

Response body for GET

Feed Element Page

Request body for POST

Feed Item Input

Request parameters for POST

Parameter Name	Type	Description	Required or Optional	Available Version
`feedElementType`	String	Feed elements are the top-level items that a feed contains. Feeds are feed element containers. The only possible value is `FeedItem`.	Required	31.0

Parameter Name	Type	Description	Required or Optional	Available Version
`originalFeedElementId`	String	If this feed element is shared, this is the original feed element 18-character ID.	Optional	31.0
`subjectId`	String	The ID of the parent this feed element is being posted to. This value can be the ID of a user, group, or record, or the string `me` to indicate the context user.	Required	31.0
`text`	String	Text of the feed element.	Required	31.0
`visibility`	String	Specifies the type of users who can see a feed element. • `AllUsers`—Visibility is not limited to internal users. • `InternalUsers`—Visibility is limited to internal users.	Optional	31.0

Response body for POST

Feed Item

Example for POST

This example posts text that mentions a user:

```
POST /chatter/feed-elements

{
   "body" : {
      "messageSegments" : [
         {
            "type" : "Text",
            "text" : "When should we meet for release planning? "
         },
         {
            "type" : "Mention",
            "id" : "005D00000016Qxp"
         }]
       },
   "feedElementType" : "FeedItem",
   "subjectId" : "005D00000016Qxp"
}
```

SEE ALSO:

Action Link Templates

Define Action Links and Post with a Feed Element

Define Action Links in a Template and Post with a Feed Element

Feed Elements, Batch Post

Post a batch of up to 500 feed elements.

Resource

```
/chatter/feed-elements/batch
```

```
/connect/communities/communityId/chatter/feed-elements/batch
```

Available version

32.0

Requires Chatter

Yes

HTTP methods

POST, HEAD

Request body for POST

Batch Collection Input

The rich input for this batch collection is a Feed Item Input

Response body for POST

Batch Results

The Batch Results response body contains Feed Item response bodies corresponding to each of the request bodies. The Feed Item response bodies are returned in the same order as the request bodies.

The request returns an HTTP 200 status code unless an error occurs that affects the entire request (such as a parsing failure). If an individual request body causes an error, the Batch Results Item `statusCode` property contains the appropriate HTTP status code.

Post a batch of feed elements

This example uses the `/chatter/feed-elements/batch` resource to post a batch of feed elements. This example uses cURL.

```
curl -H "X-PrettyPrint: 1" -F
'json={"inputs":[{"richInput":{"subjectId":"me","body":{"messageSegments":[{"type":"Text","text":"Post
 Number
1"}]},"feedElementType":"FeedItem"}},{"richInput":{"subjectId":"me","body":{"messageSegments":[{"type":"Text","text":"Post
 Number 2"}]},"feedElementType":"FeedItem"}}];type=application/json' -X POST
https://instance_name/services/data/v34.0/chatter/feed-elements/batch -H 'Authorization:
 OAuth 00DD00000007HoR!...' --insecure
```

Here's the same example showing just the request body:

```
{
  "inputs": [
    {
      "richInput": {
        "subjectId": "me",
        "body": {
          "messageSegments": [
            {
              "type": "Text",
              "text": "Post Number 1"
            }
          ]
```

```
        },
        "feedElementType": "FeedItem"
      }
    },
    {
      "richInput": {
        "subjectId": "me",
        "body": {
          "messageSegments": [
            {
              "type": "Text",
              "text": "Post Number 2"
            }
          ]
        },
        "feedElementType": "FeedItem"
      }
    }
  ]
}
```

Post a batch of feed elements with binary files

This example uses the `/chatter/feed-elements/batch` resource to post a batch of feed elements, uploads two binary files, and attach them to the feed elements in the batch. You must include the information about the binary parts in the request body. This example uses cURL.

```
curl -H "X-PrettyPrint: 1" -F 'json={"inputs":[{"binaryPartNames":["bin1"],
"richInput":{"subjectId":"me","body":{
"messageSegments":[{"type":"Text","text":"Please accept this receipt"}]},"capabilities":
 {"content" : {"description": "Receipt for expenses",
"title":"receipt.pdf"}},"feedElementType":"FeedItem"}},
{"binaryPartNames":["bin2"],"richInput":{"subjectId":"me","body":{"messageSegments":[{"type":"Text","text":"Post
 Number 2"}]},
"capabilities": {"content" : {"description": "Receipt for expenses",
"title":"receipt2.pdf"}},"feedElementType":"FeedItem"}}]};type=application/json'
-F "bin1=@/Users/jsmith/Desktop/receipt.pdf;type=application/octet-stream" -F
"bin2=@/Users/jsmith/Desktop/receipt2.pdf;type=application/octet-stream"
-X POST https://instance_name/services/data/v34.0/chatter/feed-elements/batch
-H 'Authorization: OAuth 00DD00000007HoR44QATPNzhzYEJBfU' --insecure
```

This is the same example formatted to show the request body parts:

```
POST /services/data/v34.0/chatter/feed-elements/batch HTTP/1.1
Authorization: OAuth 00DRR0000000N0g!...
User-Agent: Jakarta Commons-HttpClient/3.0.1
Host: instance_name
Content-Length: 845
Content-Type: multipart/form-data; boundary=a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq
Accept: application/json

--a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq
Content-Disposition: form-data; name="json"
Content-Type: application/json; charset=UTF-8

{
```

```
  "inputs": [
    {
      "binaryPartNames": [
        "bin1"
      ],
      "richInput": {
        "subjectId": "me",
        "body": {
          "messageSegments": [
            {
              "type": "Text",
              "text": "Please accept this receipt"
            }
          ]
        },
        "capabilities": {
          "content": {
            "description": "Receipt for expenses",
            "title": "receipt.pdf"
          }
        },
        "feedElementType": "FeedItem"
      }
    },
    {
      "binaryPartNames": [
        "bin2"
      ],
      "richInput": {
        "subjectId": "me",
        "body": {
          "messageSegments": [
            {
              "type": "Text",
              "text": "Post Number 2"
            }
          ]
        },
        "capabilities": {
          "content": {
            "description": "Receipt for expenses",
            "title": "receipt2.pdf"
          }
        },
        "feedElementType": "FeedItem"
      }
    }
  ]
}

--a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq
Content-Disposition: form-data; name="bin1"; filename="receipt.pdf"
Content-Type: application/octet-stream; charset=ISO-8859-1
```

```
...contents of receipt.pdf...


--a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq
Content-Disposition: form-data; name="bin2"; filename="receipt2.pdf"
Content-Type: application/octet-stream; charset=ISO-8859-1

...contents of receipt2.pdf...

--a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq--
```

Note: The binary part names in the `binaryPartNames` properties must match the binary part names in the `name` properties of the binary request parts.

SEE ALSO:

Using Chatter REST API Inputs

Batch Resource

Feed Elements, Batch Get

Get information about a comma-separated list of up to 500 feed elements. Errors for feed elements that can't be loaded are returned in the results.

Resource

```
/chatter/feed-elements/batch/feedElementIds
```

```
/connect/communities/communityId/chatter/feed-elements/batch/feedElementIds
```

Available version

31.0

Requires Chatter

Yes

HTTP methods

GET

Response body for GET

Batch Results

Feed Element

Access, edit, or delete a feed element. Feed items are the only type of feed element that can be edited.

Resource

```
/chatter/feed-elements/feedElementId
```

```
/connect/communities/communityId/chatter/feed-elements/feedElementId
```

Available version

31.0

Requires Chatter
Yes

HTTP methods
GET, DELETE, PATCH

PATCH is supported in version 34.0 and later.

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
`elementsPerBundle`	Integer	The maximum number of feed elements per bundle. The default and maximum value is 10.	Optional	31.0
`recentCommentCount`	Integer	The maximum number of comments to return with each feed element. The default value is 3.	Optional	31.0

Request body for PATCH
Feed Item Input

Response body for PATCH
Feed Item

Response body for GET
Generic Feed Element or Feed Item

Response body for DELETE
204: Successful Delete

SEE ALSO:

Edit a Feed Post

Edit a Question Title and Post

Feed Elements Capabilities

Access a feed element's set of capabilities.

Resource

```
/chatter/feed-elements/feedElementId/capabilities
```

```
/connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities
```

Available version
31.0

Requires Chatter
Yes

HTTP methods
GET

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
`elementsPerBundle`	Integer	The maximum number of feed elements per bundle. The default and maximum value is 10.	Optional	31.0
`recentCommentCount`	Integer	The maximum number of comments to return with each feed element. The default value is 3.	Optional	31.0

Response body for GET

Feed Element Capabilities

Feed Elements Capability, Approval

Access the approval capability of a feed element.

Resource

```
/chatter/feed-elements/feedElementId/capabilities/approval
```

```
/connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities/approval
```

Available version

32.0

Requires Chatter

Yes

HTTP methods

GET, HEAD

Response bodies for GET

Approval Capability

If the feed element doesn't support this capability, the response is 404: Not Found.

Feed Elements Capability, Associated Actions

Access the associated actions capability of a feed element.

Resource

```
/chatter/feed-elements/feedElementId/capabilities/associated-actions
```

```
/connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities/associated-actions
```

Available version

33.0

Requires Chatter

Yes

HTTP methods
GET, HEAD

Response bodies for GET
Associated Actions Capability

If the feed element doesn't support this capability, the response is 404: Not Found.

Feed Elements Capability, Banner

Access the banner capability of a feed element.

Resource

```
/chatter/feed-elements/feedElementId/capabilities/banner
```

```
/connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities/banner
```

Available version
32.0

Requires Chatter
Yes

HTTP methods
GET, HEAD

Response bodies for GET
Banner Capability

If the feed element doesn't support this capability, the response is 404: Not Found.

Feed Elements Capability, Bookmarks

Get information about whether a feed element is bookmarked, or add a bookmark to a feed element.

Resource

```
/chatter/feed-elements/feedElementId/capabilities/bookmarks
```

```
/connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities/bookmarks
```

Available version
32.0

Requires Chatter
Yes

HTTP methods
GET, PATCH

Request body for PATCH

Root XML tag

```
<bookmarksCapability>
   <isBookmarkedByCurrentUser>true</isBookmarkedByCurrentUser>
</bookmarksCapability>
```

JSON example

```
{
   "isBookmarkedByCurrentUser": true
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`isBookmarkedByCurrentUser`	Boolean	Specifies if the feed element should be bookmarked for the user (`true`) or not (`false`).	Required	32.0

Request parameter for PATCH

Parameter Name	Type	Description	Required or Optional	Available Version
`isBookmarkedByCurrentUser`	Boolean	Specifies if the feed element should be bookmarked for the user (`true`) or not (`false`).	Required	32.0

Response body for GET, PATCH

Bookmarks Capability

If the feed element doesn't support this capability, the response is 404: Not Found.

Feed Elements Capability, Bundle

Access the bundle capability of a feed element.

Resource

```
/chatter/feed-elements/feedElementId/capabilities/bundle
```

```
/connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities/bundle
```

Available version

31.0

Requires Chatter

Yes

HTTP methods

GET, HEAD

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
`elementsPerBundle`	Integer	The maximum number of feed elements per bundle. The default and maximum value is 10.	Optional	31.0
`recentCommentCount`	Integer	The maximum number of comments to return with each feed element. The default value is 3.	Optional	31.0

Response bodies for GET

One of these response bodies:

- Generic Bundle Capability
- Tracked Change Bundle Capability

If the feed element doesn't support this capability, the response is 404: Not Found.

Feed Elements Capability, Bundle Elements

Access all feed elements that are grouped together in a bundle.

Note: Bundled posts contain feed-tracked changes. In Salesforce1 downloadable apps, bundled posts are in record feeds only.

Resource

```
/chatter/feed-elements/feedElementId/capabilities/bundle/feed-elements
```

```
connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities/bundle/feed-elements
```

Available version

31.0

Requires Chatter

Yes

HTTP methods

GET

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
`elementsPerBundle`	Integer	The maximum number of feed elements per bundle. The default and maximum value is 10.	Optional	31.0
`page`	String	Specifies the page token to be used to view a page of information. Page tokens are returned as part of the response	Optional	31.0

Parameter Name	Type	Description	Required or Optional	Available Version
		class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.		
`pageSize`	Integer	Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.	Optional	31.0
`recentCommentCount`	Integer	The maximum number of comments to return with each feed element. The default value is 3.	Optional	31.0

Response body for GET

Feed Element Page

If the feed element doesn't support this capability, the response is 404: Not Found.

Feed Elements Capability, Canvas

Access the canvas capability of a feed element.

Resource

```
/chatter/feed-elements/feedElementId/capabilities/canvas
```

```
/connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities/canvas
```

Available version

32.0

Requires Chatter

Yes

HTTP methods

GET, HEAD

Response bodies for GET

Canvas Capability

If the feed element doesn't support this capability, the response is 404: Not Found.

Feed Elements Capability, Case Comment

Access the case comment capability of a feed element.

Resource

```
/chatter/feed-elements/feedElementId/capabilities/case-comment
```

```
/connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities/case-comment
```

Available version
32.0

Requires Chatter
Yes

HTTP methods
GET, HEAD

Response bodies for GET
Case Comment Capability

If the feed element doesn't support this capability, the response is 404: Not Found.

Feed Elements Capability, Chatter Likes

Access information about the chatter likes capability for a feed element.

Resource

```
/chatter/feed-elements/feedElementId/capabilities/chatter-likes
```

```
/connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities/chatter-likes
```

Available version
32.0

Requires Chatter
Yes

HTTP methods
GET

Response
Chatter Likes Capability

If the feed element doesn't support this capability, the response is 404: Not Found.

Feed Elements Capability, Chatter Likes Items

Access a page of likes for a feed element, or add a like to a feed element.

Resource

```
/chatter/feed-elements/feedElementId/capabilities/chatter-likes/items
```

```
/connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities/chatter-likes/items
```

Available version
32.0

Requires Chatter
Yes

HTTP methods
GET, POST

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
`page`	Integer	Specifies the number of the page you want returned. If you don't specify a value, the first page is returned.	Optional	32.0
`pageSize`	Integer	Specifies the number of likes per page. Valid values are between 1 and 100. If you don't specify a size, the default is 25.	Optional	32.0

Response body for POST

Like

If the feed element doesn't support this capability, the response is 404: Not Found.

Response body for GET

Like Page

If the feed element doesn't support this capability, the response is 404: Not Found.

Feed Elements Capability, Comments

Access the comments capability for a feed element

Resource

```
/chatter/feed-elements/feedElementId/capabilities/comments
```

```
/connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities/comments
```

Available version

32.0

Requires Chatter

Yes

HTTP methods

GET

Response body for GET

Comments Capability

If the feed element doesn't support this capability, the response is 404: Not Found.

Feed Elements Capability, Comments Items

Access comments for a feed element, or add a comment to a feed element.

For additional information about posting a comment, see Using POST, PATCH, or PUT for Input.

To upload a binary file to attach to a comment, you must send it in a multipart/form-data request. To send the text of the comment, you can choose to include a JSON or XML rich input body part in the multipart/form-data request. Alternately, you can choose to pass

the information in request parameter parts. For information about how to create the multipart/form-data message, see Uploading Binary Files.

Resource

```
/chatter/feed-elements/feedElementId/capabilities/comments/items
```

```
/connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities/comments/items
```

Available version

32.0

Requires Chatter

Yes

HTTP methods

GET, POST

Request body for POST

Root XML tag

`<comment>`

Properties

Name	Type	Description	Available Version
`attachment`	Depends on type.	Optional. Values are: • Attachment Input: Existing Content • Attachment Input: New File Upload **Important:** As of version 32.0, use the `capabilities` property.	24.0-31.0
`body`	Message Body Input	The body of the comment. To edit this property in a comment, make a PATCH request to `/chatter/comments/commentId`. Editing comments is supported in version 34.0 and later.	23.0
`capabilities`	Comment Capabilities Input	Capabilities of the comment, such as a file attachment	32.0

Request parameters for POST

Parameter Name	Type	Description	Required or Optional	Available Version
`text`	String	The comment text.	Required	32.0

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
`page`	String	A URL to the next page of comments.	Optional	32.0
`pageSize`	Integer	The number of comments per page. The default value is 3.	Optional	32.0

Response body for GET

Comment Page

If the feed element doesn't support this capability, the response is 404: Not Found.

Response body for POST

Comment

If the feed element doesn't support this capability, the response is 404: Not Found.

Feed Elements Capability, Content

Access the content capability of a feed element.

Resource

```
/chatter/feed-elements/feedElementId/capabilities/content
```

```
/connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities/content
```

Available version

32.0

Requires Chatter

Yes

HTTP methods

GET, HEAD

Response bodies for GET

Content Capability

If the feed element doesn't support this capability, the response is 404: Not Found.

Feed Elements Capability, Dashboard Component Snapshot

Access the dashboard component capability of a feed element.

Resource

```
/chatter/feed-elements/feedElementId/capabilities/dashboard-component-snapshot
```

```
/connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities/dashboard-component-snapshot
```

Available version

32.0

Requires Chatter
Yes

HTTP methods
GET, HEAD

Response bodies for GET
Dashboard Component Snapshot Capability

If the feed element doesn't support this capability, the response is 404: Not Found.

Feed Elements Capability, Editability

Indicates whether the context user can edit the feed element. Feed items are the only type of feed element that can be edited.

Resource

```
/chatter/feed-elements/feedElementId/capabilities/edit/is-editable-by-me
```

```
/connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities/edit/is-editable-by-me
```

Available version
34.0

Requires Chatter
Yes

HTTP methods
GET

Response body for GET
Feed Entity Is Editable

Feed Elements Capability, Email Message

Access the email message capability of a feed element.

Resource

```
/chatter/feed-elements/feedElementId/capabilities/email-message
```

```
/connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities/email-message
```

Available version
32.0

Requires Chatter
Yes

HTTP methods
GET, HEAD

Response bodies for GET
Email Message Capability

If the feed element doesn't support this capability, the response is 404: Not Found.

Feed Elements Capability, Enhanced Link

Access the enhanced link capability of a feed element.

Resource

```
/chatter/feed-elements/feedElementId/capabilities/enhanced-link
```

```
/connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities/enhanced-link
```

Available version

32.0

Requires Chatter

Yes

HTTP methods

GET, HEAD

Response bodies for GET

Enhanced Link Capability

If the feed element doesn't support this capability, the response is 404: Not Found.

Feed Elements Capability, Link

Access the link capability of a feed element.

Resource

```
/chatter/feed-elements/feedElementId/capabilities/link
```

```
/connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities/link
```

Available version

32.0

Requires Chatter

Yes

HTTP methods

GET, HEAD

Response bodies for GET

Link Capability

If the feed element doesn't support this capability, the response is 404: Not Found.

Feed Elements Capability, Moderation

Get information about a feed element's moderation capability, add a moderation flag to a feed element, or delete all moderation flags from a feed element.

Resource

```
/connect/communities/communityId/chatter/feed-elements
/feedElementId/capabilities/moderation
```

Available version
31.0

Requires Chatter
Yes

HTTP methods
GET, POST, DELETE, HEAD

Request parameters for GET and POST

Parameter Name	Type	Description	Required or Optional	Available Version
`visibility`	String	Specifies the visibility behavior of a flag for various user types. One of these values: • `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item. • `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.	Optional	31.0

Request body for POST

Root XML tag
`<moderationFlag>`

JSON

```
{
   "visibility" : "ModeratorsOnly"
}
```

Properties

Name	Type	Description	Available Version
`visibility`	String	Specifies the visibility behavior of a flag for various user types. • `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item. • `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.	30.0

Response body for GET and POST

Moderation Capability

If the feed element doesn't support this capability, the response is 404: Not Found.

Feed Elements Capability, Origin

Access the origin capability of a feed element. If a feed element has this capability, it was created by a feed action.

Resource

```
/chatter/feed-elements/feedElementId/capabilities/origin
```

```
/connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities/origin
```

Available version

33.0

Requires Chatter

Yes

HTTP methods

GET, HEAD

Response bodies for GET

Origin Capability

If the feed element doesn't support this capability, the response is 404: Not Found.

Feed Elements Capability, Poll

Access a poll for a feed element, create a poll, or vote on a poll.

Resource

```
/chatter/feed-elements/feedElementId/capabilities/poll
```

```
/connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities/poll
```

Available version

32.0

Requires Chatter

Yes

HTTP methods

GET, PATCH

Request body for PATCH

XML example

```
<pollCapability>
   <pollChoices>
      <choice>Monday</choice>
      <choice>Tuesday</choice>
   </pollChoices>
</pollCapability>
```

JSON example

```
{
   "choices" : [
      "Monday", "Tuesday"
   ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`choices`	String[]	The choices used to create a new poll. You must specify 2–10 poll choices for each poll.	Required for creating a poll	32.0
`myChoiceId`	String	ID of an existing choice on the feed poll. Used to vote on an existing poll.	Required for voting on a poll	32.0

Request parameters for PATCH

Parameter Name	Type	Description	Required or Optional	Available Version
`myChoiceId`	String	A poll choice ID.	Required	32.0

Response body for GET, PATCH

Poll Capability

If the feed element doesn't support this capability, the response is 404: Not Found.

Feed Elements Capability, Question and Answers

Access question and answers for a feed element, or update question and answers on a feed element.

Resource

```
/chatter/feed-elements/feedElementId/capabilities/question-and-answers
```

```
/connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities/question-and-answers
```

Available version

32.0

Requires Chatter

Yes

HTTP methods

GET, PATCH

Request body for PATCH

Root XML tag

```
<questionAndAnswersCapability>
   <bestAnswerId>0D7D00000000lMAKAY</bestAnswerId>
</questionAndAnswersCapability>
```

JSON example

```
{
   "bestAnswerId": "0D7D00000000lMAKAY"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`bestAnswerId`	String	A comment ID to use as a best answer for a question feed element. The best answer comment must already exist on the question feed element.	Required to PATCH the best answer. Not supported in POST requests.	32.0
`questionTitle`	String	Title for a question feed element. To edit the title of a question, make a PATCH request to `/chatter/feed-elements/`***`feedElementId`***. Editing question titles is supported in version 34.0 and later.	Required to POST or PATCH a question.	32.0

Request parameters for PATCH

Parameter Name	Type	Description	Required or Optional	Available Version
`bestAnswerId`	String	A comment ID to use as a best answer for a question feed element. The best answer comment must already exist on the question feed element.	Required	32.0

Response body for GET, PATCH

Question and Answers Capability

If the feed element doesn't support this capability, the response is 404: Not Found.

Feed Elements Capability, Recommendations

Access the recommendations capability of a feed element.

Resource

```
/chatter/feed-elements/feedElementId/capabilities/recommendations
```

```
/connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities/recommendations
```

Available version
32.0

Requires Chatter
Yes

HTTP methods
GET, HEAD

Response bodies for GET
Recommendations Capability

If the feed element doesn't support this capability, the response is 404: Not Found.

Feed Elements Capability, Record Snapshot

Access the record snapshot capability of a feed element.

Resource

```
/chatter/feed-elements/feedElementId/capabilities/record-snapshot
```

```
/connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities/record-snapshot
```

Available version
32.0

Requires Chatter
Yes

HTTP methods
GET, HEAD

Response bodies for GET
Record Snapshot Capability

If the feed element doesn't support this capability, the response is 404: Not Found.

Feed Elements Capability, Topics

Get, delete, and post topics associated with a feed element.

Resource

```
/chatter/feed-elements/feedElementId/capabilities/topics
```

```
/connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities/topics
```

Available version
32.0

Requires Chatter
Yes

HTTP methods
GET, DELETE, POST

Request body for POST

Root XML tag
`<feedItemTopic>`

JSON
The following example attaches a new topic to a feed item:

```
{ "topicName": "W3C"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`topicId`	String	ID of an existing topic	Required for existing topics if the `topicName` property isn't used	29.0
`topicName`	String	Name of a new or existing topic, up to 150 characters	Required for new topics; required for existing topics if the `topicId` property isn't used	29.0

Request parameters for POST

Parameter Name	Type	Description	Required or Optional	Available Version
`topicId`	String	ID of an existing topic.	Required for existing topics if the `topicName` property isn't used.	32.0
`topicName`	String	Name of a new or existing topic, up to 150 characters.	Required for new topics; required for existing topics if the `topicId` property isn't used.	32.0

Request parameters for DELETE

Parameter Name	Type	Description	Required or Optional	Available Version
`topicId`	String	An existing topic ID.	Required	32.0

Response body for POST

Topic

If the feed element doesn't support this capability, the response is 404: Not Found.

Response body for GET

Topics Capability

If the feed element doesn't support this capability, the response is 404: Not Found.

Feed Elements Capability, Tracked Changes

Access the tracked changes capability of a feed element.

Resource

```
/chatter/feed-elements/feedElementId/capabilities/tracked-changes
```

```
/connect/communities/communityId/chatter/feed-elements/feedElementId/capabilities/tracked-changes
```

Available version

32.0

Requires Chatter

Yes

HTTP methods

GET, HEAD

Response bodies for GET

Tracked Changes Capability

Feed Items Resources

Information about feed items. Access, search, delete, bookmark, and like feed items. Access and post comments. Access and vote in polls. And access, add, and remove topics.

Important: Feed Item Resources aren't available in version 32.0 and later. In version 32.0 and later, use Feed Elements Resources.

Feed-Items Batch

A batch of feed items. Get information about multiple feed items in one request.

Get information about the comma-separated list of up to 500 feed items. Errors for feed items that couldn't be loaded are returned in the results.

Resource

```
/chatter/feed-items/batch/feedItemIds
```

Available versions

29.0–31.0

Important: In version 32.0 and later, use Feed Elements, Batch Get.

Requires Chatter
Yes

HTTP methods
GET, HEAD

Response body for GET
Batch Results

Feed-Items Query

A paginated collection of feed items. Use this resource to search for feed items.

The query parameter (q) is required. All types of feed items are searched, including tracked changes. A search returns feed items with the search term in the feed item body, as well as feed items that have comments that contain the search term. You can also use wildcards with the query parameter.

Resource

```
/chatter/feed-items?q=searchTerm
```

Available versions
23.0–31.0

Important: In version 32.0 and later, use Feed Elements, Post and Search.

Requires Chatter
Yes

HTTP methods
GET, HEAD

Request parameters

Parameter Name	Type	Description	Required or Optional	Available Since
page	String	A generated token that indicates the view of feed items in the feed.	Optional	23.0–31.0
pageSize	Integer	Specifies the number of items per page. Valid values are from 1 through 100. If you don't specify a size, the default is 25.	Optional	23.0–31.0
q	String	Specifies the string to search. The search string must contain at least two characters, not including any wildcards. For more information about wildcards, see Wildcards. The query parameter searches all types of feed items, including tracked changes. The search returns feed items with the search term in the feed item body, as well as feed items that have comments that contain the search term.	Required	23.0–31.0
recentCommentCount	Integer	Maximum number of comments to include per feed item. The default value is 3.	Optional	29.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Since
`sort`	String	The sort order of the results. Values are: • `CreatedDateDesc` • `LastModifiedDateDesc` The default value is `CreatedDateDesc`.	Optional	28.0–31.0

Response body

Feed Item Page

Feed-Items, Specific Feed Item

Access, delete, or bookmark a specific feed item.

Resource

```
/chatter/feed-items/feedItemId
```

Available versions

23.0–31.0

Important: In version 32.0 and later, use Feed Element.

Requires Chatter

Yes

HTTP methods

GET, DELETE, PATCH, or HEAD

Request body for PATCH

The root XML tag is `<feedItem>`.

Parameter Name	Type	Description
`isBookmarkedByCurrentUser`	Boolean	Specify `true` to add the current feed item to the list of bookmarked feed items for the current user. Specify `false` to remove this feed item from the list of bookmarked feed items for the current user.

Request parameter for PATCH

Parameter Name	Type	Description
`isBookmarkedByCurrentUser`	Boolean	Specify `true` to add the current feed item to the list of bookmarked feed items for the current user. Specify `false` to

Parameter Name	Type	Description
		remove this feed item from the list of bookmarked feed items for the current user.

Response body for GET, PATCH, or HEAD

Feed Item

Example

This PATCH request bookmarks a feed item:

```
/chatter/feed-items/0D5D0000000JvckKAC?isBookmarkedByCurrentUser=true
```

Feed Items Comments

Access comments for a feed item or post a comment.

The default number of comments to retrieve is 3. To retrieve more comments increase the size of the `pageSize` parameter.

For additional information about posting a comment, see Using POST, PATCH, or PUT for Input.

To upload a binary file to attach to a comment, you must send it in a multipart/form-data request. To send the text of the comment, you can choose to include a JSON or XML rich input body part in the multipart/form-data request. Alternately, you can choose to pass the information in request parameter parts. For information about how to create the multipart/form-data message, see Uploading Binary Files.

Comments can contain up to 5000 characters.

Resource

```
/chatter/feed-items/feedItemId/comments
```

Available since version

23.0–31.0

Important: In version 32.0 and later, use Feed Elements Capability, Comments Items.

Requires Chatter

Yes

HTTP methods

GET, POST or HEAD

Request body for POST

Root XML tag

`<comment>`

Properties

Name	Type	Description	Available Version
`attachment`	Depends on type.	Optional. Values are: • Attachment Input: Existing Content	24.0-31.0

Name	Type	Description	Available Version
		• Attachment Input: New File Upload **Important:** As of version 32.0, use the `capabilities` property.	
`body`	Message Body Input	The body of the comment. To edit this property in a comment, make a PATCH request to `/chatter/comments/`***`commentId`***. Editing comments is supported in version 34.0 and later.	23.0
`capabilities`	Comment Capabilities Input	Capabilities of the comment, such as a file attachment	32.0

Request parameters for POST

Parameters for attaching an existing file

Parameter Name	Type	Description	Required or Optional	Available Version
`attachmentType`	String	The type of attachment. Must contain the value `ExistingContent`.	Required	27.0–31.0
`contentDocumentId`	String	ID of an existing file to be posted.	Required	24.0–31.0
`text`	String	Text of the comment.	Optional	23.0–31.0

Parameters for posting text

Parameter Name	Type	Description	Required or Optional	Available Version
`text`	String	The comment text.	Required	23.0–31.0

Request parameters for GET or HEAD

Parameter Name	Type	Description	Required or Optional	Available Version
`page`	String	A generated token that indicates the view of feed items in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`.	Optional	23.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
`pageSize`	Integer	Specifies the number of items per page. Valid values are from 1 through 100. If you don't specify a size, the default is 25. This parameter specifies the number of comments per page.	Optional	23.0–31.0

Response body for GET or HEAD

Comment Page

Response body for POST

Comment

Example: Posting a Comment

This POST request uses a request parameter to post a comment to a feed item:

```
/chatter/feed-items/0D5D0000000JvckKAC/comments?text=I+completely+agree
```

Example: Posting a Comment and Attaching an Existing File

This POST request posts a comment and attaches a file that has already been uploaded to the user's organization:

```
/chatter/feed-items/0D5D0000000JvckKAC/comments?text=yes&attachmentType=ExistingContent&contentDocumentId=069D00000001FHF
```

Example: Posting a Comment and Uploading and Attaching a New File

See Uploading Binary Files on page 10.

Feed-Items Likes

Used to access likes for a specific feed item, or to add a like to a specific feed item.

Resource

```
/chatter/feed-items/feedItemId/likes
```

Available versions

23.0–31.0

Important: In version 32.0 and later, use Feed Elements Capability, Chatter Likes Items.

Requires Chatter

Yes

HTTP methods

GET, POST or HEAD

Request parameters for GET or HEAD

Note: There aren't any request parameters for POST.

Parameter Name	Type	Description
`page`	Integer	Specifies the number of the page you want returned.
`pageSize`	Integer	Specifies the number of items per page. Valid values are from 1 through 100. If you don't specify a size, the default is 25.

Response body for GET or HEAD

Like Page

Response body for POST

Like

SEE ALSO:

Like a Feed Item

Feed-Items Polls

Used to access or vote in an existing poll.

Resource

```
/chatter/feed-items/feedItemId/poll
```

The *feedItemId* must have the type `PollPost`.

Available versions

26.0–31.0

Important: In version 32.0 and later, use Feed Elements Capability, Poll.

Requires Chatter

Yes

HTTP methods

GET, PATCH or HEAD

Use PATCH to either vote in an existing poll, or change the current user's vote.

Request body for PATCH

Root XML tag

```
<feedPoll>
```

Properties

Name	Type	Description	Available Version
`myChoiceId`	String	ID of an existing choice on the feed poll. Used to specify the ID of the poll choice that the current user is voting on.	26.0–31.0

Request parameters for PATCH

Name	Type	Description	Available Version
`myChoiceId`	String	ID of an existing choice on the feed poll. Used to specify the ID of the poll choice that the current user is voting on.	26.0–31.0

Response body

Feed Item Attachment: Poll

Feed-Items Topics

The topics associated with a specific feed item. Associate a new or existing topic with a specific feed item. Remove a topic from a specific feed item. Only users with the "Assign Topics" permission can add or remove existing topics from feed items. Only users with the "Create Topics" permission can add new topics to feed items.

Resource

```
/chatter/feed-items/feedItemId/topics
```

Available versions

28.0–31.0

Important: In version 32.0 and later, use Feed Elements Capability, Topics.

Requires Chatter

Yes

HTTP methods

DELETE, POST

Request parameter for DELETE

Parameter Name	Type	Description	Required or Optional	Available Version
`topicId`	String	ID of an existing topic	Required	28.0–31.0

Request body for POST

Root XML tag

```
<feedItemTopic>
```

JSON

The following example attaches a new topic to a feed item:

```
{ "topicName": "W3C"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`topicId`	String	ID of an existing topic	Required for existing topics if the `topicName` property isn't used	29.0
`topicName`	String	Name of a new or existing topic, up to 150 characters	Required for new topics; required for existing topics if the `topicId` property isn't used	29.0

Request parameters for POST

Parameter Name	Type	Description	Required or Optional	Available Version
`topicId`	String	ID of an existing topic	Required for existing topics if the `topicName` parameter isn't used	28.0–31.0
`topicName`	String	Name of a new or existing topic, up to 150 characters	Required for new topics; required for existing topics if the `topicId` parameter isn't used	28.0–31.0

Response body for DELETE

204: Successful Delete

Response body for POST

Topic

Example for POST

This example passes the ID of an existing topic in the `topidId` parameter to associate it with a feed item:

```
/chatter/feed-items/0D5D0000000JmiB/topics?topicId=0TOD0000000090r
```

Feeds Resources

Feeds are made up of feed items. There are many feed types. Each feed type has an algorithm that determines which feed items it contains.

The news feed, record feed, and user profile feed support GET and POST requests. All other feeds support GET requests.

IN THIS SECTION:

Bookmarks Feed Resources

A feed containing feed elements saved as bookmarks by the context user.

Communities Home Feed Resource
Returns a list of all feed elements associated with any managed topic in the community.

Communities Moderation Feed Resources
A feed containing feed elements and comments that have been flagged for moderation in a community.

Company Feed Resources
A feed containing all feed elements and comments from an entire company.

Favorites Feed Resources
A feed containing favorites saved by the context user.

Feeds Resource
A list of all of the feeds the context user is able to view.

Files Feed Resources
A feed containing all the feed elements that contain files posted by people or groups that the context user follows.

Filter Feed Resources
The filter feed is the news feed filtered to contain feed elements whose parent is a specified object type.

Groups Feed Resources
A feed containing all feed elements from all groups the current user either owns or is a member of.

News Feed Resources
A feed containing feed elements of interest to the context user. You can post feed items to the news feed.

People Feed Resources
All feed elements posted by all of the people the context user follows.

Record Feed Resource
The feed of the specified record, which could be a group, person, object, file and so on. You can post feed items to the record feed.

To Feed Resources
A feed containing all @mentions of the context user and posts others have made to the context user's feed.

Topics Feed Resources
The feed of the specified topic.

User Profile Feed Resources
A feed containing actions by the current user on records that can be tracked in a feed. This feed is different than `/chatter/feeds/news`, which returns more elements, including posts, group updates, and record updates. You can post feed items to the user-profile feed.

SEE ALSO:
Working with Feeds and Feed Elements

Bookmarks Feed Resources

A feed containing feed elements saved as bookmarks by the context user.

To bookmark an item, use PATCH with the Feed Elements Capability, Bookmarks resource and the `isBookmarkedByCurrentUser` parameter. For example:

```
/chatter/feed-elements/0D5D0000000Ir6zKAC/capabilities/bookmarks?isBookmarkedByCurrentUser=true
```

Available resources are:

Resource	Description
`/chatter/feeds/bookmarks/me`	Returns the URL for the bookmarks feed for the context user.
`/chatter/feeds/bookmarks/me/feed-elements`	Returns all feed elements bookmarked by the context user.

Bookmarks Feed URL

Returns the URL for the bookmarks feed for the context user.

Resource

```
/chatter/feeds/bookmarks/userId
```

```
/connect/communities/communityId/chatter/feeds/bookmarks/userId
```

Note: *userId* must be the ID of the context user or the alias `me`.

Available since version

24.0

Requires Chatter

Yes

HTTP methods

GET or HEAD

Request parameter

Parameter Name	Type	Description
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.

Response body

Feed

Bookmarks Feed Elements

Returns all feed elements bookmarked by the context user.

Resource

```
/chatter/feeds/bookmarks/userId/feed-elements
```

```
/connect/communities/communityId/chatter/feeds/bookmarks/userId/feed-elements
```

Note: *userId* must be the ID of the context user or the alias me.

Available since version
24.0

Requires Chatter
Yes

HTTP methods
GET or HEAD

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
density	String	The density of the feed. One of these values: • AllUpdates—Displays all updates from people and records the user follows and groups the user is a member of. • FewerUpdates—Displays all updates from people and records the user follows and groups the user is a member of, but hides some system-generated updates from records. The default value is AllUpdates.	Optional	31.0
elementsPerBundle	Integer	Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3. Note: Bundled posts contain feed-tracked changes. In Salesforce1 downloadable apps, bundled posts are in record feeds only.	Optional	31.0
page	String	A generated token that indicates the view of feed elements in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as nextPageURL, for example: `"nextPageUrl": "/services/data/v34.0/chatter/feeds/news/005D0000001GLowIAN/feed-elements?page=2011-03-30T17:34:50Z,0D5D0000000DSv4KAG"`	Optional	31.0
pageSize	Integer	Specifies the number of elements per page. Valid values are between 1 and 100. If you don't specify a size, the default is 25.	Optional	31.0

Parameter Name	Type	Description	Required or Optional	Available Version
`q`	String	One or more keywords to search for in the body and comments of feed elements in this feed. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.	Optional	31.0
`recentCommentCount`	Integer	Maximum number of comments to include per feed element. The default value is 3. The maximum value is 25.	Optional	31.0
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.	Optional	31.0

Response body

Feed Element Page

Bookmarks Feed Items

Returns all feed items bookmarked by the context user.

Resource

```
/chatter/feeds/bookmarks/me/feed-items
```

OR

```
/chatter/feeds/bookmarks/userId/feed-items
```

Note: *`userId`* must be the ID of the context user or the alias `me`.

Available versions

24.0–31.0

Important: In version 32.0 and later, use Bookmarks Feed Elements.

Requires Chatter

Yes

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description	Required or Optional	Available Version
`density`	String	The density of the feed. One of these values: • `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. • `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of, but hides some system-generated updates from records. The default value is `AllUpdates`.	Optional	29.0–31.0
`page`	String	A generated token that indicates the view of feed items in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`, for example: `"nextPageUrl": "/services/data/v34.0/chatter/feeds/news/005D0000001GLowIAN/feed-items?page=2011-03-30T17:34:50Z,0D5D0000000DSv4KAG"`	Optional	23.0–31.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are between 1 and 100. If you don't specify a size, the default is 25.	Optional	23.0–31.0
`q`	String	One or more keywords to search for in the body and comments of feed items in this feed. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.	Optional	28.0–31.0
`recentCommentCount`	Integer	Maximum number of comments to include per feed item. The default value is 3. The maximum value is 25.	Optional	29.0–31.0
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.	Optional	23.0–31.0

Response body
Feed Item Page

Bookmarks Feed Items Flat

This resource is deprecated as of version 26.0.

Returns a flat feed, that is, a feed where there is no hierarchy between feed items and comments: Feed items and comments are at the same level. All items are returned in chronological order.

Resource

```
/chatter/feeds/bookmarks/me/feed-items/flat
```

or

```
/chatter/feeds/bookmarks/userId/feed-items/flat
```

Note: *userId* must be the ID of the context user or the alias `me`.

Available since version
25.0–26.0

Requires Chatter
Yes

HTTP methods
GET or HEAD

Request parameters

Parameter Name	Type	Description
`numFeedItems`	Integer	Specifies the number of feed items returned. Valid values are from 0 through 100. Default value is 25. Because feed items can include comments, specifying 25 flat feed items may actually return more. For example, if each feed item has one comment, and `recentCommentCount` is greater than 1, 50 flat feed items are returned.
`page`	String	A generated token that indicates the view of feed items in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`.
`recentCommentCount`	Integer	Specifies the number of recent comments returned. Valid values are from 0 through 25. Default value is 3.
`sort`	String. One of the following: • `CreatedDateDesc`	Sorts the returned feed either by created date in ascending order, or by last modified

Parameter Name	Type	Description
	• `LastModifiedDateDesc`	date, also in ascending order. The default is `LastModifiedDateDesc`

Response body

Flat Feed

SEE ALSO:

Working with Feeds and Feed Elements

Communities Home Feed Resource

Returns a list of all feed elements associated with any managed topic in the community.

Resource

`/connect/communities/communityId/chatter/feeds/home/feed-elements`

Available since release

32.0

Requires Chatter

Yes

HTTP methods

GET, HEAD

GET parameters

Parameter Name	Type	Description	Required or Optional	Available Version
`filter`	String	Specifies the filter values that can be applied to a feed. • `AllQuestions`—Only feed elements that are questions. • `CommunityScoped`—Reserved for future use. • `SolvedQuestions`—Only feed elements that are questions and that have a best answer. • `UnansweredQuestions`—Only feed elements that are questions and that don't have any answers. • `UnsolvedQuestions`—Only feed elements that are questions and that don't have a best answer.	Optional	32.0
`page`	String	A generated token that indicates the view of feed elements in the feed. Page tokens are returned as part	Optional	29.0

Parameter Name	Type	Description	Required or Optional	Available Version
		of the response body, in one of the URL parameters, such as `nextPageURL`, for example: `"nextPageUrl": "/services/data/v34.0/chatter/feeds/news/005D0000001GLowIAN/feed-elements?page=2011-03-30T17:34:50Z,0D5D0000000DSv4KAG"`		
`pageSize`	Integer	Specifies the number of elements per page. Valid values are between 1 and 100. If you don't specify a size, the default is 25.	Optional	29.0
`q`	String	One or more keywords to search for in the body and comments of feed elements in this feed. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.	Optional	29.0
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.	Optional	29.0

Response body for GET

Feed Element Page

Communities Moderation Feed Resources

A feed containing feed elements and comments that have been flagged for moderation in a community.

Available resources are:

Resource	Description
`/connect/communities/communityId/chatter/feeds/moderation/feed-elements`	If the context user is a community moderator with the "Moderate Communities Feeds" permission, returns a list of all feed elements and comments in the community that have been flagged by community members. By default, returns 25 feed elements per page of results.

Communities Moderation Feed Elements

If the context user is a community moderator with the "Moderate Communities Feeds" permission, returns a list of all feed elements and comments in the community that have been flagged by community members. By default, returns 25 feed elements per page of results.

Resource

```
/connect/communities/communityId/chatter/feeds/moderation/feed-elements
```

Available since release

31.0

Requires Chatter

Yes

HTTP methods

GET, HEAD

GET parameters

Parameter Name	Type	Description	Required or Optional	Available Version
`page`	String	A generated token that indicates the view of feed elements in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`, for example: `"nextPageUrl": "/services/data/v34.0/chatter/feeds/news/005D0000001GLowIAN/feed-elements?page=2011-03-30T17:34:50Z,0D5D0000000DSv4KAG"`	Optional	29.0
`pageSize`	Integer	Specifies the number of elements per page. Valid values are between 1 and 100. If you don't specify a size, the default is 25.	Optional	29.0
`q`	String	One or more keywords to search for in the body and comments of feed elements in this feed. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.	Optional	29.0
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.	Optional	29.0

Response body for GET

Feed Element Page

If a feed element has comments, the Comment Page response body contains a value in the `total` property to indicate the number of comments. However the `comments` property doesn't contain a collection of comments. For example, this excerpt from a response shows a total of 2 comments, but no information about the comments:

```
      "comments": {
         "comments": [],
        "currentPageUrl": "/services/data/v34.0/connect/communities/0DBD000000002etOAA
/chatter/feed-elements/0D5D0000000LYi9KAG/comments",
         "nextPageUrl": null,
         "total": 2
      },
```

Communities Moderation Feed Items

If the context user is a community moderator with the "Moderate Communities Feeds" permission, returns a list of all feed elements and comments in the community that have been flagged by community members. By default, returns 25 feed items per page of results.

Resource

`/connect/communities/communityId/chatter/feeds/moderation/feed-items`

Available versions

29.0–31.0

Important: In version 32.0 and later, use Communities Moderation Feed Elements.

Requires Chatter

Yes

HTTP methods

GET, HEAD

GET parameters

Parameter Name	Type	Description	Required or Optional	Available Version
`page`	String	A generated token that indicates the view of feed items in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`, for example: `"nextPageUrl": "/services/data/v34.0/chatter/feeds/news/005D0000001GLowIAN/feed-items?page=2011-03-30T17:34:50Z,0D5D0000000DSv4KAG"`	Optional	29.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
`pageSize`	Integer	Specifies the number of items per page. Valid values are between 1 and 100. If you don't specify a size, the default is 25.	Optional	29.0–31.0
`q`	String	One or more keywords to search for in the body and comments of feed items in this feed. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.	Optional	29.0–31.0
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.	Optional	29.0–31.0

Response body for GET

Feed Item Page

If a feed item has comments, the Comment Page response body contains a value in the `total` property to indicate the number of comments. However the `comments` property doesn't contain a collection of comments. For example, this excerpt from a response shows a total of 2 comments, but no information about the comments:

```
      "comments": {
         "comments": [],
        "currentPageUrl": "/services/data/v34.0/connect/communities/0DBD000000002etOAA
/chatter/feed-items/0D5D0000000LYi9KAG/comments",
         "nextPageUrl": null,
         "total": 2
      },
```

Example for GET

This example makes a GET request.

```
GET /connect/communities/0DBD000000002etOAA/chatter/feeds/moderation/feed-items
```

The response is a Feed Item Page response body.

SEE ALSO:

Working with Feeds and Feed Elements

Company Feed Resources

A feed containing all feed elements and comments from an entire company.

The company feed includes posts and comments from:

- People in the company, the ones a user follows and the ones a user doesn't follow
- Public groups, including the ones a user is not a member of
- Private groups a user is a member of
- Updates for records and fields that an administrator configured for feed tracking, including the ones a user is not following. However, a user only sees updates for records they are allowed to access.

Please note the following:

- These resources are not available for communities.

Available resources are:

Resource	Description
`/chatter/feeds/company`	Returns a URL to the company feed.
`/chatter/feeds/company/feed-elements`	Returns the feed elements of a company feed.

Company Feed URL

Returns a URL to the company feed.

Resource

```
/chatter/feeds/company
```

```
/connect/communities/communityId/chatter/feeds/company
```

Available since version

23.0

Requires Chatter

Yes

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.

Response body
Feed

Company Feed Elements

Returns the feed elements of a company feed.

Resource

```
/chatter/feeds/company/feed-elements
```

```
/connect/communities/communityId/chatter/feeds/company/feed-elements
```

Available since version
31.0

Requires Chatter
Yes

HTTP methods
GET, HEAD

Request parameters

Parameter Name	Type	Description	Required or Optional	Available Version
`density`	String	The density of the feed. One of these values: • `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. • `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of, but hides some system-generated updates from records. The default value is `AllUpdates`.	Optional	31.0
`elementsPerBundle`	Integer	Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.	Optional	31.0
`page`	String	A generated token that indicates the view of feed elements in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`, for example: `"nextPageUrl": "/services/data/v34.0/chatter/feeds/news/005D0000001GLowIAN/feed-elements`	Optional	31.0

Parameter Name	Type	Description	Required or Optional	Available Version
		`?page=2011-03-30T17:34:50Z, 0D5D0000000DSv4KAG"`		
`pageSize`	Integer	Specifies the number of elements per page. Valid values are between 1 and 100. If you don't specify a size, the default is 25.	Optional	31.0
`q`	String	One or more keywords to search for in the body and comments of feed elements in this feed. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.	Optional	31.0
`recentCommentCount`	Integer	Maximum number of comments to include per feed element. The default value is 3. The maximum value is 25.	Optional	31.0
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.	Optional	31.0
`updatedSince`	String	An opaque token defining the modification time stamp of the feed and the sort order. Do not construct this token. Retrieve this token from the `updatesToken` property of the Feed Element Page response body.	Optional	31.0

Response body

Feed Element Page

Company Feed Items

Returns the feed items of a company feed.

Resource

```
/chatter/feeds/company/feed-items
```

Available versions

23.0–31.0

Important: In version 32.0 and later, use Company Feed Elements.

Requires Chatter
Yes

HTTP methods
GET, HEAD

Request parameters

Parameter Name	Type	Description	Required or Optional	Available Version
`density`	String	The density of the feed. One of these values: • `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. • `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of, but hides some system-generated updates from records. The default value is `AllUpdates`.	Optional	29.0–31.0
`page`	String	A generated token that indicates the view of feed elements in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`, for example: `"nextPageUrl": "/services/data/v34.0/chatter/feeds/news/005D0000001GLowIAN/feed-elements?page=2011-03-30T17:34:50Z,0D5D0000000DSv4KAG"`	Optional	23.0–31.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are between 1 and 100. If you don't specify a size, the default is 25.	Optional	23.0–31.0
`q`	String	One or more keywords to search for in the body and comments of feed elements in this feed. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.	Optional	28.0–31.0
`recentCommentCount`	Integer	Maximum number of comments to include per feed item. The default value is 3. The maximum value is 25.	Optional	29.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.	Optional	23.0–31.0
`updatedSince`	String	An opaque token defining the modification time stamp of the feed and the sort order. Do not construct this token. Retrieve this token from the `updatesToken` property of the Feed Element Page response body.	Optional	30.0–31.0

Response body

Feed Item Page

SEE ALSO:

Working with Feeds and Feed Elements

Favorites Feed Resources

A feed containing favorites saved by the context user.

Favorites are feed searches, list views, and topics that the context user has added as a Chatter favorite.

Available resources are:

Resource	Description
`/chatter/feeds/favorites/me`	Returns all the favorites for the context user. Creates a favorite of a Chatter feed search or topic.
`/chatter/feeds/favorites/me/favoriteId`	Returns the feed for the specified favorite. Deletes the specified favorite. Updates the last view date of the specified favorite.
`/chatter/feeds/favorites/me/favoriteId/feed-elements`	Returns a filtered feed, only showing those feed elements for the specified favorite.

List of Favorites

Returns all the favorites for the context user. Creates a favorite of a Chatter feed search or topic.

Note: The response contains feed searches that were saved using the Web UI. However, you can't use Chatter API to save a feed search.

Resource

```
/chatter/feeds/favorites/userId
```

```
/connect/communities/communityId/chatter/feeds/favorites/userId
```

Note: *userId* must be the ID of the context user or the alias `me`.

Available since version

24.0

Requires Chatter

Yes

HTTP methods

GET, HEAD, or POST

Request body for POST

Root XML tag

```
<favorite>
```

JSON for POST

```
{
   "searchText" : "release",
   "targetId" : "0TOD0000000099d"
}
```

Properties for POST

Name	Type	Description	Available Version
`searchText`	String	Text of the feed search. Use this parameter for feed searches only.	24.0
`targetId`	String	Id of the topic. Use this parameter for topics only.	28.0

Request parameters for POST

Name	Type	Description	Available Version
`searchText`	String	Text of the feed search. Use this parameter for feed searches only.	24.0
`targetId`	String	Id of the topic. Use this parameter for topics only.	28.0

Response body for GET

Favorites Page

Response body for POST

Favorite

Favorite

Returns the feed for the specified favorite. Deletes the specified favorite. Updates the last view date of the specified favorite.

Resource

```
/chatter/feeds/favorites/userId/favoriteId
```

```
/connect/communities/communityId/chatter/feeds/favorites/userId/favoriteId
```

Note: *userId* must be the ID of the context user or the alias `me`.

Available since version

24.0

Requires Chatter

Yes

HTTP methods

GET, HEAD, PATCH, or DELETE

Request parameter for GET or HEAD

Parameter Name	Type	Description
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.

Request body for PATCH

Root XML tag

```
<favorite>
```

JSON for PATCH

```
{
   "updateLastViewDate" : "true"
}
```

Properties for PATCH

Name	Type	Description	Available Version
`updateLastViewDate`	Boolean	Sets the last view date of the specified feed favorite to the	24.0

Name	Type	Description	Available Version
		current system time. Default value is `false`.	

Request parameter for PATCH

Parameter Name	Type	Description
`updateLastViewDate`	Boolean	Sets the last view date of the specified feed favorite to the current system time. Default value is `false`.

Response body for GET

Favorite

Response body for PATCH

Favorite

Favorites Feed Items

Returns a filtered feed, only showing those feed items for the specified favorite.

Resource

```
/chatter/feeds/favorites/me/favoriteId/feed-items
```

or

```
/chatter/feeds/favorites/userId/favoriteId/feed-items
```

Note: *`userId`* must be the ID of the context user or the alias `me`.

Available versions

24.0–31.0

Important: In version 32.0 and later, use Favorites Feed Elements.

Requires Chatter

Yes

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description	Required or Optional	Available Version
`density`	String	The density of the feed. One of these values: • `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. • `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of, but hides some system-generated updates from records. The default value is `AllUpdates`.	Optional	29.0–31.0
`page`	String	A generated token that indicates the view of feed items in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`, for example: `"nextPageUrl": "/services/data/v34.0/chatter/feeds/news/005D0000001GLowIAN/feed-items?page=2011-03-30T17:34:50Z,0D5D0000000DSv4KAG"`	Optional	23.0–31.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are from 1 through 100. If you don't specify a size, the default is 25.	Optional	23.0–31.0
`q`	String	One or more keywords to search for in the body and comments of feed items in this feed. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.	Optional	28.0–31.0
`recentCommentCount`	Integer	Maximum number of comments to include per feed item. The default value is 3. The maximum value is 25.	Optional	29.0–31.0
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.	Optional	23.0–31.0

Response body
Feed Item Page

SEE ALSO:
Working with Feeds and Feed Elements

Favorites Feed Elements

Returns a filtered feed, only showing those feed elements for the specified favorite.

Resource

```
/chatter/feeds/favorites/userId/favoriteId/feed-elements
```

```
/connect/communities/communityId/chatter/feeds/favorites/userId/favoriteId/feed-elements
```

Note: *userId* must be the ID of the context user or the alias `me`.

Available version
31.0

Requires Chatter
Yes

HTTP methods
GET

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
`elementsPerBundle`	Integer	The maximum number of feed elements per bundle. The default and maximum value is 10. Note: Bundled posts contain feed-tracked changes. In Salesforce1 downloadable apps, bundled posts are in record feeds only.	Optional	31.0
`page`	String	The page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.	Optional	31.0
`pageSize`	Integer	Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.	Optional	31.0

Parameter Name	Type	Description	Required or Optional	Available Version
`recentCommentCount`	Integer	The maximum number of comments to return with each feed element. The default value is 3.	Optional	31.0
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. If you pass in `null`, the default value `CreatedDescDate` is used.	Optional	31.0

Response body for GET
Feed Element Page

Feeds Resource

A list of all of the feeds the context user is able to view.

Each user may have a different set of feeds. Use this resource to determine what feeds are available. In addition, the different feeds may have unique labels as well as URLs. Use the given labels returned in this resource and avoid hard-coding any of these values.

Resource

```
/chatter/feeds/
```

```
/connect/communities/communityId/chatter/feeds/
```

Available since version
23.0

Requires Chatter
Yes

HTTP methods
GET or HEAD

Response body
Feed Directory

Files Feed Resources

A feed containing all the feed elements that contain files posted by people or groups that the context user follows.

Available resources are:

Resource	Description
`/chatter/feeds/files/me`	Returns the URL to the feed elements that contain files that are posted by users or groups the context user is following.
`/chatter/feeds/files/me/feed-elements`	Returns all feed elements posted with files by people or groups that the context user follows.

Files Feed URL

Returns the URL to the feed elements that contain files that are posted by users or groups the context user is following.

Resource

```
/chatter/feeds/files/userId
```

```
/connect/communities/communityId/chatter/feeds/files/userId
```

Note: *userId* must be the ID of the context user or the alias `me`.

Available since version

23.0

Requires Chatter

Yes

HTTP methods

GET or HEAD

Request parameter

Parameter Name	Type	Description
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.

Response body

Feed

Files Feed Elements

Returns all feed elements posted with files by people or groups that the context user follows.

Resource

```
/chatter/feeds/files/userId/feed-elements
```

```
/connect/communities/communityId/chatter/feeds/files/userId/feed-elements
```

Note: *userId* must be the ID of the context user or the alias `me`.

Available since version

31.0

Requires Chatter

Yes

HTTP methods

GET, HEAD

Request parameters

Parameter Name	Type	Description	Required or Optional	Available Version
`density`	String	The density of the feed. One of these values: • `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. • `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of, but hides some system-generated updates from records. The default value is `AllUpdates`.	Optional	31.0
`elementsPerBundle`	Integer	Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.	Optional	31.0
`page`	String	A generated token that indicates the view of feed elements in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`, for example: `"nextPageUrl": "/services/data/v34.0/chatter/feeds/news/005D0000001GLowIAN/feed-elements?page=2011-03-30T17:34:50Z,0D5D0000000DSv4KAG"`	Optional	31.0

Parameter Name	Type	Description	Required or Optional	Available Version
`pageSize`	Integer	Specifies the number of elements per page. Valid values are between 1 and 100. If you don't specify a size, the default is 25.	Optional	31.0
`q`	String	One or more keywords to search for in the body and comments of feed elements in this feed. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.	Optional	31.0
`recentCommentCount`	Integer	Maximum number of comments to include per feed element. The default value is 3. The maximum value is 25.	Optional	31.0
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.	Optional	31.0
`updatedSince`	String	An opaque token defining the modification time stamp of the feed and the sort order. Do not construct this token. Retrieve this token from the `updatesToken` property of the Feed Element Page response body.	Optional	31.0

Response body

Feed Element Page

Files Feed Items

Returns all feed items posted with files by people or groups that the context user follows.

Resource

```
/chatter/feeds/files/me/feed-items
```

or

```
/chatter/feeds/files/userId/feed-items
```

Note: *`userId`* must be the ID of the context user or the alias `me`.

Available versions

23.0–31.0

Important: In version 32.0 and later, use Files Feed Elements.

Requires Chatter
Yes

HTTP methods
GET, HEAD

Request parameters

Parameter Name	Type	Description	Required or Optional	Available Version
`density`	String	The density of the feed. One of these values: • `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. • `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of, but hides some system-generated updates from records. The default value is `AllUpdates`.	Optional	29.0–31.0
`page`	String	A generated token that indicates the view of feed elements in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`, for example: `"nextPageUrl": "/services/data/v34.0/chatter/feeds/news/005D0000001GLowIAN/feed-elements?page=2011-03-30T17:34:50Z,0D5D0000000DSv4KAG"`	Optional	23.0–31.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are between 1 and 100. If you don't specify a size, the default is 25.	Optional	23.0–31.0
`q`	String	One or more keywords to search for in the body and comments of feed elements in this feed. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.	Optional	28.0–31.0
`recentCommentCount`	Integer	Maximum number of comments to include per feed item. The default value is 3. The maximum value is 25.	Optional	29.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.	Optional	23.0–31.0
`updatedSince`	String	An opaque token defining the modification time stamp of the feed and the sort order. Do not construct this token. Retrieve this token from the `updatesToken` property of the Feed Element Page response body.	Optional	30.0–31.0

Response body

Feed Item Page

Files Feed Items Flat

This resource is deprecated as of version 26.0.

Returns a flat feed, that is, a feed where there is no hierarchy between feed items and comments: Feed items and comments are at the same level. All items are returned in chronological order.

Resource

```
/chatter/feeds/files/me/feed-items/flat
```

or

```
/chatter/feeds/files/userId/feed-items/flat
```

Note: *`userId`* must be the ID of the context user or the alias `me`.

Available since version

25.0–26.0

Requires Chatter

Yes

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
`numFeedItems`	Integer	Specifies the number of feed items returned. Valid values are from 0 through 100. Default value is 25. Because feed items can include comments, specifying 25 flat feed items may actually return more. For example, if each feed item has one comment, and `recentCommentCount` is greater than 1, 50 flat feed items are returned.
`page`	String	A generated token that indicates the view of feed items in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`.
`recentCommentCount`	Integer	Specifies the number of recent comments returned. Valid values are from 0 through 25. Default value is 3.
`sort`	String. One of the following: • `CreatedDateDesc` • `LastModifiedDateDesc`	Sorts the returned feed either by created date in ascending order, or by last modified date, also in ascending order. The default is `LastModifiedDateDesc`

Response body

Flat Feed

SEE ALSO:

Working with Feeds and Feed Elements

Filter Feed Resources

The filter feed is the news feed filtered to contain feed elements whose parent is a specified object type.

Available resources are:

Resource	Description
`/chatter/feeds/filter/`***`userId`***	A collection of feed directory objects that contain information about filter feeds available to the context user. Each object contains a URL for the filter feed, a URL for feed items or feed elements in the filter feed, and a label describing the object used to filter the feed.
`/chatter/feeds/filter/`***`userId`***`/`***`idPrefix`***	A feed object that contains information about a specific filter feed available to the context user.

Resource	Description
`/chatter/feeds/filter/userId/idPrefix/feed-elements`	Feed elements from the context user's news feed whose parent has the specified key prefix, such as the feed elements from all the groups the context user is a member of.

List of Filter Feed URLs

A collection of feed directory objects that contain information about filter feeds available to the context user. Each object contains a URL for the filter feed, a URL for feed items or feed elements in the filter feed, and a label describing the object used to filter the feed. Each URL contains the first three characters of the filter object, which is called a *key prefix* and specifies the object type. Use this resource to return a list of the possible key prefix filter feeds that the context user can apply to their news feed. See Filter Feed Items and Filter Feed Elements.

Note: The collection of feed directory objects never contains User (`005`) and Group (`0F9`) object types. However, you can use those object types to filter the feed.

Resource

```
/chatter/feeds/filter/userId
```

```
/connect/communities/communityId/chatter/feeds/filter/userId
```

`userId` must be the ID of the context user or the alias `me`.

Available since version

23.0

Requires Chatter

Yes

HTTP methods

GET or HEAD

Response body

Feed Directory

If the response body is empty, the user isn't following any records.

Specific Filter Feed URL

A feed object that contains information about a specific filter feed available to the context user. The *`keyPrefix`* in the resource URL is the first three characters of the object ID and indicates the object type.

Resource

```
/chatter/feeds/filter/userId/keyPrefix
```

```
/connect/communities/communityId/chatter/feeds/filter/userId/keyPrefix
```

`userId` must be the ID of the context user or the alias `me`.

Available since version

23.0

Requires Chatter

Yes

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.

Response body

Feed

Example response body

This request returns the URL for a filter feed items resource that's filtered by the User object (key prefix `005`).

```
GET /chatter/feeds/filter/me/005
```

Filter Feed Items

Feed items from the context user's news feed whose parent has the specified key prefix, such as the feed items from all the groups the context user is a member of. The key prefix specifies the object type and is the first three characters of an object ID.

To filter by feed items with a User parent object, use the key prefix `005`.

To filter by feed items with a Group parent object, use the key prefix `0F9`.

To filter by feed items with a parent object that the context user is following (for example, Account), use the List of Filter Feed URLs resource to return the parent objects.

Suppose the context user is a member of five groups. The following resource returns all of the feed items posted to all five of those groups:

```
/chatter/feeds/filter/me/0F9/feed-items
```

Contrast that with a request to the Record Feed Items resource, which returns all of the feed items posted to a particular group:

```
/chatter/feeds/record/0F9XXXXXXXXX/feed-items
```

The filter feed returns only the feed items that are *visible to the context user*. In the first example, the resource doesn't return the feed items for groups the context user is not a member of. In the second example, if the group is public, even if the context user is not a member, the resource returns all the group's feed items.

Resource

```
/chatter/feeds/filter/me/keyPrefix/feed-items
```

or

```
/chatter/feeds/filter/userId/keyPrefix/feed-items
```

userId must be the ID of the context user or the alias `me`.

Available versions

23.0–31.0

Important: In version 32.0 and later, use Filter Feed Elements.

Requires Chatter

Yes

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description	Required or Optional	Available Version
`density`	String	The density of the feed. One of these values: • `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. • `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of, but hides some system-generated updates from records. The default value is `AllUpdates`.	Optional	29.0–31.0
`page`	String	A generated token that indicates the view of feed elements in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`, for example: `"nextPageUrl": "/services/data/v34.0/chatter/feeds/news/005D0000001GLowIAN/feed-elements?page=2011-03-30T17:34:50Z,0D5D0000000DSv4KAG"`	Optional	23.0–31.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are between 1 and 100. If you don't specify a size, the default is 25.	Optional	23.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
`q`	String	One or more keywords to search for in the body and comments of feed elements in this feed. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.	Optional	28.0–31.0
`recentCommentCount`	Integer	Maximum number of comments to include per feed item. The default value is 3. The maximum value is 25.	Optional	29.0–31.0
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.	Optional	23.0–31.0
`updatedSince`	String	An opaque token defining the modification time stamp of the feed and the sort order. Do not construct this token. Retrieve this token from the `updatesToken` property of the Feed Element Page response body.	Optional	30.0–31.0

Response body

Feed Item Page

Example

This request gets all the feed items whose parent is the User object and that the context user can see.

```
GET https://instance_name/services/data/v34.0/chatter/feeds/filter/me/005/feed-items
```

Filter Feed Items Flat

This resource is deprecated as of version 26.0.

Returns a flat feed, that is, a feed where there is no hierarchy between feed items and comments: Feed items and comments are at the same level. All items are returned in chronological order.

Resource

```
/chatter/feeds/filter/me/feed-items/flat
```

or

```
/chatter/feeds/filter/userId/feed-items/flat
```

Note: *`userId`* must be the ID of the context user or the alias `me`.

Available since version
25.0–26.0

Requires Chatter
Yes

HTTP methods
GET or HEAD

Request parameters

Parameter Name	Type	Description
`numFeedItems`	Integer	Specifies the number of feed items returned. Valid values are from 0 through 100. Default value is 25. Because feed items can include comments, specifying 25 flat feed items may actually return more. For example, if each feed item has one comment, and `recentCommentCount` is greater than 1, 50 flat feed items are returned.
`page`	String	A generated token that indicates the view of feed items in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`.
`recentCommentCount`	Integer	Specifies the number of recent comments returned. Valid values are from 0 through 25. Default value is 3.
`sort`	String. One of the following: • `CreatedDateDesc` • `LastModifiedDateDesc`	Sorts the returned feed either by created date in ascending order, or by last modified date, also in ascending order. The default is `LastModifiedDateDesc`

Response body
Flat Feed

Filter Feed Elements

Feed elements from the context user's news feed whose parent has the specified key prefix, such as the feed elements from all the groups the context user is a member of.

The key prefix specifies the object type and is the first three characters of an object ID.

To filter by feed elements with a User parent object, use the key prefix `005`.

To filter by feed elements with a Group parent object, use the key prefix `0F9`.

To filter by feed elements with a parent object that the context user is following (for example, Account), use the List of Filter Feed URLs resource to return the parent objects.

Suppose the context user is a member of five groups. The following resource returns all of the feed elements for all five of those groups:

```
/chatter/feeds/filter/me/0F9/feed-elements
```

The filter feed returns only the feed elements that are *visible to the context user*. In this example, the resource doesn't return the feed elements for groups the context user is not a member of.

Resource

```
/chatter/feeds/filter/userId/keyPrefix/feed-elements
```

```
/connect/communities/communityId/chatter/feeds/filter/userId/keyPrefix/feed-elements
```

userId must be the ID of the context user or the alias `me`.

Available version

31.0

Requires Chatter

Yes

HTTP methods

GET

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
`density`	String	• `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. • `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of, but hides some system-generated updates from records.	Optional	31.0
`elementsPerBundle`	Integer	The maximum number of feed elements per bundle. The default and maximum value is 10. Note: Bundled posts contain feed-tracked changes. In Salesforce1 downloadable apps, bundled posts are in record feeds only.	Optional	31.0
`page`	String	The page token to use to view the page. Page tokens are returned as part of the response class, for example, `currentPageToken` or	Optional	31.0

Parameter Name	Type	Description	Required or Optional	Available Version
		`nextPageToken`. If you pass in `null`, the first page is returned.		
`pageSize`	Integer	Specifies the number of feed elements per page. Valid values are from 1 through 100. If you pass in `null`, the default size is 25.	Optional	31.0
`q`	String	One or more keywords to search for in the feed elements visible to the context user. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.	Optional	31.0
`recentCommentCount`	Integer	The maximum number of comments to return with each feed element. The default value is 3.	Optional	31.0
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. If you pass in `null`, the default value `CreatedDescDate` is used.	Optional	31.0
`updatedSince`	String	An opaque token defining the modification time stamp of the feed and the sort order.	Optional	31.0

Response body for GET

Feed Element Page

Groups Feed Resources

A feed containing all feed elements from all groups the current user either owns or is a member of.

Available resources are:

Resource	Description
`/chatter/feeds/groups/me`	Returns a URL to the feed elements for all the groups the current, signed-in user is a member of.
`/chatter/feeds/groups/me/feed-elements`	Returns all feed elements for all groups the current user either owns or is a member of.

Groups Feed URL

Returns a URL to the feed elements for all the groups the current, signed-in user is a member of.

Resource

```
/chatter/feeds/groups/userId
```

```
/connect/communities/communityId/chatter/feeds/groups/userId
```

Note: *userId* must be the ID of the context user or the alias `me`.

Response body

Feed

Groups Feed Elements

Returns all feed elements for all groups the current user either owns or is a member of.

To get the feed elements for a specific group, use the record feed resource (`/chatter/feeds/record/recordId/feed-elements`) and substitute the group ID for the ***recordId***.

Resource

```
/chatter/feeds/groups/userId/feed-elements
```

```
/connect/communities/communityId/chatter/feeds/groups/userId/feed-elements
```

Note: *userId* must be the ID of the context user or the alias `me`.

Available since version

31.0

Requires Chatter

Yes

HTTP methods

GET, HEAD

Request parameters

Parameter Name	Type	Description	Required or Optional	Available Version
`density`	String	The density of the feed. One of these values: • `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. • `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of, but hides some system-generated updates from records. The default value is `AllUpdates`.	Optional	31.0

Parameter Name	Type	Description	Required or Optional	Available Version
`elementsPerBundle`	Integer	Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.	Optional	31.0
`page`	String	A generated token that indicates the view of feed elements in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`, for example: `"nextPageUrl": "/services/data/v34.0/chatter/feeds/news/005D0000001GLowIAN/feed-elements?page=2011-03-30T17:34:50Z,0D5D0000000DSv4KAG"`	Optional	31.0
`pageSize`	Integer	Specifies the number of elements per page. Valid values are between 1 and 100. If you don't specify a size, the default is 25.	Optional	31.0
`q`	String	One or more keywords to search for in the body and comments of feed elements in this feed. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.	Optional	31.0
`recentCommentCount`	Integer	Maximum number of comments to include per feed element. The default value is 3. The maximum value is 25.	Optional	31.0
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.	Optional	31.0
`updatedSince`	String	An opaque token defining the modification time stamp of the feed and the sort order. Do not construct this token. Retrieve this token from the `updatesToken` property of the Feed Element Page response body.	Optional	31.0

Response body
Feed Element Page

Groups Feed Items

Returns all feed items for all groups the current user either owns or is a member of.

To get the feed items for a specific group, use the record feed resource (`/chatter/feeds/record/recordId/feed-items`) and substitute the group ID for the ***recordId***.

Resource

```
/chatter/feeds/groups/me/feed-items
```

or

```
/chatter/feeds/groups/userId/feed-items
```

Note: *userId* must be the ID of the context user or the alias `me`.

Available versions
23.0–31.0

Important: In version 32.0 and later, use Groups Feed Elements.

Requires Chatter
Yes

HTTP methods
GET, HEAD

Request parameters

Parameter Name	Type	Description	Required or Optional	Available Version
`density`	String	The density of the feed. One of these values: • `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. • `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of, but hides some system-generated updates from records. The default value is `AllUpdates`.	Optional	29.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
`page`	String	A generated token that indicates the view of feed elements in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`, for example: `"nextPageUrl": "/services/data/v34.0/chatter/feeds/news/005D0000001GLowIAN/feed-elements?page=2011-03-30T17:34:50Z,0D5D0000000DSv4KAG"`	Optional	23.0–31.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are between 1 and 100. If you don't specify a size, the default is 25.	Optional	23.0–31.0
`q`	String	One or more keywords to search for in the body and comments of feed elements in this feed. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.	Optional	28.0–31.0
`recentCommentCount`	Integer	Maximum number of comments to include per feed item. The default value is 3. The maximum value is 25.	Optional	29.0–31.0
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.	Optional	23.0–31.0
`updatedSince`	String	An opaque token defining the modification time stamp of the feed and the sort order. Do not construct this token. Retrieve this token from the `updatesToken` property of the Feed Element Page response body.	Optional	30.0–31.0

Response body

Feed Item Page

Groups Feed Items Flat

This resource is deprecated as of version 26.0.

Returns a flat feed, that is, a feed where there is no hierarchy between feed items and comments: Feed items and comments are at the same level. All items are returned in chronological order.

Resource

```
/chatter/feeds/groups/me/feed-items/flat
```

or

```
/chatter/feeds/groups/userId/feed-items/flat
```

Note: *userId* must be the ID of the context user or the alias `me`.

Available since version

25.0–26.0

Requires Chatter

Yes

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
`numFeedItems`	Integer	Specifies the number of feed items returned. Valid values are from 0 through 100. Default value is 25. Because feed items can include comments, specifying 25 flat feed items may actually return more. For example, if each feed item has one comment, and `recentCommentCount` is greater than 1, 50 flat feed items are returned.
`page`	String	A generated token that indicates the view of feed items in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`.
`recentCommentCount`	Integer	Specifies the number of recent comments returned. Valid values are from 0 through 25. Default value is 3.
`sort`	String. One of the following: • `CreatedDateDesc` • `LastModifiedDateDesc`	Sorts the returned feed either by created date in ascending order, or by last modified date, also in ascending order. The default is `LastModifiedDateDesc`

Response body
Flat Feed

SEE ALSO:
Groups Resources
Record Feed Resource
Working with Feeds and Feed Elements

News Feed Resources

A feed containing feed elements of interest to the context user. You can post feed items to the news feed.

Note: To match the feed displayed in the Chatter Web UI, the composition of this feed may change between releases.

Available resources are:

Resource	Description
`/chatter/feeds/news/me`	Returns a URL to the feed elements for all feeds for the context user.
`/chatter/feeds/news/me/feed-elements`	All feed elements from all groups the context user either owns or is a member of, as well as all files, records, and users the context user follows. Use this resource to get information about feed elements and to post feed elements.

News Feed URL

Returns a URL to the feed elements for all feeds for the context user.

Resource

```
/chatter/feeds/news/userId
```

```
/connect/communities/communityId/chatter/feeds/news/userId
```

Note: *userId* must be the ID of the context user or the alias `me`.

Available since version
23.0

Requires Chatter
Yes

HTTP methods
GET or HEAD

Request parameter

Parameter Name	Type	Description
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.

Response body

Feed

Example response body

```
{
   "feedElementPostUrl": "/services/data/v34.0/chatter/feed-elements",
   "feedElementsUrl":
"/services/data/v34.0/chatter/feeds/news/005xx000001Sv1mAAB/feed-elements"
}
```

News Feed Elements

All feed elements from all groups the context user either owns or is a member of, as well as all files, records, and users the context user follows. Use this resource to get information about feed elements and to post feed elements.

For information about posting a feed element, see Feed Elements, Post and Search.

For information about uploading and attaching a binary file to a feed element, see Uploading Binary Files.

Resource

```
/chatter/feeds/news/userId/feed-elements
```

```
/connect/communities/communityId/chatter/feeds/news/userId/feed-elements
```

Note: *`userId`* must be the ID of the context user or the alias `me`.

Available since version

31.0

Requires Chatter

Yes

HTTP methods

GET, HEAD

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
`density`	String	The density of the feed. One of these values: • `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. • `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of, but hides some system-generated updates from records. The default value is `AllUpdates`.	Optional	31.0
`elementsPerBundle`	Integer	Maximum number of feed elements to include in a bundle. The value must be an integer from 0 through 10. The default value is 3. Note: Bundled posts contain feed-tracked changes. In Salesforce1 downloadable apps, bundled posts are in record feeds only.	Optional	31.0
`filter`	String	Specifies the filter values that can be applied to a feed. • `AllQuestions`—Only feed elements that are questions. • `CommunityScoped`—Reserved for future use. • `SolvedQuestions`—Only feed elements that are questions and that have a best answer. • `UnansweredQuestions`—Only feed elements that are questions and that don't have any answers. • `UnsolvedQuestions`—Only feed elements that are questions and that don't have a best answer.	Optional	32.0
`page`	String	A generated token that indicates the view of feed elements in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`: `"nextPageUrl": "/services/data/v34.0/chatter/feeds/news/005D0000001GLowIAN/feed-elements?page=2011-03-30T17:34:50Z,0D5D0000000DSv4KAG"`	Optional	31.0

Parameter Name	Type	Description	Required or Optional	Available Version
`pageSize`	Integer	Specifies the number of elements per page. Valid values are from 1 through 100. If you don't specify a size, the default is 25.	Optional	31.0
`q`	String	One or more keywords to search for in the body and comments of feed elements in this feed. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.	Optional	31.0
`recentCommentCount`	Integer	Maximum number of comments to include per feed element. The default value is 3. The maximum value is 25.	Optional	31.0
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.	Optional	31.0
`updatedSince`	String	An opaque token defining the modification time stamp of the feed and the sort order. Do not construct this token. Retrieve this token from the `updatesToken` property of the Feed Element Page response body.	Optional	31.0

Response body for GET

Feed Element Page

News Feed Items

All feed items from all groups the context user either owns or is a member of, as well as all files, records, and users the context user follows. Use this resource to get information about the feed items and to post feed items, including attaching files, posting links, polls, and questions, and sharing posts.

For information about posting a feed item, see Using POST, PATCH, or PUT for Input on page 10.

For information about uploading and attaching a binary file to a feed item, see Uploading Binary Files on page 10.

As of API version 29.0, feed items can contain 5000 characters. This change is backwards compatible; all earlier API versions support the 5000 character limit.

Resource

```
/chatter/feeds/news/userId/feed-items
```

Note: *`userId`* must be the ID of the context user or the alias `me`.

Available versions
23.0–31.0

Important: In version 32.0 and later, use News Feed Elements.

Requires Chatter
Yes

HTTP methods
GET, POST, HEAD

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
`density`	String	The density of the feed. One of these values: • `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. • `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of, but hides some system-generated updates from records. The default value is `AllUpdates`.	Optional	29.0–31.0
`page`	String	A generated token that indicates the view of feed items in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`: `"nextPageUrl": "/services/data/v34.0/chatter/feeds/news/005D0000001GLowIAN/feed-items?page=2011-03-30T17:34:50Z,0D5D0000000DSv4KAG"`	Optional	23.0–31.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are from 1 through 100. If you don't specify a size, the default is 25.	Optional	23.0–31.0
`q`	String	One or more keywords to search for in the body and comments of feed items in this feed. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.	Optional	28.0–31.0
`recentCommentCount`	Integer	Maximum number of comments to include per feed item. The default value is 3. The maximum value is 25.	Optional	29.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.	Optional	23.0–31.0
`updatedSince`	String	An opaque token defining the modification time stamp of the feed and the sort order. Do not construct this token. Retrieve this token from the `updatesToken` property of the Feed Item Page response body.	Optional	30.0–31.0

Request body for POST

Root XML tag

```
<feedItem>
```

JSON

The following example attaches a poll to a feed item:

```
{
   "body":{
      "messageSegments":[
         {
            "type":"Text",
            "text":"When should we meet for release planning?"
         }
      ]
   },
   "capabilities":{
      "poll":{
         "choices":[
            "Monday",
            "Tuesday"
         ]
      }
   },
   "feedElementType":"FeedItem",
   "subjectId":"me"
}
```

Feed items support more than 1 capability. The following example attaches a file to a feed item and bookmarks it.

```
{
   "body":{
      "messageSegments":[
```

```
            {
               "type":"Text",
               "text":"Check out the latest marketing collateral."
            }
         ]
      },
      "capabilities":{
         "bookmarks":{
            "isBookmarkedByCurrentUser": true
         },
         "content":{
            "contentDocumentId": "069D00000001uQO"
         }
      },
      "feedElementType":"FeedItem",
      "subjectId":"me"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`attachment`	Depends on type.	You can specify only one of these values in the attachment parameter. Values are: • Attachment Input: Canvas App (v29.0–31.0) • Attachment Input: Existing Content (v23.0–31.0) • Attachment Input: Feed Poll (v26.0–31.0) • Attachment Input: Link (v23.0–31.0) • Attachment Input: New File Upload (v23.0–31.0) **Important:** As of API version 32.0, use the `capabilities` property.	Optional	
`body`	Message Body Input	Description of message body. The body can contain up to 25 mentions. If you specify `originalFeedElementId` to share a feed item, use the `body` property to add the first comment to the feed item.	Required unless the feed item has a content capability or a link capability	23.0

Name	Type	Description	Required or Optional	Available Version
		To edit this property in a feed item, make a PATCH request to `/chatter/feed-elements/feedElementId`. Editing feed posts is supported in version 34.0 and later.		
`capabilities`	Feed Element Capabilities Input	The capabilities that define the behavior of this feed element.	Optional	31.0
`isBookmarkedByCurrentUser`	Boolean	Specifies whether to bookmark the new feed item for the user (`true`) or not (`false`). The default value is `false`. **Important:** As of API version 32.0, use the `capabilities.bookmarks.isBookmarkedByCurrentUser` property.	Optional	24.0–31.0
`feedElementType`	String	The type of feed element this input represents. The only valid value is `FeedItem`.	Required	31.0
`originalFeedElementId`	String	To share a feed element, specify its 18-character ID. **Note:** You can't share a post and upload a file in the same request body.	Optional	31.0
`originalFeedItemId`	String	To share a feed item, specify its 18-character ID. **Important:** As of API version 32.0, use the `originalFeedElementId` property.	Optional	24.0–31.0
`subjectId`	String	The ID of the parent this feed element is being posted to. This value can be the ID of a user, group, or record, or the string `me` to indicate the context user.	Required	31.0

Name	Type	Description	Required or Optional	Available Version
`visibility`	String	Who can see the post. Valid values are: • `InternalUsers`—If Communities is enabled, only members of the default, internal community can see the feed item. • `AllUsers`—If Communities is enabled, all members of the community can see the feed item. Default values: • For external users, the default value is `AllUsers`. External users must use this value to see their posts. • For internal users, the default value is `InternalUsers`. Internal users can accept this value or use the value `AllUsers` to allow external users to see their posts. If the parent of the feed item is a User or a Group, the `visibility` of the feed item must be `AllUsers`.	Optional	26.0

Request parameters for POST

Use one of the following sets of parameters. For example, you can't post a link and attach a file in the same request.

Parameters for attaching an existing file

Parameter Name	Type	Description	Required or Optional	Available Version
`attachmentType`	String	The attachment type. Must contain the value `ExistingContent`.	Required	27.0–31.0
`contentDocumentId`	String	18-character ID of an existing file.	Required	23.0–31.0
`isBookmarkedByCurrentUser`	Boolean	Specifies whether to bookmark the new feed item for the user (`true`) or not (`false`). The default value is `false`.	Optional	26.0–31.0
`text`	String	Text of the feed item.	Optional	23.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
`visibility`	String	Who can see the post. Valid values are: • `InternalUsers`—If Communities is enabled, only members of the default, internal community can see the feed item. • `AllUsers`—If Communities is enabled, all members of the community can see the feed item. Default values: • For external users, the default value is `AllUsers`. External users must use this value to see their posts. • For internal users, the default value is `InternalUsers`. Internal users can accept this value or use the value `AllUsers` to allow external users to see their posts. If the parent of the feed item is a User or a Group, the `visibility` of the feed item must be `AllUsers`.	Optional	26.0–31.0

Parameters for attaching a new file

Parameter Name	Type	Description	Required or Optional	Available Version
`attachmentType`	String	The attachment type. Must contain the value `NewFile`.	Required	27.0–31.0
`desc`	String	Description of file being posted. **Attention:** Don't use this parameter as of version 27.0. Use `description` instead.	Optional	23.0–26.0
`description`	String	Description of file being posted.	Optional	27.0–31.0
`fileName`	String	File name of the file being posted. **Attention:** Don't use this parameter as of version 24.0. Use `title` instead.	Required	23.0–24.0
`isBookmarkedByCurrentUser`	Boolean	Specifies whether to bookmark the new feed item for the user (`true`) or not (`false`). The default value is `false`.	Optional	26.0–31.0
`text`	String	Text of the feed item.	Optional	23.0–31.0
`title`	String	File name of a file to be uploaded and posted.	Required	24.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
`visibility`	String	Who can see the post. Valid values are: • `InternalUsers`—If Communities is enabled, only members of the default, internal community can see the feed item. • `AllUsers`—If Communities is enabled, all members of the community can see the feed item. Default values: • For external users, the default value is `AllUsers`. External users must use this value to see their posts. • For internal users, the default value is `InternalUsers`. Internal users can accept this value or use the value `AllUsers` to allow external users to see their posts. If the parent of the feed item is a User or a Group, the `visibility` of the feed item must be `AllUsers`.	Optional	26.0–31.0

Parameters for posting a link

Parameter Name	Type	Description	Required or Optional	Available Version
`attachmentType`	String	The attachment type. Must contain the value `Link`.	Required	27.0–31.0
`isBookmarkedByCurrentUser`	Boolean	Specifies whether to bookmark the new feed item for the user (`true`) or not (`false`). The default value is `false`.	Optional	26.0–31.0
`text`	String	Text of the feed item.	Optional	23.0–31.0
`url`	String	URL of the link.	Required	23.0–31.0
`urlName`	String	Name of the URL.	Optional	23.0–31.0
`visibility`	String	Who can see the post. Valid values are: • `InternalUsers`—If Communities is enabled, only members of the default, internal community can see the feed item. • `AllUsers`—If Communities is enabled, all members of the community can see the feed item. Default values:	Optional	26.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
		• For external users, the default value is `AllUsers`. External users must use this value to see their posts. • For internal users, the default value is `InternalUsers`. Internal users can accept this value or use the value `AllUsers` to allow external users to see their posts. If the parent of the feed item is a User or a Group, the `visibility` of the feed item must be `AllUsers`.		

Parameters for posting a poll

Parameter Name	Type	Description	Required or Optional	Available Version
`attachmentType`	String	The attachment type. Must contain the value `Poll`.	Required	27.0–31.0
`isBookmarkedByCurrentUser`	Boolean	Specifies whether to bookmark the new feed item for the user (`true`) or not (`false`). The default value is `false`.	Optional	26.0–31.0
`pollChoice1` through `pollChoice10`	String	The choices used to create a feed item with a poll. Specify 2–10 poll choices for each poll.	Required	26.0–31.0
`text`	String	Text of the feed item.	Required	23.0–31.0
`visibility`	String	Who can see the post. Valid values are: • `InternalUsers`—If Communities is enabled, only members of the default, internal community can see the feed item. • `AllUsers`—If Communities is enabled, all members of the community can see the feed item. Default values: • For external users, the default value is `AllUsers`. External users must use this value to see their posts. • For internal users, the default value is `InternalUsers`. Internal users can accept this value or use the value `AllUsers` to allow external users to see their posts.	Optional	26.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
		If the parent of the feed item is a User or a Group, the `visibility` of the feed item must be `AllUsers`.		

Parameters for sharing a post

Parameter Name	Type	Description	Required or Optional	Available Version
`isBookmarkedByCurrentUser`	Boolean	Specifies whether to bookmark the new feed item for the user (`true`) or not (`false`). The default value is `false`.	Optional	26.0–31.0
`originalFeedItemId`	String	The 18-character ID of the feed item you want to share. Note: Sharing a post takes precedence over uploading a file. Do not share a post and upload a file in the same post.	Required	24.0–31.0
`text`	String	A comment on the shared feed item.	Optional	28.0–31.0
`visibility`	String	Who can see the post. Valid values are: • `InternalUsers`—If Communities is enabled, only members of the default, internal community can see the feed item. • `AllUsers`—If Communities is enabled, all members of the community can see the feed item. Default values: • For external users, the default value is `AllUsers`. External users must use this value to see their posts. • For internal users, the default value is `InternalUsers`. Internal users can accept this value or use the value `AllUsers` to allow external users to see their posts. If the parent of the feed item is a User or a Group, the `visibility` of the feed item must be `AllUsers`.	Optional	26.0–31.0

Parameters for posting text only

Parameter Name	Type	Description	Required or Optional	Available Version
`isBookmarkedByCurrentUser`	Boolean	Specifies whether to bookmark the new feed item for the user (`true`) or not (`false`). The default value is `false`.	Optional	26.0–31.0
`text`	String	Text of the feed item. The maximum number of characters is 1000.	Required	23.0–31.0
`visibility`	String	Who can see the post. Valid values are: • `InternalUsers`—If Communities is enabled, only members of the default, internal community can see the feed item. • `AllUsers`—If Communities is enabled, all members of the community can see the feed item. Default values: • For external users, the default value is `AllUsers`. External users must use this value to see their posts. • For internal users, the default value is `InternalUsers`. Internal users can accept this value or use the value `AllUsers` to allow external users to see their posts. If the parent of the feed item is a User or a Group, the `visibility` of the feed item must be `AllUsers`.	Optional	26.0–31.0

Parameters for posting a canvas app

Parameter Name	Type	Description	Required or Optional	Available Version
`attachmentType`	String	The attachment type. Must contain the value `Canvas`.	Required	29.0–31.0
`description`	String	Description of the canvas app	Optional	29.0–31.0
`developerName`	String	The developer name (API name) of the canvas app	Required	29.0–31.0
`height`	String	The height of the canvas app in pixels. Default height is 200 pixels.	Optional	29.0–31.0
`isBookmarkedByCurrentUser`	Boolean	Specifies whether to bookmark the new feed item for the user (`true`) or not (`false`). The default value is `false`.	Optional	29.0–31.0
`namespacePrefix`	String	The namespace prefix of the Developer Edition organization in which the canvas app was created.	Optional	29.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
`parameters`	String	Parameters passed to the canvas app in JSON format. The JSON must be URL-encoded: `parameters={'isUpdated'='true';}"`	Optional	29.0–31.0
`text`	String	Text of the feed item	Optional	29.0–31.0
`thumbnailUrl`	String	The URL to a thumbnail image for the canvas app. Maximum dimensions are 120x120 pixels.	Optional	29.0–31.0
`title`	String	The title of the link to the canvas app	Optional	29.0–31.0
`visibility`	String	Who can see the post. Valid values are: • `InternalUsers`—If Communities is enabled, only members of the default, internal community can see the feed item. • `AllUsers`—If Communities is enabled, all members of the community can see the feed item. Default values: • For external users, the default value is `AllUsers`. External users must use this value to see their posts. • For internal users, the default value is `InternalUsers`. Internal users can accept this value or use the value `AllUsers` to allow external users to see their posts. If the parent of the feed item is a User or a Group, the `visibility` of the feed item must be `AllUsers`.	Optional	29.0–31.0

Response body for GET

Feed Item Page

Response body for POST

Feed Item

Example of file post

This example updates a news feed with a post that has a new file attachment.

```
curl https://instance_name/services/data/v31.0/chatter/feeds/news/me/feed-items
     --form "feedItemFileUpload=@Desktop/textFile.txt"
     --form "title=Sample"
     --form "description=Just a Text File"
     --form "text=Please review"
     --form "attachmentType=NewFile"
     --header 'Authorization: Bearer
00DD0000000Jhd2!AQIAQBgn42jbvCW2SX8JaRO2_JjX1fNKXWBrjYQPXODtDyaT6gT4DJVs4Ki7s6GoxvO8a.Z7gIHKkfcjW6S1.ujlD413PYpF'
```

This Java example updates a news feed with a file post.

```
import java.io.File;

import org.apache.commons.httpclient.*;
import org.apache.commons.httpclient.methods.PostMethod;
import org.apache.commons.httpclient.methods.multipart.*;

...

String oauthToken = "00Dx0000000X42V!AQgAQClp6UqGWNJUf4d3LQkVCysBIS6a";
String url = "https://instance_name/services/data/v31.0/chatter/" +
             "feeds/user-profile/me/feed-items";
String text = "I love posting files to Chatter!";
File contentFile = getFile();
String description = "This is a test file that I'm posting.";
String title = "contentFile";
Part[] parts = {
        new StringPart("description", description),
        new StringPart("title", fileName),
        new StringPart("text", text),
        new FilePart("feedItemFileUpload", contentFile),
};

final PostMethod postMethod = new PostMethod(url);

try {
    postMethod.setRequestEntity(new MultipartRequestEntity(parts,
                                postMethod.getParams()));
    postMethod.setRequestHeader("Authorization", "Bearer " + oauthToken);
    postMethod.setRequestHeader("Content-type", "multipart/form-data")
    postMethod.addRequestHeader("X-PrettyPrint", "1");

    HttpClient httpClient = new HttpClient();
    httpClient.getParams().setSoTimeout(60000);

    int returnCode = httpClient.executeMethod(postMethod);

    System.out.println(postMethod.getResponseBodyAsString());

    assertTrue("Expected return code of: " + HttpStatus.SC_CREATED,
                returnCode == HttpStatus.SC_CREATED);
} finally {
    postMethod.releaseConnection();
}
```

Example of a link post

This request creates a link post using the request parameters:

```
/chatter/feeds/news/me/feed-items?text=Did+you+see+this?&url=http://www.chatter.com
```

This request body creates a link in a feed item post:

```
{ "body" : {
     "messageSegments" : [
        {
```

```
            "type": "Link",
            "url" : "http://www.chatter.com"
         },
         {
            "type": "Text",
            "text" : "Is everyone following this?"
         }
      ]
   }
}
```

This request body includes a link as an attachment to a feed item post:

```
{ "body" : {
      "messageSegments" : [
         {
            "type": "Text",
            "text" : "Is everyone following this?"
         }
      ]
   },
  "attachment" :{
     "url" : "http://www.chatter.com",
     "urlName" : "The Future of Social Enterprise"
  }
}
```

News Feed Items Flat

This resource is deprecated as of version 26.0.

Returns a flat feed, that is, a feed where there is no hierarchy between feed items and comments: Feed items and comments are at the same level. All items are returned in chronological order.

Resource

```
/chatter/feeds/news/me/feed-items/flat
```

or

```
/chatter/feeds/news/userId/feed-items/flat
```

Note: *userId* must be the ID of the context user or the alias `me`.

Available since version

25.0–26.0

Important: As of API version 32.0, use `/chatter/feeds/news/userId/feed-elements`.

Requires Chatter

Yes

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
`numFeedItems`	Integer	Specifies the number of feed items returned. Valid values are from 0 through 100. Default value is 25. Because feed items can include comments, specifying 25 flat feed items may actually return more. For example, if each feed item has one comment, and `recentCommentCount` is greater than 1, 50 flat feed items are returned.
`page`	String	A generated token that indicates the view of feed items in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`.
`recentCommentCount`	Integer	Specifies the number of recent comments returned. Valid values are from 0 through 25. Default value is 3.
`sort`	String. One of the following: • `CreatedDateDesc` • `LastModifiedDateDesc`	Sorts the returned feed either by created date in ascending order, or by last modified date, also in ascending order. The default is `LastModifiedDateDesc`

Response body

Flat Feed

SEE ALSO:

Working with Feeds and Feed Elements

People Feed Resources

All feed elements posted by all of the people the context user follows.

Available resources are:

Resource	Description
`/chatter/feeds/people/me`	Returns a URL to the feed elements for all the people the context user is following.
`/chatter/feeds/people/me/feed-elements`	Returns all the feed elements from all the people the context user follows.

People Feed URL

Returns a URL to the feed elements for all the people the context user is following.

Resource

```
/chatter/feeds/people/userId
```

```
/connect/communities/communityId/chatter/feeds/people/userId
```

Note: *userId* must be the ID of the context user or the alias `me`.

Available since version

23.0

Requires Chatter

Yes

HTTP methods

GET, HEAD

Request parameters

Parameter Name	Type	Description
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.

Response body

Feed

People Feed Elements

Returns all the feed elements from all the people the context user follows.

Resource

```
/chatter/feeds/people/userId/feed-elements
```

```
/connect/communities/communityId/chatter/feeds/people/userId/feed-elements
```

Note: *userId* must be the ID of the context user or the alias `me`.

Available since version

31.0

Requires Chatter

Yes

HTTP methods

GET, HEAD

Request parameters

Parameter Name	Type	Description	Required or Optional	Available Version
`density`	String	The density of the feed. One of these values: • `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. • `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of, but hides some system-generated updates from records. The default value is `AllUpdates`.	Optional	31.0
`elementsPerBundle`	Integer	Maximum number of feed elements to include in a bundle. The value must be an integer between 0 and 10. The default value is 3.	Optional	31.0
`page`	String	A generated token that indicates the view of feed elements in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`, for example: `"nextPageUrl": "/services/data/v34.0/chatter/feeds/news/005D0000001GLowIAN/feed-elements?page=2011-03-30T17:34:50Z,0D5D0000000DSv4KAG"`	Optional	31.0
`pageSize`	Integer	Specifies the number of elements per page. Valid values are between 1 and 100. If you don't specify a size, the default is 25.	Optional	31.0
`q`	String	One or more keywords to search for in the body and comments of feed elements in this feed. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.	Optional	31.0
`recentCommentCount`	Integer	Maximum number of comments to include per feed element. The default value is 3. The maximum value is 25.	Optional	31.0

Parameter Name	Type	Description	Required or Optional	Available Version
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.	Optional	31.0
`updatedSince`	String	An opaque token defining the modification time stamp of the feed and the sort order. Do not construct this token. Retrieve this token from the `updatesToken` property of the Feed Element Page response body.	Optional	31.0

Response body

Feed Element Page

People Feed Items

Returns all the feed items from all the people the context user follows.

Resource

```
/chatter/feeds/people/me/feed-items
```

or

```
/chatter/feeds/people/userId/feed-items
```

Note: *`userId`* must be the ID of the context user or the alias `me`.

Available versions

23.0–31.0

Important: In version 32.0 and later, use People Feed Elements.

Requires Chatter

Yes

HTTP methods

GET, HEAD

Request parameters

Parameter Name	Type	Description	Required or Optional	Available Version
`density`	String	The density of the feed. One of these values: • `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. • `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of, but hides some system-generated updates from records. The default value is `AllUpdates`.	Optional	29.0–31.0
`page`	String	A generated token that indicates the view of feed elements in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`, for example: `"nextPageUrl": "/services/data/v34.0/chatter/feeds/news/005D0000001GLowIAN/feed-elements?page=2011-03-30T17:34:50Z,0D5D0000000DSv4KAG"`	Optional	23.0–31.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are between 1 and 100. If you don't specify a size, the default is 25.	Optional	23.0–31.0
`q`	String	One or more keywords to search for in the body and comments of feed elements in this feed. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.	Optional	28.0–31.0
`recentCommentCount`	Integer	Maximum number of comments to include per feed item. The default value is 3. The maximum value is 25.	Optional	29.0–31.0
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.	Optional	23.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
`updatedSince`	String	An opaque token defining the modification time stamp of the feed and the sort order. Do not construct this token. Retrieve this token from the `updatesToken` property of the Feed Element Page response body.	Optional	30.0–31.0

Response body

Feed Item Page

People Feed Items Flat

This resource is deprecated as of version 26.0.

Returns a flat feed, that is, a feed where there is no hierarchy between feed items and comments: Feed items and comments are at the same level. All items are returned in chronological order.

Resource

```
/chatter/feeds/people/me/feed-items/flat
```

or

```
/chatter/feeds/people/userId/feed-items/flat
```

Note: *`userId`* must be the ID of the context user or the alias `me`.

Available since version

25.0–26.0

Requires Chatter

Yes

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
`numFeedItems`	Integer	Specifies the number of feed items returned. Valid values are from 0 through 100. Default value is 25. Because feed items can include comments, specifying 25 flat feed items may actually return more. For example, if each feed item has one comment, and `recentCommentCount` is greater than 1, 50 flat feed items are returned.

Parameter Name	Type	Description
`page`	String	A generated token that indicates the view of feed items in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`.
`recentCommentCount`	Integer	Specifies the number of recent comments returned. Valid values are from 0 through 25. Default value is 3.
`sort`	String. One of the following: • `CreatedDateDesc` • `LastModifiedDateDesc`	Sorts the returned feed either by created date in ascending order, or by last modified date, also in ascending order. The default is `LastModifiedDateDesc`

Response body

Flat Feed

SEE ALSO:

Working with Feeds and Feed Elements

Record Feed Resource

The feed of the specified record, which could be a group, person, object, file and so on. You can post feed items to the record feed.

Available resources are:

Resource	Description
`/chatter/feeds/record/`***`recordId`***	Returns a URL to the feed elements for all the records the current user is following, or a URL to all the feed elements of the specified *`recordId`*. Use this resource to get a group feed.
`/chatter/feeds/record/`***`recordId`***`/feed-elements`	Returns the feed elements for all the records the context user is following, or all the feed elements of the specified *`recordId`*. Use this resource to search a feed or to get the feed elements for a specific feed. To use this resource to get the feed elements for a group, specify the group ID as the *`recordId`*.

Record Feed URL

Returns a URL to the feed elements for all the records the current user is following, or a URL to all the feed elements of the specified *`recordId`*. Use this resource to get a group feed.

Resource

```
/chatter/feeds/record/recordId
```

```
/connect/communities/communityId/chatter/feeds/record/recordId
```

Note: If *recordId* is a user ID, it must be the same as the current user or the alias `me`.

Available since version

23.0

Requires Chatter

Yes

HTTP methods

GET, HEAD

Request parameters

Parameter Name	Type	Description
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.

Response body

Feed

Record Feed Elements

Returns the feed elements for all the records the context user is following, or all the feed elements of the specified *recordId*. Use this resource to search a feed or to get the feed elements for a specific feed. To use this resource to get the feed elements for a group, specify the group ID as the *recordId*.

Resource

```
/chatter/feeds/record/recordId/feed-elements
```

```
/connect/communities/communityId/chatter/feeds/record/recordId/feed-elements
```

Note: If *recordId* is a user ID, it must be the same as the current user or the alias `me`.

Available since version

23.0

Requires Chatter

Yes

HTTP methods

GET, HEAD

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
`density`	String	The density of the feed. One of these values: • `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. • `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of, but hides some system-generated updates from records. The default value is `AllUpdates`.	Optional	31.0
`elementsPerBundle`	Integer	Maximum number of feed elements to include in a bundle. The value must be an integer from 0 through 10. The default value is 3. Note: Bundled posts contain feed-tracked changes. In Salesforce1 downloadable apps, bundled posts are in record feeds only.	Optional	31.0
`filter`	String	Specifies the filter values that can be applied to a feed. • `AllQuestions`—Only feed elements that are questions. • `CommunityScoped`—Reserved for future use. • `SolvedQuestions`—Only feed elements that are questions and that have a best answer. • `UnansweredQuestions`—Only feed elements that are questions and that don't have any answers. • `UnsolvedQuestions`—Only feed elements that are questions and that don't have a best answer.	Optional	32.0
`page`	String	A generated token that indicates the view of feed elements in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`, for example: `"nextPageUrl": "/services/data/v34.0/chatter/feeds/news/005D0000001GLowIAN`	Optional	31.0

Parameter Name	Type	Description	Required or Optional	Available Version
		`/feed-elements ?page=2011-03-30T17:34:50Z, 0D5D0000000DSv4KAG"`		
`pageSize`	Integer	Specifies the number of elements per page. Valid values are between 1 and 100. If you don't specify a size, the default is 25.	Optional	31.0
`q`	String	One or more keywords to search for in the body and comments of feed elements in this feed. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.	Optional	31.0
`recentCommentCount`	Integer	Maximum number of comments to include per feed element. The default value is 3. The maximum value is 25.	Optional	31.0
`showInternalOnly`	Boolean	Specifies whether to show only feed elements from internal (non-community) users (`true`), or not (`false`). The default value is `false`.	Optional	31.0
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.	Optional	31.0
`updatedSince`	String	An opaque token defining the modification time stamp of the feed and the sort order. Do not construct this token. Retrieve this token from the `updatesToken` property of the Feed Element Page response body.	Optional	31.0

Response body for GET

Feed Element Page

Record Feed Items

Returns the feed items for all the records the context user is following, or all the feed items of the specified *`recordId`*. Use this resource to post a new feed item and to search a feed. To use this resource to get the feed items for a specific group, specify the group ID as the *`recordId`*.

For information about posting a feed item, see Using POST, PATCH, or PUT for Input.

For information about uploading a binary file and attaching it to a feed item, see Uploading Binary Files.

Note: You cannot post a feed item to a Record Feed if the record is of type `ContentDocument`.

As of API version 29.0, feed items can contain 5000 characters. This change is backwards compatible; all earlier API versions support the 5000 character limit.

Resource

```
/chatter/feeds/record/me/feed-items
```

or

```
/chatter/feeds/record/recordId/feed-items
```

Note: If *recordId* is a user ID, it must be the same as the current user or the alias `me`.

Available versions

23.0–31.0

Important: In version 32.0 and later, use Record Feed Elements.

Requires Chatter

Yes

HTTP methods

GET, POST, HEAD

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
`density`	String	The density of the feed. One of these values: • `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. • `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of, but hides some system-generated updates from records. The default value is `AllUpdates`.	Optional	29.0–31.0
`page`	String	A generated token that indicates the view of feed items in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`, for example: `"nextPageUrl": "/services/data/v34.0/chatter/feeds/news/005D0000001GLowIAN`	Optional	23.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
		`/feed-items ?page=2011-03-30T17:34:50Z, 0D5D0000000DSv4KAG"`		
`pageSize`	Integer	Specifies the number of items per page. Valid values are from 1 through 100. If you don't specify a size, the default is 25.	Optional	23.0–31.0
`q`	String	One or more keywords to search for in the body and comments of feed items in this feed. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.	Optional	28.0–31.0
`recentCommentCount`	Integer	Maximum number of comments to include per feed item. The default value is 3. The maximum value is 25.	Optional	29.0–31.0
`showInternalOnly`	Boolean	Specifies whether to show only feed items from internal (non-community) users (`true`), or not (`false`). The default value is `false`.	Optional	30.0–31.0
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.	Optional	23.0–31.0
`updatedSince`	String	An opaque token defining the modification time stamp of the feed and the sort order. Do not construct this token. Retrieve this token from the `updatesToken` property of the Feed Item Page response body.	Optional	30.0–31.0

Request body for POST

Root XML tag

```
<feedItem>
```

JSON

The following example attaches a poll to a feed item:

```
{
   "body":{
      "messageSegments":[
```

```
            {
               "type":"Text",
               "text":"When should we meet for release planning?"
            }
         ]
      },
      "capabilities":{
         "poll":{
            "choices":[
               "Monday",
               "Tuesday"
            ]
         }
      },
      "feedElementType":"FeedItem",
      "subjectId":"me"
}
```

Feed items support more than 1 capability. The following example attaches a file to a feed item and bookmarks it.

```
{
   "body":{
      "messageSegments":[
         {
            "type":"Text",
            "text":"Check out the latest marketing collateral."
         }
      ]
   },
   "capabilities":{
      "bookmarks":{
         "isBookmarkedByCurrentUser": true
      },
      "content":{
         "contentDocumentId": "069D00000001uQO"
      }
   },
   "feedElementType":"FeedItem",
   "subjectId":"me"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`attachment`	Depends on type.	You can specify only one of these values in the attachment parameter. Values are: • Attachment Input: Canvas App (v29.0–31.0) • Attachment Input: Existing Content (v23.0–31.0)	Optional	

Name	Type	Description	Required or Optional	Available Version
		• Attachment Input: Feed Poll (v26.0–31.0) • Attachment Input: Link (v23.0–31.0) • Attachment Input: New File Upload (v23.0–31.0) **Important:** As of API version 32.0, use the `capabilities` property.		
`body`	Message Body Input	Description of message body. The body can contain up to 25 mentions. If you specify `originalFeedElementId` to share a feed item, use the `body` property to add the first comment to the feed item. To edit this property in a feed item, make a PATCH request to `/chatter/feed-elements/feedElementId`. Editing feed posts is supported in version 34.0 and later.	Required unless the feed item has a content capability or a link capability	23.0
`capabilities`	Feed Element Capabilities Input	The capabilities that define the behavior of this feed element.	Optional	31.0
`isBookmarkedByCurrentUser`	Boolean	Specifies whether to bookmark the new feed item for the user (`true`) or not (`false`). The default value is `false`. **Important:** As of API version 32.0, use the `capabilities.bookmarks.isBookmarkedByCurrentUser` property.	Optional	24.0–31.0
`feedElementType`	String	The type of feed element this input represents. The only valid value is `FeedItem`.	Required	31.0

Name	Type	Description	Required or Optional	Available Version
`originalFeedElementId`	String	To share a feed element, specify its 18-character ID. Note: You can't share a post and upload a file in the same request body.	Optional	31.0
`originalFeedItemId`	String	To share a feed item, specify its 18-character ID. Important: As of API version 32.0, use the `originalFeedElementId` property.	Optional	24.0–31.0
`subjectId`	String	The ID of the parent this feed element is being posted to. This value can be the ID of a user, group, or record, or the string `me` to indicate the context user.	Required	31.0
`visibility`	String	Who can see the post. Valid values are: • `InternalUsers`—If Communities is enabled, only members of the default, internal community can see the feed item. • `AllUsers`—If Communities is enabled, all members of the community can see the feed item. Default values: • For external users, the default value is `AllUsers`. External users must use this value to see their posts. • For internal users, the default value is `InternalUsers`. Internal users can accept this value or use the value `AllUsers` to allow external users to see their posts. If the parent of the feed item is a User or a Group, the	Optional	26.0

Name	Type	Description	Required or Optional	Available Version
		`visibility` of the feed item must be `AllUsers`.		

Request parameters for POST

Use one of the following sets of parameters. For example, you can't post a link and attach a file in the same request.

Parameters for attaching an existing file

Parameter Name	Type	Description	Required or Optional	Available Version
`attachmentType`	String	The attachment type. Must contain the value `ExistingContent`.	Required	27.0–31.0
`contentDocumentId`	String	18-character ID of an existing file.	Required	23.0–31.0
`isBookmarkedByCurrentUser`	Boolean	Specifies whether to bookmark the new feed item for the user (`true`) or not (`false`). The default value is `false`.	Optional	26.0–31.0
`text`	String	Text of the feed item.	Optional	23.0–31.0
`visibility`	String	Who can see the post. Valid values are: • `InternalUsers`—If Communities is enabled, only members of the default, internal community can see the feed item. • `AllUsers`—If Communities is enabled, all members of the community can see the feed item. Default values: • For external users, the default value is `AllUsers`. External users must use this value to see their posts. • For internal users, the default value is `InternalUsers`. Internal users can accept this value or use the value `AllUsers` to allow external users to see their posts. If the parent of the feed item is a User or a Group, the `visibility` of the feed item must be `AllUsers`.	Optional	26.0–31.0

Parameters for attaching a new file

Parameter Name	Type	Description	Required or Optional	Available Version
`attachmentType`	String	The attachment type. Must contain the value `NewFile`.	Required	27.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
`desc`	String	Description of file being posted. **Attention:** Don't use this parameter as of version 27.0. Use `description` instead.	Optional	23.0–26.0
`description`	String	Description of file being posted.	Optional	27.0–31.0
`fileName`	String	File name of the file being posted. **Attention:** Don't use this parameter as of version 24.0. Use `title` instead.	Required	23.0–24.0
`isBookmarkedByCurrentUser`	Boolean	Specifies whether to bookmark the new feed item for the user (`true`) or not (`false`). The default value is `false`.	Optional	26.0–31.0
`text`	String	Text of the feed item.	Optional	23.0–31.0
`title`	String	File name of a file to be uploaded and posted.	Required	24.0–31.0
`visibility`	String	Who can see the post. Valid values are: • `InternalUsers`—If Communities is enabled, only members of the default, internal community can see the feed item. • `AllUsers`—If Communities is enabled, all members of the community can see the feed item. Default values: • For external users, the default value is `AllUsers`. External users must use this value to see their posts. • For internal users, the default value is `InternalUsers`. Internal users can accept this value or use the value `AllUsers` to allow external users to see their posts. If the parent of the feed item is a User or a Group, the `visibility` of the feed item must be `AllUsers`.	Optional	26.0–31.0

Parameters for posting a link

Parameter Name	Type	Description	Required or Optional	Available Version
`attachmentType`	String	The attachment type. Must contain the value `Link`.	Required	27.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
`isBookmarkedByCurrentUser`	Boolean	Specifies whether to bookmark the new feed item for the user (`true`) or not (`false`). The default value is `false`.	Optional	26.0–31.0
`text`	String	Text of the feed item.	Optional	23.0–31.0
`url`	String	URL of the link.	Required	23.0–31.0
`urlName`	String	Name of the URL.	Optional	23.0–31.0
`visibility`	String	Who can see the post. Valid values are: • `InternalUsers`—If Communities is enabled, only members of the default, internal community can see the feed item. • `AllUsers`—If Communities is enabled, all members of the community can see the feed item. Default values: • For external users, the default value is `AllUsers`. External users must use this value to see their posts. • For internal users, the default value is `InternalUsers`. Internal users can accept this value or use the value `AllUsers` to allow external users to see their posts. If the parent of the feed item is a User or a Group, the `visibility` of the feed item must be `AllUsers`.	Optional	26.0–31.0

Parameters for posting a poll

Parameter Name	Type	Description	Required or Optional	Available Version
`attachmentType`	String	The attachment type. Must contain the value `Poll`.	Required	27.0–31.0
`isBookmarkedByCurrentUser`	Boolean	Specifies whether to bookmark the new feed item for the user (`true`) or not (`false`). The default value is `false`.	Optional	26.0–31.0
`pollChoice1` through `pollChoice10`	String	The choices used to create a feed item with a poll. Specify 2–10 poll choices for each poll.	Required	26.0–31.0
`text`	String	Text of the feed item.	Required	23.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
`visibility`	String	Who can see the post. Valid values are: • `InternalUsers`—If Communities is enabled, only members of the default, internal community can see the feed item. • `AllUsers`—If Communities is enabled, all members of the community can see the feed item. Default values: • For external users, the default value is `AllUsers`. External users must use this value to see their posts. • For internal users, the default value is `InternalUsers`. Internal users can accept this value or use the value `AllUsers` to allow external users to see their posts. If the parent of the feed item is a User or a Group, the `visibility` of the feed item must be `AllUsers`.	Optional	26.0–31.0

Parameters for sharing a post

Parameter Name	Type	Description	Required or Optional	Available Version
`isBookmarkedByCurrentUser`	Boolean	Specifies whether to bookmark the new feed item for the user (`true`) or not (`false`). The default value is `false`.	Optional	26.0–31.0
`originalFeedItemId`	String	The 18-character ID of the feed item you want to share. Note: Sharing a post takes precedence over uploading a file. Do not share a post and upload a file in the same post.	Required	24.0–31.0
`text`	String	A comment on the shared feed item.	Optional	28.0–31.0
`visibility`	String	Who can see the post. Valid values are: • `InternalUsers`—If Communities is enabled, only members of the default, internal community can see the feed item.	Optional	26.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
		• `AllUsers`—If Communities is enabled, all members of the community can see the feed item. Default values: • For external users, the default value is `AllUsers`. External users must use this value to see their posts. • For internal users, the default value is `InternalUsers`. Internal users can accept this value or use the value `AllUsers` to allow external users to see their posts. If the parent of the feed item is a User or a Group, the `visibility` of the feed item must be `AllUsers`.		

Parameters for posting text only

Parameter Name	Type	Description	Required or Optional	Available Version
`isBookmarkedByCurrentUser`	Boolean	Specifies whether to bookmark the new feed item for the user (`true`) or not (`false`). The default value is `false`.	Optional	26.0–31.0
`text`	String	Text of the feed item. The maximum number of characters is 1000.	Required	23.0–31.0
`visibility`	String	Who can see the post. Valid values are: • `InternalUsers`—If Communities is enabled, only members of the default, internal community can see the feed item. • `AllUsers`—If Communities is enabled, all members of the community can see the feed item. Default values: • For external users, the default value is `AllUsers`. External users must use this value to see their posts. • For internal users, the default value is `InternalUsers`. Internal users can accept this value or use the value `AllUsers` to allow external users to see their posts.	Optional	26.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
		If the parent of the feed item is a User or a Group, the `visibility` of the feed item must be `AllUsers`.		

Parameters for posting a canvas app

Parameter Name	Type	Description	Required or Optional	Available Version
`attachmentType`	String	The attachment type. Must contain the value `Canvas`.	Required	29.0–31.0
`description`	String	Description of the canvas app	Optional	29.0–31.0
`developerName`	String	The developer name (API name) of the canvas app	Required	29.0–31.0
`height`	String	The height of the canvas app in pixels. Default height is 200 pixels.	Optional	29.0–31.0
`isBookmarkedByCurrentUser`	Boolean	Specifies whether to bookmark the new feed item for the user (`true`) or not (`false`). The default value is `false`.	Optional	29.0–31.0
`namespacePrefix`	String	The namespace prefix of the Developer Edition organization in which the canvas app was created.	Optional	29.0–31.0
`parameters`	String	Parameters passed to the canvas app in JSON format. The JSON must be URL-encoded: `parameters={'isUpdated'='true';}"`	Optional	29.0–31.0
`text`	String	Text of the feed item	Optional	29.0–31.0
`thumbnailUrl`	String	The URL to a thumbnail image for the canvas app. Maximum dimensions are 120x120 pixels.	Optional	29.0–31.0
`title`	String	The title of the link to the canvas app	Optional	29.0–31.0
`visibility`	String	Who can see the post. Valid values are: • `InternalUsers`—If Communities is enabled, only members of the default, internal community can see the feed item. • `AllUsers`—If Communities is enabled, all members of the community can see the feed item. Default values: • For external users, the default value is `AllUsers`. External users must use this value to see their posts.	Optional	29.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
		• For internal users, the default value is `InternalUsers`. Internal users can accept this value or use the value `AllUsers` to allow external users to see their posts. If the parent of the feed item is a User or a Group, the `visibility` of the feed item must be `AllUsers`.		

Response body for GET

Feed Item Page

Response body for POST

Feed Item

Example for GET

This request searches for feed items that contain the string `track`:

```
/chatter/feeds/record/me/feed-items?q=track
```

In the Feed Item Page response body, the string `track` is found in the `feed-items/items/body/text` property.

Record Feed Items Flat

This resource is deprecated as of version 26.0.

Returns a flat feed, that is, a feed where there is no hierarchy between feed items and comments: Feed items and comments are at the same level. All items are returned in chronological order.

Resource

```
/chatter/feeds/record/me/feed-items/flat
```

or

```
/chatter/feeds/record/userId/feed-items/flat
```

Note: *`userId`* must be the ID of the context user or the alias `me`.

Available since version

25.0–26.0

Requires Chatter

Yes

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
`numFeedItems`	Integer	Specifies the number of feed items returned. Valid values are from 0 through 100. Default value is 25. Because feed items can include comments, specifying 25 flat feed items may actually return more. For example, if each feed item has one comment, and `recentCommentCount` is greater than 1, 50 flat feed items are returned.
`page`	String	A generated token that indicates the view of feed items in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`.
`recentCommentCount`	Integer	Specifies the number of recent comments returned. Valid values are from 0 through 25. Default value is 3.
`sort`	String. One of the following: • `CreatedDateDesc` • `LastModifiedDateDesc`	Sorts the returned feed either by created date in ascending order, or by last modified date, also in ascending order. The default is `LastModifiedDateDesc`

Response body

Flat Feed

SEE ALSO:

Records Resources

Working with Feeds and Feed Elements

Get a Record Feed

To Feed Resources

A feed containing all @mentions of the context user and posts others have made to the context user's feed.

Available resources are:

Resource	Description
`/chatter/feeds/to/me`	Returns a URL to the feed elements for the feed of the context user.
`/chatter/feeds/to/me/feed-elements`	Returns the feed elements for all @mentions of the context user, and all posts made to the user's feed.

To Feed URL

Returns a URL to the feed elements for the feed of the context user.

Resource

```
/chatter/feeds/to/userId
```

```
/connect/communities/communityId/chatter/feeds/to/userId
```

Note: *userId* must be the ID of the context user or the alias `me`.

Available since version

23.0

Requires Chatter

Yes

HTTP methods

GET, HEAD

Request parameters

Parameter Name	Type	Description
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.

Response body

Feed

To Feed Elements

Returns the feed elements for all @mentions of the context user, and all posts made to the user's feed. The To Feed includes posts that mention you and posts that other users make to your wall. Your own posts are included only if they have a comment. The comment can be from you or from another user.

Resource

```
/chatter/feeds/to/userId/feed-elements
```

```
/connect/communities/communityId/chatter/feeds/to/userId/feed-elements
```

Note: *userId* must be the ID of the context user or the alias `me`.

Available since version

31.0

Requires Chatter
Yes

HTTP methods
GET, HEAD

Request parameters

Parameter Name	Type	Description	Required or Optional	Available Version
`density`	String	The density of the feed. One of these values: • `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. • `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of, but hides some system-generated updates from records. The default value is `AllUpdates`.	Optional	31.0
`elementsPerBundle`	Integer	Maximum number of feed elements to include in a bundle. The value must be an integer from 0 through 10. The default value is 3. Note: Bundled posts contain feed-tracked changes. In Salesforce1 downloadable apps, bundled posts are in record feeds only.	Optional	31.0
`page`	String	A generated token that indicates the view of feed elements in the feed.Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`, for example: `"nextPageUrl": "/services/data/v34.0/chatter/feeds/news/005D0000001GLowIAN/feed-elements?page=2011-03-30T17:34:50Z,0D5D0000000DSv4KAG"`	Optional	31.0
`pageSize`	Integer	The number of feed elements per page. Valid values are between 1 and 100. The default size is 25.	Optional	31.0
`q`	String	One or more keywords to search for in the body and comments of feed elements in this feed. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.	Optional	31.0

Parameter Name	Type	Description	Required or Optional	Available Version
`recentCommentCount`	Integer	Maximum number of comments to include per feed element. The default value is 3. The maximum value is 25.	Optional	31.0
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.	Optional	31.0

Response body for GET

Feed Element Page

To Feed Items

Returns the feed items for all @mentions of the context user, and all posts made to the user's feed. The To Feed includes posts that mention you and posts that other users make to your wall. Your own posts are included only if they have a comment. The comment can be from you or from another user.

Resource

```
/chatter/feeds/to/me/feed-items
```

OR

```
/chatter/feeds/to/userId/feed-items
```

Note: *`userId`* must be the ID of the context user or the alias `me`.

Available versions

23.0–31.0

Important: In version 32.0 and later, use To Feed Elements.

Requires Chatter

Yes

HTTP methods

GET, HEAD

Request parameters

Parameter Name	Type	Description	Required or Optional	Available Version
`density`	String	The density of the feed. One of these values: • `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. • `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of, but hides some system-generated updates from records. The default value is `AllUpdates`.	Optional	29.0–31.0
`page`	String	A generated token that indicates the view of feed items in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`, for example: `"nextPageUrl": "/services/data/v34.0/chatter/feeds/news/005D0000001GLowIAN/feed-items?page=2011-03-30T17:34:50Z,0D5D0000000DSv4KAG"`	Optional	23.0–31.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are from 1 through 100. If you don't specify a size, the default is 25.	Optional	23.0–31.0
`q`	String	One or more keywords to search for in the body and comments of feed items in this feed. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.	Optional	28.0–31.0
`recentCommentCount`	Integer	Maximum number of comments to include per feed item. The default value is 3. The maximum value is 25.	Optional	29.0–31.0
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.	Optional	23.0–31.0

Response body for GET
Feed Item Page

To Feed Items Flat

This resource is deprecated as of version 26.0.

Returns a flat feed, that is, a feed where there is no hierarchy between feed items and comments: Feed items and comments are at the same level. All items are returned in chronological order.

Resource

```
/chatter/feeds/to/me/feed-items/flat
```

or

```
/chatter/feeds/to/userId/feed-items/flat
```

Note: *userId* must be the ID of the context user or the alias `me`.

Available since version
25.0–26.0

Requires Chatter
Yes

HTTP methods
GET or HEAD

Request parameters

Parameter Name	Type	Description
`numFeedItems`	Integer	Specifies the number of feed items returned. Valid values are from 0 through 100. Default value is 25. Because feed items can include comments, specifying 25 flat feed items may actually return more. For example, if each feed item has one comment, and `recentCommentCount` is greater than 1, 50 flat feed items are returned.
`page`	String	A generated token that indicates the view of feed items in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`.
`recentCommentCount`	Integer	Specifies the number of recent comments returned. Valid values are from 0 through 25. Default value is 3.
`sort`	String. One of the following: • `CreatedDateDesc`	Sorts the returned feed either by created date in ascending order, or by last modified

Parameter Name	Type	Description
	• `LastModifiedDateDesc`	date, also in ascending order. The default is `LastModifiedDateDesc`

Response body

Flat Feed

SEE ALSO:

Working with Feeds and Feed Elements

Topics Feed Resources

The feed of the specified topic.

Available resources are:

Resource	Description
`/chatter/feeds/topics/`***`topicId`***	Returns a URL to the feed items for the specified topic.
`/chatter/feeds/topics/`***`topicId`***`/feed-elements`	Returns the feed elements for all @mentions of the context user, and all posts made to the user's feed.

Topics Feed URL

Returns a URL to the feed items for the specified topic.

Resource

```
/chatter/feeds/topics/topicId
```

```
/connect/communities/communityId/chatter/feeds/topics/topicId
```

Available since version

28.0

Requires Chatter

Yes

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date.

Parameter Name	Type	Description
		• `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.

Response body

Feed

Topics Feed Elements

Returns all feed elements for the specified topic.

Resource

```
/chatter/feeds/topics/topicId/feed-elements
```

```
/connect/communities/communityId/chatter/feeds/topics/topicId/feed-elements
```

Available since version

31.0

Requires Chatter

Yes

HTTP methods

GET, HEAD

Request parameters

Parameter Name	Type	Description	Required or Optional	Available Version
`filter`	String	Specifies the filter values that can be applied to a feed. • `AllQuestions`—Only feed elements that are questions. • `CommunityScoped`—Reserved for future use. • `SolvedQuestions`—Only feed elements that are questions and that have a best answer. • `UnansweredQuestions`—Only feed elements that are questions and that don't have any answers. • `UnsolvedQuestions`—Only feed elements that are questions and that don't have a best answer.	Optional	32.0
`page`	String	A generated token that indicates the view of feed elements in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`.	Optional	31.0

Parameter Name	Type	Description	Required or Optional	Available Version
`pageSize`	Integer	Specifies the number of feed elements per page. Valid values are between 1 and 100. The default value is 25.	Optional	31.0
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.	Optional	31.0

Response body

Feed Element Page

Topics Feed Items

Returns all feed items for the specified topic. To post a topic, use the Feed-Items Topics resource.

Resource

```
/chatter/feeds/topics/topicId/feed-items
```

Available versions

28.0–31.0

Important: In version 32.0 and later, use Topics Feed Elements.

Requires Chatter

Yes

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description	Required or Optional	Available Version
`page`	String	A generated token that indicates the view of feed items in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`.	Optional	28.0–31.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are from 1 through 100. If you don't specify a size, the default is 25.	Optional	28.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.	Optional	28.0–31.0

Response body

Feed Item Page

SEE ALSO:

Working with Feeds and Feed Elements

User Profile Feed Resources

A feed containing actions by the current user on records that can be tracked in a feed. This feed is different than `/chatter/feeds/news`, which returns more elements, including posts, group updates, and record updates. You can post feed items to the user-profile feed.

Note: To match the feed displayed in the Chatter Web UI, the composition of this feed may change between releases.

Available resources are:

Resource	Description
`/chatter/feeds/user-profile/me`	Returns a URL to the feed elements for the user profile feed.
`/chatter/feeds/user-profile/me/feed-elements`	Returns the feed elements for all actions on all records that can be tracked in a feed.

User Profile Feed URL

Returns a URL to the feed elements for the user profile feed.

Resource

```
/chatter/feeds/user-profile/userId
```

```
/connect/communities/communityId/chatter/feeds/user-profile/userId
```

Response body

Feed

User Profile Feed Elements

Returns the feed elements for all actions on all records that can be tracked in a feed.

Resource

```
/chatter/feeds/user-profile/userId/feed-elements
```

```
/connect/communities/communityId/chatter/feeds/user-profile/userId/feed-elements
```

Available since version

31.0

Requires Chatter

Yes

HTTP methods

GET, HEAD

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
`density`	String	The density of the feed. One of these values: • `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. • `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of, but hides some system-generated updates from records. The default value is `AllUpdates`.	Optional	31.0
`elementsPerBundle`	Integer	Maximum number of feed elements to include in a bundle. The value must be an integer from 0 through 10. The default value is 3. Note: Bundled posts contain feed-tracked changes. In Salesforce1 downloadable apps, bundled posts are in record feeds only.	Optional	31.0
`filter`	String	The only value is `CommunityScoped`. Filters the feed to include only feed elements that are scoped to communities. Feed elements that are always visible in all communities are filtered out. Currently, feed elements scoped to communities have a User or a Group parent record. However, other parent record types could be scoped to communities in the future.		
`page`	String	A generated token that indicates the view of feed elements in the feed. Page tokens are returned as part	Optional	31.0

Parameter Name	Type	Description	Required or Optional	Available Version
		of the response body, in one of the URL parameters, such as `nextPageURL`: `"nextPageUrl": "/services/data/v34.0/chatter/feeds/news/005D0000001GLowIAN/feed-elements?page=2011-03-30T17:34:50Z,0D5D0000000DSv4KAG"`		
`pageSize`	Integer	Specifies the number of feed elements per page. Valid values are from 1 through 100. The default value is 25.	Optional	31.0
`q`	String	One or more keywords to search for in the body and comments of feed elements in this feed. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.	Optional	31.0
`recentCommentCount`	Integer	Maximum number of comments to include per feed element. The default value is 3. The maximum value is 25.	Optional	31.0
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.	Optional	31.0

Response body for GET

Feed Element Page

User Profile Feed Items

Returns the feed items for all actions on all records that can be tracked in a feed. You can also use this resource to post a new feed item.

For information about posting a feed item, see Using POST, PATCH, or PUT for Input.

To upload a binary file, you must send it as a body part in a multipart/form-data request. You can send information such as the text of a post or comment as a JSON or XML rich input body part in the same multipart/form-data request. Alternately, you can choose to send that information in request parameters. If you pass both a rich input request body and request parameters, the request parameters are ignored. For information about how to create the multipart/form-data message, see Uploading Binary Files.

As of API version 29.0, feed items can contain 5000 characters. This change is backwards compatible; all earlier API versions support the 5000 character limit. In API versions 23.0 and 24.0, posts to the User Profile Feed Items resource created user status updates; they didn't create feed items. Therefore, the character limit for the User Profile Feed Items resource for those API versions is still 1000 characters.

Resource

```
/chatter/feeds/user-profile/me/feed-items
```

or

```
/chatter/feeds/user-profile/userId/feed-items
```

Available versions

23.0–31.0

Important: In version 32.0 and later, use User Profile Feed Elements.

Requires Chatter

Yes

HTTP methods

GET, POST or HEAD

Request body for POST

Root XML tag

```
<feedItem>
```

JSON

The following example attaches a poll to a feed item:

```
{
   "body":{
      "messageSegments":[
         {
            "type":"Text",
            "text":"When should we meet for release planning?"
         }
      ]
   },
   "capabilities":{
      "poll":{
         "choices":[
            "Monday",
            "Tuesday"
         ]
      }
   },
   "feedElementType":"FeedItem",
   "subjectId":"me"
}
```

Feed items support more than 1 capability. The following example attaches a file to a feed item and bookmarks it.

```
{
   "body":{
      "messageSegments":[
         {
```

```
            "type":"Text",
            "text":"Check out the latest marketing collateral."
         }
      ]
   },
   "capabilities":{
      "bookmarks":{
         "isBookmarkedByCurrentUser": true
      },
      "content":{
         "contentDocumentId": "069D00000001uQO"
      }
   },
   "feedElementType":"FeedItem",
   "subjectId":"me"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`attachment`	Depends on type.	You can specify only one of these values in the attachment parameter. Values are: • Attachment Input: Canvas App (v29.0–31.0) • Attachment Input: Existing Content (v23.0–31.0) • Attachment Input: Feed Poll (v26.0–31.0) • Attachment Input: Link (v23.0–31.0) • Attachment Input: New File Upload (v23.0–31.0) **Important:** As of API version 32.0, use the `capabilities` property.	Optional	
`body`	Message Body Input	Description of message body. The body can contain up to 25 mentions. If you specify `originalFeedElementId` to share a feed item, use the `body` property to add the first comment to the feed item. To edit this property in a feed item, make a PATCH request to	Required unless the feed item has a content capability or a link capability	23.0

Name	Type	Description	Required or Optional	Available Version
		`/chatter/feed-elements/feedElementId`. Editing feed posts is supported in version 34.0 and later.		
`capabilities`	Feed Element Capabilities Input	The capabilities that define the behavior of this feed element.	Optional	31.0
`isBookmarkedByCurrentUser`	Boolean	Specifies whether to bookmark the new feed item for the user (`true`) or not (`false`). The default value is `false`. **Important:** As of API version 32.0, use the `capabilities.bookmarks.isBookmarkedByCurrentUser` property.	Optional	24.0–31.0
`feedElementType`	String	The type of feed element this input represents. The only valid value is `FeedItem`.	Required	31.0
`originalFeedElementId`	String	To share a feed element, specify its 18-character ID. **Note:** You can't share a post and upload a file in the same request body.	Optional	31.0
`originalFeedItemId`	String	To share a feed item, specify its 18-character ID. **Important:** As of API version 32.0, use the `originalFeedElementId` property.	Optional	24.0–31.0
`subjectId`	String	The ID of the parent this feed element is being posted to. This value can be the ID of a user, group, or record, or the string `me` to indicate the context user.	Required	31.0
`visibility`	String	Who can see the post. Valid values are: • `InternalUsers`—If Communities is enabled, only	Optional	26.0

Name	Type	Description	Required or Optional	Available Version
		members of the default, internal community can see the feed item. • `AllUsers`—If Communities is enabled, all members of the community can see the feed item. Default values: • For external users, the default value is `AllUsers`. External users must use this value to see their posts. • For internal users, the default value is `InternalUsers`. Internal users can accept this value or use the value `AllUsers` to allow external users to see their posts. If the parent of the feed item is a User or a Group, the `visibility` of the feed item must be `AllUsers`.		

Request parameters for POST

Use one of the following sets of parameters. For example, you can't post a link and attach a file in the same request.

Parameters for attaching an existing file

Parameter Name	Type	Description	Required or Optional	Available Version
`attachmentType`	String	The attachment type. Must contain the value `ExistingContent`.	Required	27.0–31.0
`contentDocumentId`	String	18-character ID of an existing file.	Required	23.0–31.0
`isBookmarkedByCurrentUser`	Boolean	Specifies whether to bookmark the new feed item for the user (`true`) or not (`false`). The default value is `false`.	Optional	26.0–31.0
`text`	String	Text of the feed item.	Optional	23.0–31.0
`visibility`	String	Who can see the post. Valid values are: • `InternalUsers`—If Communities is enabled, only members of the default, internal community can see the feed item.	Optional	26.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
		• `AllUsers`—If Communities is enabled, all members of the community can see the feed item. Default values: • For external users, the default value is `AllUsers`. External users must use this value to see their posts. • For internal users, the default value is `InternalUsers`. Internal users can accept this value or use the value `AllUsers` to allow external users to see their posts. If the parent of the feed item is a User or a Group, the `visibility` of the feed item must be `AllUsers`.		

Parameters for attaching a new file

Parameter Name	Type	Description	Required or Optional	Available Version
`attachmentType`	String	The attachment type. Must contain the value `NewFile`.	Required	27.0–31.0
`desc`	String	Description of file being posted. **Attention:** Don't use this parameter as of version 27.0. Use `description` instead.	Optional	23.0–26.0
`description`	String	Description of file being posted.	Optional	27.0–31.0
`fileName`	String	File name of the file being posted. **Attention:** Don't use this parameter as of version 24.0. Use `title` instead.	Required	23.0–24.0
`isBookmarkedByCurrentUser`	Boolean	Specifies whether to bookmark the new feed item for the user (`true`) or not (`false`). The default value is `false`.	Optional	26.0–31.0
`text`	String	Text of the feed item.	Optional	23.0–31.0
`title`	String	File name of a file to be uploaded and posted.	Required	24.0–31.0
`visibility`	String	Who can see the post. Valid values are: • `InternalUsers`—If Communities is enabled, only members of the default, internal community can see the feed item. • `AllUsers`—If Communities is enabled, all members of the community can see the feed item.	Optional	26.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
		Default values: • For external users, the default value is `AllUsers`. External users must use this value to see their posts. • For internal users, the default value is `InternalUsers`. Internal users can accept this value or use the value `AllUsers` to allow external users to see their posts. If the parent of the feed item is a User or a Group, the `visibility` of the feed item must be `AllUsers`.		

Parameters for posting a link

Parameter Name	Type	Description	Required or Optional	Available Version
`attachmentType`	String	The attachment type. Must contain the value `Link`.	Required	27.0–31.0
`isBookmarkedByCurrentUser`	Boolean	Specifies whether to bookmark the new feed item for the user (`true`) or not (`false`). The default value is `false`.	Optional	26.0–31.0
`text`	String	Text of the feed item.	Optional	23.0–31.0
`url`	String	URL of the link.	Required	23.0–31.0
`urlName`	String	Name of the URL.	Optional	23.0–31.0
`visibility`	String	Who can see the post. Valid values are: • `InternalUsers`—If Communities is enabled, only members of the default, internal community can see the feed item. • `AllUsers`—If Communities is enabled, all members of the community can see the feed item. Default values: • For external users, the default value is `AllUsers`. External users must use this value to see their posts. • For internal users, the default value is `InternalUsers`. Internal users can accept this value or use the value `AllUsers` to allow external users to see their posts.	Optional	26.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
		If the parent of the feed item is a User or a Group, the `visibility` of the feed item must be `AllUsers`.		

Parameters for posting a poll

Parameter Name	Type	Description	Required or Optional	Available Version
`attachmentType`	String	The attachment type. Must contain the value `Poll`.	Required	27.0–31.0
`isBookmarked ByCurrentUser`	Boolean	Specifies whether to bookmark the new feed item for the user (`true`) or not (`false`). The default value is `false`.	Optional	26.0–31.0
`pollChoice1` through `pollChoice10`	String	The choices used to create a feed item with a poll. Specify 2–10 poll choices for each poll.	Required	26.0–31.0
`text`	String	Text of the feed item.	Required	23.0–31.0
`visibility`	String	Who can see the post. Valid values are: • `InternalUsers`—If Communities is enabled, only members of the default, internal community can see the feed item. • `AllUsers`—If Communities is enabled, all members of the community can see the feed item. Default values: • For external users, the default value is `AllUsers`. External users must use this value to see their posts. • For internal users, the default value is `InternalUsers`. Internal users can accept this value or use the value `AllUsers` to allow external users to see their posts. If the parent of the feed item is a User or a Group, the `visibility` of the feed item must be `AllUsers`.	Optional	26.0–31.0

Parameters for sharing a post

Parameter Name	Type	Description	Required or Optional	Available Version
`isBookmarkedByCurrentUser`	Boolean	Specifies whether to bookmark the new feed item for the user (`true`) or not (`false`). The default value is `false`.	Optional	26.0–31.0
`originalFeedItemId`	String	The 18-character ID of the feed item you want to share. Note: Sharing a post takes precedence over uploading a file. Do not share a post and upload a file in the same post.	Required	24.0–31.0
`text`	String	A comment on the shared feed item.	Optional	28.0–31.0
`visibility`	String	Who can see the post. Valid values are: • `InternalUsers`—If Communities is enabled, only members of the default, internal community can see the feed item. • `AllUsers`—If Communities is enabled, all members of the community can see the feed item. Default values: • For external users, the default value is `AllUsers`. External users must use this value to see their posts. • For internal users, the default value is `InternalUsers`. Internal users can accept this value or use the value `AllUsers` to allow external users to see their posts. If the parent of the feed item is a User or a Group, the `visibility` of the feed item must be `AllUsers`.	Optional	26.0–31.0

Parameters for posting text only

Parameter Name	Type	Description	Required or Optional	Available Version
`isBookmarkedByCurrentUser`	Boolean	Specifies whether to bookmark the new feed item for the user (`true`) or not (`false`). The default value is `false`.	Optional	26.0–31.0
`text`	String	Text of the feed item. The maximum number of characters is 1000.	Required	23.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
`visibility`	String	Who can see the post. Valid values are: • `InternalUsers`—If Communities is enabled, only members of the default, internal community can see the feed item. • `AllUsers`—If Communities is enabled, all members of the community can see the feed item. Default values: • For external users, the default value is `AllUsers`. External users must use this value to see their posts. • For internal users, the default value is `InternalUsers`. Internal users can accept this value or use the value `AllUsers` to allow external users to see their posts. If the parent of the feed item is a User or a Group, the `visibility` of the feed item must be `AllUsers`.	Optional	26.0–31.0

Parameters for posting a canvas app

Parameter Name	Type	Description	Required or Optional	Available Version
`attachmentType`	String	The attachment type. Must contain the value `Canvas`.	Required	29.0–31.0
`description`	String	Description of the canvas app	Optional	29.0–31.0
`developerName`	String	The developer name (API name) of the canvas app	Required	29.0–31.0
`height`	String	The height of the canvas app in pixels. Default height is 200 pixels.	Optional	29.0–31.0
`isBookmarkedByCurrentUser`	Boolean	Specifies whether to bookmark the new feed item for the user (`true`) or not (`false`). The default value is `false`.	Optional	29.0–31.0
`namespacePrefix`	String	The namespace prefix of the Developer Edition organization in which the canvas app was created.	Optional	29.0–31.0
`parameters`	String	Parameters passed to the canvas app in JSON format. The JSON must be URL-encoded: `parameters={'isUpdated'='true';}"`	Optional	29.0–31.0
`text`	String	Text of the feed item	Optional	29.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
`thumbnailUrl`	String	The URL to a thumbnail image for the canvas app. Maximum dimensions are 120x120 pixels.	Optional	29.0–31.0
`title`	String	The title of the link to the canvas app	Optional	29.0–31.0
`visibility`	String	Who can see the post. Valid values are: • `InternalUsers`—If Communities is enabled, only members of the default, internal community can see the feed item. • `AllUsers`—If Communities is enabled, all members of the community can see the feed item. Default values: • For external users, the default value is `AllUsers`. External users must use this value to see their posts. • For internal users, the default value is `InternalUsers`. Internal users can accept this value or use the value `AllUsers` to allow external users to see their posts. If the parent of the feed item is a User or a Group, the `visibility` of the feed item must be `AllUsers`.	Optional	29.0–31.0

Request parameters for GET or HEAD

Parameter Name	Type	Description	Required or Optional	Available Version
`density`	String	The density of the feed. One of these values: • `AllUpdates`—Displays all updates from people and records the user follows and groups the user is a member of. • `FewerUpdates`—Displays all updates from people and records the user follows and groups the user is a member of, but hides some system-generated updates from records. The default value is `AllUpdates`.	Optional	29.0–31.0
`page`	String	A generated token that indicates the view of feed items in the feed. Page tokens are returned as part of the	Optional	23.0–31.0

Parameter Name	Type	Description	Required or Optional	Available Version
		response body, in one of the URL parameters, such as `nextPageURL`, for example: `"nextPageUrl": "/services/data/v34.0/chatter/feeds/news/005D0000001GLowIAN/feed-items?page=2011-03-30T17:34:50Z,0D5D0000000DSv4KAG"`		
`pageSize`	Integer	Specifies the number of items per page. Valid values are from 1 through 100. If you don't specify a size, the default is 25.	Optional	23.0–31.0
`q`	String	One or more keywords to search for in the body and comments of feed items in this feed. The search string can contain wildcards and must contain at least two characters that aren't wildcards. See Wildcards.	Optional	28.0–31.0
`recentCommentCount`	Integer	Maximum number of comments to include per feed item. The default value is 3. The maximum value is 25.	Optional	29.0–31.0
`sort`	String	Specifies the order returned by the sort, such as by date created or last modified: • `CreatedDateDesc`—Sorts by most recent creation date. • `LastModifiedDateDesc`—Sorts by most recent activity. The default value is `CreatedDescDate`.	Optional	23.0–31.0

Response body for GET

Feed Item Page

Response body for POST

Feed Item

User-Profile Feed Items Flat

This resource is deprecated as of version 26.0.

Returns a flat feed, that is, a feed where there is no hierarchy between feed items and comments: Feed items and comments are at the same level. All items are returned in chronological order.

Resource

```
/chatter/feeds/user-profile/me/feed-items/flat
```

or

```
/chatter/feeds/user-profile/userId/feed-items/flat
```

Available since version

25.0–26.0

Requires Chatter

Yes

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
`numFeedItems`	Integer	Specifies the number of feed items returned. Valid values are from 0 through 100. Default value is 25. Because feed items can include comments, specifying 25 flat feed items may actually return more. For example, if each feed item has one comment, and `recentCommentCount` is greater than 1, 50 flat feed items are returned.
`page`	String	A generated token that indicates the view of feed items in the feed. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`.
`recentCommentCount`	Integer	Specifies the number of recent comments returned. Valid values are from 0 through 25. Default value is 3.
`sort`	String. One of the following: • `CreatedDateDesc` • `LastModifiedDateDesc`	Sorts the returned feed either by created date in ascending order, or by last modified date, also in ascending order. The default is `LastModifiedDateDesc`

Response body

Flat Feed

SEE ALSO:

Working with Feeds and Feed Elements

Files Connect Repository Resources

A files connect repository is an external content repository that's connected to Salesforce. Use these resources to get a list of repositories, to get information about a repository or a repository file, and to get the content of a repository file. Also use these resources to get information about the files and folders in a repository folder.

Available resources:

Resource	Description
`/connect/content-hub/repositories`	Get a list of files connect repositories.
`/connect/content-hub/repositories/`***`repositoryId`***	Get information about a files connect repository.
`/connect/content-hub/repositories/`***`repositoryId`***`/files/`***`repositoryFileId`***	Get information about a files connect repository file.
`/connect/content-hub/repositories/`***`repositoryId`***`/files/`***`repositoryFileId`***`/content`	Get the content of the repository file from the external system. The content is streamed as the body of the response.
`/connect/content-hub/repositories/`***`repositoryId`***`/folders/`***`repositoryFolderId`***`/items`	Get information about the files and folders in a repository folder.

Use these files resources to work with references to repository files:

Resource	Description
`/chatter/files/`***`fileId`***	Get information about or update the version of an existing reference.
`/chatter/files/`***`fileId`***`/content`	Get the content of the reference. The content is streamed as the body of the response.
`/chatter/users/me/files/`	Create a reference.

Files Connect Repository List

Get a list of files connect repositories.

Resource

```
/connect/content-hub/repositories
```

Available version

32.0

Requires Chatter

No

HTTP methods

GET

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
`canBrowseOnly`	Boolean	Retrieves only repositories that support browsing.	Optional	32.0
`canSearchOnly`	Boolean	Retrieves only repositories that support searching.	Optional	32.0
`page`	Integer	Specifies the page token to be used to view a page of information. Page tokens are returned as part of the response class, such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.	Optional	32.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are between 1 and 100. If you pass in `null`, the default is 25.	Optional	32.0

Response body for GET

Files Connect Repository Collection

Files Connect Repository

Get information about a files connect repository.

Resource

```
/connect/content-hub/repositories/repositoryId
```

Available version

32.0

Requires Chatter

No

HTTP methods

GET, HEAD

Response body

Files Connect Repository

Files Connect Repository File

Get information about a files connect repository file.

Resource

```
/connect/content-hub/repositories/repositoryId/files/repositoryFileId
```

Available version
32.0

Requires Chatter
No

HTTP methods
GET, HEAD

Response body
Repository File Detail

Files Connect Repository File Content

Get the content of the repository file from the external system. The content is streamed as the body of the response.

Resource

```
/connect/content-hub/repositories/repositoryId/files/repositoryFileId/content
```

Available version
32.0

Requires Chatter
No

HTTP methods
GET, HEAD

Response body
Binary stream of the file content.

Files Connect Repository Folder Items

Get information about the files and folders in a repository folder.

Resource

```
/connect/content-hub/repositories/repositoryId/folders/repositoryFolderId/items
```

Available version
32.0

Requires Chatter
No

HTTP methods
GET, HEAD

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
`page`	Integer	Specifies the page token to be used to view a page of information. Page tokens are returned as part of the response class,	Optional	32.0

Parameter Name	Type	Description	Required or Optional	Available Version
		such as `currentPageToken` or `nextPageToken`. If you pass in `null`, the first page is returned.		
`pageSize`	Integer	Specifies the number of items per page. Valid values are between 1 and 100. If you pass in `null`, the default is 25.	Optional	32.0

Response body for GET

Repository Folder Items Collection

Files Resources

Information about files, including content, rendered version, and sharing.

Available resources are:

Resource	Description
`/chatter/files/`***`fileId`***	Get information about a specified file, including references to external files. Upload a new version of an existing file, including references to external files. Rename a file, move a file to a different folder, and delete a file.
`/chatter/files/`***`fileId`***`/content`	Returns the content of the file, including references to external files. The content is streamed as the body of the response.
`/chatter/files/`***`fileId`***`/file-shares`	Returns information about the objects with which the specified file has been shared. Objects can be users, groups, or records. Can also be used to share the specified file with users.
`/chatter/files/`***`fileId`***`/file-shares/link`	A description of a file shared as a link. Create, access, and delete a file's share link.
`/chatter/files/`***`fileId`***`/rendition`	A rendition of the file. A rendition is a binary preview of the file. The format of the binary depends on the format specified in the request.
`/chatter/files/batch/`***`fileIds`***	Information about a list of files, specified by file IDs. Get information about a list of files and delete a list of files.

File Information

Get information about a specified file, including references to external files. Upload a new version of an existing file, including references to external files. Rename a file, move a file to a different folder, and delete a file.

To upload a new version of an existing file, make a POST request. You must upload the new version as a binary part in a multipart/form-data request. See Uploading Binary Files on page 10. To upload a new file to the Files tab, use `/chatter/users/me/files` on page 373.

Resource

```
/chatter/files/fileId
```

```
/connect/communities/communityId/chatter/files/fileId
```

Available since version

24.0

Requires Chatter

Yes

HTTP methods

GET, POST, PATCH, DELETE, or HEAD

Note: POST, PATCH, and DELETE available since version 26.0.

Request body for POST

Uploads to SharePoint 2010 are limited to 3 MB.

Root XML tag

```
<fileInput>
```

JSON

```
{
     "desc" : "Employee Survey Results",
     "title" : "emp_surv_results"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`contentModifiedDate`	Date	File-specific last modified date that can be used to set a client-side modified date.	Optional	32.0
`desc`	String	Description of the file to be uploaded.	Optional	24.0
`isInMyFileSync`	Boolean	`true` if the file is synced withSalesforce Files Sync; `false` otherwise.	Optional	31.0
`isMajorVersion`	Boolean	`true` if the file is a major version; `false` if the file is a minor version. Major versions can't be replaced.	Optional	31.0
`parentFolderId`	String	ID of the containing folder.	Optional	31.0
`repositoryFileId`	String	ID of the external file.	Optional	32.0
`repositoryId`	String	ID of the external data source.	Optional	32.0

Name	Type	Description	Required or Optional	Available Version
`title`	String	Title of the file.	Optional	24.0

Request parameters for POST

Uploads to SharePoint 2010 are limited to 3 MB.

Name	Type	Description	Available Version
`desc`	String	Description of the file to be uploaded.	26.0
`isDataSync`	Boolean	The request is automatically generated by an automated client.	31.0
`isInMyFileSync`	Boolean	`true` if the file is synced withSalesforce Files Sync; `false` otherwise.	31.0
`isMajorVersion`	Boolean	`true` if the file is a major version; `false` if the file is a minor version. Major versions can't be replaced.	31.0
`title`	String	Title of the file.	26.0

Request parameters for GET

Get information about the specified file.

Name	Type	Description	Available Version
`versionNumber`	Integer	Specify an existing version number for the file. If not specified, the latest version is returned.	23.0

Request parameters for PATCH

Update the title or sync state of a file or move a file to a different folder.

Name	Type	Description	Available Version
`desc`	String	A new description for the file	32.0
`title`	String	A new title for the file	26.0
`isInMyFileSync`	Boolean	`true` if the file is synced withSalesforce Files Sync; `false` otherwise.	31.0
`parentFolderId`	String	ID of the containing folder	31.0

Response body for GET, POST, PATCH, and HEAD

File Detail

Response body for DELETE
Returns a status code of 204.

Example
This PATCH request updates the title of a file:

```
/chatter/files/069D00000001FHF?title=A+New+Title
```

File Content

Returns the content of the file, including references to external files. The content is streamed as the body of the response.

Resource

```
/chatter/files/fileId/content
```

```
/connect/communities/communityId/chatter/files/fileId/content
```

Available since version
24.0

Requires Chatter
Yes

HTTP methods
GET

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
`isDataSync`	Boolean	The request is automatically generated by an automated client.	Optional	31.0
`versionNumber`	Integer	Specify an existing version number for the file. If not specified, the latest version is returned.	Optional	24.0

Response body
Binary stream of the file content.

File Shares

Returns information about the objects with which the specified file has been shared. Objects can be users, groups, or records. Can also be used to share the specified file with users.

Resource

```
/chatter/files/fileId/file-shares
```

```
/connect/communities/communityId/chatter/files/fileId/file-shares
```

Available since version
24.0

Requires Chatter
Yes

HTTP methods
GET, HEAD, POST

POST is available in versions 30.0 and later.

Request parameters for GET
In versions 30.0 and later, the first share returned in a GET request is the organization record share. In communities, the first share returned is the network record share.

Name	Type	Description	Required or Optional	Available Since
`page`	Integer	An integer greater than 0 specifying a page of topics. The default value is zero.	Optional	24.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are between 1 and 100. The default value is 25.	Optional	24.0

Request body for POST

Root XML tag
`<fileShares>`

JSON example

```
{
  "message": "I shared a document with you.",
  "shares": [
     { "id": "005D0000001Az4l", "sharingType": "V" },
     { "id": "005D0000001B2ny", "sharingType": "C" }
   ]
}
```

Properties

Name	Type	Description	Available Version
`message`	String	Private message body that's sent to the recipients. If Salesforce Communities is enabled, an email is sent instead of a private message.	30.0
`shares`	Share Input[]	Collection of Share Input request bodies	30.0

Request parameters for POST

Parameters for sharing a file with users

Name	Type	Description	Required or Optional	Available Since
`id` ***(1–9)***	String	List of up to 9 user IDs to share this file with. This list must match the list of sharing types.	Required	30.0
`message`	String	Private message body that's sent to the recipients. If Salesforce Communities is enabled, an email is sent instead of a private message.	Optional	30.0
`sharingType` ***(1–9)***	String	List of sharing types, either `C` for collaborator or `V` for viewer. This list must match the list of user IDs.	Required	30.0

Response body for GET and POST

File Shares Page

Example for POST

This example uses request parameters to share a file with two users.

```
POST /chatter/files/069D00000001JDU/file-shares?
id1=005D0000001LL8O&id2=005D0000001QdEL&sharingType1=C&sharingType2=V
```

This example uses a request body to share the same file with the same users.

```
POST /chatter/files/069D00000001JDU/file-shares

{
  "shares": [
     { "id": "005D0000001LL8O", "sharingType": "C" },
     { "id": "005D0000001QdEL", "sharingType": "V" }
   ]
}
```

Files Shares Link

A description of a file shared as a link. Create, access, and delete a file's share link.

Resource

```
/chatter/files/fileId/file-shares/link
```

```
/connect/communities/communityId/chatter/files/fileId/file-shares/link
```

Available since version

30.0

Requires Chatter

Yes

HTTP methods

GET, HEAD, PUT, DELETE

Response body for GET or PUT
File Share Link

Response body for DELETE
Returns a status code of 204.

File Rendition

A rendition of the file. A rendition is a binary preview of the file. The format of the binary depends on the format specified in the request.

Note: For shared files, renditions process asynchronously after upload. For private files, renditions process when the first file preview is requested, and aren't available immediately after the file is uploaded.

Resource

```
/chatter/files/fileId/rendition
```

```
/connect/communities/communityId/chatter/files/fileId/rendition
```

Available since version
24.0

Requires Chatter
Yes

HTTP methods
GET

Request parameters

Name	Type	Description
`page`	Integer	Specify a rendition of a specific page. The first page is 0. If you don't specify a specific page, a rendition of the first page is returned. For a PDF file rendition, the rendition is for the entire document. The only valid value is 0: you can't get the rendition of a single page.
`type`	String	Specify the type of rendition to be returned. Default value is `THUMB120BY90`. Valid values are: • `FLASH` • `PDF` • `THUMB120BY90` • `THUMB240BY180` • `THUMB720BY480`

Response body
Binary stream of the rendition.

File Information, Batch

Information about a list of files, specified by file IDs. Get information about a list of files and delete a list of files.

The number of file IDs you can specify is limited by the length of the URL. In general, you can specify up to 100 file IDs in a comma-separated list.

Resource

```
/chatter/files/batch/fileIds
```

```
/connect/communities/communityId/chatter/files/batch/fileIds
```

Available since version
26.0

Requires Chatter
Yes

HTTP methods
GET, DELETE

Response body
Batch Result Item

Folders Resources

Information about a folder or the contents of a folder. Modify and delete folders, add items to folders, and sync root folders.

Available resources:

Resource	Description
`/chatter/folders/folderId`	Get information about the specified folder. Move, rename, and delete a folder. Sync the root folder.
`/chatter/folders/folderId/items`	Get information about the contents of a folder. Add a file to a folder and create a folder. Sync the root folder.

Folder Information

Get information about the specified folder. Move, rename, and delete a folder. Sync the root folder. Deleting a folder deletes all the folders within that folder and removes all files from the deleted folders so they are no longer synced.

Resource

```
/chatter/folders/folderId
```

Note: You can use `root` as the *folderId*.

Available since version
30.0

Requires Chatter
Yes

HTTP methods
GET, HEAD, PATCH, DELETE

Request body for PATCH
Folder Input

Request parameters for PATCH

Move a folder

Parameter	Type	Description	Required or Optional	Available Version
`isInMyFileSync`	Boolean	`true` if the folder and all folders in this folder are synced with Salesforce Files Sync; `false` otherwise. Setting `isInMyFileSync` is valid only for the root folder. Once set to `true`, `isInMyFileSync` can't be set to `false`.	Optional	33.0
`parentFolderId`	String	The ID of the new parent folder. If `null`, the folder is removed from its current folder. If the specified ID is `root`, the folder is moved to the root of the user's folder hierarchy.	Required	30.0

Rename a folder

Parameter	Type	Description	Required or Optional	Available Version
`isInMyFileSync`	Boolean	`true` if the folder and all folders in this folder are synced with Salesforce Files Sync; `false` otherwise. Setting `isInMyFileSync` is valid only for the root folder. Once set to `true`, `isInMyFileSync` can't be set to `false`.	Optional	33.0
`name`	String	The new name for the folder	Required	30.0

Response body for GET and PATCH
Folder

Folder Contents

Get information about the contents of a folder. Add a file to a folder and create a folder. Sync the root folder.

To create a folder, you can use a request body or request parameters in a POST request.

To create a file, you must send it as a body part in a multipart/form-data POST request. You can send the description and title information in a JSON or XML rich input body in a part of the multipart/form-data request. Alternately, you can send the description, title, and type in request parameters. See Uploading Binary Files on page 10. To add a file to the Files tab, use `/chatter/users/me/files`.

Resource

```
/chatter/folders/folderId/items
```

Note: You can use `root` as the *folderId*.

Available since version

30.0

Requires Chatter

Yes

HTTP methods

GET, HEAD, POST

Request parameters for GET

Parameter	Type	Description	Required or Optional	Available Version
`filter`	String	Specifies a filter for items within a folder. The value `IsInMyFileSync` filters out the items that aren't synced in a folder.	Optional	33.0
`page`	Integer	An integer specifying a page of items. The default value is 0, which returns the first page.	Optional	30.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are from 1 through 100. If you don't specify a size, the default is 25.	Optional	30.0

Response body for GET

Folder Item Page

Request body for POST

Folder Item Input

Request parameters for POST

Add a file to a folder

Parameter	Type	Description	Required or Optional	Available Version
`desc`	String	The description of the file	Optional	30.0
`isInMyFileSync`	Boolean	`true` if the folder and all folders in this folder are synced with Salesforce Files Sync; `false` otherwise. Setting `isInMyFileSync` is valid only for the root folder. Once set to `true`, `isInMyFileSync` can't be set to `false`.	Optional	33.0

Parameter	Type	Description	Required or Optional	Available Version
`title`	String	The title of the file	Optional	30.0
`type`	String	The item type. To add a file to a folder, the value must be `File`.	Required	30.0

Create a folder

Parameter	Type	Description	Required or Optional	Available Version
`folderPath`	String	The path to the folder	Required	30.0
`isInMyFileSync`	Boolean	`true` if the folder and all folders in this folder are synced with Salesforce Files Sync; `false` otherwise. Setting `isInMyFileSync` is valid only for the root folder. Once set to `true`, `isInMyFileSync` can't be set to `false`.	Optional	33.0
`type`	String	The item type. To create a folder, the value must be `Folder`.	Required	30.0

Response body for POST

Folder Item

Example for adding a file to a folder

```
{
   "file" :
      {
         "description" : "Yearly Report for Universal Containers Box Division",
         "title" : "Yearly.txt"
      },
   "type" : "File"
}
```

Example for creating a folder

```
{
   "folder" :
      {
         "path" : "my_documents/my_folder"
      },
   "type" : "Folder"
}
```

Groups Resources

Information about groups, such as the group's members, photo, and the groups in the organization. Create and delete a group, add members to a group, and change the group photo.

Note: The groups resource is not a feed. To view the feed for a group, or post to it, use the record feed resource with a `groupId`.

Available resources:

Resource	Description
`/chatter/group-membership-requests/requestId`	Update the status of a request by a user to join a private group or get information about a request to join a private group.
`/chatter/group-memberships/membershipId`	Access information about a member of a group, remove a member from a group, and update a group member's role.
`/chatter/group-memberships/batch/membershipIds`	Access information about a list of group members.
`/chatter/group-records/groupRecordId`	Get information about a group record or remove the association between a record and a group.
`/chatter/groups/`	A list of all the groups in the organization. Get information about groups or create a group.
`/chatter/groups/batch/group_list`	Returns information about more than one group at the same time.
`/chatter/groups/groupId`	Information about a group.
`/chatter/groups/groupId/announcements`	Information about the announcements posted to the specified group. Get information or post an announcement to a group.
`/chatter/groups/groupID/files`	Returns information about the files posted to the specified group.
`/chatter/groups/groupId/members`	Members of a specified group. Get the members and add a member to a group. To add a member, the context user must be the group owner or moderator.
`/chatter/groups/groupId/members/requests`	Request membership in a private group or get the status of requests to a join a private group.
`/chatter/groups/groupId/my-settings`	Information about the context user's Chatter settings for the specified group.
`/chatter/groups/groupId/photo`	The photo for the specified group. Get, post, crop, and delete the photo.
`/chatter/groups/groupId/records`	Get the list of records associated with a group. Associate a record with a group.
`/chatter/groups/groupId/topics`	Returns up to five topics most recently used in the specified group.

SEE ALSO:

Groups Feed Resources

List of Groups

A list of all the groups in the organization. Get information about groups or create a group.

Resource

```
/chatter/groups/
```

```
/connect/communities/communityId/chatter/groups/
```

Available version

23.0

POST available in 29.0

Requires Chatter

Yes

HTTP methods

GET, HEAD, POST

Request parameters for GET

Parameter Name	Type	Description	Available
`archiveStatus`	String	Returns the list of groups filtered by the archive status you specify: • `All`—All groups, including groups that are archived and groups that are not archived. • `Archived`—Only groups that are archived. • `NotArchived`—Only groups that are not archived. The default value is `NotArchived`.	29.0
`page`	Integer	An integer specifying a page of items. The default value is 0, which returns the first page.	23.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are between 1 and 250. The default value is 25.	23.0
`q`	String	Specifies the string to search. The search string must contain at least two characters, not including any wildcards. For more information about wildcards, see Wildcards.	23.0

Request body for POST

Root XML Tag

```
<group>
```

JSON

```
{
   "announcement" : "0D5D0000000K4XT",
   "canHaveChatterGuests" : "true",
   "description" : "This group is building a powerful API.",
   "information" : {
      "text" : "Read this section before you post to the group.",
      "title" : "Information"
   },
   "isArchived" : "false",
   "isAutoArchiveDisabled" : "false",
   "name" : "Chatter API",
   "owner" : "00590000000DmPw",
   "visibility" : "PublicAccess"
}
```

Properties

Name	Type	Description	Available Version
`announcement`	String	An announcement ID. An announcement displays in a designated location in the Salesforce UI until 11:59 p.m. on its expiration date, unless it's deleted or replaced by another announcement.	31.0
`canHaveChatterGuests`	Boolean	`true` if this group allows Chatter customers, `false` otherwise. After this property is set to `true`, it cannot be set to `false`.	29.0
`description`	String	The "Description" section of the group	29.0
`information`	Group Information Input	The "Information" section of the groupIn the Web UI, the "Information" section is above the "Description" section. If the group is private, the "Information" section is visible only to members. The "Description" section is visible to everyone.	28.0
`isArchived`	Boolean	`true` if the group is archived, `false` otherwise. Defaults to `false`.	29.0
`isAutoArchiveDisabled`	Boolean	`true` if automatic archiving is turned off for the group, `false` otherwise. Defaults to `false`.	29.0
`name`	String	The name of the group	29.0
`owner`	String	The ID of the group owner. This property is available for PATCH requests only.	29.0

Name	Type	Description	Available Version
`visibility`	String	Specifies the group visibility type. One of the following values: • `PrivateAccess`—Only members of the group can see posts to this group. • `PublicAccess`—All users within the community can see posts to this group. • `Unlisted`—Reserved for future use.	29.0

Request parameters for POST

Parameter Name	Type	Description	Required or Optional	Available
`canHaveChatterGuests`	Boolean	`true` if this group allows Chatter customers, `false` otherwise. After this property is set to `true`, it cannot be set to `false`.	Optional	29.0
`description`	String	The "Description" section of the group	Optional	29.0
`informationText`	String	The text in the "Information" section of the group	Optional	29.0
`informationTitle`	String	The title of the "Information" section of the group	Optional	29.0
`name`	String	The name of the group	Required	29.0
`visibility`	String	Specifies the group visibility type. One of the following: • `PrivateAccess`—Only members of the group can see posts to this group. • `PublicAccess`—All users within the community can see posts to this group. • `Unlisted`—Reserved for future use.	Required	29.0

Response body for GET

Group Page

Response body for POST

Group Detail

Example for POST

This request creates a public group:

```
POST /chatter/groups?name=New+Group&visibility=PublicAccess
```

SEE ALSO:

Create a Group

Group Information

Information about a group.

Resource

```
/chatter/groups/groupId
```

```
/connect/communities/communityId/chatter/groups/groupId
```

Available version

23.0

PATCH is available in 28.0

Requires Chatter

Yes

HTTP methods

GET, DELETE, HEAD, PATCH

DELETE is supported in 29.0

PATCH is supported in 28.0

Request body for PATCH

Group Input

Root XML Tag

```
<group>
```

JSON

```
{
   "announcement" : "0D5D0000000K4XT",
   "canHaveChatterGuests" : "true",
   "description" : "This group is building a powerful API.",
   "information" : {
      "text" : "Read this section before you post to the group.",
      "title" : "Information"
   },
   "isArchived" : "false",
   "isAutoArchiveDisabled" : "false",
   "name" : "Chatter API",
   "owner" : "00590000000DmPw",
   "visibility" : "PublicAccess"
}
```

Properties

Name	Type	Description	Available Version
`announcement`	String	An announcement ID. An announcement displays in a designated location in the Salesforce UI until 11:59 p.m. on its expiration date, unless it's deleted or replaced by another announcement.	31.0
`canHaveChatterGuests`	Boolean	`true` if this group allows Chatter customers, `false` otherwise. After this property is set to `true`, it cannot be set to `false`.	29.0
`description`	String	The "Description" section of the group	29.0
`information`	Group Information Input	The "Information" section of the groupIn the Web UI, the "Information" section is above the "Description" section. If the group is private, the "Information" section is visible only to members. The "Description" section is visible to everyone.	28.0
`isArchived`	Boolean	`true` if the group is archived, `false` otherwise. Defaults to `false`.	29.0
`isAutoArchiveDisabled`	Boolean	`true` if automatic archiving is turned off for the group, `false` otherwise. Defaults to `false`.	29.0
`name`	String	The name of the group	29.0
`owner`	String	The ID of the group owner. This property is available for PATCH requests only.	29.0
`visibility`	String	Specifies the group visibility type. One of the following values: • `PrivateAccess`—Only members of the group can see posts to this group. • `PublicAccess`—All users within the community can see posts to this group. • `Unlisted`—Reserved for future use.	29.0

Group Information Input

JSON

```
{
   "text" : "Post questions about the API.",
   "title" : "API Questions"
}
```

Properties

Name	Type	Description	Available Version
`text`	String	The text in the "Information" section of the group. HTML is not supported. The maximum length is 4000 characters (bytes).	28.0
`title`	String	The title in the "Information" section of the group. The maximum length is 240 characters (bytes).	28.0

Request parameters for PATCH

Property	Type	Description	Required or Optional	Available Version
`announcement`	String	The ID of a feed item to use as an announcement. An announcement displays in a designated location in the Salesforce UI until 11:59 p.m. on its expiration date, unless it's deleted or replaced by another announcement.	Optional	31.0
`canHaveChatterGuests`	Boolean	`true` if this group allows Chatter customers, `false` otherwise. After this property is set to `true`, it cannot be set to `false`.	Optional	29.0
`description`	String	The "Description" section of the group	Optional	29.0
`informationText`	String	The "Information" section of the group. In the Web UI, the "Information" section is above the "Description" section. If the group is private, the "Information" section is visible only to members. The "Description" section is visible to everyone.	Optional	28.0
`informationTitle`	String	The title of the "Information" section of the group	Optional	28.0
`isArchived`	String	`true` if the group is archived, `false` otherwise. Defaults to `false`.	Optional	29.0
`isAutoArchiveDisabled`	Boolean	`true` if automatic archiving is turned off for the group, `false` otherwise. Defaults to `false`.	Optional	29.0
`name`	String	The name of the group	Optional	29.0
`owner`	String	The ID of the group owner. This property is available for PATCH requests only.	Optional	29.0

Property	Type	Description	Required or Optional	Available Version
`visibility`	String	Specifies the group visibility type. One of the following values: • `PrivateAccess`—Only members of the group can see posts to this group. • `PublicAccess`—All users within the community can see posts to this group. • `Unlisted`—Reserved for future use.	Optional	29.0

Response body for GET and PATCH

Group Detail

Example for PATCH

This request body updates the text and title of the "Information" section of a group:

```
{
   "information" : {
      "text" : "Post questions about the API to this group. If you have the answer to
a question, please post it. This is a private R&D group.",
      "title" : "API Questions Group"
   }
}
```

Here is the same example using request parameters:

```
https://instance_name/services/data/34.0/chatter/groups/groupId
?informationText=Post+questions+about+the+API+to+this+group.+If+you+have+the+answer+to+a+question%2C%20
+please+post+it.+This+is+a+private+R%26amp%3Bamp%3BD+group.&informationTitle=API+Questions+Group
```

The response is a Group Detail response body.

Batch Group Information

Returns information about more than one group at the same time.

Resource

```
/chatter/groups/batch/group_list
```

```
/connect/communities/communityId/chatter/groups/batch/group_list
```

Where *group_list* is a comma separated list of up to 500 group IDs.

Available since version

23.0

Requires Chatter

Yes

HTTP methods

GET or HEAD

Response body
Batch Results

Group Announcements

Information about the announcements posted to the specified group. Get information or post an announcement to a group.

Use an announcement to highlight information. Users can discuss, like, and post comments on announcements in the group feed. Group members receive an email notification when you post an announcement, same as for other posts, depending on their selected group email notification frequency. Deleting the feed post deletes the announcement.

To get information about a specific announcement, update the expiration date of an announcement, or delete an announcement, use the Announcements Resource (`/chatter/announcements/`***`announcementId`***).

Resource

```
/chatter/groups/groupID/announcements
```

```
/connect/communities/communityId/chatter/groups/groupID/announcements
```

Available since version
31.0

Requires Chatter
Yes

HTTP methods
GET, POST, HEAD

Request parameters for GET

Parameter Name	Type	Description	Available Since
`page`	Integer	An integer specifying a page of items. The default value is 0, which returns the first page.	31.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are from 1 through 100. If you don't specify a size, the default is 25.	31.0

Request body for POST

Root XML tag
`<announcement>`

JSON example

```
{   "body":
      {
         "messageSegments": [
         {
            "text": "Please install the updates for all iOS devices.",
            "type": "Text"
```

```
            }
            ]
        }
    "expirationDate": "2014-02-22T00:00:00.000Z",
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`body`	Message Body Input	Text of the announcement.	Required	31.0
`expirationDate`	Date	The Salesforce UI displays an announcement until 11:59 p.m. on this date unless another announcement is posted first. The Salesforce UI ignores the time value in the `expirationDate`. However, you can use the time value to create your own display logic in your own UI.	Required	31.0

Request parameters for POST

Parameter Name	Type	Description	Available Since
`expirationDate`	Date	The Salesforce UI displays an announcement until 11:59 p.m. on this date unless another announcement is posted first. The Salesforce UI ignores the time value in the `expirationDate`. However, you can use the time value to create your own display logic in your own UI.	31.0

Response body for GET

Announcement Page

Response body for POST

Announcement

Example for POST

This request posts an announcement to a group:

```
POST /chatter/groups/0F9D00000000I1d/announcements
```

This is the request body:

```
{   "body":
      {
         "messageSegments": [
         {
            "text": "Please install the updates for all iOS devices.",
            "type": "Text"
          }
          ]
      }
   "expirationDate": "2014-02-22T00:00:00.000Z",
}
```

Group Files

Returns information about the files posted to the specified group.

Resource

```
/chatter/groups/groupID/files
```

```
/connect/communities/communityId/chatter/groups/groupID/files
```

Available since version

24.0

Requires Chatter

Yes

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description	Available Since
`page`	Integer	An integer specifying a page of items. The default value is 0, which returns the first page.	24.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are from 1 through 100. If you don't specify a size, the default is 25.	24.0
`q`	String	A query string to filter the results. The string must contain at least two characters, not including wildcards. For information about wildcards, see Wildcards.	27.0

Response body
File Summary Page

Group Members

Members of a specified group. Get the members and add a member to a group. To add a member, the context user must be the group owner or moderator.

Resource

```
/chatter/groups/groupId/members
```

```
/connect/communities/communityId/chatter/groups/groupId/members
```

Available since version
23.0

Requires Chatter
Yes

HTTP methods
GET, POST or HEAD

Request body for POST

Root XML tag
`<member>`

Properties

Name	Type	Description	Available Version
`role`	String	The role of the member. One of these values: • `GroupManager` • `StandardMember`	29.0
`userId`	String	The user ID of the member to be added to the group.	23.0

Request parameter for POST

Parameter Name	Type	Description	Available Version
`role`	String	The role of the member. One of these values: • `GroupManager` • `StandardMember`	29.0
`userId`	String	ID of the user to add to the group.	23.0

Request parameters for GET or HEAD

Parameter Name	Type	Description	Available Version
`page`	Integer	An integer specifying a page of items. The default value is 0, which returns the first page.	23.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are between 1 and 1000. If you don't specify a size, the default is 25.	23.0

Response body for GET or HEAD

Group Member Page

Response body for POST

Group Member

Example: Add a member to a group

This POST request adds a member to a group:

```
/chatter/groups/0F9D00000000I4O/members?userId=005D0000001LLO1
```

The response is a Group Member response body:

```
{
   "id": "0FBD00000003LRxOAM",
   "role": "StandardMember",
   "url": "/services/data/v34.0/chatter/group-memberships/0FBD00000003LRxOAM",
   "user": {
      "companyName": "Universal Containers",
      "firstName": "Gordon",
      "id": "005D0000001LLO1IAO",
      "isActive": true,
      "isInThisCommunity": true,
      "lastName": "Johnson",
      "motif": {
         "color": "1797C0",
         "largeIconUrl": "/img/icon/profile64.png",
         "mediumIconUrl": "/img/icon/profile32.png",
         "smallIconUrl": "/img/icon/profile16.png"
      },
      "mySubscription": {
         "id": "0E8D0000000T5ZUKA0",
         "url": "/services/data/v34.0/chatter/subscriptions/0E8D0000000T5ZUKA0"
      },
      "name": "Gordon Johnson",
      "photo": {
         "fullEmailPhotoUrl":
"https://instance_name/ncsphoto/SqRfG_-PUTY6KtfOp_f0cPhKrC7plNhUw5eOz7Aj1u-oFL0K-ff2oe04ZAmT2MqC",

         "largePhotoUrl": "https://instance_name/profilephoto/729D00000000LJL/F",
```

```
            "photoVersionId": "729D00000000LJLIA2",
            "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000LJL/T",
            "standardEmailPhotoUrl":
"https://instance_name/ncsphoto/SqRfG_-PUTY6KtfOp_f0cPhKrC7plNhUw5eOz7Aj1u-6d5rE2mNaMaBZfGnFX65O",

            "url": "/services/data/v34.0/chatter/users/005D0000001LLO1IAO/photo"
        },
        "title": "VP Sales",
        "type": "User",
        "url": "/services/data/v34.0/chatter/users/005D0000001LLO1IAO",
        "userType": "Internal"
    }
}
```

Note: If the group is private, the response is 403. To request to join the group, send a POST request to the Group Members—Private resource (`/chatter/groups/groupId/members/requests`).

SEE ALSO:

Join a Group

Group Members—Private

Request membership in a private group or get the status of requests to a join a private group.

Note: To request membership in a public group use the Group Members resource.

Resource

```
/chatter/groups/groupId/members/requests
```

```
/connect/communities/communityId/chatter/groups/groupId/members/requests
```

Available since version

27.0

Requires Chatter

Yes

HTTP methods

GET, HEAD, and POST

Request parameter for GET

Parameter Name	Type	Description	Available Since
`status`	String	Optional. If you include this parameter, results are filtered to include those that match the specified status. Valid values: • `Accepted` • `Declined` • `Pending`	27.0

Response body for GET or HEAD

Group Membership Request Collection

Response body for POST

Group Membership Request

This table lists HTTP response code descriptions that are unique to this resource. For descriptions of other HTTP response codes, see Status Codes and Error Responses.

HTTP Response Code	Description
201	One of the following: • Success • User already has a membership request for this group
204	User is already a member of this group.
403	One of the following: • Group is private: `INSUFFICIENT_ACCESS_OR_READONLY` • User is an external user: `INSUFFICIENT_ACCESS`

Example for GET

This example gets information about all the requests to join that have been accepted for this private group:

```
/chatter/groups/0F9D00000000izfKAA/members/requests?status=Accepted
```

The JSON Group Membership Request Collection response body includes one request from the user Gordon Johnson:

```
{
   "requests": [{
      "createdDate": "2012-10-19T00:23:23.000Z",
      "id": "0I5D000000002fmKAA",
      "lastUpdateDate": "2012-10-19T00:24:06.000Z",
      "requestedGroup": {
         "id": "0F9D00000000izfKAA",
         "url": "/services/data/v34.0/chatter/groups/0F9D00000000izfKAA"
      },
      "responseMessage": null,
      "status": "Accepted",
     "url": "/services/data/v34.0/chatter/group-membership-requests/0I5D000000002fmKAA",

      "user": {
         "companyName": "Universal Containers",
         "firstName": "Gordon",
         "id": "005D0000001LLO1IAO",
         "isActive": true,
         "lastName": "Johnson",
         "motif": {
            "color": "1797C0",
            "largeIconUrl": "/img/icon/profile64.png",
            "mediumIconUrl": "/img/icon/profile32.png",
            "smallIconUrl": "/img/icon/profile16.png"
```

```
         },
         "mySubscription": {
            "id": "0E8D0000000T5ZUKA0",
            "url": "/services/data/v34.0/chatter/subscriptions/0E8D0000000T5ZUKA0"
         },
         "name": "Gordon Johnson",
         "photo": {
          "fullEmailPhotoUrl": "https://instance_name/ncsphoto/SqRfG_-PUTY6KtfOp_f0cPhK
rC7plNhUw5eOz7Aj1u-zWLMxsSybO7lwUPg5_Xne",
            "largePhotoUrl": "https://instance_name/profilephoto/729D00000000LJL/F",
            "photoVersionId": "729D00000000LJLIA2",
            "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000LJL/T",
          "standardEmailPhotoUrl": "https://instance_name/ncsphoto/SqRfG_-PUTY6KtfOp_f0
cPhKrC7plNhUw5eOz7Aj1u_2Vnyzx4jZ2zFiKBz_sTt2",
            "url": "/services/data/v34.0/chatter/users/005D0000001LLO1IAO/photo"
         },
         "title": "VP Sales",
         "type": "User",
         "url": "/services/data/v34.0/chatter/users/005D0000001LLO1IAO",
         "userType": "Internal"
      }
   }],
   "total": 1
}
```

Example for POST

This example requests membership in a private group with the ID `0F9D00000000lpK`:

```
/chatter/groups/0F9D00000000lpK/members/requests
```

The Group Membership Request response body is in JSON format:

```
{
   "createdDate": "2013-04-02T23:44:25.000Z",
   "id": "0I5D000000002kOKAQ",
   "lastUpdateDate": "2013-04-02T23:44:25.000Z",
   "requestedGroup": {
      "id": "0F9D00000000lpKKAQ",
      "url": "/services/data/v34.0/chatter/groups/0F9D00000000lpKKAQ"
   },
   "responseMessage": null,
   "status": "Pending",
   "url": "/services/data/v34.0/chatter/group-membership-requests/0I5D000000002kOKAQ",
   "user": {
      "companyName": "Universal Containers",
      "firstName": "Gordon",
      "id": "005D0000001LLO1IAO",
      "isActive": true,
      "isInThisCommunity": true,
      "lastName": "Johnson",
      "motif": {
         "color": "1797C0",
         "largeIconUrl": "/img/icon/profile64.png",
         "mediumIconUrl": "/img/icon/profile32.png",
         "smallIconUrl": "/img/icon/profile16.png"
```

```
      },
      "mySubscription": null,
      "name": "Gordon Johnson",
      "photo": {
         "fullEmailPhotoUrl":
"https://instance_name/ncsphoto/SqRfG_-PUTY6KtfOp_f0cPhKrC7plNhUw5eOz7Aj1u-oFL0K-ff2oe04ZAmT2MqC",

         "largePhotoUrl": "https://instance_name/profilephoto/729D00000000LJL/F",
         "photoVersionId": "729D00000000LJLIA2",
         "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000LJL/T",
         "standardEmailPhotoUrl":
"https://instance_name/ncsphoto/SqRfG_-PUTY6KtfOp_f0cPhKrC7plNhUw5eOz7Aj1u-6d5rE2mNaMaBZfGnFX65O",

         "url": "/services/data/v34.0/chatter/users/005D0000001LLO1IAO/photo"
      },
      "title": "VP Sales",
      "type": "User",
      "url": "/services/data/v34.0/chatter/users/005D0000001LLO1IAO",
      "userType": "Internal"
   }
}
```


SEE ALSO:

Request to Join a Private Group

Group Membership Requests—Private

Update the status of a request by a user to join a private group or get information about a request to join a private group. Get the value of ***requestId*** from the Group Membership Request response from a POST request to `/chatter/groups/groupId/members/requests`.

Resource

```
/chatter/group-membership-requests/requestId
```

```
/connect/communities/communityId/chatter/group-membership-requests/requestId
```

Available since version

27.0

Requires Chatter

Yes

HTTP methods

GET, HEAD, and PATCH

Request body for PATCH

Group Membership Request Update

Request parameters for PATCH

Parameter Name	Type	Description	Available Since
`responseMessage`	String	Use this property to provide a message to the user if their membership request is declined. The value of this property is used only when the value of the `status` property is `Declined`. The maximum length is 756 characters.	27.0
`status`	String	Valid values: • `Accepted` • `Declined`	27.0

Response body for PATCH

Group Membership Request

Response body for GET or HEAD

Group Membership Request

Example for PATCH

This example updates a request to join a private group from `Pending` to `Accepted`:

```
/chatter/group-membership-requests/0I5D000000002fwKAA?status=Accepted
```

This request returns a Group Membership Request response body.

SEE ALSO:

Approve or Reject a Request to Join a Private Group

Group Memberships Resources

Information about memberships in Chatter groups.

Available resources are:

Resource	Description
`/chatter/group-memberships/`***`membershipId`***	Access information about a member of a group, remove a member from a group, and update a group member's role.
`/chatter/group-memberships/batch/`***`membershipIds`***	Access information about a list of group members.

Group Memberships Information

Access information about a member of a group, remove a member from a group, and update a group member's role.

Membership IDs are returned from the `/chatter/groups/`***`groupId`***`/memberships` resource.

Resource

```
/chatter/group-memberships/membershipId
```

```
/connect/communities/communityId/chatter/group-memberships/membershipId
```

Available since version

23.0

Requires Chatter

Yes

HTTP methods

GET, DELETE, HEAD, PATCH

PATCH is supported in version 29.0.

Request body for PATCH

Root XML tag

```
<member>
```

JSON example

```
{
    "role" : "GroupManager",
    "userId" : "005B0000000Ge16"
}
```

Properties

Name	Type	Description	Available Version
`role`	String	The role of the member. One of these values: • `GroupManager` • `StandardMember`	29.0
`userId`	String	The user ID of the member to be added to the group.	23.0

Note: To update a group member's role in a PATCH request, pass only the `role` property, don't pass the `userId` property.

Request parameters for PATCH

Name	Type	Description	Available Version
`role`	String	The role of the member. One of these values: • `GroupManager` • `StandardMember`	29.0

Response body for GET or PATCH
Group Member

Batch Group Memberships

Access information about a list of group members. The members don't have to belong to the same group. Each membership ID is associated with a specific group.

Membership IDs are returned in the response bodies of several groups resources.

The following groups resource lists every member of a group in its own Group Member response body. The `id` property of every Group Member response body contains a membership Id:

```
/chatter/groups/groupId/members
```

Resource

```
/chatter/group-memberships/batch/membershipIds
```

```
/connect/communities/communityId/chatter/group-memberships/batch/membershipIds
```

Where *membershipIds* is a comma separated list of up to 500 membership IDs.

Available since version
27.0

Requires Chatter
Yes

HTTP methods
GET, HEAD

Response body for GET or HEAD
Batch Results

Example for GET

This request gets the group membership information for two users:

```
/chatter/group-memberships/batch/0FBD00000003GAPOA2,0FBD00000003GAKOA
```

This is the Batch Results response body, which contains nested User Summary response bodies:

```
{
   "hasErrors": true,
   "results": [
      {
         "result": {
            "id": "0FBD00000003GAPOA2",
            "role": "StandardMember",
           "url": "/services/data/v34.0/chatter/group-memberships/0FBD00000003GAPOA2",

            "user": {
               "companyName": "Universal Containers",
               "firstName": "Gordon",
               "id": "005D0000001LLO1IAO",
               "isActive": true,
               "isInThisCommunity": true,
```

```
                "lastName": "Johnson",
                "motif": {
                   "color": "1797C0",
                   "largeIconUrl": "/img/icon/profile64.png",
                   "mediumIconUrl": "/img/icon/profile32.png",
                   "smallIconUrl": "/img/icon/profile16.png"
                },
                "mySubscription": {
                   "id": "0E8D0000000T5ZUKA0",
                  "url": "/services/data/v34.0/chatter/subscriptions/0E8D0000000T5ZUKA0"

                },
                "name": "Gordon Johnson",
                "photo": {
                   "fullEmailPhotoUrl":
"https://instance_name/ncsphoto/SqRfG_-PUTY6KtfOp_f0cPhKrC7plNhUw5eOz7Aj1u-oFL0K-ff2oe04ZAmT2MqC",

                 "largePhotoUrl": "https:/instance_name/profilephoto/729D00000000LJL/F",

                   "photoVersionId": "729D00000000LJLIA2",
                   "smallPhotoUrl":
"https://instance_name/profilephoto/729D00000000LJL/T",
                   "standardEmailPhotoUrl":
"https://instance_name/ncsphoto/SqRfG_-PUTY6KtfOp_f0cPhKrC7plNhUw5eOz7Aj1u-6d5rE2mNaMaBZfGnFX65O",

                   "url": "/services/data/v34.0/chatter/users/005D0000001LLO1IAO/photo"
                },
                "title": "VP Sales",
                "type": "User",
                "url": "/services/data/v34.0/chatter/users/005D0000001LLO1IAO",
                "userType": "Internal"
             }
          },
          "statusCode": 200
       },
       {
          "result": [{
             "errorCode": "NOT_FOUND",
             "message": "The requested resource does not exist"
          }],
          "statusCode": 404
       }
    ]
}
```


Note that the second group membership wasn't found. Batch requests return a status code for each subrequest.

Group Photo

The photo for the specified group. Get, post, crop, and delete the photo.

To use an image from the Files page as a group photo (existing content), pass the file ID in the `fileId` property of the request body or in the `fileId` request parameter. Images uploaded on the Group page don't have a file ID and can't be used as the `fileId`.

To upload a binary file as a group photo, you must send it in a multipart/form-data message. For information about how to create the multipart/form-data message, see Uploading Binary Files.

Resource

```
/chatter/groups/groupId/photo
```

```
/connect/communities/communityId/chatter/groups/groupId/photo
```

Available since version

23.0

Requires Chatter

Yes

HTTP methods

GET, POST, DELETE, or HEAD

Request body for POST

Root XML tag

```
<photo>
```

JSON example

```
{   "cropSize" : "120",
    "cropX" : "0",
    "cropY" : "0",
    "fileId" : "069D00000001IOh"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`cropSize`	Integer	The length, in pixels, of any edge of the crop square.	Required to crop an uploaded file or an existing file.	29.0
`cropX`	Integer	The position X, in pixels, from the left edge of the image to the start of the crop square. Top left is position (0,0).	Required to crop an uploaded file or an existing file.	29.0
`cropY`	Integer	The position Y, in pixels, from the top edge of the image to the start of the crop square. Top left is position (0,0).	Required to crop an uploaded file or an existing file.	29.0
`fileId`	String	18–character ID of an existing file. The key prefix must be 069 and the file size must be less than 2 MB. Note: Images uploaded on the Group page and on the User page don't have file IDs and therefore can't be used.	Required to choose an existing file.	25.0

Name	Type	Description	Required or Optional	Available Version
`versionNumber`	Integer	Version number of an existing file. If not provided, the latest version is used.	Optional	25.0

Request parameters for POST

Parameter Name	Type	Description	Required or Optional	Available Version
`cropSize`	Integer	The length, in pixels, of any edge of the crop square.	Required to crop an uploaded file or an existing file.	29.0
`cropX`	Integer	The position X, in pixels, from the left edge of the image to the start of the crop square. Top left is position (0,0).	Required to crop an uploaded file or an existing file.	29.0
`cropY`	Integer	The position Y, in pixels, from the top edge of the image to the start of the crop square. Top left is position (0,0).	Required to crop an uploaded file or an existing file.	29.0
`fileId`	String	18 character ID of an existing file. The key prefix must be 069 and the file size must be less than 2 MB. Note: Files uploaded on the Group page and on the User Profile page don't have a file ID and can't be posted or cropped. Files uploaded on the Files page have a file ID and can be posted and cropped.	Required to choose an existing file.	25.0
`versionNumber`	Integer	Version number of the existing content. If not provided, the latest version is used.	Optional	25.0

Note: Photos are processed asynchronously and may not be visible right away.

Response body for GET or HEAD

Photo

Response body for POST

Photo

SEE ALSO:

Update and Crop a Group Photo

Group Records

Get the list of records associated with a group. Associate a record with a group.

Resource

```
/chatter/groups/groupID/records
```

```
/connect/communities/communityId/chatter/groups/groupID/records
```

Available since version

33.0

Requires Chatter

Yes

HTTP methods

GET, HEAD, POST

POST is supported in version 34.0 and later.

Request parameters for GET

Parameter Name	Type	Description	Available Since
`page`	Integer	An integer specifying a page of items. The default value is 0, which returns the first page.	33.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are from 1 through 100. If you don't specify a size, the default is 25.	33.0

Request body for POST

Root XML tag

`<groupRecord>`

JSON example

```
{"recordId" : "001D000000Io9cD"}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`recordId`	String	ID of the record.	Required	34.0

Request parameters for POST

Parameter Name	Type	Description	Available Since
`recordId`	String	ID of the record.	34.0

Response body for GET
Group Record Page

Response body for POST
Group Record

Group Record

Get information about a group record or remove the association between a record and a group.

Resource

```
/chatter/group-records/groupRecordId
```

```
/connect/communities/communityId/chatter/group-records/groupRecordId
```

Available version
34.0

Requires Chatter
Yes

HTTP methods
GET, DELETE, HEAD

Response body for GET
Group Record

Response body for DELETE
204: Successful Delete

Group Settings

Information about the context user's Chatter settings for the specified group.

Resource

```
/chatter/groups/groupId/my-settings
```

```
/connect/communities/communityId/chatter/groups/groupId/my-settings
```

Available since version
27.0

Requires Chatter
Yes

HTTP methods
GET, HEAD, or PATCH

Request body for PATCH

Root XML tag
`<groupChatterSettings>`

Properties

A user's Chatter settings.

Name	Type	Description	Available Version
`emailFrequency`	String	The frequency with which a group member receives email from a group. Valid values are: • `EachPost` • `DailyDigest` • `WeeklyDigest` • `Never` In Communities, the Email on Each Post option is disabled once a group exceeds 10,000 members. All members who had this option selected are automatically switched to Daily Digest.	27.0

Request parameter for PATCH

Parameter Name	Type	Description	Available Since
`emailFrequency`	String	The frequency with which a group member receives email from a group. Valid values are: • `EachPost` • `DailyDigest` • `WeeklyDigest` • `Never`	27.0

Response body for GET, HEAD, or PATCH
Group Chatter Settings

Group Topics

Returns up to five topics most recently used in the specified group.

Resource

```
/chatter/groups/groupID/topics
```

```
/connect/communities/communityId/chatter/groups/groupID/topics
```

Available since version

28.0

Requires Chatter

Yes

HTTP methods

GET or HEAD

Response body

Topic Collection

Influence Resource

List of the percentile thresholds used to categorize an organization's users by their influence in Chatter.

Users are ranked based on the number of posts and comments made, and comments and likes received. The percentile associated with a category represents the bottom threshold for that influencer category. For example: `0.25` defines the cutoff for the `Active Influencer` category, which means it includes users with an influence rank equal to or greater than the 25th percentile (25%) and less than the next highest category threshold.

Influence Thresholds

Resource

```
/chatter/influence/thresholds
```

```
/connect/communities/communityId/chatter/influence/thresholds
```

Available since version

24.0

Requires Chatter

Yes

HTTP method

GET or HEAD

Response body

Influence Thresholds

Example response body

```
{"thresholds": [
   {
      "name": "Top Influencer",
      "threshold": "0.8",
```

```
    }
    {
       "name": "Active Influencer",
       "threshold": "0.25",
    }
    {
       "name": "Observer",
       "threshold": "0.0",
    }
]}
```

Likes Resource

Information about the specified like. Also used to delete a like.

You can find the ID for a specific like from any feed, such as a news feed or a record feed.

Like

Resource

```
/chatter/likes/likeId
```

```
/connect/communities/communityId/chatter/likes/likeId
```

Available since version

23.0

Requires Chatter

Yes

HTTP methods

GET, DELETE or HEAD

Response body for GET or HEAD

Like

Managed Topics Resources

Managed topics for a community. Get a list of managed topics. Create, delete, or reorder managed topics in a community.

Available resources:

Resource	Description
/connect/communities/communityId/managed-topics	Get all the managed topics for a community. Create a managed topic. Reorder managed topics.
/connect/communities/communityId/managed-topics/managedTopicId	Get information about or delete a managed topic in a community.

Managed Topics

Get all the managed topics for a community. Create a managed topic. Reorder managed topics.

Only community managers (users with the "Create and Set Up Communities" or "Manage Communities" permission) can create or reorder managed topics.

Resource

```
/connect/communities/communityId/managed-topics
```

Available version

32.0

Requires Chatter

No

HTTP methods

GET, POST, PATCH, HEAD

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
`managedTopicType`	String	Type of managed topic. • `Featured`—Topics that are featured, for example, on the community home page, but don't provide overall navigation. • `Navigational`—Topics that display in a navigational menu in the community. A topic can be associated with up to two managed topic types, so a topic can be both a `Featured` topic and a `Navigational` topic.	Optional	32.0

Response body for GET

Managed Topic Collection

Request body for POST

Root XML tag

`<managedTopic>`

JSON example

```
{
"managedTopicType" :"Featured",
"recordId" :"0TOD00000000cwk"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`managedTopicType`	String	Type of managed topic. • `Featured`—Topics that are featured, for example, on the community home page, but don't provide overall navigation. • `Navigational`—Topics that display in a navigational menu in the community. A topic can be associated with up to two managed topic types, so a topic can be both a `Featured` topic and a `Navigational` topic. You can create up to 25 managed topics per `managedTopicType`.	Required	32.0
`name`	String	Name of the managed topic.	Required for new topics You must specify either `name` or `recordId`. You can't specify both.	32.0
`recordId`	String	ID of the record to become a managed topic. In version 32.0, `recordId` must be a topic ID.	Required for existing topics if the `Name` property isn't used You must specify either `name` or `recordId`. You can't specify both.	32.0

Request parameters for POST

Parameter Name	Type	Description	Required or Optional	Available Version
`managedTopicType`	String	Type of managed topic. • `Featured`—Topics that are featured, for example, on the community home page, but don't provide overall navigation.	Required	32.0

Parameter Name	Type	Description	Required or Optional	Available Version
		• `Navigational`—Topics that display in a navigational menu in the community. A topic can be associated with up to two managed topic types, so a topic can be both a `Featured` topic and a `Navigational` topic. You can create up to 25 managed topics per `managedTopicType`.		
`name`	String	Name of the managed topic.	Required for new topics You must specify either `name` or `recordId`. You can't specify both.	32.0
`recordId`	String	ID of the record to become a managed topic. In version 32.0, `recordId` must be a topic ID.	Required if the `Name` property isn't used You must specify either `name` or `recordId`. You can't specify both.	32.0

Response body for POST

Managed Topic

Request body for PATCH

Root XML tag

```
<managedTopicPositionCollection>
```

JSON example

```
{
"managedTopicPositions" : [
  {
    "managedTopicId" :"0mtD0000000001aIAA",
    "position" :"0"
  },
  {
    "managedTopicId" :"0mtD0000000002qIAA",
    "position" :"2"
  }
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`managedTopicPositions`	Managed Topic Position Input[]	Collection of managed topic position request bodies. This collection can include `Featured` and `Navigational` managed topics and doesn't need to include all managed topics. If you don't include all managed topics in `managedTopicPositions`, the managed topics are reordered by respecting the positions indicated in `managedTopicPositions` and then by pushing down any managed topics that aren't included in `managedTopicPositions` to the next available position. See Reorder managed topics example.	Required	32.0

Reorder managed topics example

If you have these managed topics:

Managed Topic	Position
ManagedTopicA	0
ManagedTopicB	1
ManagedTopicC	2
ManagedTopicD	3
ManagedTopicE	4

And you reorder managed topics by including this information in Managed Topic Position Collection Input:

Managed Topic	Position
ManagedTopicD	0
ManagedTopicE	2

The result is:

Managed Topic	Position
ManagedTopicD	0
ManagedTopicA	1
ManagedTopicE	2
ManagedTopicB	3
ManagedTopicC	4

Response body for PATCH
Managed Topic Collection

Managed Topic

Get information about or delete a managed topic in a community.

Only community managers (users with the "Create and Set Up Communities" or "Manage Communities" permission) can delete managed topics.

Resource

```
/connect/communities/communityId/managed-topics/managedTopicId
```

Available version
32.0

Requires Chatter
No

HTTP methods
GET, DELETE, HEAD

Response body for GET
Managed Topic

Response body for DELETE
204: Successful Delete

Mentions Resources

Information about users and groups that a user can mention in a feed item body or comment body. When a user or group is mentioned, they receive notification.

Use the Mentions Completions resource to get a list of mention proposals. Use the Mentions Validations resource to determine which of those proposals are valid for the context user.

IN THIS SECTION:

Mentions Completions
Information about users and groups to mention in a feed item body or comment body. Use this resource to generate a list of proposed mentions that a user can choose from. A mention is an "@" character followed by a user or group name. When a user or group is mentioned, they receive a notification.

Mentions Validations
Information about whether the specified mentions are valid for the context user. For example, the context user can't mention private groups he doesn't belong to. If such a group were included in the list of mention validations, the `hasErrors` property of the Mention Validations response body would be `true` and the group would have a `validationStatus` of `Disallowed` in its mention validation.

Mentions Completions

Information about users and groups to mention in a feed item body or comment body. Use this resource to generate a list of proposed mentions that a user can choose from. A mention is an "@" character followed by a user or group name. When a user or group is mentioned, they receive a notification.

Resource

```
/chatter/mentions/completions
```

```
/connect/communities/communityId/chatter/mentions/completions
```

Available since version

29.0

Requires Chatter

Yes

HTTP methods

GET, HEAD

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Since
`contextId`	String	A feed item ID (for a mention in a comment) or a feed subject ID (for a mention in a feed item) that narrows search results, with more useful results listed first.	Optional	29.0
`page`	Integer	The number of the page you want returned. Numbering starts at 0. If you pass `0` or omit the property, the first page is returned. Attempting to page beyond 500 results returns an empty response.	Optional	29.0
`pageSize`	Integer	The number of items per page. Valid values are between 1 and 100. The default size is 25. Attempting to retrieve more than 500 results returns an empty response.	Optional	29.0

Parameter Name	Type	Description	Required or Optional	Available Since
`q`	String	A search term. Searches for matching user and group names. To search for a user, a minimum of 1 character is required. To search for a group, a minimum of 2 characters is required. This parameter does not support wildcards.	Required	29.0
`type`	String	The type of mention completion. One of these values: • `All`—All mention completions, regardless of the type of record to which the mention refers. • `Group`—Mention completions for groups. • `User`—Mention completions for users. The default value is `All`.	Optional	29.0

Response body for GET

Mention Completion Page

Example for GET

This example specifies a feed item ID in the `contextId` parameter to request mention completions for a comment on the feed item:

```
GET /chatter/mentions/completions?contextId=0D5D0000000Hwky&q=g
```

This is the response body:

```
{
   "currentPageUrl":
"/services/data/v34.0/chatter/mentions/completions?contextId=0D5D0000000Hwky&q=g",
   "mentionCompletions": [
      {
         "additionalLabel": null,
         "description": "VP Sales",
         "name": "Gordon Johnson",
         "photoUrl": "https://instance_name/profilephoto/729D00000000LJL/T",
         "recordId": "005D0000001LLO1IAO"
      },
      {
         "additionalLabel": null,
         "description": "A place for constructive cr...",
         "name": "Airing of Grievances",
         "photoUrl": "https://instance_name/profilephoto/0F9/T",
         "recordId": "0F9D00000000izfKAA"
      }
   ],
   "nextPageUrl": null,
```

```
    "previousPageUrl": null
}
```

SEE ALSO:

Get a List of @Mention Proposals

Mentions Validations

Information about whether the specified mentions are valid for the context user. For example, the context user can't mention private groups he doesn't belong to. If such a group were included in the list of mention validations, the `hasErrors` property of the Mention Validations response body would be `true` and the group would have a `validationStatus` of `Disallowed` in its mention validation.

Resource

```
/chatter/mentions/validations
```

```
/connect/communities/communityId/chatter/mentions/validations
```

Available since version

29.0

Requires Chatter

Yes

HTTP methods

GET, HEAD

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Since
`parentId`	String	The feed item parent ID (for new feed items) or feed item ID (for comments).	Required	29.0
`recordIds`	String[]	A comma separated list of IDs to be mentioned. The maximum value is 25.	Required	29.0
`visibility`	String	Specifies the type of users who can see a feed item. • `AllUsers`—Visibility is not limited to internal users. • `InternalUsers`—Visibility is limited to internal users.	Required	29.0

Response body for GET

Mention Validation

Usage

First make a request to Mentions Completions to get a list of proposed mentions. Pass those proposed mentions to the Mentions Validations resource to determine whether the mentions are valid for the context user.

SEE ALSO:

Get a List of @Mention Proposals

Motifs Resources

Returns URLs for sets of small, medium, and large motif icons. Records have different sets of motif icons depending on the record type. The motif can also include the record type's base color.

Available resources are:

Resource	Description
`/connect/motifs/idOrPrefix`	Returns a set of small, medium, and large motif icons for the specified record. Can also return the base color for the specified record.
`/connect/motifs/batch/idOrPrefixList`	Returns a set of small, medium, and large motif icons for the specified records. Can also contain the base color for each specified record.

Motifs Resource

Returns a set of small, medium, and large motif icons for the specified record. Can also return the base color for the specified record.

Specify either an ID or key prefix for *idOrPrefix*.

Resource

```
/connect/motifs/idOrPrefix
```

```
/connect/communities/communityId/motifs/idOrPrefix
```

Available since release

27.0

Requires Chatter

No

HTTP methods

GET or HEAD

Response body

Motif

Example of GET

Request for user motifs: `/connect/motifs/005D0000001LLO1`

Response:

```
{
   "color": "1797C0",
   "largeIconUrl": "/img/icon/profile64.png",
   "mediumIconUrl": "/img/icon/profile32.png",
   "smallIconUrl": "/img/icon/profile16.png"
}
```

Batch Motifs Resource

Returns a set of small, medium, and large motif icons for the specified records. Can also contain the base color for each specified record.

Specify a comma-separated list of IDs and key prefixes for *idOrPrefixList*. You can mix IDs and key prefixes in the same request.

Resource

```
/connect/motifs/batch/idOrPrefixList
```

```
/connect/communities/communityId/motifs/batch/idOrPrefixList
```

Available since release

27.0

Requires Chatter

No

HTTP methods

GET or HEAD

Response body

Batch Results

Example of GET

Request for motifs for a user and a file: `/connect/motifs/batch/005D0000001LLO1,069D00000001FHF`

Response:

```
{"results": [
   {
      "result": {
         "color": "1797C0",
         "largeIconUrl": "/img/icon/profile64.png",
         "mediumIconUrl": "/img/icon/profile32.png",
         "smallIconUrl": "/img/icon/profile16.png"
      },
      "statusCode": 200
   },
   {
      "result": {
         "color": "993333",
         "largeIconUrl": "/img/content/content64.png",
         "mediumIconUrl": "/img/content/content32.png",
         "smallIconUrl": "/img/icon/files16.png"
      },
      "statusCode": 200
```

```
    }
]}
```

Push Notifications Resource

Send a mobile push notification to connected apps on users' devices. This resource is only accessible when the session is established with a connected app that is developed in the same organization or installed in the same package as the recipient's connected app.

Resource

```
/connect/notifications/push
```

Available version

31.0

Requires Chatter

No

HTTP methods

POST

Request body for POST

Root XML tag

`<pushNotification>`

JSON example

```
{    "appName" : "TestApp",
     "namespace" : "abc",
     "userIds" : ["005x00000013dPK", "005x00000013dPL"],
     "payload" : "{'aps':{'alert':'test', 'badge':0, 'sound':'default'}}"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`appName`	String	The API name of the connected app that the push notification is sent to.	Required	31.0
`namespace`	String	The namespace of the connected app that the push notification is sent to.	Required if the namespace is set	31.0
`payload`	String	The push notification payload in JSON format.	Required	31.0
`userIds`	String[]	The push notification recipient user ids.	Required	31.0

Request parameters for POST

Parameter Name	Type	Description	Required or Optional	Available Version
`appName`	String	The API name of the connected app that the push notification is sent to.	Required	31.0
`namespace`	String	The namespace of the connected app that the push notification is sent to.	Required if the namespace is set	31.0
`payload`	String	The push notification payload in JSON format.	Required	31.0
`userIds`	String[]	The push notification recipient user ids.	Required	31.0

Question and Answers Suggestions Resource

Get question and answers suggestions.

Resource

```
/connect/question-and-answers/suggestions
```

```
/connect/communities/communityId/question-and-answers/suggestions
```

Available version

32.0

Requires Chatter

No

HTTP methods

GET

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
`includeArticles`	Boolean	Specify `true` to include knowledge articles in the search results. To return only questions, specify `false`.	Optional	32.0
`maxResults`	Integer	The maximum number of results to return for each type of item. Possible values are 1–10. The default value is 5.	Optional	32.0
`q`	String	Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards. The maximum length is 255 characters.	Required	32.0

Parameter Name	Type	Description	Required or Optional	Available Version
`subjectId`	String	Specify a subject ID to search only questions on that object. If the ID is a topic or a user, the ID is ignored.	Optional	32.0

Response body for GET

Question and Answers Suggestion Collection

Recommendation Definition Resources

Recommendation definitions allow you to create custom recommendations that appear in communities, encouraging users to watch videos, take training and more. Use these resources to get information about, create, modify, and delete recommendation definitions. Also get information about, upload, modify, and delete recommendation definition photos.

Only community managers (users with the "Create and Set Up Communities" or "Manage Communities" permission) and users with the "Modify All Data" permission can access, create, and delete recommendation definitions and scheduled recommendations. These recommendations appear directly in the feed in communities in the Salesforce1 mobile browser app. They also appear, by default, on the right side of the home page in communities using the Summer '15 or later version of the Napili template.

Available resources:

Resource	Description
`/connect/recommendation-definitions`	Get a list of recommendation definitions or create a recommendation definition.
`/connect/recommendation-definitions/`***`recommendationDefinitionId`***	Get information about, modify, or delete a recommendation definition.
`/connect/recommendation-definitions/`***`recommendationDefinitionId`***`/photo`	Get information about, upload, modify, or delete the photo for a recommendation definition.

Recommendation Definitions

Get a list of recommendation definitions or create a recommendation definition.

Only community managers (users with the "Create and Set Up Communities" or "Manage Communities" permission) and users with the "Modify All Data" permission can access, create, and delete recommendation definitions and scheduled recommendations. These recommendations appear directly in the feed in communities in the Salesforce1 mobile browser app. They also appear, by default, on the right side of the home page in communities using the Summer '15 or later version of the Napili template.

Resource

```
/connect/recommendation-definitions
```

```
/connect/communities/communityId/recommendation-definitions
```

Available version

33.0

Requires Chatter
No

HTTP methods
GET, HEAD, POST

Request body for POST

JSON example

```
{
   "actionUrl" : "https://www.example.com",
   "actionUrlName" : "Launch",
   "explanation" : "Click this link to learn more!",
   "name" : "Corporate Training Project",
   "title" : "Want to learn more about Corporate Training Project?"}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`actionUrl`	String	The URL for acting on the recommendation, for example, the URL to join a group.	Required	33.0
`actionUrlName`	String	The text label for the action URL in the user interface, for example, "Launch."	Required	34.0
`explanation`	String	The explanation, or body, of the recommendation.	Required	33.0
`name`	String	The name of the recommendation definition. The name is displayed in Setup.	Required	33.0
`title`	String	The title of the recommendation definition.	Optional	33.0

Request parameters for POST

Parameter Name	Type	Description	Required or Optional	Available Version
`actionUrl`	String	The URL for acting on the recommendation, for example, the URL to join a group.	Required	33.0
`actionUrlName`	String	The text label for the action URL in the user interface, for example, "Launch."	Required	34.0
`explanation`	String	The explanation, or body, of the recommendation.	Required	33.0

Parameter Name	Type	Description	Required or Optional	Available Version
`name`	String	The name of the recommendation definition. The name is displayed in Setup.	Required	33.0
`title`	String	The title of the recommendation definition.	Optional	33.0

Response body for POST

Recommendation Definition

Response body for GET

Recommendation Definition Collection

Recommendation Definition

Get information about, modify, or delete a recommendation definition.

Only community managers (users with the "Create and Set Up Communities" or "Manage Communities" permission) and users with the "Modify All Data" permission can access, create, and delete recommendation definitions and scheduled recommendations. These recommendations appear directly in the feed in communities in the Salesforce1 mobile browser app. They also appear, by default, on the right side of the home page in communities using the Summer '15 or later version of the Napili template.

Resource

```
/connect/recommendation-definitions/recommendationDefinitionId
```

```
/connect/communities/communityId/recommendation-definitions/recommendationDefinitionId
```

Available version

33.0

Requires Chatter

No

HTTP methods

GET, HEAD, PATCH, DELETE

Request body for PATCH

JSON example

```
{
   "actionUrl" : "https://www.example.com",
   "actionUrlName" : "Launch",
   "explanation" : "Click this link to learn more!",
   "name" : "Corporate Training Project",
   "title" : "Want to learn more about Corporate Training Project?"}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`actionUrl`	String	The URL for acting on the recommendation, for example, the URL to join a group.	Optional	33.0
`actionUrlName`	String	The text label for the action URL in the user interface, for example, "Launch."	Optional	34.0
`explanation`	String	The explanation, or body, of the recommendation.	Optional	33.0
`name`	String	The name of the recommendation definition. The name is displayed in Setup.	Optional	33.0
`title`	String	The title of the recommendation definition.	Optional	33.0

Request parameters for PATCH

Parameter Name	Type	Description	Required or Optional	Available Version
`actionUrl`	String	The URL for acting on the recommendation, for example, the URL to join a group.	Optional	33.0
`actionUrlName`	String	The text label for the action URL in the user interface, for example, "Launch."	Optional	34.0
`explanation`	String	The explanation, or body, of the recommendation.	Optional	33.0
`name`	String	The name of the recommendation definition. The name is displayed in Setup.	Optional	33.0
`title`	String	The title of the recommendation definition.	Optional	33.0

Response body for GET and PATCH

Recommendation Definition

Response for DELETE

204: Successful Delete

Recommendation Definition Photo

Get information about, upload, modify, or delete the photo for a recommendation definition.

Only community managers (users with the "Create and Set Up Communities" or "Manage Communities" permission) and users with the "Modify All Data" permission can access, create, and delete recommendation definitions and scheduled recommendations. These recommendations appear directly in the feed in communities in the Salesforce1 mobile browser app. They also appear, by default, on the right side of the home page in communities using the Summer '15 or later version of the Napili template.

To use an image from the Files page as a recommendation definition photo (existing content), pass the file ID in the `fileId` property of the request body or in the `fileId` request parameter.

To upload a binary file as a recommendation definition photo, you must send it in a multipart/form-data message. For information about how to create the multipart/form-data message, see Uploading Binary Files.

Resource

```
/connect/recommendation-definitions/recommendationDefinitionId/photo
```

Available version

33.0

Requires Chatter

Yes

HTTP methods

GET, HEAD, PUT, DELETE

Request body for PUT

Root XML tag

```
<photo>
```

JSON example

```
{   "cropSize" : "120",
    "cropX" : "0",
    "cropY" : "0",
    "fileId" : "069D00000001IOh"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`cropSize`	Integer	The length, in pixels, of any edge of the crop square.	Required to crop an uploaded file or an existing file.	29.0
`cropX`	Integer	The position X, in pixels, from the left edge of the image to the start of the crop square. Top left is position (0,0).	Required to crop an uploaded file or an existing file.	29.0
`cropY`	Integer	The position Y, in pixels, from the top edge of the image to the start of the crop square. Top left is position (0,0).	Required to crop an uploaded file or an existing file.	29.0
`fileId`	String	18–character ID of an existing file. The key prefix must be 069 and the file size must be less than 2 MB. Note: Images uploaded on the Group page and on the	Required to choose an existing file.	25.0

Name	Type	Description	Required or Optional	Available Version
		User page don't have file IDs and therefore can't be used.		
`versionNumber`	Integer	Version number of an existing file. If not provided, the latest version is used.	Optional	25.0

Request parameters for PUT

Parameter Name	Type	Description	Required or Optional	Available Version
`cropSize`	Integer	The length, in pixels, of any edge of the crop square.	Required to crop an uploaded file or an existing file.	29.0
`cropX`	Integer	The position X, in pixels, from the left edge of the image to the start of the crop square. Top left is position (0,0).	Required to crop an uploaded file or an existing file.	29.0
`cropY`	Integer	The position Y, in pixels, from the top edge of the image to the start of the crop square. Top left is position (0,0).	Required to crop an uploaded file or an existing file.	29.0
`fileId`	String	18 character ID of an existing file. The key prefix must be 069 and the file size must be less than 2 MB. Note: Files uploaded on the Group page and on the User Profile page don't have a file ID and can't be posted or cropped. Files uploaded on the Files page have a file ID and can be posted and cropped.	Required to choose an existing file.	25.0

Parameter Name	Type	Description	Required or Optional	Available Version
`versionNumber`	Integer	Version number of the existing content. If not provided, the latest version is used.	Optional	25.0

Note: Photos are processed asynchronously and may not be visible right away.

Response body for GET and PUT

Photo

Records Resources

Get information about the followers of a record. Add and remove topics from records.

Available resources are:

IN THIS SECTION:

Followers on Records Resource

Get information about the followers of the specified record.

Topics on Records Resource

The topics assigned to the specified record. Get information about topics assigned to the specified record, and add and remove topics from the specified record. Administrators must enable topics for objects before users can add topics to records of that object type.

SEE ALSO:

Record Feed Resource

Followers on Records Resource

Get information about the followers of the specified record.

Resource

```
/chatter/records/recordId/followers
```

```
/connect/communities/communityId/chatter/records/recordId/followers
```

As of version 29.0, you can use a topic ID for the *recordId*.

Available since version

23.0

Requires Chatter

Yes

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
`page`	Integer	An integer specifying a page of items. The default value is 0, which returns the first page.
`pageSize`	Integer	Specifies the number of items per page. Valid values are between 1 and 1000. If you don't specify a size, the default is 25.

Response body

Subscription Page

Topics on Records Resource

The topics assigned to the specified record. Get information about topics assigned to the specified record, and add and remove topics from the specified record. Administrators must enable topics for objects before users can add topics to records of that object type.

Resource

```
/connect/records/recordId/topics
```

```
/connect/communities/communityId/records/recordId/topics
```

Available since version

30.0

Requires Chatter

No

HTTP methods

GET, DELETE, POST

Request parameters for DELETE

Parameter Name	Type	Description	Required or Optional	Available Version
`topicId`	String	ID of an existing topic	Required	30.0

Response body for DELETE

204: Successful Delete

Request body for POST

Root XML tag

```
<topicAssignment>
```

JSON

The following example assigns a topic to a record:

```
{ "topicId": "0TOD0000000090r"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`topicId`	String	ID of an existing topic.	Required for existing topics if the `topicName` property isn't used.	30.0
`topicName`	String	Name of a new or existing topic.	Required for new topics; required for existing topics if the `topicId` property isn't used.	30.0

Request parameters for POST

Parameter Name	Type	Description	Required or Optional	Available Version
`topicId`	String	ID of an existing topic	Required for existing topics if the `topicName` parameter isn't used	30.0
`topicName`	String	Name of a new or existing topic	Required for new topics; required for existing topics if the `topicId` parameter isn't used	30.0

Response body for POST

Topic

Example for POST

This example passes the ID of an existing topic in the `topidId` parameter to associate it with an opportunity record:

```
POST /connect/records/006D000000CtYBc/topics?topicId=0TOD0000000090r
```

Scheduled Recommendations Resources

Get information about, create, update, or delete scheduled recommendations.

Only community managers (users with the "Create and Set Up Communities" or "Manage Communities" permission) and users with the "Modify All Data" permission can access, create, and delete recommendation definitions and scheduled recommendations. These recommendations appear directly in the feed in communities in the Salesforce1 mobile browser app. They also appear, by default, on the right side of the home page in communities using the Summer '15 or later version of the Napili template.

Available resources:

Resource	Description
`/connect/scheduled-recommendations`	Get a list of scheduled recommendations or create a scheduled recommendation.

Resource	Description
`/connect/scheduled-recommendations/scheduledRecommendationId`	Get information about, update, or delete a scheduled recommendation.

Scheduled Recommendations

Get a list of scheduled recommendations or create a scheduled recommendation.

Only community managers (users with the "Create and Set Up Communities" or "Manage Communities" permission) and users with the "Modify All Data" permission can access, create, and delete recommendation definitions and scheduled recommendations. These recommendations appear directly in the feed in communities in the Salesforce1 mobile browser app. They also appear, by default, on the right side of the home page in communities using the Summer '15 or later version of the Napili template.

Resource

```
/connect/scheduled-recommendations
```

```
/connect/communities/communityId/scheduled-recommendations
```

Available version

33.0

Requires Chatter

No

HTTP methods

GET, HEAD, POST

Request body for POST

JSON example for creating a new scheduled recommendation

```
{
   "rank" : "2",
   "recommendationDefinitionId" : "0RDD00000004CEhOAM"}
```

JSON example for updating an existing scheduled recommendation

```
{   "rank" : "2"}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`enabled`	Boolean	Indicates whether scheduling is enabled. If `true`, the recommendation is enabled and appears in communities. If `false`, recommendations in feeds in the Salesforce1 mobile browser app aren't removed, but no new recommendations appear. In communities using the Summer '15 or later version of the Napili template, disabled recommendations no longer appear.	Optional	34.0

Name	Type	Description	Required or Optional	Available Version
`rank`	Integer	Relative rank of the scheduled recommendation indicated by ascending whole numbers starting with 1. Setting the rank is comparable to an insertion into an ordered list. The scheduled recommendation is inserted into the position specified by the `rank`. The `rank` of all the scheduled recommendations after it is pushed down. See Ranking scheduled recommendations example. If the specified `rank` is larger than the size of the list, the scheduled recommendation is put at the end of the list. The `rank` of the scheduled recommendation is the size of the list, instead of the one specified. If a `rank` is not specified, the scheduled recommendation is put at the end of the list.	Optional	33.0
`recommendationDefinitionId`	String	ID of the recommendation definition that this scheduled recommendation schedules.	Required for creating a scheduled recommendation. You can't specify a `recommendationDefinitionId` when updating an existing scheduled recommendation.	33.0

Request parameters for POST

Parameter Name	Type	Description	Required or Optional	Available Version
`enabled`	Boolean	Indicates whether scheduling is enabled. If `true`, the recommendation is enabled and appears in communities. If `false`, recommendations in feeds in the Salesforce1 mobile browser app aren't removed, but no new recommendations appear. In communities using the Summer	Optional	34.0

Parameter Name	Type	Description	Required or Optional	Available Version
		'15 or later version of the Napili template, disabled recommendations no longer appear.		
`rank`	Integer	Relative rank of the scheduled recommendation indicated by ascending whole numbers starting with 1. Setting the rank is comparable to an insertion into an ordered list. The scheduled recommendation is inserted into the position specified by the `rank`. The `rank` of all the scheduled recommendations after it is pushed down. See Ranking scheduled recommendations example. If the specified `rank` is larger than the size of the list, the scheduled recommendation is put at the end of the list. The `rank` of the scheduled recommendation is the size of the list, instead of the one specified. If a `rank` is not specified, the scheduled recommendation is put at the end of the list.	Optional	33.0
`recommendationDefinitionId`	String	ID of the recommendation definition that this scheduled recommendation schedules.	Required	33.0

Response body for POST

Scheduled Recommendation

Response body for GET

Scheduled Recommendation Collection

Scheduled Recommendation

Get information about, update, or delete a scheduled recommendation.

Only community managers (users with the "Create and Set Up Communities" or "Manage Communities" permission) and users with the "Modify All Data" permission can access, create, and delete recommendation definitions and scheduled recommendations. These recommendations appear directly in the feed in communities in the Salesforce1 mobile browser app. They also appear, by default, on the right side of the home page in communities using the Summer '15 or later version of the Napili template.

Resource

```
/connect/scheduled-recommendations/scheduledRecommendationId
```

```
/connect/communities/communityId/scheduled-recommendations/scheduledRecommendationId
```

Available version

33.0

Requires Chatter

No

HTTP methods

GET, HEAD, PATCH, DELETE

Request body for PATCH

JSON example for creating a new scheduled recommendation

```
{
   "rank" : "2",
   "recommendationDefinitionId" : "0RDD00000004CEhOAM"}
```

JSON example for updating an existing scheduled recommendation

```
{   "rank" : "2"}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`enabled`	Boolean	Indicates whether scheduling is enabled. If `true`, the recommendation is enabled and appears in communities. If `false`, recommendations in feeds in the Salesforce1 mobile browser app aren't removed, but no new recommendations appear. In communities using the Summer '15 or later version of the Napili template, disabled recommendations no longer appear.	Optional	34.0
`rank`	Integer	Relative rank of the scheduled recommendation indicated by ascending whole numbers starting with 1. Setting the rank is comparable to an insertion into an ordered list. The scheduled recommendation is inserted into the position specified by the `rank`. The `rank` of all the scheduled recommendations after it is pushed down. See Ranking scheduled recommendations example.	Optional	33.0

Name	Type	Description	Required or Optional	Available Version
		If the specified `rank` is larger than the size of the list, the scheduled recommendation is put at the end of the list. The `rank` of the scheduled recommendation is the size of the list, instead of the one specified. If a `rank` is not specified, the scheduled recommendation is put at the end of the list.		
`recommendationDefinitionId`	String	ID of the recommendation definition that this scheduled recommendation schedules.	Required for creating a scheduled recommendation. You can't specify a `recommendationDefinitionId` when updating an existing scheduled recommendation.	33.0

Request parameters for PATCH

Parameter Name	Type	Description	Required or Optional	Available Version
`enabled`	Boolean	Indicates whether scheduling is enabled. If `true`, the recommendation is enabled and appears in communities. If `false`, recommendations in feeds in the Salesforce1 mobile browser app aren't removed, but no new recommendations appear. In communities using the Summer '15 or later version of the Napili template, disabled recommendations no longer appear.	Optional	34.0
`rank`	Integer	Relative rank of the scheduled recommendation indicated by ascending whole numbers starting with 1. Setting the rank is comparable to an insertion into an ordered list. The scheduled recommendation is inserted into the position specified by the `rank`. The `rank` of all the scheduled	Optional	33.0

Parameter Name	Type	Description	Required or Optional	Available Version
		recommendations after it is pushed down. See Ranking scheduled recommendations example. If the specified `rank` is larger than the size of the list, the scheduled recommendation is put at the end of the list. The `rank` of the scheduled recommendation is the size of the list, instead of the one specified. If a `rank` is not specified, the scheduled recommendation is put at the end of the list.		

Response body for PATCH

Scheduled Recommendation

Response for DELETE

204: Successful Delete

Deleting a scheduled recommendation is comparable to a deletion in an ordered list. All scheduled recommendations after the deleted scheduled recommendation receive a new, higher rank automatically.

Subscriptions Resource

Information about the specified subscription. Also used to delete a subscription, for example, to unfollow a record or a topic.

A subscription ID is returned as part of the response body for `follower` and `following` resources, for example, `/records/`***`recordId`***`/followers`. In addition, subscriptions IDs are also returned in many summary response bodies, such as group summary or user summary.

Resource

```
/chatter/subscriptions/subscriptionId
```

```
/connect/communities/communityId/chatter/subscriptions/subscriptionId
```

Available since version

23.0

Requires Chatter

Yes

HTTP methods

GET, DELETE, or HEAD

Response body for GET

Subscription

SEE ALSO:

Stop Following a Record

Topics Resources

List of topics for the organization, list of suggested topics for a specified string of text, feed item, or record, list of trending topics for the organization, information about a specified topic, endorsements for a specific topic, information about the files posted recently to a specified topic, information about the groups recently contributing to a specified topic, knowledgeable people for a specified topic, and list of topics related to a specified topic.

Available resources:

Resource	Description
`/connect/topics`	List of topics for the organization.
`/connect/topics/topicId`	Information about the specified topic. Get information about a topic, update a topic's name or description, merge topics, and delete a topic.
`/connect/topics/suggestions`	List of suggested topics for a specified string of text, feed item, or record.
`/connect/topics/trending`	List of the top five trending topics for the organization.
`/connect/topics/topicId/endorsements`	Endorsements on knowledgeable people for the specified topic. Get endorsements for a specified topic and endorse a user for the specified topic.
`/connect/topic-endorsements/endorsementId`	An endorsement. Get information about an endorsement and remove an endorsement.
`/connect/topics/topicId/files`	Returns information about the five files most recently posted to the specified topic.
`/connect/topics/topicId/groups`	Returns information about the five groups that most recently contributed to the specified topic.
`/connect/topics/topicId/knowledgeable-users`	People who are knowledgeable about topics. Get a list of knowledgeable people for the specified topic.
`/connect/topics/topicId/related-topics`	List of five topics most closely related to the specified topic.
`/connect/topics/topicId/topic-opt-outs`	Topic opt outs. Get a list of topic opt outs for the specified topic. Hide yourself from the Knowledgeable People list for the specified topic.
`/connect/topic-opt-outs/topicOptOutId`	Topic opt out. After hiding yourself from the Knowledgeable People list, show yourself again.

List of Topics

List of topics for the organization.

Resource

```
/connect/topics
```

```
/connect/communities/communityId/topics
```

Available since version

29.0

In version 28.0, use `/chatter/topics`.

Requires Chatter

No

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Values	Description	Required or Optional	Available Version
`exactMatch`	Boolean	Use with the `q` parameter and specify `true` to find a topic by its exact, case-insensitive name. Default value is `false`.	Optional	28.0
`page`	Integer	An integer greater than 0 specifying a page of topics. The default value is zero.	Optional	28.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are between 1 and 100. The default value is 25.	Optional	28.0
`q`	String	A query string to filter the results. The string must contain at least two characters, not including wildcards.	Optional	28.0
`sort`	String	One of the following values: • `popularDesc`—Sorts topics by popularity with the most	Optional	28.0

Parameter Name	Values	Description	Required or Optional	Available Version
		popular first. This value is the default. • `alphaAsc`—Sorts topics alphabetically.		

Response body

Topic Collection

Example for GET

This example requests a list of topics sorted alphabetically:

```
/connect/topics?sort=alphaAsc
```

Topics Suggestions

List of suggested topics for a specified string of text, feed item, or record.

Resource

```
/connect/topics/suggestions
```

```
/connect/communities/communityId/topics/suggestions
```

Available since version

29.0

In version 28.0, use `/chatter/topics/suggestions`.

Requires Chatter

No

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Values	Description	Required or Optional
`maxResults`	Integer	Maximum number of topic suggestions that get returned. The default is 5. Value must be greater than 0 and less than or equal to 25.	Optional
`recordId`	String	18-character ID of a feed item or record from which to extract topics. Administrators must enable topics for objects to see topic suggestions for records of that object type. Topics for	Required if the `text` parameter isn't used

Parameter Name	Values	Description	Required or Optional
		objects is available in version 30.0.	
`text`	String	Text from which to extract topics.	Required if the `recordId` parameter isn't used

Response body

Topic Suggestion Collection

Example for GET

This example gets topic suggestions for the text "Working on the planning meeting for our new initiative."

```
/connect/topics/suggestions?text=Working+on+the+planning+meeting+for+our+new+initiative.
```

Trending Topics

List of the top five trending topics for the organization.

The more frequently people add a specific topic to their posts and comments and comment on or like posts with the same topic over a short period of time, the more likely it is to become a trending topic. For example, if your coworkers are attending the upcoming Dreamforce conference and have started discussing it in Chatter, you may see a trending topic for Dreamforce. A trending topic is not solely based on popularity and usually relates to a one-time or infrequent event that has a spike in activity, such as a conference or a project deadline.

Note: In API versions earlier than 28.0, topics began with a hashtag (#).

Resource

```
/connect/topics/trending
```

```
/connect/communities/communityId/topics/trending
```

Available since version

29.0

In versions 24.0–28.0, use `/chatter/topics/trending` and `/connect/communities/communityId/chatter/topics/trending`.

Requires Chatter

No

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Values	Description
`maxResults`	Integer	Maximum number of trending topics that get returned. The default is 5. Value must be greater than 0 and less than 100.

Response body

Topic Collection

Example response body (version 28.0 and later)

```
{"currentPageUrl" : "/services/data/v34.0/connect/topics/trending?maxResults=5",
"nextPageUrl" : null,
"topics" : [
   {
      "createdDate": "2013-01-01T20:43:31.000Z",
      "id" : "0TOD0000000090qOAA",
      "description" : "Application Programming Interface",
      "talkingAbout" : 527,
      "name" : "API",
      "url": "/services/data/v34.0/connect/topics/0TOD0000000090qOAA"
   },
   {
      "createdDate": "2012-11-14T05:24:23.000Z",
      "id" : "0TOD000000006RxOAI",
      "description" : "The cloud computing event of the year.",
      "talkingAbout" : 424,
      "name" : "Dreamforce",
      "url": "/services/data/v34.0/connect/topics/0TOD000000006RxOAI"
   },
   {
      "createdDate": "2013-02-25T10:47:51.000Z",
      "id" : "0TOD0000000090rOAA",
      "description" : "All things related to mobile.",
      "talkingAbout" : 87,
      "name" : "Mobile",
      "url": "/services/data/v34.0/connect/topics/0TOD0000000090rOAA"
   }
]}
```

Example response body (version 24.0–27.0)

```
{"topics" : [
   {
      "name" : "#API"
   },
   {
      "name" : "#Dreamforce"
   },
   {
      "name" : "#Mobile"
   }
]}
```

Topic

Information about the specified topic. Get information about a topic, update a topic's name or description, merge topics, and delete a topic.

Users with the "Edit Topics" or "Modify All Data" permission can edit topic names and descriptions. Users with the "Delete Topics" or "Modify All Data" permission can merge and delete topics.

Resource

```
/connect/topics/topicId
```

```
/connect/communities/communityId/topics/topicId
```

Available since version

29.0

In version 28.0, use `/chatter/topics/topicId`.

Requires Chatter

No

HTTP methods

GET, PATCH, DELETE, and HEAD

Request body for PATCH

Root XML tag

```
<topic>
```

JSON

```
{
   "description" : "World Wide Web Consortium (W3C) is the main international standards
 organization for the Web."
   "name" : "W3C"}
```

Properties

Name	Type	Description	Available Version
`description`	String	Text description of the topic	28.0
`idsToMerge`	String[]	List of up to five topic IDs to merge with the topic	33.0
`name`	String	Name of the topic Use this property to change only the capitalization and spacing of the topic name.	28.0

Request parameters for PATCH

Parameter Name	Type	Description	Available Since
`description`	String	Text description of the topic	28.0
`idsToMerge`	String[]	List of up to five topic IDs to merge with the topic	33.0
`name`	String	Name of the topic Use this property to change only the capitalization and spacing of the topic name.	28.0

Response body for GET, HEAD, or PATCH

Topic

Response body for DELETE

204: Successful Delete

Topic deletion is asynchronous. If a topic is requested before the deletion completes, the response is 200: Successful and the `isBeingDeleted` property of the Topic output is `true` in version 33.0 and later. If a topic is requested after the deletion completes, the response is 404: Not found.

Example for PATCH

This example uses request parameters to update the description of a topic:

```
/connect/topics/0TOD000000006RxOAI?description=Edit+requests
```

This is the same request using a JSON request body:

```
/connect/topics/0TOD000000006RxOAI

{
   "description" : "Edit requests"
}
```

Topics, Endorse People

Endorsements on knowledgeable people for the specified topic. Get endorsements for a specified topic and endorse a user for the specified topic.

Resource

```
/connect/topics/topicId/endorsements
```

```
/connect/communities/communityId/topics/topicId/endorsements
```

Available since version

30.0

Requires Chatter

Yes

HTTP methods

GET, HEAD, POST

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Since
`endorseeId`	String	Specifies the ID of the user who received endorsements.	Optional	31.0
`endorserId`	String	Specifies the ID of the user who endorsed others.	Optional	31.0
`page`	Integer	An integer greater than 0 specifying a page of	Optional	30.0

Parameter Name	Type	Description	Required or Optional	Available Since
		endorsements. The default value is zero.		
`pageSize`	Integer	Specifies the number of items per page. Valid values are between 1 and 100. The default value is 25.	Optional	30.0

Request body for POST

Root XML tag

```
<topicEndorsement>
```

JSON

```
{
   "userId" : "005B0000000Ge16"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`userId`	String	ID of the user to be endorsed for the specified topic.	Required	30.0

Request parameters for POST

Parameter Name	Type	Description	Available Since
`userId`	String	ID of the user to endorse	30.0

Response body for GET

Topic Endorsement Collection

Response body for POST

Topic Endorsement

Topic Endorsements

An endorsement. Get information about an endorsement and remove an endorsement.

Resource

```
/connect/topic-endorsements/endorsementId
```

```
/connect/communities/communityId/topic-endorsements/endorsementId
```

Available since version

30.0

Requires Chatter

Yes

HTTP methods

GET, HEAD, DELETE

Response body for GET

Topic Endorsement

Response body for DELETE

204: Successful Delete

Example for DELETE

When you endorse someone for a topic, an endorsement record is created with the key prefix `0en`. To remove an endorsement, delete the endorsement record.

```
DELETE /connect/topic-endorsements/0enD0000000003UIAQ
```

This is the response:

```
HTTP 204
```

Topic Files

Returns information about the five files most recently posted to the specified topic.

Resource

```
/connect/topics/topicId/files
```

```
/connect/communities/communityId/topics/topicId/files
```

Available since version

29.0

In version 28.0, use `/chatter/topics/topicId/files`.

Requires Chatter

No

HTTP methods

GET or HEAD

Response body

File Summary Page

Example response body

This GET request returns information about every file in the specified topic:

```
/connect/topics/0TOD000000009Q2/files
```

The response is:


```
{
   "currentPageUrl": null,
   "files": [{
      "checksum": "a65b71bc6a31fa95dd0a9232c9f58c86",
      "contentSize": 100377,
      "contentUrl": null,
      "description": null,
      "downloadUrl":
"/services/data/v34.0/chatter/files/069D00000001E32IAE/content?versionNumber=1",
      "fileExtension": "png",
      "fileType": "Png",
      "flashRenditionStatus": "Na",
      "id": "069D00000001E32IAE",
      "isInMyFileSync": false,
      "mimeType": "image/png",
      "modifiedDate": "2012-07-17T21:47:27.000Z",
      "motif": {
         "color": "993333",
         "largeIconUrl": "/img/content/content64.png",
         "mediumIconUrl": "/img/content/content32.png",
         "smallIconUrl": "/img/icon/files16.png"
      },
      "mySubscription": {
         "id": "0E8D0000000T5JrKAK",
         "url": "/services/data/v34.0/chatter/subscriptions/0E8D0000000T5JrKAK"
      },
      "name": "Recruitment Poster",
      "origin": "Chatter",
      "owner": {
         "companyName": "SeattleApps",
         "firstName": "Rene",
         "id": "005D0000001LFNnIAO",
         "isActive": true,
         "lastName": "Holmes",
         "motif": {
            "color": "1797C0",
            "largeIconUrl": "/img/icon/profile64.png",
            "mediumIconUrl": "/img/icon/profile.png",
            "smallIconUrl": "/img/icon/profile.png"
      },
         "mySubscription": null,
         "name": "Rene Holmes",
         "photo": {
            "fullEmailPhotoUrl":
"https://instance_name/ncsphoto/SqRfG_-PUTY6KtfOp_f0cEw-MQI3xmkJ21oNEOQ-5sZVyX4qlsJTXfOWM77RW7yt",

            "largePhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/F",
            "photoVersionId": "729D00000000BDV",
            "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/T",
            "standardEmailPhotoUrl":
"https://instance_name/ncsphoto/SqRfG_-PUTY6KtfOp_f0cEw-MQI3xmkJ21oNEOQ-5sbktFTIf9vWhFTTALJiaFz_",

            "url": "/services/data/v34.0/chatter/users/005D0000001LL8OIAW/photo"
```

```
        },
        "title": null,
        "type": "User",
        "url": "/services/data/v34.0/chatter/users/005D0000001LFNnIAO",
        "userType": "Internal"
      },
      "parentFolder" : "null",
      "pdfRenditionStatus": "Na",
     "renditionUrl": "/services/data/v34.0/chatter/files/069D00000001E32IAE/rendition",

      "thumb120By90RenditionStatus": "Success",
      "thumb240By180RenditionStatus": "Success",
      "thumb720By480RenditionStatus": "Success",
      "title": "Recruitment Poster",
      "type": "ContentDocument",
      "url": "/services/data/v34.0/chatter/files/069D00000001E32IAE?versionNumber=1",
      "versionNumber": "1"
   }],
   "nextPageUrl": null,
   "previousPageUrl": null
}
```


Topic Groups

Returns information about the five groups that most recently contributed to the specified topic.

Resource

```
/connect/topics/topicId/groups
```

```
/connect/communities/communityId/topics/topicId/groups
```

Available since version

29.0

In version 28.0, use `/chatter/topics/topicId/groups`.

Requires Chatter

Yes

HTTP methods

GET or HEAD

Response body

Group Page

Topics, Knowledgeable People List

People who are knowledgeable about topics. Get a list of knowledgeable people for the specified topic.

Resource

```
/connect/topics/topicId/knowledgeable-users
```

```
/connect/communities/communityId/topics/topicId/knowledgeable-users
```

Available since version
30.0

Requires Chatter
Yes

HTTP methods
GET or HEAD

Request parameters

Parameter Name	Values	Description	Required or Optional	Available Version
`page`	Integer	An integer greater than 0 specifying a page of topics. The default value is zero.	Optional	30.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are between 1 and 100. The default value is 25.	Optional	30.0

Response body
Knowledgeable People Collection

Related Topics

List of five topics most closely related to the specified topic.

Resource

```
/connect/topics/topicId/related-topics
```

```
/connect/communities/communityId/topics/topicId/related-topics
```

Available since version
29.0

In version 28.0, use `/chatter/topics/topicId/relatedtopics`.

Requires Chatter
No

HTTP methods
GET or HEAD

Response body
Topic Collection

Topics, Opt Outs from Knowledgeable People List

Topic opt outs. Get a list of topic opt outs for the specified topic. Hide yourself from the Knowledgeable People list for the specified topic.

Resource

```
/connect/topics/topicId/topic-opt-outs
```

```
/connect/communities/communityId/topics/topicId/topic-opt-outs
```

Available since version
30.0

Requires Chatter
Yes

HTTP methods
GET, HEAD, POST

Response body for GET
Topic Opt Out Collection

Response body for POST
Topic Opt Out

Topics, Opt Out from Knowledgeable People List

Topic opt out. After hiding yourself from the Knowledgeable People list, show yourself again.

Resource

```
/connect/topic-opt-outs/topicOptOutId
```

```
/connect/communities/communityId/topic-opt-outs/topicOptOutId
```

Available since version
30.0

Requires Chatter
Yes

HTTP methods
GET, HEAD, DELETE

Response body for GET
Topic Opt Out

Response body for DELETE
204: Successful Delete

Example for DELETE
When you opt out of the knowledgeable people list for a topic, a topic opt out record is created with the key prefix `0eb`. To show yourself in the knowledgeable people list after you've opted out, delete the topic opt out record.

```
DELETE /connect/topic-opt-outs/0ebD0000000003oIAA
```

This is the response:

```
HTTP 204
```

User Profiles Resource

Returns details about a user's profile. The user profile data populates the profile page (also called the Chatter profile page). This data includes user information (such as address, manager, and phone number), some user capabilities (permissions), and a set of subtab apps, which are custom tabs on the profile page.

For information about subtab apps, search for "subtab apps" at help.salesforce.com.

The capabilities information provides some information about the context user's capabilities in relation to another user, called the subject user. For example, the capabilities information indicates whether the context user can chat with the subject user and whether the context user can direct message the subject user. Specify the subject user as the *userID*.

Resource

```
/connect/user-profiles/userID
```

```
/connect/communities/communityId/user-profiles/userID
```

Available since version

29.0

Requires Chatter

Yes

HTTP methods

GET or HEAD

Parameters

Parameter Name	Type	Description	Available Since
`userId`	String	ID of the user	29.0
`communityId`	String	ID of the community	29.0

Response body for GET

User Profile

Users Resources

Information about the user, such as who is following the user, the user's files or recommendations. Also used for posting feed items and updating conversation status.

Available resources are:

Resource	Description
`/chatter/users`	Returns information about all users in an organization.
`/chatter/users/me`	Information about the context or specified user's Chatter profile.

Resource	Description
/chatter/users/batch/***user_list***	Returns information about the specified users' profiles.
/chatter/users/me/conversations	Returns all private conversations for the context user. Also used to search across all conversations.
/chatter/users/me/conversations/***conversationId***	Returns information about the specified private conversation for the context user. Also used to search within a specific conversation.
/chatter/users/me/conversations/***conversationId***/mark-read	Changes the status of the specified conversation.
/chatter/users/me/conversations/unread-count	Returns the number of conversations that have unread messages for the context or specified user.
/chatter/users/me/files/	Post a file, including references to external files, to Salesforce Files Sync or get information about files a user owns. These files don't include files shared with a user, files a user follows, or general organization files.
/chatter/users/me/files/filter/groups	Returns information about files posted to groups that the specified user is a member of.
/chatter/users/me/files/filter/sharedwithme	Returns information about files that have been shared with the specified user.
/chatter/users/me/followers	Returns information about the followers for the specified user.
/chatter/users/me/following	Returns a list of people, groups, records, topics, and files that the specified user is following. Also used to follow records.
/chatter/users/me/groups	The groups the specified user is a member of.
/chatter/users/me/knowledgeable-about-topics	Topics that the specified user is knowledgeable about. Get a list of topics for the specified user.
/chatter/users/me/messages	Returns all the messages for all the private conversations for the context user. Also used to search across all messages and post a message.
/chatter/users/me/messages/***messageId***	Returns the specified Chatter private message.
/chatter/users/me/photo	The user photo. Get, post, and crop the photo.
/chatter/users/me/recommendations	Returns the user, group, file, record, and custom recommendations for the context user.
/chatter/users/me/recommendations/***action***	Returns the recommendations for the specified action for the context user.
/chatter/users/me/recommendations/***action***/***objectCategory***	Returns the recommendations for the specified action and object category for the context user.

Resource	Description
/chatter/users/me/recommendations/***action***/***idPrefix***	Returns the recommendations for the specified action and key prefix for the context user.
/chatter/users/me/recommendations/***action***/***objectID***	Returns the recommendation for the specified action and object ID for the context user. Also used to delete a recommendation.
/chatter/users/***userId***/recommendations/***action***/***objectEnum***	Returns the static recommendation for the specified action and object type for the context user. Also used to delete a static recommendation.
/connect/communities/***communityId*** /chatter/users/***userId***/reputation	Returns the specified community user's reputation.
/chatter/users/me/settings	Information about the context user's global Chatter settings.
/chatter/users/me/status	Returns the most recent user status feed item. Also used to update or delete user status.
/chatter/users/me/topics	Topics recently used by the specified user. Get up to five topics most recently used by the specified user.

User Information

Returns information about all users in an organization.

Resource

```
/chatter/users
```

```
/connect/communities/communityId/chatter/users
```

Available since version

23.0

Requires Chatter

Yes

HTTP methods

GET or HEAD

Request parameters

Parameter	Type	Description	Available Version
page	Integer	An integer specifying a page of items. The default value is 0, which returns the first page.	23.0
pageSize	Integer	The number of items per page. Valid values are between 1 and 250. The default value is 25.	23.0

Parameter	Type	Description	Available Version
`q`	String	Specifies the string to search. The search string must contain at least two characters, not including any wildcards. For more information about wildcards, see Wildcards. This parameter searches user names, not email addresses, titles, and so on.	23.0
`searchContextId`	String	Specify a feed item ID that provides context for the search results. Use this parameter for @mention completion. Passing a feed item ID for `searchContextId` matches users already mentioned in the thread earlier in the returned collection. When using this parameter, you can't query more than 500 results and you can't use wildcards in the search term.	28.0

Response body

User Page

In versions earlier than 28.0, the `user` property in the User Page response body can contain a User Summary or a User Detail response body. In version 28.0 and later, it always contains User Detail. The fields that the context user is not allowed to see are set to `null`.

User Profile Information

Information about the context or specified user's Chatter profile.

Resource

```
/chatter/users/userId
```

```
/connect/communities/communityId/chatter/users/userId
```

Available version

23.0

PATCH available in 29.0

Requires Chatter

Yes

HTTP methods

GET, HEAD, PATCH

Request body for PATCH

Root XML tag

```
<user>
```

Properties

Name	Type	Description	Available Version
`aboutMe`	String	The `aboutMe` property of a User Detail response body. The maximum length is 1000 characters. This property populates the "About Me" section of the user profile, which is visible to all members of a community or an organization.	29.0

JSON example

```
{
   "aboutMe": "Staff Technical Writer responsible for the Connect API and Connect in
 Apex documentation."
}
```

Request parameters for PATCH

Name	Type	Description	Available Version
`aboutMe`	String	The `aboutMe` property of a User Detail response body. The maximum length is 1000 characters. This property populates the "About Me" section of the user profile, which is visible to all members of a community or an organization.	29.0

Response body for GET

- User Detail—Versions 26.0 and later. If the user is external, the properties that the User Detail response body shares with the User Summary response body can have non-null values. Other properties are always `null`.
- User Summary—Versions 25.0 and earlier.

Response body for PATCH

User Detail

Example request parameters for PATCH

This example uses a parameter to update the `aboutMe` property. In the UI, this property populates the "About Me" section in the user profile.

```
/chatter/users/me?aboutMe=Staff+Technical+Writer+at+Seattle+Apps
```

Example request body for PATCH

This example uses a request body to update the `aboutMe` property.

```
/chatter/users/me
```

Request body:

```
{
   "aboutMe": "Staff Technical Writer at Seattle Apps"
}
```

SEE ALSO:

Get Chatter Activity Statistics for a User

Update a User Profile

Batch User Information

Returns information about the specified users' profiles.

Resource

```
/chatter/users/batch/user_list
```

```
/connect/communities/communityId/chatter/users/batch/user_list
```

Where *user_list* is a comma separated list of up to 500 user IDs.

Available since version

23.0

Requires Chatter

Yes

HTTP methods

GET or HEAD

Response body

Batch Results

SEE ALSO:

Get User Information for Multiple Users

User Conversations, General

Returns all private conversations for the context user. Also used to search across all conversations. See User Conversations, Specific on page 371 and User Messages, General on page 382.

Resource

```
/chatter/users/userId/conversations
```

```
/connect/communities/communityId/chatter/users/userId/conversations
```

Note: *userId* must be the ID of the context user or the alias `me`.

Available since version
23.0

Requires Chatter
Yes

HTTP methods
GET or HEAD

Request parameters for GET

Parameter Name	Type	Required or Optional	Description	Available Version
`page`	String	Optional	A page token that indicates where retrieval begins. Get this token from the `NextPageUrl` property in the response body.	23.0
`pageSize`	Integer	Optional	Specifies the number of items per page. Valid values are from 1 through 100. If you don't specify a size, the default is 25.	23.0
`q`	String	Optional	Specifies the string to search. The search string must contain at least two characters, not including any wildcards. For more information about wildcards, see Wildcards. This parameter is optional. Note: The query parameter only searches message text bodies, not user names, titles, and so on.	24.0

Response body
Conversation Summary Collection

User Conversations, Specific

Returns information about the specified private conversation for the context user. Also used to search within a specific conversation.

Note: You can also use the Conversations resource to change the status of a specified conversation. See Change Status of Conversations on page 372.

Resource

```
/chatter/users/userId/conversations/conversationId
```

```
/connect/communities/communityId/chatter/users/userId/conversations/conversationId
```

Note: *userId* must be the ID of the context user or the alias `me`.

Available since version

23.0

Requires Chatter

Yes

HTTP methods

GET, PATCH, or HEAD

Request parameter for GET

Parameter Name	Type	Required or Optional	Description	Available Version
`q`	String	Optional	Specifies the string to search. The search string must contain at least two characters, not including any wildcards. For more information about wildcards, see Wildcards. This parameter is optional. Note: The query parameter only searches message text bodies, not user names, titles, and so on.	24.0

Request body for PATCH

Root XML tag

```
<conversation>
```

Properties

Name	Type	Description	Available Version
`read`	Boolean	Changes the conversation status to unread (`false`) or read (`true`).	24.0

Request parameters for PATCH

Parameter Name	Type	Description
`read`	Boolean	Changes the conversation status to unread (`false`) or read (`true`).

Response body for GET or HEAD

Conversation Detail

Change Status of Conversations

Changes the status of the specified conversation. Also see Using POST, PATCH, or PUT for Input. You can also use PATCH on the conversations resource to change the status of a specified conversation.

Resource

```
/chatter/users/me/conversations/conversationId/mark-read
```

Available versions

23.0–29.0

Requires Chatter

Yes

HTTP methods

POST or PUT

Request body

Root XML tag

```
<conversation>
```

Properties

Name	Type	Description	Available Version
`read`	Boolean	Changes the conversation status to unread (`false`) or read (`true`).	24.0

Request parameters

Parameter Name	Type	Description
`read`	Boolean	Changes the conversation status to unread (`false`) or read (`true`).

Get Unread Count for Conversations

Returns the number of conversations that have unread messages for the context or specified user.

Resource

```
/chatter/users/userId/conversations/unread-count
```

```
/connect/communities/communityId/chatter/users/userId/conversations/unread-count
```

Available since version

23.0

HTTP methods

GET or HEAD

Request body

Conversation Unread Count

Users Files, General

Post a file, including references to external files, to Salesforce Files Sync or get information about files a user owns. These files don't include files shared with a user, files a user follows, or general organization files.

When you post a file, it is uploaded to the Files tab as a private file and is available only to the file owner.

To upload a file, you must send it in a multipart/form-data request. You can include the description and title of the file in the multipart/form-data request as a JSON or XML request body. You can also send the information as request parameters. For information about how to create the multipart/form-data message, see Uploading Binary Files.

Important: Be aware that this resource may return a large amount of data and may take a while to process.

Resource

```
/chatter/users/userId/files/
```

```
/connect/communities/communityId/chatter/users/userId/files/
```

Available since version

24.0

Requires Chatter

Yes

HTTP methods

GET, POST, or HEAD

Request body for POST

Describes a new file input. To upload a file, provide the binary file in a multipart/form-data body part where `name="fileData"`. See Uploading Binary Files.

Root XML tag

```
<fileInput>
```

JSON

```
{
   "desc":"Employee Survey Results",
   "title":"emp_surv_results"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`contentModifiedDate`	Date	File-specific last modified date that can be used to set a client-side modified date.	Optional	32.0
`desc`	String	Description of the file to be uploaded.	Optional	24.0
`isInMyFileSync`	Boolean	`true` if the file is synced withSalesforce Files Sync; `false` otherwise.	Optional	31.0
`isMajorVersion`	Boolean	`true` if the file is a major version; `false` if the file is a minor version. Major versions can't be replaced.	Optional	31.0
`parentFolderId`	String	ID of the containing folder.	Optional	31.0
`repositoryFileId`	String	ID of the external file.	Optional	32.0
`repositoryId`	String	ID of the external data source.	Optional	32.0
`title`	String	Title of the file.	Optional	24.0

Request parameters for POST

Name	Type	Description	Required or Optional	Available version
`contentModifiedDate`	Date	File-specific last modified date that can be used to set a client-side modified date.	Optional	32.0
`desc`	String	Description of the file	Optional	24.0
`isDataSync`	Boolean	The request is automatically generated by an automated client	Optional	31.0
`isInMyFileSync`	Boolean	`true` if the file is synced withSalesforce Files Sync; `false` otherwise. The default value is `false`.	Optional	31.0

Name	Type	Description	Required or Optional	Available version
`isMajorVersion`	Boolean	`true` if the file is a major version; `false` if the file is a minor version. Major versions can't be replaced. The default value is `false`.	Optional	31.0
`repository FileId`	String	ID of the external file.	Optional	32.0
`repositoryId`	String	ID of the external data source.	Optional	32.0
`title`	String	Title of the file	Optional	24.0

Request parameters for GET

Parameter Name	Type	Description	Available Since
`page`	Integer	An integer specifying a page of items. The default value is 0, which returns the first page.	24.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are from 1 through 100. If you don't specify a size, the default is 25.	24.0
`q`	String	A query string to filter the results. The string must contain at least two characters, not including wildcards. For information about wildcards, see Wildcards.	27.0

Response body for GET

File Summary Page

Response body for POST

File Summary

Example POST request

This example creates a new file and uploads it to Salesforce Files Sync. The example shows the body parts in a multipart/form-data request. The first body part is a File Input request body. The second is the content of the file. For more information, see Uploading Binary Files.

```
POST /services/data/v34.0/chatter/users/me/files

Authorization: OAuth
00DD0000000Jhd2!AQIAQC.lh4qTQcBhOPm4TZom5IaOOZLVPVK4wI_rPYJvmE8r2VW8XA.
OZ7S29JEM_7Ctq1lst2dzoV.owisJc0KacUbDxyae
User-Agent: Jakarta Commons-HttpClient/3.0.1
Host: instance_name
```

```
Content-Length: 489
Content-Type: multipart/form-data; boundary=a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq
Accept: application/json

--a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq
Content-Type: application/json; charset=UTF-8
Content-Disposition: form-data; name="json"

{
   "title":"BoatPrices"
}

--a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq
Content-Disposition: form-data; name="fileData"; filename="BoatPrices.txt"
Content-Type: application/octet-stream; charset=ISO-8859-1

...contents of BoatPrices.txt...

--a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq--
```

User Files, Filtered by Group

Returns information about files posted to groups that the specified user is a member of.

Resource

```
/chatter/users/userId/files/filter/groups
```

```
/connect/communities/communityId/chatter/users/userId/files/filter/groups
```

Available since version

24.0

Requires Chatter

Yes

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description	Available Since
page	Integer	An integer specifying a page of items. The default value is 0, which returns the first page.	24.0
pageSize	Integer	Specifies the number of items per page. Valid values are from 1 through 100. If you don't specify a size, the default is 25.	24.0
q	String	A query string to filter the results. The string must contain at least two characters, not	27.0

Parameter Name	Type	Description	Available Since
		including wildcards. For information about wildcards, see Wildcards.	

Response body

File Summary Page

User Files, Filtered by Sharing

Returns information about files that have been shared with the specified user.

Resource

```
/chatter/users/userId/files/filter/shared-with-me
```

```
/connect/communities/communityId/chatter/users/userId/files/filter/shared-with-me
```

Note: The resource can also be accessed with the final keyword as one word:

```
/chatter/users/userid/files/filter/sharedwithme
```

Available since version

24.0

Requires Chatter

Yes

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description	Available Since
`page`	Integer	An integer specifying a page of items. The default value is 0, which returns the first page.	24.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are from 1 through 100. If you don't specify a size, the default is 25.	24.0
`q`	String	A query string to filter the results. The string must contain at least two characters, not including wildcards. For information about wildcards, see Wildcards.	27.0

Response body
File Summary Page

User Followers

Returns information about the followers for the specified user.

Resource

```
/chatter/users/userId/followers
```

```
/connect/communities/communityId/chatter/users/userId/followers
```

Available since version
23.0

Requires Chatter
Yes

HTTP methods
GET or HEAD

Request parameters

Parameter Name	Type	Description
`page`	Integer	Specifies the page number to return. The default value is 0, which returns the first page.
`pageSize`	Integer	Specifies the number of items per page. Valid values are between 1 and 1000. If you don't specify a size, the default is 25.

Response body
Follower Page

Following

Returns a list of people, groups, records, topics, and files that the specified user is following. Also used to follow records. See Using POST, PATCH, or PUT for Input.

Note: You can't use the users resource to add a user to a group. Use the groups resource instead.

Resource

```
/chatter/users/userId/following
```

```
/connect/communities/communityId/chatter/users/userId/following
```

Available since version
23.0

Following topics is available since version 29.0.

Requires Chatter
Yes

HTTP methods
GET, POST, or HEAD

Request body for POST

Root XML tag
`<following>`

Properties

Name	Type	Description	Available Version
`subjectId`	String	The ID of what to follow, for example, a user ID, a record ID, a file ID, a topic ID, and so on. Note: Use Group Member Input for adding members to groups.	23.0 Topic IDs are available in 29.0

Request parameter for POST

Parameter Name	Type	Description	Available
`subjectId`	String	This request enables the user to follow the specified record. This could be a user ID, a file ID, a topic ID, and so on.	23.0 Topic IDs are available in 29.0

Request parameters for GET or HEAD
All parameters are optional.

Parameter Name	Type	Description
`filterType`	String	Specifies the key prefix to filter the type of objects returned. The key prefix is the three-character prefix code in the object ID. Object IDs are prefixed with three-character codes that specify the type of the object. For example, User objects have a prefix of 005 and Group objects have a prefix of 0F9. You don't need to generate the key prefixes yourself. Use the List of Filter Feed URLs to see the available key prefixes.

Parameter Name	Type	Description
`page`	Integer	Specifies the page number to return. The default value is 0, which returns the first page.
`pageSize`	Integer	Specifies the number of items per page. Valid values are between 1 and 1000. If you don't specify a size, the default is 25.

Response body for GET or HEAD

Following Page

Response body for POST

Subscription

SEE ALSO:

Follow a Record

Get a List of Records a User Follows

User Groups

The groups the specified user is a member of.

Resource

```
/chatter/users/userId/groups
```

```
/connect/communities/communityId/chatter/users/userId/groups
```

Available since version

23.0

Requires Chatter

Yes

HTTP methods

GET, HEAD

Request parameters

Parameter Name	Type	Description	Required or Optional	Available Version
`page`	Token	A generated token that indicates the view of feed items. Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`.	Optional	23.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are between 1 and 250. If you don't specify a size, the default is 25.	Optional	23.0

Parameter Name	Type	Description	Required or Optional	Available Version
`q`	String	Specifies the string to search. The search string must contain at least two characters, not including wildcards. See Wildcards.	Optional	30.0

Response body

User Group Page

Example

This GET request returns information about all groups the context user is a member of whose names or descriptions contain the letters `co`.

```
/chatter/users/me/groups?q=co
```

Topics a User is Knowledgeable About

Topics that the specified user is knowledgeable about. Get a list of topics for the specified user.

Resource

```
/chatter/users/userId/knowledgeable-about-topics
```

```
/connect/communities/communityId/chatter/users/userId/knowledgeable-about-topics
```

Available since version

30.0

Requires Chatter

Yes

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Values	Description	Required or Optional	Available Version
`page`	Integer	An integer greater than 0 specifying a page of topics. The default value is zero.	Optional	30.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are between 1 and 100. The default value is 25.	Optional	30.0

Response body

Topics People Are Knowledgeable About Collection

User Messages, General

Returns all the messages for all the private conversations for the context user. Also used to search across all messages and post a message. See Using POST, PATCH, or PUT for Input. See also User Conversations, General on page 369.

Resource

```
/chatter/users/userId/messages
```

```
/connect/communities/communityId/chatter/users/userId/messages
```

Note: *userId* must be the ID of the context user or the alias `me`.

Available since version

23.0

Requires Chatter

Yes

HTTP methods

GET, HEAD, POST

Request body for POST

Root XML tag

`<message>`

Properties

Name	Type	Description	Available Version
`body`	String	Text of the message body	23.0
`inReplyTo`	String	ID of an existing message that identifies which conversation this message is part of.You must specify either `recipients` or `inReplyTo`. You must specify one or the other, not both.	23.0
`recipients`	User Message Recipient List Input	List of users who are the intended message recipients, up to 9. Comma-separated list of user IDs.You must specify either `recipients` or `inReplyTo`. You must specify one or the other, not both.	23.0

Example XML format:

```
<message>
   <body>Text of the message</body>
```

```
   <recipients>
      <recipient>userID</recipient>
      <recipient>userID</recipient>
   </recipients>
   <inReplyTo>messageID</inReplyTo>
</message>
```

Example JSON format:

```
{
   "body": "Text of the message",
   "recipients": ["userID","userID"],
   "inReplyTo": "messageID"
}
```

Request parameters for POST

Parameter Name	Type	Required or Optional	Description	Available Version
`inReplyTo`	String	You must specify either `recipients` or `inReplyTo`. You cannot specify both.	ID of an existing message that identifies which conversation this message is part of	23.0
`recipients`	String	You must specify either `recipients` or `inReplyTo`. You cannot specify both.	List of users who are the intended message recipients, up to 9. Comma-separated list of user IDs.	23.0
`text`	String	Required	The text of the message, up to 10,000 characters	23.0

Request parameters for GET

Parameter Name	Type	Required or Optional	Description	Available Version
`page`	String	Optional	A page token that indicates where retrieval begins. Get this token from the `NextPageUrl` property in the response body.	23.0
`pageSize`	Integer	Optional	Specifies the number of items per page. Valid values are from 1 through 100. If you don't specify a size, the default is 25.	23.0

Parameter Name	Type	Required or Optional	Description	Available Version
q	String	Optional	Specifies the string to search. The search string must contain at least two characters, not including any wildcards. For more information about wildcards, see Wildcards. This parameter is optional. Note: The query parameter only searches message text bodies, not user names, titles, and so on.	24.0

Response body for GET or HEAD

Message Collection

Response body for POST

Message

Example response body for GET or HEAD

```
{
   "currentPageUrl": "/services/data/v34.0/chatter/users/me/messages",
   "messages": [
      {
         "body": {
            "messageSegments": [],
            "text": "Will you be able to meet about your Dreamforce presentation this
week?"
         },
         "conversationId": "03MD000000002LTMAY",
         "conversationUrl":
"/services/data/v34.0/chatter/users/me/conversations/03MD000000002LTMAY",
         "id": "03JD000000003XGMAY",
         "recipients": [
            {
               "companyName": "SeattleApps",
               "firstName": "Rene",
               "id": "005D0000001LFNnIAO",
               "isActive": true,
               "lastName": "Holmes",
               "motif": {
                  "color": "1797C0",
                  "largeIconUrl": "/img/icon/profile64.png",
                  "mediumIconUrl": "/img/icon/profile32.png",
```

```
                  "smallIconUrl": "/img/icon/profile16.png"
               },
               "mySubscription": null,
               "name": "Rene Holmes",
               "photo": {
                  "fullEmailPhotoUrl":
"https://instance_name/ncsphoto/SqRfG_-PUTY6KtfOp_f0cEw-MQI3xmkJ21oNEOQ-5sZVyX4qlsJTXfOWM77RW7yt",

                  "largePhotoUrl":
"https://instance_name/profilephoto/729D00000000XA5/F",
                  "photoVersionId": "729D00000000XA5",
                  "smallPhotoUrl":
"https://instance_name/profilephoto/729D00000000XA5/T",
                  "standardEmailPhotoUrl":
"https://instance_name/ncsphoto/SqRfG_-PUTY6KtfOp_f0cEw-MQI3xmkJ21oNEOQ-5sbktFTIf9vWhFTTALJiaFz_",

                  "url": "/services/data/v34.0/chatter/users/005D0000001LLO6IAO/photo"
               },
               "title": "Senior Manager",
               "type": "User",
               "url": "/services/data/v34.0/chatter/users/005D0000001LFNnIAO",
               "userType": "Internal"
            },
            {
               "companyName": null,
               "firstName": "Desmond",
               "id": "005D0000001LJvOIAW",
               "isActive": true,
               "lastName": "Catillo",
               "motif": {
                  "color": "1797C0",
                  "largeIconUrl": "/img/icon/profile64.png",
                  "mediumIconUrl": "/img/icon/profile32.png",
                  "smallIconUrl": "/img/icon/profile16.png"
               },
               "mySubscription": null,
               "name": "Desmond Catillo",
               "photo": {

                  "fullEmailPhotoUrl":
"https://instance_name/ncsphoto/SqRfG_-PUTY6KtfOp_f0cEw-MQI3xmkJ21oNEOQ-5sZVyX4qlsJTXfOWM77RW7yt",

                  "largePhotoUrl":
"https://instance_name/profilephoto/729D00000000XA5/F",
                  "photoVersionId": "729D00000000XA5",
                  "smallPhotoUrl":
"https://instance_name/profilephoto/729D00000000XA5/T",
                  "standardEmailPhotoUrl":
"https://instance_name/ncsphoto/SqRfG_-PUTY6KtfOp_f0cEw-MQI3xmkJ21oNEOQ-5sbktFTIf9vWhFTTALJiaFz_",

                  "url": "/services/data/v34.0/chatter/users/005D0000001LLO1IAO/photo"
               },
               "title": "Senior Manager",
               "type": "User",
```

```
                "url": "/services/data/v34.0/chatter/users/005D0000001LJvOIAW",
                "userType": "Internal"
            }
         ],
         "sender": {
            "companyName": "SeattleApps",
            "firstName": "Rene",
            "id": "005D0000001LFNnIAO",
            "isActive": true,
            "lastName": "Holmes",
            "motif": {
               "color": "1797C0",
               "largeIconUrl": "/img/icon/profile64.png",
               "mediumIconUrl": "/img/icon/profile32.png",
               "smallIconUrl": "/img/icon/profile16.png"
            },
            "mySubscription": null,
            "name": "Rene Holmes",
            "photo": {
               "fullEmailPhotoUrl":
"https://instance_name/ncsphoto/SqRfG_-PUTY6KtfOp_f0cEw-MQI3xmkJ21oNEOQ-5sZVyX4qlsJTXfOWM77RW7yt",
              "largePhotoUrl": "https://instance_name/profilephoto/729D00000000XA5/F",

               "photoVersionId": "729D00000000XA5",
              "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000XA5/T",

               "standardEmailPhotoUrl":
"https://instance_name/ncsphoto/SqRfG_-PUTY6KtfOp_f0cEw-MQI3xmkJ21oNEOQ-5sbktFTIf9vWhFTTALJiaFz_",

               "url": "/services/data/v34.0/chatter/users/005D0000001LLO6IAO/photo"
            },
            "title": "Senior Manager",
            "type": "User",
            "url": "/services/data/v34.0/chatter/users/005D0000001LFNnIAO",
            "userType": "Internal"
         },
         "sentDate": "2012-07-20T21:27:07.000Z",
         "url": "/services/data/v34.0/chatter/users/me/messages/03JD000000003XGMAY"
      },
      {
         "body": {
            "messageSegments": [],
            "text": "Rene Holmes has shared Recruitment Poster.png with you on
Chatter.\n\nPlease review.\n\nTo view the file, go to:
https://na1-blitz01.soma.salesforce.com/069D00000001E32"
         },
         "conversationId": "03MD000000002LTMAY",
         "conversationUrl":
"/services/data/v34.0/chatter/users/me/conversations/03MD000000002LTMAY",
         "id": "03JD000000003VAMAY",
         "recipients": [
            {
               "companyName": "SeattleApps",
               "firstName": "Rene",
```

```
            "id": "005D0000001LFNnIAO",
            "isActive": true,
            "lastName": "Holmes",
            "motif": {
               "color": "1797C0",
               "largeIconUrl": "/img/icon/profile64.png",
               "mediumIconUrl": "/img/icon/profile32.png",
               "smallIconUrl": "/img/icon/profile16.png"
            },
            "mySubscription": null,
            "name": "Rene Holmes",
            "photo": {
               "fullEmailPhotoUrl":
"https://instance_name/ncsphoto/SqRfG_-PUTY6KtfOp_f0cEw-MQI3xmkJ21oNEOQ-5sZVyX4qlsJTXfOWM77RW7yt",

               "largePhotoUrl":
"https://instance_name/profilephoto/729D00000000XA5/F",
               "photoVersionId": "729D00000000XA5",
               "smallPhotoUrl":
"https://instance_name/profilephoto/729D00000000XA5/T",
               "standardEmailPhotoUrl":
"https://instance_name/ncsphoto/SqRfG_-PUTY6KtfOp_f0cEw-MQI3xmkJ21oNEOQ-5sbktFTIf9vWhFTTALJiaFz_",

               "url": "/services/data/v34.0/chatter/users/005D0000001LLO6IAO/photo"

            },
            "title": "Senior Manager",
            "type": "User",
            "url": "/services/data/v34.0/chatter/users/005D0000001LFNnIAO",
            "userType": "Internal"
         },
         {
            "companyName": null,
            "firstName": "Desmond",
            "id": "005D0000001LJvOIAW",
            "isActive": true,
            "lastName": "Catillo",
            "motif": {
               "color": "1797C0",
               "largeIconUrl": "/img/icon/profile64.png",
               "mediumIconUrl": "/img/icon/profile32.png",
               "smallIconUrl": "/img/icon/profile16.png"
            },
            "mySubscription": null,
            "name": "Desmond Catillo",
            "photo": {
               "fullEmailPhotoUrl":
"https://instance_name/ncsphoto/SqRfG_-PUTY6KtfOp_f0cEw-MQI3xmkJ21oNEOQ-5sZVyX4qlsJTXfOWM77RW7yt",

               "largePhotoUrl":
"https://instance_name/profilephoto/729D00000000XA5/F",
               "photoVersionId": "729D00000000XA5",
               "smallPhotoUrl":
"https://instance_name/profilephoto/729D00000000XA5/T",
```

```
                    "standardEmailPhotoUrl":
"https://instance_name/ncsphoto/SqRfG_-PUTY6KtfOp_f0cEw-MQI3xmkJ21oNEOQ-5sbktFTIf9vWhFTTALJiaFz_",

                    "url": "/services/data/v34.0/chatter/users/005D0000001LLO1IAO/photo"
                },
                "title": "Senior Manager",
                "type": "User",
                "url": "/services/data/v34.0/chatter/users/005D0000001LJvOIAW",
                "userType": "Internal"
            }
         ],
         "sender": {
            "companyName": "SeattleApps",
            "firstName": "Rene",
            "id": "005D0000001LFNnIAO",
            "isActive": true,
            "lastName": "Holmes",
            "motif": {
               "color": "1797C0",
               "largeIconUrl": "/img/icon/profile64.png",
               "mediumIconUrl": "/img/icon/profile32.png",
               "smallIconUrl": "/img/icon/profile16.png"
            },
            "mySubscription": null,
            "name": "Rene Holmes",
            "photo": {
               "fullEmailPhotoUrl":
"https://instance_name/ncsphoto/SqRfG_-PUTY6KtfOp_f0cEw-MQI3xmkJ21oNEOQ-5sZVyX4qlsJTXfOWM77RW7yt",

              "largePhotoUrl": "https://instance_name/profilephoto/729D00000000XA5/F",

                "photoVersionId": "729D00000000XA5",
              "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000XA5/T",

                "standardEmailPhotoUrl":
"https://instance_name/ncsphoto/SqRfG_-PUTY6KtfOp_f0cEw-MQI3xmkJ21oNEOQ-5sbktFTIf9vWhFTTALJiaFz_",

                "url": "/services/data/v34.0/chatter/users/005D0000001LLO6IAO/photo"
            },
            "title": "Senior Manager",
            "type": "User",
            "url": "/services/data/v34.0/chatter/users/005D0000001LFNnIAO",
            "userType": "Internal"
         },
         "sentDate": "2012-07-18T19:01:51.000Z",
         "url": "/services/data/v34.0/chatter/users/me/messages/03JD000000003VAMAY"
      },
   "nextPageUrl": null
}
```


SEE ALSO:

Send a Private Message

User Messages, Specific

Returns the specified Chatter private message.

Resource

```
/chatter/users/userId/messages/messageId
```

```
/connect/communities/communityId/chatter/users/userId/messages/messageId
```

Note: `userId` must be the ID of the context user or the alias `me`.

Available since version

23.0

Requires Chatter

Yes

HTTP methods

GET or HEAD

Response body

Message

User Photo

The user photo. Get, post, and crop the photo.

To use an image from the Files page as a user photo, pass the file ID in the `fileId` property of the request body or in the `fileId` request parameter. Images uploaded on the User page don't have a file ID and can't be used as the `fileId`.

To upload a binary file as the user photo, you must send it in a multipart/form-data message. For information about how to create the multipart/form-data message, see Uploading Binary Files.

To display user profile photos in a feed, cache the user photos. Then use the `photoVersionId` property of the Photo response body to determine when you need to update a photo. This technique helps you avoid running over limits and may improve mobile client performance.

Resource

```
/chatter/users/userId/photo
```

```
/connect/communities/communityId/chatter/users/userId/photo
```

Available since version

23.0

Requires Chatter

Yes

HTTP methods

GET, POST or HEAD

Request body for POST

Root XML tag

`<photo>`

JSON example

```
{   "cropSize" : "120",
    "cropX" : "0",
    "cropY" : "0",
    "fileId" : "069D00000001IOh"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`cropSize`	Integer	The length, in pixels, of any edge of the crop square.	Required to crop an uploaded file or an existing file.	29.0
`cropX`	Integer	The position X, in pixels, from the left edge of the image to the start of the crop square. Top left is position (0,0).	Required to crop an uploaded file or an existing file.	29.0
`cropY`	Integer	The position Y, in pixels, from the top edge of the image to the start of the crop square. Top left is position (0,0).	Required to crop an uploaded file or an existing file.	29.0
`fileId`	String	18–character ID of an existing file. The key prefix must be 069 and the file size must be less than 2 MB. Note: Images uploaded on the Group page and on the User page don't have file IDs and therefore can't be used.	Required to choose an existing file.	25.0
`versionNumber`	Integer	Version number of an existing file. If not provided, the latest version is used.	Optional	25.0

Request parameters for POST

Parameter Name	Type	Description	Required or Optional	Available Version
`cropSize`	Integer	The length, in pixels, of any edge of the crop square.	Required to crop an uploaded file or an existing file.	29.0
`cropX`	Integer	The position X, in pixels, from the left edge of the image to the start of the crop square. Top left is position (0,0).	Required to crop an uploaded file or an existing file.	29.0

Parameter Name	Type	Description	Required or Optional	Available Version
`cropY`	Integer	The position Y, in pixels, from the top edge of the image to the start of the crop square. Top left is position (0,0).	Required to crop an uploaded file or an existing file.	29.0
`fileId`	String	18 character ID of an existing file. The key prefix must be 069 and the file size must be less than 2 MB. Note: Files uploaded on the Group page and on the User Profile page don't have a file ID and can't be posted or cropped. Files uploaded on the Files page have a file ID and can be posted and cropped.	Required to choose an existing file.	25.0
`versionNumber`	Integer	Version number of the existing content. If not provided, the latest version is used.	Optional	25.0

Note: Photos are processed asynchronously and may not be visible right away.

Response body for GET or HEAD

Photo

Response body for POST

Photo

SEE ALSO:

Update and Crop a User Photo

User Recommendations, General

Returns the user, group, file, record, and custom recommendations for the context user.

Resource

```
/chatter/users/userId/recommendations
```

```
/connect/communities/communityId/chatter/users/userId/recommendations
```

Note: *userId* must be the ID of the context user or the alias `me`.

Available since version

24.0

Requires Chatter

Yes

HTTP methods

GET or HEAD

Request parameters for GET

Parameter Name	Type	Description	Available Version
`contextAction`	String	Action that the user just performed. Supported values are: • `follow` • `view` Use `contextAction` and `contextObjectId` together to get new recommendations based on the action just performed.	33.0
`contextObjectId`	String	ID of the object that the user just perfomed an action on. • If `contextAction` is `follow`, `contextObjectId` is user ID, file ID, or record ID. • If `contextAction` is `view`, `contextObjectId` is user ID, file ID, group ID, or record ID. Use `contextAction` and `contextObjectId` together to get new recommendations based on the action just performed.	33.0

Parameter Name	Type	Description	Available Version
`followed`	String	ID of a user. Provides new recommendations based on the context of this user ID. Important: Use `contextAction` and `contextObjectId` in version 33.0 and later.	23.0–32.0
`maxResults`	Integer	Maximum number of recommendation results; default is 10. Value must be greater than 0.	23.0
`viewed`	String	ID of a file. Provides new recommendations based on the context of this file ID. Important: Use `contextAction` and `contextObjectId` in version 33.0 and later.	23.0–33.0

Response body

Recommendations

User Recommendations for a Specific Action

Returns the recommendations for the specified action for the context user.

Resource

```
/chatter/users/userId/recommendations/action
```

```
/connect/communities/communityId/chatter/users/userId/recommendations/action
```

Note: *`userId`* must be the ID of the context user or the alias `me`.

Valid values for *`action`* and resulting recommendations:

- `follow` returns recommendations to follow users, files, and records.
- `join` returns recommendations to join groups.
- `view` returns recommendations to view users, files, groups, and records (version 25.0 and later). In version 34.0 and later, `view` also returns custom recommendations.

Available since version

24.0

Requires Chatter
Yes

HTTP methods
GET or HEAD

Request parameters for GET

Parameter Name	Type	Description	Available Version
`contextAction`	String	Action that the user just performed. Supported values are: • `follow` • `view` Use `contextAction` and `contextObjectId` together to get new recommendations based on the action just performed.	33.0
`contextObjectId`	String	ID of the object that the user just perfomed an action on. • If `contextAction` is `follow`, `contextObjectId` is user ID, file ID, or record ID. • If `contextAction` is `view`, `contextObjectId` is user ID, file ID, group ID, or record ID. Use `contextAction` and `contextObjectId` together to get new recommendations based on the action just performed.	33.0
`followed`	String	ID of a user. Provides new recommendations based on the context of this user ID. **Important:** Use `contextAction` and `contextObjectId` in version 33.0 and later.	23.0–32.0
`maxResults`	Integer	Maximum number of recommendation results;	23.0

Parameter Name	Type	Description	Available Version
		default is 10. Value must be greater than 0.	
`viewed`	String	ID of a file. Provides new recommendations based on the context of this file ID. **Important:** Use `contextAction` and `contextObjectId` in version 33.0 and later.	23.0–33.0

Response body

Recommendations

User Recommendations for a Specific Action and Object Category

Returns the recommendations for the specified action and object category for the context user.

Resource

```
/chatter/users/userId/recommendations/action/objectCategory
```

```
/connect/communities/communityId/chatter/users/userId/recommendations/action/objectCategory
```

Note: *`userId`* must be the ID of the context user or the alias `me`.

- Valid values for *`action`*:
  - `follow`
  - `join`
  - `view` (version 25.0 and later)
- Valid values for *`objectCategory`*:
  - If *`action`* is `follow`, *`objectCategory`* is `users`, `files`, or `records`.
  - If *`action`* is `join`, *`objectCategory`* is `groups`.
  - If *`action`* is `view`, *`objectCategory`* is `users`, `files`, `groups`, `records`, or `custom` (version 34.0 and later).

Available since version

23.0

Requires Chatter

Yes

HTTP methods

GET, HEAD

Request parameters for GET

Parameter Name	Type	Description	Available Version
`contextAction`	String	Action that the user just performed. Supported values are: • `follow` • `view` Use `contextAction` and `contextObjectId` together to get new recommendations based on the action just performed.	33.0
`contextObjectId`	String	ID of the object that the user just perfomed an action on. • If `contextAction` is `follow`, `contextObjectId` is user ID, file ID, or record ID. • If `contextAction` is `view`, `contextObjectId` is user ID, file ID, group ID, or record ID. Use `contextAction` and `contextObjectId` together to get new recommendations based on the action just performed.	33.0
`followed`	String	ID of a user. Provides new recommendations based on the context of this user ID. **Important:** Use `contextAction` and `contextObjectId` in version 33.0 and later.	23.0–32.0
`maxResults`	Integer	Maximum number of recommendation results; default is 10. Value must be greater than 0.	23.0

Parameter Name	Type	Description	Available Version
`viewed`	String	ID of a file. Provides new recommendations based on the context of this file ID. **Important:** Use `contextAction` and `contextObjectId` in version 33.0 and later.	23.0–33.0

Response body for GET

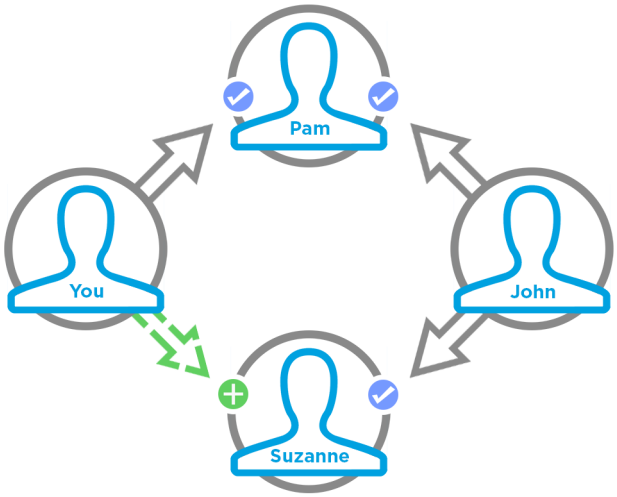
Recommendations

Example request URL to get new recommendations based on an action just performed

If you just followed Pam, who has a user ID of 005D0000001GLowIAB, use:

```
/chatter/users/me/recommendations/follow/users?contextAction=follow&contextObjectId=005D0000001GLowIAB
```

to get only recommendations of users who are followed by the same people who follow Pam. In this example, John follows Pam so you get a recommendation to follow Suzanne since John also follows Suzanne.



Example request URL to get a list of files viewed by the same people who viewed the file you're viewing

This URL provides the same list that displays in the People Also Viewed area on the file detail page for the given file ID.

```
/chatter/users/me/recommendations/view/files?contextAction=view&contextObjectId=069D00000000xNXIAY
```

SEE ALSO:

Get Recommendations of Users to Follow

User Recommendations for a Specific Action and Key Prefix

Returns the recommendations for the specified action and key prefix for the context user.

Resource

```
/chatter/users/userId/recommendations/action/idPrefix
```

```
/connect/communities/communityId/chatter/users/userId/recommendations/action/idPrefix
```

Note: *userId* must be the ID of the context user or the alias `me`.

- Valid values for *action*:
  - `follow`
  - `join`
  - `view`
- The key prefix, or *idPrefix* in the resource URL, is the first three characters of the object ID and indicates the object type. Valid values for *idPrefix* in this URL are:
  - If *action* is `follow`, *idPrefix* is the key prefix for users (005), files (069), or records (such as 001 or 003).
  - If *action* is `join`, *idPrefix* is the key prefix for groups (0F9).
  - If *action* is `view`, *idPrefix* is the key prefix for users (005), files (069), groups (0F9), records (such as 001 or 003), or custom recommendations (0RD) in version 34.0 and later.

Available since version

26.0

Requires Chatter

Yes

HTTP methods

GET or HEAD

Request parameters for GET

Parameter Name	Type	Description	Available Version
`contextAction`	String	Action that the user just performed. Supported values are: • `follow` • `view` Use `contextAction` and `contextObjectId` together to get new recommendations based on the action just performed.	33.0

Parameter Name	Type	Description	Available Version
`contextObjectId`	String	ID of the object that the user just perfomed an action on. • If `contextAction` is `follow`, `contextObjectId` is user ID, file ID, or record ID. • If `contextAction` is `view`, `contextObjectId` is user ID, file ID, group ID, or record ID. Use `contextAction` and `contextObjectId` together to get new recommendations based on the action just performed.	33.0
`followed`	String	ID of a user. Provides new recommendations based on the context of this user ID. **Important:** Use `contextAction` and `contextObjectId` in version 33.0 and later.	23.0–32.0
`maxResults`	Integer	Maximum number of recommendation results; default is 10. Value must be greater than 0.	23.0
`viewed`	String	ID of a file. Provides new recommendations based on the context of this file ID. **Important:** Use `contextAction` and `contextObjectId` in version 33.0 and later.	23.0–33.0

Example request URL

```
/chatter/users/me/recommendations/view/001
```

Response body for GET

Recommendations

User Recommendations for a Specific Action and Object ID

Returns the recommendation for the specified action and object ID for the context user. Also used to delete a recommendation.

Resource

```
/chatter/users/userId/recommendations/action/objectId
```

```
/connect/communities/communityId/chatter/users/userId/recommendations/action/objectId
```

Note: *userId* must be the ID of the context user or the alias `me`.

- Valid values for *action*:
  - `follow`
  - `join`
  - `view` (version 25.0 and later)
- Valid values for *objectId*:
  - If *action* is `follow`, *objectId* is a user ID, file ID, or record ID.
  - If *action* is `join`, *objectId* is a group ID.
  - If *action* is `view`, *objectId* is a user ID, file ID, group ID, record ID, or custom recommendation ID (version 34.0 and later).

Available since version

24.0

Requires Chatter

Yes

HTTP methods

GET, DELETE, or HEAD

DELETE is used to remove a recommendation and is only valid for the following resources:

- `/chatter/users/userId/recommendations/follow/recommendedUserId`
- `/chatter/users/userId/recommendations/follow/recommendedFileId`
- `/chatter/users/userId/recommendations/follow/recommendedRecordId`
- `/chatter/users/userId/recommendations/join/recommendedGroupId`
- `/chatter/users/userId/recommendations/view/customRecommendationId`

Note: *userId* must be the ID of the context user or the alias `me`.

Response body for GET

Recommendations

Response body for DELETE

204: Successful Delete

User Recommendations for a Specific Action and Object Type

Returns the static recommendation for the specified action and object type for the context user. Also used to delete a static recommendation.

Resource

```
/chatter/users/userId/recommendations/action/objectEnum
```

```
/connect/communities/communityId/chatter/users/userId/recommendations/action/objectEnum
```

Note: *userId* must be the ID of the context user or the alias `me`.

- Valid values for *action*:
  - `view`
- Valid values for *objectEnum*:
  - `Today`—Static recommendations that don't have an ID, for example, the Today app recommendation.

Available version
34.0

Requires Chatter
Yes

HTTP methods
GET, DELETE, HEAD

Response body for GET
Recommendations

Response body for DELETE
204: Successful Delete

User Reputation

Returns the specified community user's reputation.

Resource

```
/connect/communities/communityId/chatter/users/userId/reputation
```

Available version
32.0

Requires Chatter
Yes

HTTP methods
GET, HEAD

Response body
Reputation

User Settings

Information about the context user's global Chatter settings.

Resource

```
/chatter/users/userId/settings
```

```
/connect/communities/communityId/chatter/users/userId/settings
```

Note: The *userId* must be the ID of the context user or the alias `me`.

Available since version

27.0

Requires Chatter

Yes

HTTP methods

GET, HEAD, or PATCH

Request body for PATCH

Root XML tag

```
<userChatterSettings>
```

Properties

Name	Type	Description	Available Version
`defaultGroup EmailFrequency`	String	The default frequency with which a user receives email from a group when they join. Valid values are: • `EachPost` • `DailyDigest` • `WeeklyDigest` • `Never` In Communities, the Email on Each Post option is disabled once a group exceeds 10,000 members. All members who had this option selected are automatically switched to Daily Digest.	27.0

Request parameter for PATCH

Parameter Name	Type	Description	Available Since
`defaultGroup EmailFrequency`	String	The default frequency with which a user receives email from a group when they join. This setting does not affect groups a user is already a member of. Valid values are: • `EachPost`	27.0

Parameter Name	Type	Description	Available Since
		• `DailyDigest` • `WeeklyDigest` • `Never`	

Response body for GET, HEAD, or PATCH

User Chatter Settings

Response body example for GET

```
{"defaultGroupEmailFrequency": "Never"}
```

User Status

Note: This resource is no longer available as of version 25.0. Instead, use any feed resource, with the feed-item, such as `/chatter/feeds/news/me/feed-items`.

Returns the most recent user status feed item. Also used to update or delete user status. See Using POST, PATCH, or PUT for Input.

Resource

```
/chatter/users/me/status
```

or

```
/chatter/users/userId/status
```

Available in versions

23.0–24.0

Requires Chatter

Yes

HTTP methods

GET, DELETE, POST or HEAD

Request body for POST

Root XML tag

`<userStatus>`

Properties

Name	Type	Description	Available Version
`body`	Message Body Input	Description of message body.	23.0–24.0

Request parameter for POST

Name	Type	Description
`text`	String *`Text to post`*	Text of the item to be posted to the user's status

Response body for GET or POST
User Status

Topics Recently Used

Topics recently used by the specified user. Get up to five topics most recently used by the specified user.

Resource

```
/chatter/users/userId/topics
```

```
/connect/communities/communityId/chatter/users/userId/topics
```

Available since version
28.0

Requires Chatter
Yes

HTTP methods
GET or HEAD

Response body
Topic Collection

Zones Resources

Information about Chatter Answers zones.

Available resources:

Resource	Description
`/connect/zones`	A list of Chatter Answers zones in the organization. Get information about zones available to the context user.
`/connect/zones/zoneId`	Information about the specified Chatter Answers zone.
`/connect/zones/zoneId/searchResults`	Search results from a search in a Chatter Answers zone.

List of Zones

A list of Chatter Answers zones in the organization. Get information about zones available to the context user.

Resource

```
/connect/zones/
```

```
/connect/communities/communityId/zones/
```

Available version
29.0

Requires Chatter
No

HTTP methods
GET

Request parameters for GET

Parameter Name	Type	Description	Available Version
`page`	Integer	An integer specifying a page of items. The default value is 0, which returns the first page.	29.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are between 1 and 250. The default value is 25.	29.0

Response body for GET
Zone Page

Zone Information

Information about the specified Chatter Answers zone.

Resource

```
/connect/zones/zoneId
```

```
/connect/communities/communityId/zones/zoneId
```

Available version
29.0

Requires Chatter
No

HTTP methods
GET

Response body for GET
Zone

Zone Search Results

Search results from a search in a Chatter Answers zone.

Resource

```
/connect/zones/zoneId/search-results
```

```
/connect/communities/communityId/zones/zoneId/search-results
```

Available version
29.0

Requires Chatter
No

HTTP methods

GET

Request parameters for GET

Parameter Name	Type	Description	Available Version
`page`	Integer	An integer specifying a page of items. The default value is 0, which returns the first page.	29.0
`pageSize`	Integer	Specifies the number of items per page. Valid values are between 1 and 100. The default value is 20.	29.0

Response body for GET

Zone Search Result

CHAPTER 6 Chatter REST API Request Bodies

To perform a POST, PATCH, or PUT request, pass query parameters or create a request body formatted in either XML or JSON. This chapter lists the request bodies. The query parameters are listed with each resource.

To create a JSON request body, specify the properties of the request body in JSON format.

This is an example of a Comment request body.

```
{
 "body" : "Let's look for a new solution."
}
```

If a request body is top-level, it has a root XML tag listed. To create an XML request body, nest the properties as XML tags inside the root XML tag.

This is the same Connect request body in XML format:

```
<comment>
 <body>Let's look for a new solution.</body>
</comment>
```

Action Link Definition Input

The definition of an action link. An action link is a button on a feed element. Clicking an action link can take a user to a Web page, initiate a file download, or invoke an API call to Salesforce or to an external server. An action link includes a URL and an HTTP method, and can include a request body and header information, such as an OAuth token for authentication. Use action links to integrate Salesforce and third-party services into the feed so that users can take action to drive productivity and accelerate innovation.

Usage

You can use context variables in the `actionUrl`, `headers`, and `requestBody` properties. Use context variables to pass information about the user who executed the action link to your server-side code. Salesforce substitutes the value when the action link is executed.

These are the available context variables:

Context Variable	Description
`{!actionLinkId}`	The ID of the action link the user executed.
`{!actionLinkGroupId}`	The ID of the action link group containing the action link the user executed.
`{!communityId}`	The ID of the community in which the user executed the action link. The value for your internal organization is the empty key `"000000000000000000"`.
`{!communityUrl}`	The URL of the community in which the user executed the action link. The value for your internal organization is empty string `""`.

Context Variable	Description
`{!orgId}`	The ID of the organization in which the user executed the action link.
`{!userId}`	The ID of the user that executed the action link.

Root XML tag

```
<actionLinkDefinition>
```

JSON example

This example doesn't use a template:

```
{
   "actionType":"Api",
   "actionUrl":"/services/data/v34.0/chatter/feed-elements",
   "groupDefault":"true",
   "labelKey":"Post",
   "method":"Post",
   "requiresConfirmation":"true",
   "requestBody":"{\"subjectId\": \"me\",\"feedElementType\": \"FeedItem\",\"body\":
{\"messageSegments\": [{\"type\": \"Text\",\"text\": \"Testing a post created via an
API action link.\"}]}}",
   "headers":[
   {
      "name":"Content-Type", "value":"application/json"
   }]
}
```

This example uses a template:

```
{
   "templateId":"07gD00000004C9r",
   "templateBindings":[
      {
         "key":"Bindings.version",
         "value":"v33.0"
      }
   ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`actionType`	String	Defines the type of action link. Values are: • `Api`—The action link calls a synchronous API at the action URL. Salesforce sets the status to `SuccessfulStatus` or	Required Can be defined in an action link template.	33.0

Name	Type	Description	Required or Optional	Available Version
		`FailedStatus` based on the HTTP status code returned by your server. • `ApiAsync`—The action link calls an asynchronous API at the action URL. The action remains in a `PendingStatus` state until a third party makes a request to `/connect/action-links/`***`actionLinkId`*** to set the status to `SuccessfulStatus` or `FailedStatus` when the asynchronous operation is complete. • `Download`—The action link downloads a file from the action URL. • `Ui`—The action link takes the user to a web page at the action URL. Use `Ui` if you need to load a page before the user performs an action, for example, to have the user provide input or view something before the action happens.		
`actionUrl`	String	The action link URL. For example, a `Ui` action link URL is a Web page. A `Download` action link URL is a link to the file to download. `Ui` and `Download` action link URLs are provided to clients. An `Api` or `ApiAsync` action link URL is a REST resource. `Api` and `ApiAsync` action link URLs aren't provided to clients. Links to Salesforce can be relative. All other links must be absolute and start with `https://`. Tip: To avoid issues due to upgrades or changing functionality in your API, we recommend using a versioned API for `actionUrl`, for example, `https://www.example.com/api/v1/exampleResource`. If your API isn't versioned, you can use the `expirationDate` property of the Action Link Group Definition Input to avoid issues due to upgrades or changing functionality in your API.	Required Can be defined in an action link template.	33.0

Name	Type	Description	Required or Optional	Available Version
`excludedUserId`	String	ID of a single user to exclude from performing the action. If you specify an `excludedUserId`, you can't specify a `userId`.	Optional Can be defined in an action link template using the `User Visibility` and `Custom User Alias` fields.	33.0
`groupDefault`	Boolean	`true` if this action is the default action link in the action link group; `false` otherwise. There can be only one default action link per action link group. The default action link gets distinct styling in the Salesforce UI.	Optional Can be defined in an action link template.	33.0
`headers`	Request Header Input[]	The request headers for the `Api` and `ApiAsync` action link types. See Authentication.	Optional Can be defined in an action link template.	33.0
`labelKey`	String	Key for the set of labels to show in the user interface. A set includes labels for these states: NewStatus, PendingStatus, SuccessStatus, FailedStatus. For example, if you use the `Approve` key, you get these labels: Approve, Pending, Approved, Failed. For a complete list of keys and labels, see Action Links Labels. If none of the predefined labels work for your action link, use a custom label. To use a custom label, create an action link template. See Create Action Link Templates.	Required Can be defined in an action link template.	33.0
`method`	String	One of these HTTP methods: • `HttpDelete`—Returns HTTP 204 on success. Response body or output class is empty. • `HttpGet`—Returns HTTP 200 on success.	Required Can be defined in an action link template.	33.0

Name	Type	Description	Required or Optional	Available Version
		• `HttpHead`—Returns HTTP 200 on success. Response body or output class is empty. • `HttpPatch`—Returns HTTP 200 on success or HTTP 204 if the response body or output class is empty. • `HttpPost`—Returns HTTP 201 on success or HTTP 204 if the response body or output class is empty. Exceptions are the batch posting resources and methods, which return HTTP 200 on success. • `HttpPut`—Return HTTP 200 on success or HTTP 204 if the response body or output class is empty.		
`requestBody`	String	The request body for `Api` and `ApiAsync` action link types. Note: Escape quotation mark characters in the `requestBody` value.	Optional Can be defined in an action link template.	33.0
`requires Confirmation`	Boolean	`true` to require the user to confirm the action; `false` otherwise.	Required Can be defined in an action link template.	33.0
`userId`	String	The ID of the user who can execute the action. If not specified or `null`, any user can execute the action. If you specify a `userId`, you can't specify an `excludedUserId`.	Optional Can be defined in an action link template using the `User Visibility` and `Custom User Alias` fields.	33.0

SEE ALSO:

Define Action Links and Post with a Feed Element

Action Link Group Definition Input

The definition of an action link group. All action links must belong to a group. Action links in a group are mutually exclusive and share some properties. Define stand-alone actions in their own action group.

Information in the action link definition can be sensitive to a third party (for example, OAuth bearer token headers). For this reason, only requests made via the connected app that created the action link definition can read, modify, or delete the definition. In addition, the user making the request must have created the definition or have "View All Data" permission.

Root XML tag

```
<actionLinkGroup>
```

JSON example

This example instantiates an action link group definition without a template:

```
{
  "actionLinks": [
  {
    "actionType": "Api",
    "labelKey": "Like",
    "groupDefault": "true",
    "actionUrl": "https://test.com/this",
    "method": "HttpPost",
    "requiresConfirmation": "false"
  },
  {
    "actionType": "Api",
    "labelKey": "Unlike",
    "groupDefault": "false",
    "actionUrl": "https://test.com/that",
    "method": "HttpPost",
    "requiresConfirmation": "false"
  }
  ],
  "executionsAllowed": "OncePerUser",
  "expirationDate": "2014-07-07T23:59:11.168Z",
  "category": "Primary"
}
```

This example instantiates an action link group definition from a template:

```
{
   "templateId":"07gD00000004C9r",
   "templateBindings":[
      {
         "key":"Bindings.version",
         "value":"v33.0"
      }
   ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`actionLinks`	Action Link Definition Input[]	The action links in the group. Within an action link group, action links are displayed in the order listed in the `actionLinks` property of the Action Link Group Definition Input request body. Within a feed item, action link groups are displayed in the order specified in the `actionLinkGroupIds` property of the Associated Actions Capability Input request body.	Required to instantiate this action link group without a template. To instantiate from a template, don't specify a value.	33.0
`category`	String	Indicates the priority and relative locations of action links in an associated feed item. Values are: • `Primary`—The action link group is displayed in the body of the feed element. • `Overflow`—The action link group is displayed in the overflow menu of the feed element. You can create up to three action links in a `Primary` group and up to four in an `Overflow` group.	Required to instantiate this action link group without a template. To instantiate from a template, don't specify a value.	33.0
`executionsAllowed`	String	Defines the number of times an action link can be executed. Values are: • `Once`—An action link can be executed only once across all users. • `OncePerUser`—An action link can be executed only once for each user. • `Unlimited`—An action link can be executed an unlimited number of times by each user. If the action link's `actionType` is `Api` or `ApiAsync`, you can't use this value.	Required to instantiate this action link group without a template. To instantiate from a template, don't specify a value.	33.0
`expirationDate`	Date	ISO 8601 date string, for example, 2011-02-25T18:24:31.000Z, that	Required to instantiate this action link group without a template.	33.0

Name	Type	Description	Required or Optional	Available Version
		represents the date and time this action link group is removed from associated feed items and can no longer be executed. The `expirationDate` must be within one year of the creation date. If the action link group definition includes an OAuth token, it is a good idea to set the expiration date of the action link group to the same value as the expiration date of the OAuth token so that users can't execute the action link and get an OAuth error. To set a date when instantiating from a template, see Set the Action Link Group Expiration Time.	Optional to instantiate from a template.	
`templateBindings`	Action Link Template Binding Input[]	A collection of key-value pairs to fill in binding variable values or a custom user alias from an action link template. To instantiate this action link group from an action link template that uses binding variables, you must provide values for all the variables. See Define Binding Variables.	To instantiate without a template, don't specify a value. Required to instantiate this action link group from a template that uses binding variables.	33.0
`templateId`	String	The ID of the action link group template from which to instantiate this action link group.	To instantiate without a template, don't specify a value. Required to instantiate this action link group from a template.	33.0

SEE ALSO:

Define Action Links and Post with a Feed Element

Define Action Links and Post with a Feed Element

Define Action Links in a Template and Post with a Feed Element

Action Link Input

The desired execution status of an action link.

Root XML tag

```
<actionLink>
```

JSON example

```
{  "status" : "SuccessfulStatus"  }
```

Properties

Name	Type	Description	Required or Optional	Available Version
`status`	String	The current status of the action. Values are: • `FailedStatus`—The action link execution failed. • `NewStatus`—The action link is ready to be executed. Available for `Download` and `Ui` action links only. • `PendingStatus`—The action link is executing. Choosing this value triggers the API call for `Api` and `ApiAsync` action links. • `SuccessfulStatus`—The action link executed successfully.	Required	33.0

SEE ALSO:

Update the Status of an Action Link or Invoke an Action Link

Action Link Template Binding Input

A key-value pair to fill in a binding variable value from an action link template.

JSON example

```
{
   "key":"version",
   "value":"33.0"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`key`	String	The name of the binding variable key specified in the action link template in	Required	33.0

Name	Type	Description	Required or Optional	Available Version
		Setup. For example, if the binding variable in the template is `{!Binding.firstName}`, the key is `firstName`		
`value`	String	The value of the binding variable key. For example, if the key is `firstName`, this value could be `Joan`.	Required	33.0

Announcement Input

An announcement displays in a designated location in the Salesforce UI until 11:59 p.m. on its expiration date, unless it's deleted or replaced by another announcement.

Root XML tag

```
<announcement>
```

JSON example

```
{   "body":
      {
         "messageSegments": [
         {
            "text": "Please install the updates for all iOS devices.",
            "type": "Text"
          }
          ]
      }
   "expirationDate": "2014-02-22T00:00:00.000Z",
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`body`	Message Body Input	Text of the announcement.	Required	31.0
`expirationDate`	Date	The Salesforce UI displays an announcement until 11:59 p.m. on this date unless another announcement is posted first. The Salesforce UI ignores the time value in the `expirationDate`. However, you can use the time value to create your own display logic in your own UI.	Required	31.0

Associated Actions Capability Input

Describes action links that can be created or updated on a feed element. To associate an action link group with a feed element, the client must be the same connected app that created the action link group definition and the user must either be the user who created the definition or have "View all Data" permission.

An action link is a button on a feed element. Clicking an action link can take a user to a Web page, initiate a file download, or invoke an API call to Salesforce or to an external server. An action link includes a URL and an HTTP method, and can include a request body and header information, such as an OAuth token for authentication. Use action links to integrate Salesforce and third-party services into the feed so that users can take action to drive productivity and accelerate innovation.

JSON example

```
"associatedActions": {
    "actionLinkGroupIds":["0Agxx00000000d1CAA", "0Agxx00000000d2DFY"]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`actionLinkGroupIds`	String[]	The action link group IDs to associate with the feed element. Associate one `Primary` and up to 10 total action link groups to a feed item. Action link groups are returned in the order specified in this property. Get an action link group ID in the response from a request to /connect/action-link-group-definitions.	Required	33.0

Attachment Input: Canvas App

Important: This request body isn't available in version 32.0 and later. In version 32.0 and later, use Canvas Capability Input.

XML

In XML request bodies, `attachmentType` is an attribute of the `<attachment>` element.

```
<attachment attachmentType="Canvas">
   <developerName>TravelApp</developerName>
</attachment>
```

JSON

```
{"attachment" :
  {
    "attachmentType" : "Canvas",
    "developerName" : "TravelApp",
    "namespacePrefix" : "dev_org",
    "parameters" : "{&quot;itinerary&quot;:&quot;123&quot;}",
```

```
    "height" : "250px",
    "title" : "TravelItinerary",
  }
}
```

Properties

Name	Type	Description	Available Version
`attachmentType`	String	The attachment type. Must contain the value `Canvas`.	29.0–31.0
`description`	String	Optional. The description of the canvas app.	29.0–31.0
`developerName`	String	The developer name (API name) of the canvas app	29.0–31.0
`height`	String	Optional. The height of the canvas app in pixels. Default height is 200 pixels.	29.0–31.0
`namespacePrefix`	String	Optional. The namespace prefix of the Developer Edition organization in which the canvas app was created.	29.0–31.0
`parameters`	String	Optional. Parameters passed to the canvas app in JSON format. The JSON must be escaped.	29.0–31.0
`thumbnailUrl`	String	Optional. The URL to a thumbnail image for the canvas app. Maximum dimensions are 120x120 pixels.	29.0–31.0
`title`	String	Optional. The title of the link used to call the canvas app.	29.0–31.0

Attachment Input: Existing Content

Important: This request body isn't available in version 32.0 and later. In version 32.0 and later, use Content Capability Input.

XML

In XML request bodies, `attachmentType` is an attribute of the `<attachment>` element.

```
<attachment attachmentType="ExistingContent">
   <contentDocumentId>069...</contentDocumentId>
</attachment>
```

JSON

```
{ "attachment":
   {
      "attachmentType":"ExistingContent",
      "contentDocumentId": "069D00000001FHF"
   }
}
```

Properties

Name	Type	Description	Available Version
`attachmentType`	String	The attachment type. Must contain the value `ExistingContent`.	27.0–31.0
`contentDocumentId`	String	ID of the existing content	23.0–31.0

Attachment Input: Feed Poll

Important: This request body isn't available in version 32.0 and later. In version 32.0 and later, use Poll Capability Input.

XML

Note that in XML, the choices for the poll are each wrapped in a `<choice>` element and `attachmentType` is an attribute of the `<attachment>` element. For example:

```
<attachment attachmentType="Poll">
   <pollChoices>
      <choice>One</choice>
      <choice>Two</choice>
   </pollChoices>
</attachment>
```

JSON

```
{ "attachment":
   {
      "attachmentType":"Poll",
      "pollChoices": [
         "Monday",
         "Tuesday"
      ]
   }
}
```

Properties

Name	Type	Description	Available Version
`attachmentType`	String	The attachment type. Must contain the value `Poll`.	27.0–31.0
`pollChoices`	String []	A list of choices used in the feed poll.	26.0–31.0

Attachment Input: Link

Important: This request body isn't available in version 32.0 and later. In version 32.0 and later, use Link Capability Input.

XML

The `attachmentType` property is an attribute of the `<attachment>` element. For example:

```
<attachment attachmentType="Link">
   <url>http://www.google.com</url>
   <urlName>A search engine</urlName>
</attachment>
```

JSON

```
{ "attachment":
   {
      "attachmentType":"Link",
      "url": "http://www.salesforce.com",
      "urlName": "Salesforce"
   }
}
```

Properties

Name	Type	Description	Available Version
`attachmentType`	String	The attachment type. Must contain the value `Link`.	27.0–31.0
`url`	String	URL included in the attachment.	23.0–31.0
`urlName`	String	Optional. Name of the URL. If not provided, a name is generated from the domain name of the URL.	23.0–31.0

Attachment Input: New File Upload

Important: This request body isn't available in version 32.0 and later. In version 32.0 and later, use Content Capability Input.

To upload a new file, provide the binary file in a multipart/form-data body part where `name="feedItemFileUpload"`. See Uploading Binary Files.

XML

The `attachmentType` property is an attribute of the `<attachment>` element. For example:

```
<attachment attachmentType="NewFile">
   <description>TPS Report (with cover)</description>
       <title>TPS.txt</title>
</attachment>
```

JSON

```
{ "attachment":
  {
     "attachmentType":"NewFile",
     "description":"TPS Report (with cover)",
```

```
    "title":"TPS.txt"
  }
}
```

Properties

Name	Type	Description	Available Version
`attachmentType`	String	The attachment type. Must contain the value `NewFile`.	27.0–31.0
`desc`	String	Description of the file. **Attention:** This parameter should not be used as of version 27.0. Use `description` instead.	23.0–26.0
`description`	String	Optional. Description of the file.	27.0–31.0
`title`	String	Title of the file.	24.0–31.0

Batch Collection Input

Describes a collection of inputs for a batch request.

JSON example

```
{
   "inputs": [
   {
      "richInput": {
         "subjectId": "me",
         "body": {
            "messageSegments": [
            {
               "type": "Text",
               "text": "Post Number 1"
            }]
         },
         "feedElementType": "FeedItem"
      }
   }, {
      "richInput": {
         "subjectId": "me",
         "body": {
            "messageSegments": [
            {
               "type": "Text",
               "text": "Post Number 2"
            }]
         },
         "feedElementType": "FeedItem"
```

```
        }
    }]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`inputs`	Batch Input[]	Collection of inputs (request bodies) to be processed together. Every Batch Input request body's `richInput` property must contain the same type of request body.	Required	32.0

Batch Input

A single batch input to nest in a batch collection input.

JSON example

Example without a binary part:

```
{
   "richInput": {
      "subjectId": "me",
      "body": {
         "messageSegments": [
         {
            "type": "Text",
            "text": "Post Number 1"
         }]
      },
      "feedElementType": "FeedItem"
   }
}
```

Example with a binary part:

```
{
      "binaryPartNames": [
        "bin1"
      ],
      "richInput": {
        "subjectId": "me",
        "body": {
          "messageSegments": [
            {
              "type": "Text",
              "text": "Please accept this receipt"
            }
          ]
        },
        "capabilities": {
```

```
          "content": {
            "description": "Receipt for expenses",
            "title": "receipt.pdf"
          }
        },
        "feedElementType": "FeedItem"
      }
    }
```

In this example, the `name` parameter in the `Content-Disposition` header of the binary body part must have the value `bin1`.

Properties

Name	Type	Description	Required or Optional	Available Version
`binaryPartNames`	String[]	List of the `name` parameters in the `Content-Disposition` header of the binary body parts to include with this input. If this value exists, the corresponding binary part must be included in the request. Note: When used with the `/chatter/feed-elements/batch` resource, this property can contain only one value, not a list of values.	Optional	32.0
`richInput`		An input body (*request body*) for the request. The `richInput` property must contain the request body that the resource expects. The `/chatter/feed-elements/batch` resource expects a Feed Item Input request body.	Required	32.0

Bookmarks Capability Input

Create or update a bookmark on a feed element.

Root XML tag

```
<bookmarksCapability>
   <isBookmarkedByCurrentUser>true</isBookmarkedByCurrentUser>
</bookmarksCapability>
```

JSON example

```
{
   "isBookmarkedByCurrentUser": true
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`isBookmarkedByCurrentUser`	Boolean	Specifies if the feed element should be bookmarked for the user (`true`) or not (`false`).	Required	32.0

Canvas Capability Input

Create or update a canvas app associated with a feed element.

Root XML tag

```
<canvasCapability>
   <developerName>true</developerName>
   <title>true</title>
</canvasCapability>
```

JSON example

```
{
   "developerName": "myNewApp",
   "title": "My New App"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`description`	String	A description of the canvas app. The maximum size is 255 characters.	Optional	32.0
`developerName`	String	The API name (developer name) of the connected app.	Required	32.0
`height`	String	The height of the canvas app in pixels.	Optional	32.0
`namespacePrefix`	String	A unique namespace prefix for the canvas app.	Optional	32.0
`parameters`	String	JSON parameters passed to the canvas app.	Optional	32.0
`thumbnailUrl`	String	A thumbnail URL to a preview image. The maximum thumbnail size is 120 pixels by 120 pixels.	Optional	32.0
`title`	String	A title for the canvas link.	Required	32.0

Comment Input

Root XML tag

`<comment>`

JSON

The following example attaches an existing file to a comment:

```
{
   "body" : {
      "messageSegments" : [
        {
           "text" : "FY14 Sales Projections",
           "type" : "Text"
        }
      ]
   },
   "capabilities": {
      "contentDocumentId":"069D00000001FHF"
   }
}
```

Properties

Name	Type	Description	Available Version
`attachment`	Depends on type.	Optional. Values are: • Attachment Input: Existing Content • Attachment Input: New File Upload **Important:** As of version 32.0, use the `capabilities` property.	24.0-31.0
`body`	Message Body Input	The body of the comment. To edit this property in a comment, make a PATCH request to `/chatter/comments/`***`commentId`***. Editing comments is supported in version 34.0 and later.	23.0

Name	Type	Description	Available Version
`capabilities`	Comment Capabilities Input	Capabilities of the comment, such as a file attachment	32.0

SEE ALSO:

- Post a Comment
- Post a Comment and Include a File from Salesforce
- Post an @Mention
- Post Multiple @Mentions
- Edit a Comment

Comment Capabilities Input

Root XML tag

`<commentCapabilities>`

Properties

Name	Type	Description	Available Version
`content`	Content Capability Input	A file attachment.	32.0

Content Capability Input

Create or update a file associated with a feed element.

Root XML tag

```
<contentCapability>
   <contentDocumentId>069D00000001pyS</contentDocumentId>
</contentCapability>
```

JSON example

```
{
   "contentDocumentId": "069D00000001pyS"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`contentDocumentId`	String	ID of the existing content.	Required for existing content Not supported for new content	32.0
`description`	String	Description of the file to be uploaded.	Optional for new content Not supported for existing content	32.0
`title`	String	Title of the file. This value is used as the file name for new content. For example, if the title is My Title, and the file is a .txt file, the file name is My Title.txt.	Required for new content Not supported for existing content	32.0

Datacloud Order Input

A list of Data.com identification numbers that are used to purchase contacts or companies by a specific `userType`.

Root XML tag

```
<DatacloudOrder>
```

JSON example

```
{
   "companyIds":[
      "23456",
      "24548"
   ],
   "userType":"Monthly"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`companyIds`	String	The list of companies to purchase. If `contactIds` are included with `companyIds`, your purchase fails.	Required	32.0
`contactIds`	String	The list of contacts to purchase. We recommend that you include only contacts in your list. If `companyIds` are included with `contactIds`, your purchase fails.	Required	32.0

Name	Type	Description	Required or Optional	Available Version
userType	String	The Data.com user type. • `Monthy`—A user type that's assigned monthly point limits for purchasing Data.com records. Only the assigned user can use monthly points. Points expire at the end of the month. Monthly is the default setting for `DataCloudUserType`. • `Listpool`— A user type that allows users to draw from a pool of points to purchase Data.com records.	Optional—If a user type isn't specified, Monthly is the default.	32.0

Dashboard Component Snapshot Input

Root XML tag

```
<dashboardComponentSnapshot>
```

JSON

```
{
   "parentId" : "0F9D00000000I4O",
   "feedItemBody" : {
      "messageSegments" : [
      {
         "text" : "Sales Heat",
         "type" : "Text"
      }
      ]
   }
}
```

Properties

Name	Type	Description	Available Version
`parentId`	String	A user, group, or dashboard ID to which you are posting the snapshot.	25.0
`filterId1`	String	Optional. A dashboard filter item ID.	25.0
`filterId2`	String	Optional. A second dashboard filter item ID.	25.0
`filterId3`	String	Optional. A third dashboard filter item ID.	25.0
`feedItemBody`	Message Body Input	Contains the text that accompanies the feed item associated with the dashboard component snapshot.	25.0

Name	Type	Description	Available Version
`runningUserId`	String	Optional. A user ID from which the view of the dashboard will be created, if other than the creator.	25.0

Feed Element Capabilities Input

A container for all capabilities that can be included when creating a new feed element.

In API versions 30.0 and earlier, every feed item can have comments, likes, topics, and so on. In versions 31.0 and later, every feed item (and feed element) can have a unique set of *capabilities*. If a capability property exists on a feed element, that capability is available, even if the capability property doesn't have a value. For example, if the `ChatterLikes` capability property exists on a feed element (with or without a value), the context user can like that feed element. If the capability property doesn't exist, it isn't possible to like that feed element. A capability can also contain associated data. For example, the `Moderation` capability contains data about moderation flags.

XML example

```
<capabilities>
   <associatedActions>
      <actionLinkGroupIds>
         <actionLinkGroupId/>0Agxx00000000d1CAA<actionLinkGroupId/>
         <actionLinkGroupId/>0Agxx00000000d2DFY<actionLinkGroupId/>
      </actionLinkGroupIds>
   </associatedActions>
   <bookmarksCapability>
      <isBookmarkedByCurrentUser>false</isBookmarkedByCurrentUser>
   </bookmarksCapability>
   <contentCapability>
      <contentDocumentId>069D00000001pyS</contentDocumentId>
   </contentCapability>
</capabilities>
```

JSON example

```
"capabilities": {
   "associatedActions": {
      "actionLinkGroupIds":["0Agxx00000000d1CAA", "0Agxx00000000d2DFY"]
   },
   "bookmarks": {
      "isBookmarkedByCurrentUser": false
   },
   "content": {
      "contentDocumentId": "069D00000001pyS"
   }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`associatedActions`	Associated Actions Capability Input	Describes action links that can be created or updated on a feed element. To associate an action link group with a feed element, the client must be the same connected app that created the action link group definition and the user must either be the user who created the definition or have "View all Data" permission.	Optional	31.0
`bookmarks`	Bookmarks Capability Input	Create or update a bookmark on a feed element.	Optional	32.0
`canvas`	Canvas Capability Input	Create or update a canvas app associated with a feed element.	Optional	32.0
`content`	Content Capability Input	Attach or update a file with a feed element. Use this class to attach a new file or update a file that has already been uploaded to Salesforce.	Optional	32.0
`link`	Link Capability Input	Create or update a link attached to a feed element.	Optional	32.0
`poll`	Poll Capability Input	Create, update, or vote on a poll attached to a feed element.	Optional	32.0
`questionAndAnswers`	Question and Answers Capability Input	Create or edit a question feed element or set the best answer of the existing question feed element.	Optional	32.0

SEE ALSO:

Edit a Question Title and Post

Feed Favorite Input

Root XML tag

`<favorite>`

JSON for POST

```
{
   "searchText" : "release",
   "targetId" : "0TOD0000000099d"
}
```

Properties for POST

Name	Type	Description	Available Version
`searchText`	String	Text of the feed search. Use this parameter for feed searches only.	24.0
`targetId`	String	Id of the topic. Use this parameter for topics only.	28.0

JSON for PATCH

```
{
   "updateLastViewDate" : "true"
}
```

Properties for PATCH

Name	Type	Description	Available Version
`updateLastViewDate`	Boolean	Sets the last view date of the specified feed favorite to the current system time. Default value is `false`.	24.0

Feed Item Input

Root XML tag

`<feedItem>`

JSON

The following example attaches a poll to a feed item:

```
{
   "body":{
      "messageSegments":[
         {
            "type":"Text",
            "text":"When should we meet for release planning?"
         }
      ]
   },
   "capabilities":{
      "poll":{
         "choices":[
            "Monday",
            "Tuesday"
         ]
      }
   },
```

```
   "feedElementType":"FeedItem",
   "subjectId":"me"
}
```

Feed items support more than 1 capability. The following example attaches a file to a feed item and bookmarks it.

```
{
   "body":{
      "messageSegments":[
         {
            "type":"Text",
            "text":"Check out the latest marketing collateral."
         }
      ]
   },
   "capabilities":{
      "bookmarks":{
         "isBookmarkedByCurrentUser": true
      },
      "content":{
         "contentDocumentId": "069D00000001uQO"
      }
   },
   "feedElementType":"FeedItem",
   "subjectId":"me"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`attachment`	Depends on type.	You can specify only one of these values in the attachment parameter. Values are: • Attachment Input: Canvas App (v29.0–31.0) • Attachment Input: Existing Content (v23.0–31.0) • Attachment Input: Feed Poll (v26.0–31.0) • Attachment Input: Link (v23.0–31.0) • Attachment Input: New File Upload (v23.0–31.0) **Important:** As of API version 32.0, use the `capabilities` property.	Optional	

Name	Type	Description	Required or Optional	Available Version
`body`	Message Body Input	Description of message body. The body can contain up to 25 mentions. If you specify `originalFeedElementId` to share a feed item, use the `body` property to add the first comment to the feed item. To edit this property in a feed item, make a PATCH request to `/chatter/feed-elements/feedElementId`. Editing feed posts is supported in version 34.0 and later.	Required unless the feed item has a content capability or a link capability	23.0
`capabilities`	Feed Element Capabilities Input	The capabilities that define the behavior of this feed element.	Optional	31.0
`isBookmarkedByCurrentUser`	Boolean	Specifies whether to bookmark the new feed item for the user (`true`) or not (`false`). The default value is `false`. **Important:** As of API version 32.0, use the `capabilities.bookmarks.isBookmarkedByCurrentUser` property.	Optional	24.0–31.0
`feedElementType`	String	The type of feed element this input represents. The only valid value is `FeedItem`.	Required	31.0
`originalFeedElementId`	String	To share a feed element, specify its 18-character ID. **Note:** You can't share a post and upload a file in the same request body.	Optional	31.0
`originalFeedItemId`	String	To share a feed item, specify its 18-character ID. **Important:** As of API version 32.0, use the `originalFeedElementId` property.	Optional	24.0–31.0

Name	Type	Description	Required or Optional	Available Version
`subjectId`	String	The ID of the parent this feed element is being posted to. This value can be the ID of a user, group, or record, or the string `me` to indicate the context user.	Required	31.0

Name	Type	Description	Required or Optional	Available Version
`visibility`	String	Who can see the post. Valid values are: • `InternalUsers`—If Communities is enabled, only members of the default, internal community can see the feed item. • `AllUsers`—If Communities is enabled, all members of the community can see the feed item. Default values: • For external users, the default value is `AllUsers`. External users must use this value to see their posts. • For internal users, the default value is `InternalUsers`. Internal users can accept this value or use the value `AllUsers` to allow external users to see their posts. If the parent of the feed item is a User or a Group, the `visibility` of the feed item must be `AllUsers`.	Optional	26.0

SEE ALSO:

- Post a Feed Item
- Post a Feed Element and Include a File from Salesforce
- Post an @Mention
- Post Multiple @Mentions
- Define Action Links and Post with a Feed Element
- Define Action Links in a Template and Post with a Feed Element
- Edit a Feed Post
- Edit a Question Title and Post
- Share a Feed Item
- Post to a Group

Feed Item Topic Input

Root XML tag

`<feedItemTopic>`

JSON

The following example attaches a new topic to a feed item:

```
{ "topicName": "W3C"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`topicId`	String	ID of an existing topic	Required for existing topics if the `topicName` property isn't used	29.0
`topicName`	String	Name of a new or existing topic, up to 150 characters	Required for new topics; required for existing topics if the `topicId` property isn't used	29.0

Feed Poll Input

Important: This request body isn't available in version 32.0 and later. In version 32.0 and later, use Poll Capability Input.

Root XML tag

`<feedPoll>`

Properties

Name	Type	Description	Available Version
`myChoiceId`	String	ID of an existing choice on the feed poll. Used to specify the ID of the poll choice that the current user is voting on.	26.0–31.0

File Input

Describes a new file input. To upload a file, provide the binary file in a multipart/form-data body part where `name="fileData"`. See Uploading Binary Files.

Root XML tag

`<fileInput>`

JSON

```
{
   "desc":"Employee Survey Results",
   "title":"emp_surv_results"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`contentModifiedDate`	Date	File-specific last modified date that can be used to set a client-side modified date.	Optional	32.0
`desc`	String	Description of the file to be uploaded.	Optional	24.0
`isInMyFileSync`	Boolean	`true` if the file is synced withSalesforce Files Sync; `false` otherwise.	Optional	31.0
`isMajorVersion`	Boolean	`true` if the file is a major version; `false` if the file is a minor version. Major versions can't be replaced.	Optional	31.0
`parentFolderId`	String	ID of the containing folder.	Optional	31.0
`repositoryFileId`	String	ID of the external file.	Optional	32.0
`repositoryId`	String	ID of the external data source.	Optional	32.0
`title`	String	Title of the file.	Optional	24.0

File Shares Input

Root XML tag

`<fileShares>`

JSON example

```
{
  "message": "I shared a document with you.",
  "shares": [
     { "id": "005D0000001Az4l", "sharingType": "V" },
     { "id": "005D0000001B2ny", "sharingType": "C" }
   ]
}
```

Properties

Name	Type	Description	Available Version
`message`	String	Private message body that's sent to the recipients. If Salesforce Communities is enabled, an email is sent instead of a private message.	30.0
`shares`	Share Input[]	Collection of Share Input request bodies	30.0

Folder Input

Root XML tag

`<folderInput>`

JSON example

```
{
   "path" : "my_documents/my_folder"
}
```

Properties

Name	Type	Description	Available Version
`isInMyFileSync`	Boolean	`true` if the folder and all folders in this folder are synced with Salesforce Files Sync; `false` otherwise. Setting `isInMyFileSync` is valid only for the root folder. Once set to `true`, `isInMyFileSync` can't be set to `false`.	33.0
`name`	String	The name of the folder. Either `path` or `name` is required. If you specify both, `path` is used and `name` is ignored.	30.0
`parentFolderId`	String	The ID of the parent folder. Either `parentFolderId` or `path` is required. You can't include both.	30.0
`path`	String	The path of the folder. For example, `my documents/folder`.	30.0

Folder Item Input

Root XML tag

`<folderItem>`

JSON example

```
{
   "file" :
      {
         "desc" : "Yearly Report for Universal Containers Box Division",
         "title" : "Yearly.txt"
      },
   "type" : "File"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`file`	File Input	A file in a folder	Required to POST a `File`	30.0
`folder`	Folder Input	A folder in a folder	Required to POST a `Folder`	30.0
`type`	String	The item type. One of these values: • `File` • `Folder`	Required	30.0

Following Input

Root XML tag

`<following>`

Properties

Name	Type	Description	Available Version
`subjectId`	String	The ID of what to follow, for example, a user ID, a record ID, a file ID, a topic ID, and so on. Note: Use Group Member Input for adding members to groups.	23.0 Topic IDs are available in 29.0

SEE ALSO:

Follow a Record

Generic Batch Collection Input

Root XML Tag

```
<batch>
```

JSON

```
{
   "batchRequests" : [
    {
       "method" : "Get",
       "url" : "/v34.0/chatter/feeds/news/me"
    },
    {
       "method" : "Get",
       "url" : "/v34.0/chatter/feeds/user-profile/me"
    }
   ]
}
```

Properties

Name	Type	Description	Available Version
`batchRequests`	Generic Batch Input[]	Collection of Batch Request Input request bodies containing the subrequest URLs to execute.	28.0
`haltOnError`	Boolean	The default value is `false`. If the value is `false` and a subrequest in the batch does not complete, Salesforce attempts to execute subsequent subrequests in the batch. If the value is `true` and a subrequest in the batch does not complete due to an HTTP response in the 400 or 500 range, Salesforce halts execution. It does not attempt to execute subsequent subrequests in the batch, and it returns an HTTP 412 status code and a `BATCH_PROCESSING_HALTED` error message for each subsequent subrequest indicating that a previous request was unsuccessful. The top-level request to `/connect/batch` returns HTTP 200 and the `hasErrors` property in the response is set to `true`.	28.0

SEE ALSO:

Like a Feed Item and Return the Total Number of Likes

Generic Batch Input

Root XML Tag

```
<request>
```

JSON

```
{
   "method" : "Get",
   "url" : "/v28.0/chatter/feeds/news/me"
}
```

Properties

Name	Type	Description	Available Version
`binaryPartName`	String	The name of the binary part in the multipart request. When there are multiple binary parts uploaded in one batch request, this value is used to map a request to its binary part. To prevent name collisions, use a unique value for each `binaryPartName` property in a batch request. This property is optional. If this value exists, a `binaryPartNameAlias` value must also exist.	28.0
`binaryPartNameAlias`	String	The `name` parameter in the Content-Disposition header of the binary body part. Different resources expect different values. See Uploading Binary Files. This property is optional. If this value exists, a `binaryPartName` value must also exist.	28.0
`method`	String	One of these HTTP methods: • `HttpDelete`—Returns HTTP 204 on success. Response body or output class is empty. • `HttpGet`—Returns HTTP 200 on success. • `HttpHead`—Returns HTTP 200 on success. Response body or output class is empty. • `HttpPatch`—Returns HTTP 200 on success or HTTP 204 if the response body or output class is empty. • `HttpPost`—Returns HTTP 201 on success or HTTP 204 if the response body or output class is empty. Exceptions are the batch	28.0

Name	Type	Description	Available Version
		posting resources and methods, which return HTTP 200 on success. • `HttpPut`—Return HTTP 200 on success or HTTP 204 if the response body or output class is empty.	
`richInput`		The input body for the request. The type depends on the request specified in the `url` property. This property is optional.	28.0
`url`	String	The Chatter API resource to request. This value is called a *subrequest*. The subrequest URL can start with: • `/services/data/`***`version`***, for example, `/services/data/v34.0/chatter/users/me`. • `/`***`version`***, for example, `/v34.0/chatter/users/me`. The version number must be less than or equal to the version of the request given to the `/connect/batch` request URL. Additional information about subrequests: • The URL can include any query string parameters that the subrequest supports. The query string must be URL-encoded. • You can use parameters to filter response bodies. • You cannot apply headers at the subrequest level.	28.0

Group Chatter Settings Input

Root XML tag

`<groupChatterSettings>`

Properties

A user's Chatter settings.

Name	Type	Description	Available Version
`emailFrequency`	String	The frequency with which a group member receives email from a group. Valid values are: • `EachPost` • `DailyDigest` • `WeeklyDigest` • `Never` In Communities, the Email on Each Post option is disabled once a group exceeds 10,000 members. All members who had this option selected are automatically switched to Daily Digest.	27.0

Group Information Input

JSON

```
{
   "text" : "Post questions about the API.",
   "title" : "API Questions"
}
```

Properties

Name	Type	Description	Available Version
`text`	String	The text in the "Information" section of the group. HTML is not supported. The maximum length is 4000 characters (bytes).	28.0
`title`	String	The title in the "Information" section of the group. The maximum length is 240 characters (bytes).	28.0

Group Input

Root XML Tag

```
<group>
```

JSON

```
{
   "announcement" : "0D5D0000000K4XT",
```

```
   "canHaveChatterGuests" : "true",
   "description" : "This group is building a powerful API.",
   "information" : {
      "text" : "Read this section before you post to the group.",
      "title" : "Information"
   },
   "isArchived" : "false",
   "isAutoArchiveDisabled" : "false",
   "name" : "Chatter API",
   "owner" : "00590000000DmPw",
   "visibility" : "PublicAccess"
}
```

Properties

Name	Type	Description	Available Version
`announcement`	String	An announcement ID. An announcement displays in a designated location in the Salesforce UI until 11:59 p.m. on its expiration date, unless it's deleted or replaced by another announcement.	31.0
`canHaveChatterGuests`	Boolean	`true` if this group allows Chatter customers, `false` otherwise. After this property is set to `true`, it cannot be set to `false`.	29.0
`description`	String	The "Description" section of the group	29.0
`information`	Group Information Input	The "Information" section of the groupIn the Web UI, the "Information" section is above the "Description" section. If the group is private, the "Information" section is visible only to members. The "Description" section is visible to everyone.	28.0
`isArchived`	Boolean	`true` if the group is archived, `false` otherwise. Defaults to `false`.	29.0
`isAutoArchiveDisabled`	Boolean	`true` if automatic archiving is turned off for the group, `false` otherwise. Defaults to `false`.	29.0
`name`	String	The name of the group	29.0
`owner`	String	The ID of the group owner. This property is available for PATCH requests only.	29.0

Name	Type	Description	Available Version
`visibility`	String	Specifies the group visibility type. One of the following values: • `PrivateAccess`—Only members of the group can see posts to this group. • `PublicAccess`—All users within the community can see posts to this group. • `Unlisted`—Reserved for future use.	29.0

SEE ALSO:

Create a Group

Group Member Input

Root XML tag

`<member>`

JSON example

```
{
    "role" : "GroupManager",
    "userId" : "005B0000000Ge16"
}
```

Properties

Name	Type	Description	Available Version
`role`	String	The role of the member. One of these values: • `GroupManager` • `StandardMember`	29.0
`userId`	String	The user ID of the member to be added to the group.	23.0

SEE ALSO:

Join a Group

Group Membership Request Update

Root XML tag

`<groupMembershipRequestUpdate>`

Properties

Name	Type	Description	Available Version
`responseMessage`	String	Use this property to provide a message to the user if their membership request is declined. The value of this property is used only when the value of the `status` property is `Declined`. The maximum length is 756 characters.	27.0
`status`	String	Valid values: • `Accepted` • `Declined`	27.0

SEE ALSO:

Approve or Reject a Request to Join a Private Group

Group Record Input

Record to add to a Chatter group.

Root XML tag

`<groupRecord>`

JSON example

```
{"recordId" : "001D000000Io9cD"}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`recordId`	String	ID of the record.	Required	34.0

Link Capability Input

Create or update a link attached to a feed element.

Root XML tag

```
<linkCapability>
   <url>http://www.example.com</url>
   <urlName>Example</urlName>
</linkCapability>
```

JSON example

```
{
   "url": "http://www.example.com",
   "urlName": "Example"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`url`	String	URL of the link.	Required	32.0
`urlName`	String	Name of the link.	Optional	32.0

Managed Topic Input

A managed topic.

Root XML tag

`<managedTopic>`

JSON example

```
{
"managedTopicType" :"Featured",
"recordId" :"0TOD00000000cwk"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`managedTopicType`	String	Type of managed topic. • `Featured`—Topics that are featured, for example, on the community home page, but don't provide overall navigation. • `Navigational`—Topics that display in a navigational menu in the community. A topic can be associated with up to two managed topic types, so a topic can be both a `Featured` topic and a `Navigational` topic. You can create up to 25 managed topics per `managedTopicType`.	Required	32.0

Name	Type	Description	Required or Optional	Available Version
`name`	String	Name of the managed topic.	Required for new topics You must specify either `name` or `recordId`. You can't specify both.	32.0
`recordId`	String	ID of the record to become a managed topic. In version 32.0, `recordId` must be a topic ID.	Required for existing topics if the `Name` property isn't used You must specify either `name` or `recordId`. You can't specify both.	32.0

Managed Topic Position Collection Input

A collection of relative positions of managed topics.

Root XML tag

```
<managedTopicPositionCollection>
```

JSON example

```
{
"managedTopicPositions" : [
  {
    "managedTopicId" :"0mtD0000000001aIAA",
    "position" :"0"
  },
  {
    "managedTopicId" :"0mtD0000000002qIAA",
    "position" :"2"
  }
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`managedTopicPositions`	Managed Topic Position Input[]	Collection of managed topic position request bodies. This collection can include `Featured` and `Navigational` managed topics and doesn't need to include all managed topics.	Required	32.0

Name	Type	Description	Required or Optional	Available Version
		If you don't include all managed topics in `managedTopicPositions`, the managed topics are reordered by respecting the positions indicated in `managedTopicPositions` and then by pushing down any managed topics that aren't included in `managedTopicPositions` to the next available position. See Reorder managed topics example.		

Reorder managed topics example

If you have these managed topics:

Managed Topic	Position
ManagedTopicA	0
ManagedTopicB	1
ManagedTopicC	2
ManagedTopicD	3
ManagedTopicE	4

And you reorder managed topics by including this information in Managed Topic Position Collection Input:

Managed Topic	Position
ManagedTopicD	0
ManagedTopicE	2

The result is:

Managed Topic	Position
ManagedTopicD	0
ManagedTopicA	1
ManagedTopicE	2
ManagedTopicB	3
ManagedTopicC	4

Managed Topic Position Input

The relative position of a managed topic.

Root XML tag

```
<managedTopicPosition>
```

JSON example

```
{
"managedTopicId" :"0mtD0000000001aIAA",
"position" :"0"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
managedTopicId	String	ID of existing managed topic.	Required	32.0
position	Integer	Relative position of the managed topic, indicated by zero-indexed, ascending whole numbers.	Required	32.0

Message Body Input

Name	Type	Description	Available Version
messageSegments	Segments[]	Collection of message segments, based on type. Segments can be one of the following: • Message Segment Input: Hash Tag • Message Segment Input: Link • Message Segment Input: Mention • Message Segment Input: Text	23.0

SEE ALSO:

Edit a Comment

Edit a Feed Post

Edit a Question Title and Post

Message Segment Input: Hash Tag

Represents an inline hash tag. The use of this segment is optional because hash tags are also parsed out of Text segments.

Name	Type	Description	Available Version
`tag`	String	Text of the hash tag without the hash tag. Note: Closing square brackets ( ]) are not supported in hash tag text. If the text contains a closing square bracket ( ]), the hash tag ends at the bracket.	23.0 Multiple word hash tags are available in version 28.0.
`type`	String	Value is `Hashtag`.	23.0

Message Segment Input: Link

Represents an inline link to a URL. If a link is provided as input without whitespace text before it, a space is inserted. If a link is provided as input without whitespace text after it, a space is inserted. The use of this segment is optional because links are also parsed out of Text segments.

Name	Type	Description	Available Version
`type`	String	Value is `Link`.	23.0
`url`	URL	URL to be used for the link.	23.0

Message Segment Input: Mention

Represents an inline mention of a user or group. You must provide a user or group ID. No spaces are added. When creating a feed item or a comment, you can include up to 25 mentions.

Name	Type	Description	Available Version
`id`	String	ID of the user or group to be mentioned.	23.0 Groups are available in 29.0
`type`	String	Value is `Mention`	23.0

Message Segment Input: Text

Represents plain inline text. Text that looks like a link and is surrounded by space is converted into a link segment. Text prefixed with a hash and surrounded by square brackets ([]) is converted into a hash tag segment. Multiple word hash tags are available in version 28.0 and higher.

Name	Type	Description	Available Version
`text`	String	Plain text for this segment.	23.0
`type`	String	Only value is `Text`.	23.0

SEE ALSO:

- Edit a Comment
- Edit a Feed Post
- Edit a Question Title and Post

Moderation Flag Input

Root XML tag

```
<moderationFlag>
```

JSON

```
{
   "visibility" : "ModeratorsOnly"
}
```

Properties

Name	Type	Description	Available Version
`visibility`	String	Specifies the visibility behavior of a flag for various user types. • `ModeratorsOnly`—The flag is visible only to users with moderation permissions on the flagged element or item. • `SelfAndModerators`—The flag is visible to the creator of the flag and to users with moderation permissions on the flagged element or item.	30.0

Photo Input

Describes a photo input.

Root XML tag

```
<photo>
```

JSON example

```
{    "cropSize" : "120",
     "cropX" : "0",
     "cropY" : "0",
     "fileId" : "069D00000001IOh"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`cropSize`	Integer	The length, in pixels, of any edge of the crop square.	Required to crop an uploaded file or an existing file.	29.0
`cropX`	Integer	The position X, in pixels, from the left edge of the image to the start of the crop square. Top left is position (0,0).	Required to crop an uploaded file or an existing file.	29.0
`cropY`	Integer	The position Y, in pixels, from the top edge of the image to the start of the crop square. Top left is position (0,0).	Required to crop an uploaded file or an existing file.	29.0
`fileId`	String	18–character ID of an existing file. The key prefix must be 069 and the file size must be less than 2 MB. Note: Images uploaded on the Group page and on the User page don't have file IDs and therefore can't be used.	Required to choose an existing file.	25.0
`versionNumber`	Integer	Version number of an existing file. If not provided, the latest version is used.	Optional	25.0

SEE ALSO:

Update and Crop a Group Photo

Update and Crop a User Photo

Poll Capability Input

Create, update, or vote on a poll attached to a feed element.

XML example

```
<pollCapability>
   <pollChoices>
```

```
      <choice>Monday</choice>
      <choice>Tuesday</choice>
   </pollChoices>
</pollCapability>
```

JSON example

```
{
   "choices" : [
      "Monday", "Tuesday"
   ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`choices`	String[]	The choices used to create a new poll. You must specify 2–10 poll choices for each poll.	Required for creating a poll	32.0
`myChoiceId`	String	ID of an existing choice on the feed poll. Used to vote on an existing poll.	Required for voting on a poll	32.0

Push Notification Input

A push notification.

Root XML tag

`<pushNotification>`

JSON example

```
{    "appName" : "TestApp",
     "namespace" : "abc",
     "userIds" : ["005x00000013dPK", "005x00000013dPL"],
     "payload" : "{'aps':{'alert':'test', 'badge':0, 'sound':'default'}}"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`appName`	String	The API name of the connected app that the push notification is sent to.	Required	31.0
`namespace`	String	The namespace of the connected app that the push notification is sent to.	Required if the namespace is set	31.0
`payload`	String	The push notification payload in JSON format.	Required	31.0
`userIds`	String[]	The push notification recipient user ids.	Required	31.0

Question and Answers Capability Input

Create or edit a question feed element or set the best answer of the existing question feed element.

Root XML tag

```
<questionAndAnswersCapability>
   <bestAnswerId>0D7D00000000lMAKAY</bestAnswerId>
</questionAndAnswersCapability>
```

JSON example

```
{
   "bestAnswerId": "0D7D00000000lMAKAY"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`bestAnswerId`	String	A comment ID to use as a best answer for a question feed element. The best answer comment must already exist on the question feed element.	Required to PATCH the best answer. Not supported in POST requests.	32.0
`questionTitle`	String	Title for a question feed element. To edit the title of a question, make a PATCH request to `/chatter/feed-elements/`***`feedElementId`***. Editing question titles is supported in version 34.0 and later.	Required to POST or PATCH a question.	32.0

SEE ALSO:

Edit a Question Title and Post

Recommendation Definition Input

A recommendation definition.

JSON example

```
{
   "actionUrl" : "https://www.example.com",
   "actionUrlName" : "Launch",
   "explanation" : "Click this link to learn more!",
   "name" : "Corporate Training Project",
   "title" : "Want to learn more about Corporate Training Project?"}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`actionUrl`	String	The URL for acting on the recommendation, for example, the URL to join a group.	Required for creating a recommendation definition. Optional for updating an existing recommendation definition.	33.0
`actionUrlName`	String	The text label for the action URL in the user interface, for example, "Launch."	Required for creating a recommendation definition. Optional for updating an existing recommendation definition.	34.0
`explanation`	String	The explanation, or body, of the recommendation.	Required for creating a recommendation definition. Optional for updating an existing recommendation definition.	33.0
`name`	String	The name of the recommendation definition. The name is displayed in Setup.	Required for creating a recommendation definition. Optional for updating an existing recommendation definition.	33.0
`title`	String	The title of the recommendation definition.	Optional	33.0

Request Header Input

An HTTP request header name and value pair.

JSON example

```
{
   "name": "Content-Type",
   "value": "application/json"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`name`	String	The name of the request header.	Required	33.0
`value`	String	The value of the request header.	Required	33.0

Scheduled Recommendation Input

A scheduled recommendation.

JSON example for creating a new scheduled recommendation

```
{
   "rank" : "2",
   "recommendationDefinitionId" : "0RDD00000004CEhOAM"}
```

JSON example for updating an existing scheduled recommendation

```
{   "rank" : "2"}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`enabled`	Boolean	Indicates whether scheduling is enabled. If `true`, the recommendation is enabled and appears in communities. If `false`, recommendations in feeds in the Salesforce1 mobile browser app aren't removed, but no new recommendations appear. In communities using the Summer '15 or later version of the Napili template, disabled recommendations no longer appear.	Optional	34.0
`rank`	Integer	Relative rank of the scheduled recommendation indicated by ascending whole numbers starting with 1. Setting the rank is comparable to an insertion into an ordered list. The scheduled recommendation is inserted	Optional	33.0

Name	Type	Description	Required or Optional	Available Version
		into the position specified by the `rank`. The `rank` of all the scheduled recommendations after it is pushed down. See Ranking scheduled recommendations example. If the specified `rank` is larger than the size of the list, the scheduled recommendation is put at the end of the list. The `rank` of the scheduled recommendation is the size of the list, instead of the one specified. If a `rank` is not specified, the scheduled recommendation is put at the end of the list.		
`recommendationDefinitionId`	String	ID of the recommendation definition that this scheduled recommendation schedules.	Required for creating a scheduled recommendation. You can't specify a `recommendationDefinitionId` when updating an existing scheduled recommendation.	33.0

Ranking scheduled recommendations example

If you have these scheduled recommendations:

Scheduled Recommendations	Rank
ScheduledRecommendationA	1
ScheduledRecommendationB	2
ScheduledRecommendationC	3

And you include this information in the Scheduled Recommendation Input:

Scheduled Recommendation	Rank
ScheduledRecommendationD	2

The result is:

Scheduled Recommendation	Rank
ScheduledRecommendationA	1
ScheduledRecommendationD	2
ScheduledRecommendationB	3
ScheduledRecommendationC	4

Share Input

JSON example

```
{
  "id": "005D0000001Az4l",
  "sharingType": "V"
}
```

Properties

Name	Type	Description	Available Version
`id`	String	User ID to share a file with	30.0
`sharingType`	String	Either `C` for collaborator or `V` for viewer	30.0

Topic Assignment Input

Represents a topic to assign to a feed item or record. Administrators must enable topics for objects before users can add topics to records of that object type.

Root XML tag

`<topicAssignment>`

JSON

The following example assigns a topic to a record:

```
{ "topicId": "0TOD0000000090r"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`topicId`	String	ID of an existing topic.	Required for existing topics if the `topicName` property isn't used.	30.0

Name	Type	Description	Required or Optional	Available Version
`topicName`	String	Name of a new or existing topic.	Required for new topics; required for existing topics if the `topicId` property isn't used.	30.0

Topic Endorsement Input

Endorse a user who's knowledgeable about a topic.

Root XML tag

```
<topicEndorsement>
```

JSON

```
{
   "userId" : "005B0000000Ge16"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
`userId`	String	ID of the user to be endorsed for the specified topic.	Required	30.0

Topic Input

Root XML tag

```
<topic>
```

JSON

```
{
   "description" : "World Wide Web Consortium (W3C) is the main international standards
 organization for the Web."
   "name" : "W3C"}
```

Properties

Name	Type	Description	Available Version
`description`	String	Text description of the topic	28.0
`idsToMerge`	String[]	List of up to five topic IDs to merge with the topic	33.0

Name	Type	Description	Available Version
`name`	String	Name of the topic Use this property to change only the capitalization and spacing of the topic name.	28.0

User Chatter Settings Input

A user's global Chatter settings.

Root XML tag

```
<userChatterSettings>
```

Properties

Name	Type	Description	Available Version
`defaultGroupEmailFrequency`	String	The default frequency with which a user receives email from a group when they join. Valid values are: • `EachPost` • `DailyDigest` • `WeeklyDigest` • `Never` In Communities, the Email on Each Post option is disabled once a group exceeds 10,000 members. All members who had this option selected are automatically switched to Daily Digest.	27.0

User Input

Root XML tag

```
<user>
```

Properties

Name	Type	Description	Available Version
`aboutMe`	String	The `aboutMe` property of a User Detail response body. The maximum length is 1000 characters. This property populates the "About Me" section of the user profile,	29.0

Name	Type	Description	Available Version
		which is visible to all members of a community or an organization.	

JSON example

```
{
   "aboutMe": "Staff Technical Writer responsible for the Connect API and Connect in Apex documentation."
}
```

SEE ALSO:

Update a User Profile

User Message

Root XML tag

`<message>`

Properties

Name	Type	Description	Available Version
`body`	String	Text of the message body	23.0
`inReplyTo`	String	ID of an existing message that identifies which conversation this message is part of.You must specify either `recipients` or `inReplyTo`. You must specify one or the other, not both.	23.0
`recipients`	User Message Recipient List Input	List of users who are the intended message recipients, up to 9. Comma-separated list of user IDs.You must specify either `recipients` or `inReplyTo`. You must specify one or the other, not both.	23.0

Example XML format:

```
<message>
   <body>Text of the message</body>
```

```
   <recipients>
      <recipient>userID</recipient>
      <recipient>userID</recipient>
   </recipients>
   <inReplyTo>messageID</inReplyTo>
</message>
```

Example JSON format:

```
{
   "body": "Text of the message",
   "recipients": ["userID","userID"],
   "inReplyTo": "messageID"
}
```

SEE ALSO:

Send a Private Message

User Message Recipient Input

Root XML tag

`<recipient>`

Properties

Name	Type	Description	Available Version
recipient	String	18-character ID of the user who receives the message.	23.0

User Message Recipient List Input

Root XML tag

`<message>`

Properties

Name	Type	Description	Available Version
recipients	User Message Recipient Input[]	Collection of recipients who receive the message.	23.0

User Status Input

Note: Setting a user status is no longer available as of version 25.0. Post feed items to the feed instead, using Feed Item Input.

Root XML tag

`<userStatus>`

Properties

Name	Type	Description	Available Version
`body`	Message Body Input	Description of message body.	23.0–24.0

User Stream (Conversation) Input

Root XML tag

```
<conversation>
```

Properties

Name	Type	Description	Available Version
`read`	Boolean	Changes the conversation status to unread (`false`) or read (`true`).	24.0

CHAPTER 7 Chatter REST API Response Bodies

The successful execution of a request to a Chatter REST API resource can return a response body in either JSON or XML format.

A request to a Chatter REST API resource always returns an HTTP response code, whether the request was successful or not.

Action Link Definition

The definition of an action link. To read this information, a request must use the same connected app that was used to create the action link definition.

Property Name	Type	Description	Filter Group and Version	Available Version
`actionUrl`	String	The action link URL. For example, a `Ui` action link URL is a Web page. A `Download` action link URL is a link to the file to download. `Ui` and `Download` action link URLs are provided to clients. An `Api` or `ApiAsync` action link URL is a REST resource. `Api` and `ApiAsync` action link URLs aren't provided to clients. Links to Salesforce can be relative. All other links must be absolute and start with `https://`.	Small, 33.0	33.0
`createdDate`	Date	An ISO 8601 format date string, for example, 2011-02-25T18:24:31.000Z..	Small, 33.0	33.0
`excludedUserId`	String	ID of a single user to exclude from performing the action. If you specify an `excludedUserId`, you can't specify a `userId`.	Small, 33.0	33.0
`groupDefault`	Boolean	`true` if this action is the default action link in the action link group; `false` otherwise. There can be only one default action link per action link group. The default action link gets distinct styling in the Salesforce UI.	Small, 33.0	33.0
`headers`	Request Header[]	The request headers for the `Api` and `ApiAsync` action link types.	Big, 33.0	33.0
`id`	String	18-character ID for the action link definition.	Small, 33.0	33.0

Property Name	Type	Description	Filter Group and Version	Available Version
`label`	String	A custom label to display on the action link button. A `label` value can be set only in an action link template. Action links have four statuses: NewStatus, PendingStatus, SuccessStatus, and FailedStatus. These strings are appended to the label for each status: • *label* • *label* Pending • *label* Success • *label* Failed For example, if the value of `label` is "See Example," the values of the four action link states are: See Example, See Example Pending, See Example Success, and See Example Failed. An action link can use either `label` or `labelKey` to generate label names, it can't use both. If `label` has a value, the value of `labelKey` is `None`. If `labelKey` has a value other than `None`, the value of `label` is `null`.	Small, 34.0	34.0
`labelKey`	String	Key for the set of labels to show in the user interface. A set includes labels for these states: NewStatus, PendingStatus, SuccessStatus, FailedStatus. For example, if you use the `Approve` key, you get these labels: Approve, Pending, Approved, Failed. See Action Links Labels.	Small, 33.0	33.0
`method`	String	The HTTP method. One of these values: • `HttpDelete`—Returns HTTP 204 on success. Response body or output class is empty. • `HttpGet`—Returns HTTP 200 on success. • `HttpHead`—Returns HTTP 200 on success. Response body or output class is empty. • `HttpPatch`—Returns HTTP 200 on success or HTTP 204 if the response body or output class is empty.	Medium, 33.0	33.0

Property Name	Type	Description	Filter Group and Version	Available Version
		• `HttpPost`—Returns HTTP 201 on success or HTTP 204 if the response body or output class is empty. Exceptions are the batch posting resources and methods, which return HTTP 200 on success. • `HttpPut`—Return HTTP 200 on success or HTTP 204 if the response body or output class is empty.		
`modifiedDate`	Date	ISO8601 date string, for example, 2011-02-25T18:24:31.000Z.	Small, 33.0	33.0
`requestBody`	String	The request body for `Api` and `ApiAsync` action link types.	Big, 33.0	33.0
`requires Confirmation`	Boolean	`true` to require the user to confirm the action; `false` otherwise.	Small, 33.0	33.0
`templateId`	String	The ID of the action link template from which to instantiate this action link. If the action link isn't associated with a template, the value is `null`.	Medium, 33.0	33.0
`type`	String	Defines the type of action link. Values are: • `Api`—The action link calls a synchronous API at the action URL. Salesforce sets the status to `SuccessfulStatus` or `FailedStatus` based on the HTTP status code returned by your server. • `ApiAsync`—The action link calls an asynchronous API at the action URL. The action remains in a `PendingStatus` state until a third party makes a request to `/connect/action-links/`***`actionLinkId`*** to set the status to `SuccessfulStatus` or `FailedStatus` when the asynchronous operation is complete. • `Download`—The action link downloads a file from the action URL. • `Ui`—The action link takes the user to a web page at the action URL.	Small, 33.0	33.0

Property Name	Type	Description	Filter Group and Version	Available Version
`userId`	String	The ID of the user who can execute the action. If not specified or `null`, any user can execute the action. If you specify a `userId`, you can't specify an `excludedUserId`.	Small, 33.0	33.0

Action Link Diagnostic

Any diagnostic information that is returned from the REST endpoint when an action link has been executed. Diagnostic information is given only for users who can access the action link.

Property Name	Type	Description	Filter Group and Version	Available Version
`diagnosticInfo`	String	Any diagnostic information that is returned from the REST endpoint when an action link has been executed. Diagnostic information is given only for users who can access the action link.	Small, 33.0	33.0
`url`	String	The Chatter REST API URL for this action link diagnostic.	Small, 33.0	33.0

Action Link Group Definition

The definition of an action link group. All action links must belong to a group. Action links in a group are mutually exclusive and share some properties. Define stand-alone actions in their own action group.

Property Name	Type	Description	Filter Group and Version	Available Version
`actionLinks`	Action Link Definition[]	A collection of action link definitions that make up the action link group. Within an action link group, action links are displayed in the order listed in the `actionLinks` property of the Action Link Group Definition Input request body. Within a feed item, action link groups are displayed in the order specified in the `actionLinkGroupIds` property of the Associated Actions Capability Input request body.	Medium, 33.0	33.0

Property Name	Type	Description	Filter Group and Version	Available Version
`category`	String	Indicates the priority and location of the action links. Values are: • `Primary`—The action link group is displayed in the body of the feed element. • `Overflow`—The action link group is displayed in the overflow menu of the feed element.	Medium, 33.0	33.0
`createdDate`	Date	ISO8601 date string, for example, 2011-02-25T18:24:31.000Z	Small, 33.0	33.0
`executionsAllowed`	String	Defines the number of times an action link can be executed. Values are: • `Once`—An action link can be executed only once across all users. • `OncePerUser`—An action link can be executed only once for each user. • `Unlimited`—An action link can be executed an unlimited number of times by each user. If the action link's `actionType` is `Api` or `ApiAsync`, you can't use this value.	Medium, 33.0	33.0
`expirationDate`	Date	ISO 8601 date string, for example, 2011-02-25T18:24:31.000Z, that represents the date and time this action group expires and can no longer be executed. If the value is `null`, there isn't an expiration date.	Medium, 33.0	33.0
`id`	String	18-character ID of the action link group definition	Small, 33.0	33.0
`modifiedDate`	Date	ISO8601 date string, for example, 2011-02-25T18:24:31.000Z	Small, 33.0	33.0
`templateId`	String	The ID of the action link group template from which to instantiate this action link group, or `null` if this group isn't associated with a template.	Medium, 33.0	33.0
`url`	String	The Chatter REST API URL for this action link group definition.	Small, 33.0	33.0

Address

Property	Type	Description	Filter Group and Version	Available Version
`city`	String	Name of the city	Small, 29.0	23.0
`country`	String	Name of the country	Small, 29.0	23.0
`formattedAddress`	String	Address formatted by locale	Small, 29.0	25.0
`state`	String	Name of the state, province, or so on	Small, 29.0	23.0
`street`	String	Street number	Small, 29.0	23.0
`zip`	String	Zip or postal code	Small, 29.0	23.0

Announcement

An announcement displays in a designated location in the Salesforce UI until 11:59 p.m. on its expiration date, unless it's deleted or replaced by another announcement.

Property Name	Type	Description	Filter Group and Version	Available Version
`expirationDate`	Date	The Salesforce UI displays an announcement until 11:59 p.m. on this date unless another announcement is posted first. The Salesforce UI ignores the time value in the `expirationDate`. However, you can use the time value to create your own display logic in your own UI.	Small, 31.0	31.0
`feedElement`	Feed Item	The feed element that contains the body of the announcement and its associated comments, likes, and so on.	Small, 31.0	31.0
`id`	String	18-character ID of the announcement.	Small, 31.0	31.0
`url`	String	The URL to the announcement.	Small, 33.0	33.0

Announcement Page

A paginated collection of Announcement response bodies.

Property Name	Type	Description	Filter Group and Version	Available Version
`announcements`	Announcement[]	A collection of Announcement response bodies. An announcement displays in a designated location in the Salesforce UI until 11:59 p.m. on its expiration date, unless it's deleted or replaced by another announcement.	Small, 31.0	31.0
`currentPageUrl`	String	Chatter REST API URL identifying the current page.	Medium, 31.0	31.0
`nextPageUrl`	String	Chatter REST API URL identifying the next page or `null` if there isn't a next page. Check whether this value is `null` before getting another page. If a page doesn't exist, a `ConnectApi.NotFoundException` error is returned.	Small, 31.0	31.0
`previousPageUrl`	String	Chatter REST API URL identifying the previous page or `null` if there isn't a previous page.	Medium, 31.0	31.0

Approval Capability

If a feed element has this capability, it includes information about an approval.

Property Name	Type	Description	Filter Group and Version	Available Version
`id`	String	The work item ID. The work item ID is `null` if there isn't a pending work item associated with the approval record.	Small, 32.0	32.0
`postTemplateFields`	Approval Post Template Field[]	The details of the approval post template field.	Small, 32.0	32.0
`processInstanceStepId`	String	The process instance step ID. The associated record represents one step in an approval process.	Small, 32.0	32.0
`status`	String	The status of the approval. One of these values: • `Approved` • `Fault` • `Held` • `NoResponse`	Small, 32.0	32.0

Property Name	Type	Description	Filter Group and Version	Available Version
		• `Pending` • `Reassigned` • `Rejected` • `Removed` • `Started`		

Approval Post Template Field

Description of an approval post template field.

Property	Type	Description	Filter Group and Version	Available Version
`displayName`	String	The field name	Small, 29.0	28.0
`displayValue`	String	The field value or `null` if the field is set to `null`.	Small, 29.0	28.0
`record`	Reference	A record ID If no record exists or if the reference is `null`, this value is `null`.	Small, 29.0	28.0

Article Item

Article item in question and answer search results.

Property Name	Type	Description	Filter Group and Version	Available Version
`id`	String	ID of the article.	Small, 32.0	32.0
`rating`	Double	Rating of the article.	Small, 32.0	32.0
`title`	String	Title of the article.	Small, 32.0	32.0
`urlLink`	String	Link URL of the article.	Small, 32.0	32.0
`viewCount`	Integer	Number of votes given to the article.	Small, 32.0	32.0

Associated Actions Capability

If a feed element has this capability, it has platform actions associated with it.

Property Name	Type	Description	Filter Group and Version	Available Version
`platformActionGroups`	Platform Action Group[]	The platform action groups associated with a feed element. Platform action groups are returned in the order specified in the Associated Actions Capability Input request body.	Medium, 33.0	33.0

Banner Capability

If this feed element has this capability, it has a banner motif and style.

Property Name	Type	Description	Filter Group and Version	Available Version
`motif`	Motif	A banner motif.	Small, 31.0	31.0
`style`	String	Decorates a feed item with a color and set of icons. • `Announcement`—An announcement displays in a designated location in the Salesforce UI until 11:59 p.m. on its expiration date, unless it's deleted or replaced by another announcement.	Small, 31.0	31.0

Batch Result Item

Property	Type	Description	Filter Group and Version	Available Version
`result`	The type depends on the object type of the batch. **Important:** If the result is an error, the type is a collection.	Dependent on the type of batch. Value could be one of the following: • Feed Item for `/chatter/feed-elements/batch` (v32.0) • Error Message (v23.0) • File Summary for `/chatter/files/batch/fileIds` (v26.0) • Group Member for `/chatter/group-memberships/batch/membershipIds`	Small, 29.0	

Property	Type	Description	Filter Group and Version	Available Version
		(v27.0) • Group for `/chatter/groups/batch/`***`groupIds`*** (v23.0) • Motif for `/connect/motifs/batch/`***`idOrPrefixList`*** (v27.0) • User Detail for `/chatter/users/batch/`***`userIds`*** (v23.0) • For `/connect/batch`, the value can be any response body depending on the sub-request in the batch. (v28.0)		
`statusCode`	Integer	An HTTP status code indicating the status of this individual request in the batch	Small, 29.0	23.0

Batch Results

Property	Type	Description	Filter Group and Version	Available Version
`hasErrors`	Boolean	`true` if at least one of the results in the result set is an HTTP status code in the 400 or 500 range; `false` otherwise	Small, 29.0	28.0
`results`	Batch Result Item[]	Collection of batch result items	Small, 29.0	23.0

Bookmarks Capability

If a feed element has this capability, the context user can bookmark it.

Property Name	Type	Description	Filter Group and Version	Available Version
`isBookmarkedByCurrentUser`	Boolean	Indicates whether the feed element has been bookmarked by the current user (`true`) or not (`false`).	Small, 32.0	32.0

Case Comment Capability

If a feed element has this capability, it has a case comment on the case feed.

Property Name	Type	Description	Filter Group and Version	Available Version
`actorType`	String	Specifies the type of user who made the comment. • `Customer`—if a Chatter customer made the comment • `CustomerService`—if a service representative made the comment	Small, 32.0	32.0
`createdBy`	User Detail or User Summary	Information about the user who created the comment.	Small, 32.0	32.0
`createdDate`	Date	ISO8601 date string, for example, 2011-02-25T18:24:31.000Z.	Small, 32.0	32.0
`eventType`	String	Specifies an event type for a comment in the case feed. • `NewInternal`—A case comment that has newly been marked Internal Only. • `NewPublished`—A newly published case comment. • `NewPublishedByCustomer`—A case comment by a customer that was newly published. • `PublishExisting`—An existing case comment that was republished. • `PublishExistingByCustomer`—An existing case comment by a customer that was republished. • `UnpublishExistingByCustomer`—An existing case comment by a customer that was unpublished. • `UnpublishExsiting`—An existing case comment that was unpublished.	Small, 32.0	32.0

Property Name	Type	Description	Filter Group and Version	Available Version
		Note: Unfortunately, this typo is in the code, not the documentation. Use this spelling in your code.		
`id`	String	18-character ID of case comment.	Small, 32.0	32.0
`published`	Boolean	Specifies whether the comment has been published.	Small, 32.0	32.0
`text`	String	Text of the case comment.	Small, 32.0	32.0

Canvas Capability

If a feed element has this capability, it renders a canvas app.

Property Name	Type	Description	Filter Group and Version	Available Version
`description`	String	A description of the canvas app. The maximum size is 255 characters.	Small, 32.0	32.0
`developerName`	String	The API name (developer name) of the connected app.	Small, 32.0	32.0
`height`	String	The height of the canvas app in pixels.	Small, 32.0	32.0
`icon`	ConnectApi.Icon	The icon for the canvas app.	Small, 32.0	32.0
`namespacePrefix`	String	A unique namespace prefix for the canvas app.	Small, 32.0	32.0
`parameters`	String	JSON parameters passed to the canvas app.	Small, 32.0	32.0
`thumbnailUrl`	String	A thumbnail URL to a preview image. The maximum thumbnail size is 120 pixels by 120 pixels.	Small, 32.0	32.0
`title`	String	A title for the canvas link.	Small, 32.0	32.0

Chatter Activity

Property	Type	Description	Filter Group and Version	Available Version
`commentCount`	Integer	Total number of comments in the organization or community made by the user	Small, 29.0	23.0

Property	Type	Description	Filter Group and Version	Available Version
`commentReceivedCount`	Integer	Total number of comments in the organization or community received by the user	Small, 29.0	23.0
`likeReceivedCount`	Integer	Total number of likes on posts and comments in the organization or community received by the user	Small, 29.0	23.0
`postCount`	Integer	Total number of posts in the organization or community made by the user	Small, 29.0	23.0

Chatter Directory

Property	Type	Description	Filter Group and Version	Available Version
`feeds`	String	Returns a URL to the general feeds resource for the organization	Small, 29.0	24.0
`groups`	String	Returns a URL to the general groups resource for the organization	Small, 29.0	24.0
`organization`	String	Note: This property is no longer available as of version 26.0. Use the `organization` property in the Connect Directory instead. Returns a URL to the general connect organization resource		24.0–26.0
`users`	String	Returns a URL to the general users resource for the organization	Small, 29.0	24.0

Chatter Influence

Property	Type	Description	Filter Group and Version	Available Version
`percentile`	String	Percentile value for the user's influence rank within the organization or community	Small, 29.0	24.0
`rank`	Integer	Number indicating the user's influence rank, relative to all other users within the organization or community	Small, 29.0	24.0

Chatter Likes Capability

If a feed element has this capability, the context user can like it. Exposes information about existing likes.

Property Name	Type	Description	Filter Group and Version	Available Version
`isLikedBy CurrentUser`	Boolean	Indicates whether the feed element is liked by the current user (`true`) or not (`false`).	Small, 32.0	32.0
`page`	Like Page	Likes information for this feed element.	Small, 32.0	32.0
`likesMessage`	Like Message Body	A message body that describes who likes the feed element.	Small, 32.0	32.0
`myLike`	Reference	If the context user has liked the feed element, this property is a reference to the specific like, `null` otherwise.	Medium, 32.0	32.0

ClientInfo

Property	Type	Description	Filter Group and Version	Available Version
`applicationName`	String	Name of the connected app used for authentication. For more information, see Understanding Connected Apps.	Small, 29.0	23.0
`applicationUrl`	String	Value from the `Info URL` field of the connected app used for authentication	Small, 29.0	23.0

Comment

Property	Type	Description	Filter Group and Version	Available Version
`attachment`	Depends on type of comment	If the comment does not contain an attachment or if the attachment, returns `null`. Otherwise, returns Feed Item Attachment: Content. **Important:** As of verson 32.0, use the `capabilities` property.	Medium, 29.0	24.0-31.0
`body`	Feed Item Body	Message body for the comment	Small, 29.0	23.0
`capabilities`	Comment Capabilities	Capabilities associated with the comment.	Medium, 32.0	32.0
`clientInfo`	ClientInfo	Information about the connected app used to authenticate the connection	Medium, 29.0	23.0
`createdDate`	String	ISO8601 date string, for example, 2011-02-25T18:24:31.000Z	Small, 29.0	23.0

Property	Type	Description	Filter Group and Version	Available Version
`feedElement`	Reference	The feed element on which the comment is posted.	Medium, 32.0	32.0
`feedItem`	Reference	The feed item on which the comment is posted. **Important:** As of version 32.0, use the `feedElement` property.	Medium, 29.0	23.0-31.0
`id`	String	18-character ID of the comment	Small, 29.0	23.0
`isDeletable`	Boolean	**Note:** This property is no longer available as of version 25.0. Use the `isDeleteRestricted` property instead. Specifies if the context user can delete the comment		23.0–24.0
`isDelete Restricted`	Boolean	Returns `true` if a comment cannot be deleted by the context user. If it returns `false`, possibly the context user can delete the comment.	Medium, 29.0	25.0
`likes`	Like Page	Page of likes associated with the comment	Medium, 29.0	24.0
`likesMessage`	Like Message Body	A message body that describes who likes the comment.	Medium, 29.0	27.0
`myLike`	Reference	If the context user has liked the comment, returns a reference to that like or returns `null`	Medium, 29.0	23.0
`parent`	Reference	Information about the parent of the feed-item associated with this comment	Big, 29.0	23.0
`relative CreatedDate`	String	The created date formatted as a relative, localized string, for example, “17m ago” or “Yesterday.”	Medium, 29.0	28.0
`type`	String	Indicates the type of comment. Valid values are: • `ContentComment`—the comment contains an attachment • `TextComment`—the comment contains only text	Small, 29.0	24.0
`url`	String	URL to this comment	Small, 29.0	23.0
`user`	User Summary	Information about the comment author	Small, 29.0	23.0

Comment Capabilities

A container for all capabilities that can be included with a comment.

Property Name	Type	Description	Filter Group and Version	Available Version
`content`	Content Capability	Content, such as a file attachment, associated with this feed element. Most Content Capability properties are null if the content has been deleted from the feed element or if the access has changed to private.	Small, 32.0	32.0
`edit`	Edit Capability	If a comment has this capability, users who have permission can edit it.	Small, 34.0	34.0

Comment Page

Property	Type	Description	Filter Group and Version	Available Version
`comments`	Comment[]	Collection of comments **Important:** Use the `items` property as of version 32.0.	Small, 29.0	23.0-31.0
`currentPageUrl`	String	URL for the current page of comments, that is, the most recent comments. The default is 25 comments per page.	Medium, 29.0	23.0
`items`	Comment[]	Collection of comments	Small, 32.0	32.0
`nextPageUrl`	String	URL to the next page of comments, that is, older comments. The default is 25 comments per page.	Small, 29.0	23.0
`total`	Integer	Total number of comments across all pages for the parent feed item	Small, 29.0	23.0

Comments Capability

If a feed element has this capability, the context user can add a comment.

Property Name	Type	Description	Filter Group and Version	Available Version
`page`	Comment Page	The comments information for this feed element.	Small, 32.0	32.0

Community

Property	Type	Description	Filter Group and Version	Available Version
`allowChatterAccessWithoutLogin`	Boolean	Specifies if guest users can access public groups in the community without logging in.	Small, 31.0	31.0
`allowMembersToFlag`	Boolean	Specifies whether members of the community can flag content	Small, 30.0	30.0
`description`	String	Community's description	Small, 29.0	26.0
`id`	String	18-character ID of the community	Small, 29.0	26.0
`invitationsEnabled`	Boolean	Specifies whether members of the community can invite other members to the community, `true` if they can, `false` otherwise	Small, 29.0	26.0
`knowledgeableEnabled`	Boolean	Specifies whether knowledgeable people and endorsements are available for topics (`true`), or not (`false`).	Small, 30.0	30.0
`name`	String	Community name	Small, 29.0	26.0
`nicknameDisplayEnabled`	Boolean	Specifies whether nicknames are displayed in the community.	Small, 32.0	32.0
`privateMessagesEnabled`	Boolean	Specifies whether members of the community can send and receive private messages to and from other members of the community (`true`) or not (`false`).	Small, 30.0	30.0
`reputationEnabled`	Boolean	Specifies whether reputation is calculated and displayed for members of the community.	Small, 31.0	31.0
`sendWelcomeEmail`	Boolean	Specifies whether a welcome email is automatically generated and sent to any new member. `true` if email is sent, `false` otherwise	Small, 29.0	26.0
`sendWelcomeEmail`	Boolean	Specifies whether a welcome email is automatically generated and sent to any new member. `true` if email is sent, `false` otherwise	Small, 29.0	26.0
`siteUrl`	String	Site URL for the community, which is the custom domain plus a URL prefix	Small, 30.0	30.0
`status`	String	Community's status. Values are: • `Live` • `Inactive` • `UnderConstruction`	Small, 29.0	28.0
`urlPathPrefix`	String	Community-specific URL prefix	Small, 29.0	26.0
`url`	String	Full URL to community	Small, 29.0	26.0

Community Page

Property	Type	Description	Filter Group and Version	Available Version
`communities`	Community[]	List of communities available to the current user	Small, 29.0	26.0
`total`	Integer	Total number of communities	Small, 29.0	26.0

Connect Directory

Property	Type	Description	Filter Group and Version	Available Version
`communities`	Community[]	List of communities in the current organization	Small, 29.0	26.0
`organization`	String	URL to the Chatter organization	Small, 29.0	25.0

Content Capability

If a feed element has this capability, it has a file attachment.

If content is deleted from a feed element after it's posted or if the access to the content is changed to private, the Content Capability exists, however most of its properties are null.

Property Name	Type	Description	Filter Group and Version	Available Version
`checksum`	String	MD5 checksum for the file.	Small, 32.0	32.0
`contentHub Repository`	Files Connect Repository	The external data source associated with this file, or `null` if it doesn't point to an external file.	Small, 32.0	32.0
`contentUrl`	String	URL of the content for links and Google docs.	Small, 32.0	32.0
`description`	String	Description of the attachment.	Small, 32.0	32.0
`downloadUrl`	String	URL to the content.	Small, 32.0	32.0
`external DocumentUrl`	String	The external URL this file points to, or `null` if it does not point to an external file.	Small, 32.0	32.0
`fileExtension`	String	Extension of the file.	Small, 32.0	32.0
`fileSize`	String	Size of the file in bytes. If size cannot be determined, returns `Unknown`.	Small, 32.0	32.0
`fileType`	String	Type of file.	Small, 32.0	32.0

Property Name	Type	Description	Filter Group and Version	Available Version
`hasPdfPreview`	Boolean	`true` if the file has a PDF preview available, `false` otherwise.	Medium, 32.0	32.0
`id`	String	18-character ID of the content.	Small, 32.0	32.0
`isInMyFileSync`	Boolean	`true` if the file is synced withSalesforce Files Sync; `false` otherwise.	Small, 32.0	32.0
`mimeType`	String	MIME type of the file.	Small, 32.0	32.0
`renditionUrl`	String	URL to the rendition resource for the file. Renditions are processed asynchronously and may not be available immediately after the file has been uploaded.	Small, 32.0	32.0
`renditionUrl 240By180`	String	URL to the 240x180 size rendition resource for the file. Renditions are processed asynchronously and may not be available immediately after the file has been uploaded.	Small, 32.0	32.0
`renditionUrl 720By480`	String	URL to the 720x480 size rendition resource for the file. Renditions are processed asynchronously and may not be available immediately after the file has been uploaded.	Small, 32.0	32.0
`repository FileUrl`	String	The URL of the Chatter REST API resource giving access to the external repository file properties, or `null` if this file doesn't point to an external file.	Small, 32.0	32.0
`textPreview`	String	Text preview of the file if available, `null` otherwise. The maximum number of characters is 200.	Big, 32.0	32.0
`thumb120By90 RenditionStatus`	String	The status of the rendering of the 120x90 pixel sized preview image of the file. Should be either Processing, Failed, Success, or Na if unavailable.	Medium, 32.0	32.0
`thumb240By180 RenditionStatus`	String	The status of the rendering of the 240x180 pixel sized preview image of the file. Should be either Processing, Failed, Success, or Na if unavailable.	Medium, 32.0	32.0
`thumb720By480 RenditionStatus`	String	The status of the rendering of the 720x480 pixel sized preview image of the file. Should be either Processing, Failed, Success, or Na if unavailable.	Medium, 32.0	32.0

Property Name	Type	Description	Filter Group and Version	Available Version
`title`	String	Title of the file.	Small, 32.0	32.0
`versionId`	String	Version ID of the file.	Medium, 32.0	32.0

Conversation Detail

Property	Type	Description	Filter Group and Version	Available Version
`conversationId`	String	18-character ID of the conversation	Medium, 29.0	23.0
`conversationUrl`	String	URL to the conversation detail page	Small, 29.0	23.0
`members`	User Summary[]	Collection of users participating in this conversation	Small, 29.0	23.0
`messages`	Message Collection	Messages in this conversation	Small, 29.0	23.0
`read`	Boolean	Does the context user have unread messages in this conversation?	Small, 29.0	23.0

Conversation Summary

Property	Type	Description	Filter Group and Version	Available Version
`id`	String	18-character ID of the conversation	Small, 29.0	23.0
`latestMessage`	Message	The most recent message in this conversation	Small, 29.0	23.0
`members`	User Summary[]	Collection of users participating in the conversation	Small, 29.0	23.0
`read`	Boolean	Does the context user have unread messages in this conversation?	Small, 29.0	23.0
`url`	String	URL to the conversation detail page	Small, 29.0	23.0

Conversation Summary Collection

Property	Type	Description	Filter Group and Version	Available Version
`conversations`	Conversation Summary[]	Collection of zero or more conversations	Small, 29.0	23.0

Property	Type	Description	Filter Group and Version	Available Version
`currentPageUrl`	String	URL to the current page of conversations	Medium, 29.0	23.0
`nextPageUrl`	String	URL to the next page of messages, that is, older conversations	Small, 29.0	23.0

Conversation Unread Count

Property	Type	Description	Filter Group and Version	Available Version
`hasMore`	Boolean	`true` if the actual number of unread conversations is greater than 50, `false` otherwise	Small, 29.0	23.0
`unreadCount`	Integer	Number of unread conversations, that is, conversations that have unread messages. Is an exact value up to the maximum of 50, and is capped at 50 if the value is greater.	Small, 29.0	23.0

Dashboard Component Snapshot

Represents both dashboard component snapshots and alerts you receive when a dashboard component value crosses a threshold.

Property Name	Type	Description	Filter Group and Version	Available Version
`componentId`	String	18-character ID of the dashboard component.	Small, 32.0	32.0
`componentName`	String	The dashboard component name.	Small, 32.0	32.0
`dashboardBodyText`	String	Display this text next to the actor in the feed element.Use this text in place of the default body text.	Small, 32.0	32.0
`dashboardId`	String	18-character ID of the dashboard.	Small, 32.0	32.0
`dashboardName`	String	The name of the dashboard.	Small, 32.0	32.0
`fullSizeImageUrl`	String	The source URL to retrieve the full-size image of a snapshot. Access this URL with OAuth credentials.	Small, 32.0	32.0
`lastRefreshDate`	Date	ISO-8601 formatted date specifying when this dashboard component was last refreshed.	Small, 32.0	32.0

Property Name	Type	Description	Filter Group and Version	Available Version
`lastRefreshDateDisplayText`	String	Display text for the last refresh date, for example, "Last Refreshed on October 31, 2013."	Small, 32.0	32.0
`runningUser`	User Summary	The running user of the dashboard at the time the snapshot was posted. This value may be `null`. Each dashboard has a running user, whose security settings determine which data to display in a dashboard.	Small, 32.0	32.0
`thumbnailUrl`	String	The source URL to retrieve the thumbnail image of a snapshot. Access this URL with OAuth credentials.	Small, 32.0	32.0

Dashboard Component Snapshot Capability

If a feed element has this capability, it has a dashboard component snapshot. A snapshot is a static image of a dashboard component at a specific point in time.

Property Name	Type	Description	Filter Group and Version	Available Version
`dashboardComponentSnapshot`	Dashboard Component Snapshot	The dashboard component snapshot.	Small, 32.0	32.0

Datacloud Company

The Data.com company record. If you own a company record, you see all the fields. If you don't own the company record, some fields are hidden by asterisks (***).

Property Name	Type	Description	Filter Group and Version	Available Version
`activeContacts`	Integer	The number of active contacts that the company has.	Small, 32.0	32.0
`address`	ConnectApi.Address	The address of the company.	Small, 32.0	32.0
`annualRevenue`	Double	The amount of money that the company makes in one year.	Small, 32.0	32.0
`companyId`	String	The unique numerical identifier for the company.	Small, 32.0	32.0

Property Name	Type	Description	Filter Group and Version	Available Version
`description`	String	The description of the company.	Small, 32.0	32.0
`dunsNumber`	String	A randomly generated nine-digit number that's assigned by Dun & Bradstreet (D&B) to identify unique business establishments.	Small, 32.0	32.0
`industry`	String	A description for the area of focus in which the company does business.	Small, 32.0	32.0
`isInactive`	Boolean	Whether the company information is no longer accurate (true) or the company information is up-to-date (false).	Small, 32.0	32.0
`isOwned`	Boolean	Whether the company has been purchased by you or your organization (`true`) or not (`false`).	Small, 32.0	32.0
`naicsCode`	String	North American Industry Classification System (NAICS) codes were created to provide details about a business's service orientation. The code descriptions are focused on what a business does.	Small, 32.0	32.0
`naicsDescription`	String	A description of the NAICS classification.	Small, 32.0	32.0
`name`	String	The company's name.	Small, 32.0	32.0
`numberOfEmployees`	Integer	The number of employees who are working for the company.	Small, 32.0	32.0
`ownership`	String	The type of ownership of the company: • Public • Private • Government • Other	Small, 32.0	32.0
`phoneNumbers`	ConnectApi.PhoneNumber	A numerical string that contains a corporate telephone number for the company.	Small, 32.0	32.0
`sic`	String	Standard Industrial Codes (SIC) is a numbering convention that indicates what type of service a business provides.	Small, 32.0	32.0

Property Name	Type	Description	Filter Group and Version	Available Version
`sicDescription`	String	A description of the SIC classification.	Small, 32.0	32.0
`site`	String	The type of location of the company, such as "Headquarters."	Small, 32.0	32.0
`tickerSymbol`	String	The symbol that uniquely identifies companies that are traded on public stock exchanges.	Small, 32.0	32.0
`tradeStyle`	String	The symbol that uniquely identifies companies that are traded on public stock exchanges.	Small, 32.0	32.0
`updatedDate`	Date	The date when the information was last updated for this company.	Small, 32.0	32.0
`website`	String	The URL to the company's website.	Small, 32.0	32.0
`yearStarted`	String	The year when the company was founded.	Small, 32.0	32.0

Datacloud Company Collection

Purchase details for all companies that were purchased for a specific order.

Property Name	Type	Description	Filter Group and Version	Available Version
`companies`	ConnectApi.DatacloudCompany	A list of all companies for a specific order.	Small, 32.0	32.0
`currentPageUrl`	String	The URL to the current page of companies.	Medium, 32.0	32.0
`nextPageUrl`	String	The URL to the next page of companies. If there isn't a next page, value is `null`.	Small, 32.0	32.0
`previousPageUrl`	String	The URL to the previous page of companies. The value is `null` if there isn't a previous page.	Medium, 32.0	32.0
`total`	Integer	Number of companies in the order. Users can set how many companies are displayed on a page. The default page size is 25.	Small, 32.0	32.0

Datacloud Contact

Information about Data.com contacts. If you own a contact, you see all the fields. If you don't own the contact record, some fields are hidden by asterisks (***).

Property Name	Type	Description	Filter Group and Version	Available Version
`address`	ConnectApi.Address[]	A postal address for the company where this contacts works.	Small, 32.0	32.0
`companyId`	String	The unique numerical identifier for the company at which the contact works.	Small, 32.0	32.0
`companyName`	String	The name of the company at which the contact works.	Small, 32.0	32.0
`contactId`	String	The unique numerical identifier for a contact.	Small, 32.0	32.0
`department`	String	The name of the department to which the contact is assigned.	Small, 32.0	32.0
`email`	String	An email address for this contact.	Small, 32.0	32.0
`firstName`	String	The first name of a contact.	Small, 32.0	32.0
`isInactive`	Boolean	A `true` or `false` response. True means that the record is no longer active. False means that the contact is still valid and active.	Small, 32.0	32.0
`isOwned`	Boolean	Specifies whether your organization owns this contact.	Small, 32.0	32.0
`lastName`	String	The last name of a contact.	Small, 32.0	32.0
`level`	String	A human resource label designating a person's level in the company.	Small, 32.0	32.0
`phoneNumbers`	ConnectApi.PhoneNumber[]	A numerical string that contains the direct-dial phone number for the contact.	Small, 32.0	32.0
`title`	String	The job title for a contact.	Small, 32.0	32.0
`updatedDate`	Date	The date when the information for this contacts was updated.	Small, 32.0	32.0

Datacloud Contact Collection

Purchase details for all contacts that were purchased for a specific order.

Property Name	Type	Description	Filter Group and Version	Available Version
`contacts`	ConnectApi.DatacloudContact[]	A list of all the contacts for a specific *`orderId`*.	Small, 32.0	32.0
`currentPageUrl`	String	The current page of contacts.	Medium, 32.0	32.0
`nextPageUrl`	String	The URL to the next page of contacts. If there's only one page of contacts, the value is null.	Small, 32.0	32.0
`previousPageUrl`	String	The URL to the previous page of contacts. The value is null when there's only one page of contacts.	Medium, 32.0	32.0
`total`	Integer	Number of contacts that are associated with this order. The total number of contacts can be greater than the number of contacts that are displayed on a single page.	Small, 32.0	32.0

Datacloud Order

Information about the Data.com records that you purchased and details of the purchase.

Property Name	Type	Description	Filter Group and Version	Available Version
`entityUrl`	String	A URL to a list of purchased records for a specific order.	Small, 32.0	32.0
`id`	String	The `orderId` that identifies a specific order.	Small, 32.0	32.0
`purchaseCount`	Integer	The number of records that were purchased for this order.	Small, 32.0	32.0
`purchaseDate`	Date	The date when records were purchased.	Small, 32.0	32.0
`url`	String	A URL for the order information for a specific order.	Small, 32.0	32.0

Datacloud Purchase Usage

Information about the Data.com purchase usage for an organization. Shows the credits that are used and available for Monthly and Listpool users.

Property Name	Type	Description	Filter Group and Version	Available Version
listpoolCreditsAvailable	Integer	Listpool credits that are available for all users in the Listpool group.	Small, 32.0	32.0
listpoolCreditsUsed	Integer	Listpool credits that were used by all the users in the Listpool group.	Small, 32.0	32.0
monthlyCreditsAvailable	Integer	Monthly credits that are assigned to a specific user.	Small, 32.0	32.0
monthlyCreditsUsed	Integer	Monthly credits that were used by a specific user.	Small, 32.0	32.0

Edit Capability

If a feed element or comment has this capability, it can be edited by users who have permission.

Property Name	Type	Description	Filter Group and Version	Available Version
`isEdit Restricted`	Boolean	Specifies whether editing this feed element or comment is restricted. If `true`, the context user can't edit this feed element or comment. If `false`, the context user may or may not have permission to edit this feed element or comment. To determine if a feed element or comment is editable by the context user, use the Feed Elements, Editability or Comment Capability, Editability resource.	Small, 34.0	34.0
`isEditable ByMeUrl`	String	The URL to check if the context user is able to edit this feed element or comment.	Medium, 34.0	34.0
`lastEditedBy`	User Detail	Who last edited this feed element or comment.	Small, 34.0	34.0
`lastEditedDate`	Date	The most recent edit date of this feed element or comment.	Small, 34.0	34.0
`latestRevision`	Integer	The most recent revision of this feed element or comment.	Small, 34.0	34.0
`relativeLast EditedDate`	String	Relative last edited date, for example, "2h ago."	Small, 34.0	34.0

Email Address

An email address for an email message sent from a case record in Case Feed.

Property	Type	Description	Filter Group and Version	Available Version
`displayName`	String	Display name for an email address	Small, 29.0	29.0
`emailAddress`	String	Email address	Small, 29.0	29.0

Email Message Capability

If a feed element has this capability, it has an email message from a case.

Property Name	Type	Description	Filter Group and Version	Available Version
`direction`	String	The direction of the email message. Values are: • `Inbound`—An inbound message (sent by a customer). • `Outbound`—An outbound message (sent to a customer by a support agent).	Small, 32.0	32.0
`emailMessageId`	String	The ID of the email message.	Small, 32.0	32.0
`subject`	String	The subject of the email message.	Small, 32.0	32.0
`textBody`	String	The body of the email message.	Big, 32.0	32.0
`toAddresses`	Email Address[]	The To address of the email message.	Small, 32.0	32.0

Enhanced Link Capability

If a feed element has this capability, it has a link that may contain supplemental information like an icon, a title, and a description.

Property Name	Type	Description	Filter Group and Version	Available Version
`description`	String	A description with a 500 character limit.	Small, 32.0	32.0
`icon`	Icon	A icon.	Small, 32.0	32.0
`linkRecordId`	String	A ID associated with the link if the link URL refers to a Salesforce record.	Small, 32.0	32.0
`linkUrl`	String	A link URL to a detail page if available content can't display inline.	Small, 32.0	32.0

Property Name	Type	Description	Filter Group and Version	Available Version
`title`	String	A title to a detail page.	Small, 32.0	32.0

Error Message

Property	Type	Description	Filter Group and Version	Available Version
`errorCode`	String	An error code with information about the error, for example, `INSUFFICIENT_PRIVILEGES`	Small, 29.0	23.0
`message`	String	Description of error	Small, 29.0	23.0

Error with Output

Property	Type	Description	Filter Group and Version	Available Version
`message`	String	Description of the error	Small, 29.0	27.0
`output`	Any response body	The response body described in the requested resource. For example, if a successful request returns a Like response body but an error triggers the Error with Output response body, the value of the `output` property is a Like response body.	Small, 29.0	27.0

Explanation Details

Represents a detailed explanation for some recommendations.

Property	Type	Description	Filter Group and Version	Available Version
`data`	Various	More information about the recommendation. The value of the `type` property determines what is returned. • `UserFollowsSameUsers`—people that both the user and recommended person are following. See User Summary. • `GroupMembersFollowed`—people that the user is following and that are members of the recommended group. See User Summary.	Small, 29.0	24.0

Property	Type	Description	Filter Group and Version	Available Version
		• `UserViewingSameRecords`—records that the user and the recommended person have recently viewed. See Record Summary. • `RecordParentOfFollowed`—records that the user is following and that are child records of the recommended record. See Record Summary.		
`summary`	String	Summary explanation for recommendation.	Small, 29.0	24.0
`type`	String	Indicates the reason for the recommendation. • `Custom`—Custom recommendations • `FilePopular`—Files with many followers or views • `FileViewedTogether`—Files often viewed at the same time as other files the context user views • `FollowedTogetherWithFollowees`—Users often followed with the same people the context user follows • `GroupMembersFollowed`—Groups with members the context user follows • `GroupNew`—Recently created groups • `GroupPopular`—Groups with many active members • `ItemViewedTogether`—Records often viewed at the same time as other records the context user views • `PopularApp`—Applications that are popular • `RecordOwned`—Records owned by the context user • `RecordParentOfFollowed`—Parent records of records the context user follows • `RecordViewed`—Records the context user recently viewed • `UserDirectReport`—Users who report to the context user • `UserFollowedTogether`—Users often followed at the same time as other users the context user follows • `UserFollowsSameUsers`—Users who follow the same users as the context user • `UserManager`—The context user's manager	Small, 29.0	24.0

Property	Type	Description	Filter Group and Version	Available Version
		• `UserNew`—Recently created users • `UserPeer`—Users who report to the same manager as the context user • `UserPopular`—Users with many followers • `UserViewingSameRecords`—Users who view the same records as the context user		

Explanation Summary

Represents a summary explanation for a recommendation.

Property	Type	Description	Filter Group and Version	Available Version
`detailsUrl`	String	URL for explanation details or `null` if the recommendation doesn't have a detailed explanation.	Small, 29.0	24.0
`summary`	String	Summary explanation for recommendation.	Small, 29.0	23.0
`type`	String	Indicates the reason for the recommendation. • `Custom`—Custom recommendations • `FilePopular`—Files with many followers or views • `FileViewedTogether`—Files often viewed at the same time as other files the context user views • `FollowedTogetherWithFollowees`—Users often followed with the same people the context user follows • `GroupMembersFollowed`—Groups with members the context user follows • `GroupNew`—Recently created groups • `GroupPopular`—Groups with many active members • `ItemViewedTogether`—Records often viewed at the same time as other records the context user views • `PopularApp`—Applications that are popular • `RecordOwned`—Records owned by the context user • `RecordParentOfFollowed`—Parent records of records the context user follows	Small, 29.0	24.0

Property	Type	Description	Filter Group and Version	Available Version
		• `RecordViewed`—Records the context user recently viewed • `UserDirectReport`—Users who report to the context user • `UserFollowedTogether`—Users often followed at the same time as other users the context user follows • `UserFollowsSameUsers`—Users who follow the same users as the context user • `UserManager`—The context user's manager • `UserNew`—Recently created users • `UserPeer`—Users who report to the same manager as the context user • `UserPopular`—Users with many followers • `UserViewingSameRecords`—Users who view the same records as the context user		

Favorite

Property	Type	Description	Filter Group and Version	Available Version
`community`	Reference	Information about the community. Returns `null` if favorite exists in default community or communities aren't enabled.	Medium, 29.0	26.0
`createdBy`	User Summary	Information about the original creator of the favorite	Medium, 29.0	24.0
`feedUrl`	String	URL to the feed item for this favorite	Small, 29.0	24.0
`id`	String	18-character ID of the favorite	Small, 29.0	24.0
`lastViewDate`	String	ISO8601 date string, for example, 2011-02-25T18:24:31.000Z	Medium, 29.0	24.0
`name`	String	Name of the favorite	Small, 29.0	24.0
`searchText`	String	If the favorite is from a search, contains the search text, if not, contains `null`.	Medium, 29.0	24.0
`target`	Reference	A reference to the topic if applicable, `null` otherwise	Small, 29.0	28.0
`type`	String	One of the following values: • `ListView` • `Search`	Small, 29.0	24.0

Property	Type	Description	Filter Group and Version	Available Version
		• `Topic`		
`url`	String	URL to this favorite	Small, 29.0	24.0
`user`	User Summary	Information about the user who saved this favorite	Medium, 29.0	24.0

Favorites Page

Property	Type	Description	Filter Group and Version	Available Version
`favorites`	Favorite[]	Collection of one or more favorites	Small, 29.0	24.0
`total`	Integer	Total number of favorites	Small, 29.0	24.0

Features

Features that are available to the context user. If a feature is disabled, the resources associated with that feature are unavailable and some feature-related properties on objects are set to `null`.

Property	Type	Description	Filter Group and Version	Available Version
`chatter`	Boolean	Specifies whether Chatter is enabled for an organization or not	Small, 29.0	27.0
`chatterActivity`	Boolean	Specifies whether the user details include information about Chatter activity	Small, 29.0	24.0
`chatterAnswers`	Boolean	Specifies whether Chatter Answers is enabled	Small, 29.0	29.0
`chatter GlobalInfluence`	Boolean	Specifies whether the user details include global Chatter activity	Small, 29.0	24.0
`chatter GroupRecords`	Boolean	Specifies whether Chatter groups can have records associated with them	Small, 33.0	33.0
`chatter GroupRecord Sharing`	Boolean	Specifies whether Chatter records are implicitly shared among group members when records are added to groups	Small, 33.0	33.0
`chatter Messages`	Boolean	Specifies whether Chatter messages are enabled for the organization	Small, 29.0	24.0
`chatterTopics`	Boolean	Specifies whether Chatter topics are enabled	Small, 29.0	28.0
`communities Enabled`	Boolean	Specifies whether Salesforce Communities is enabled	Small, 31.0	31.0

Property	Type	Description	Filter Group and Version	Available Version
communityModeration	Boolean	Specifies whether community moderation is enabled for this organization	Small, 29.0	29.0
communityReputation	Boolean	Specifies whether reputation is enabled for communities in this organization.	Small, 32.0	32.0
dashboardComponentSnapshots	Boolean	Specifies whether the user can post dashboard component snapshots	Small, 29.0	24.0
defaultCurrencyIsoCode	String	The ISO code of the default currency. Applicable only when multiCurrency is false.	Small, 29.0	28.0
feedPolling	Boolean	Reserved for future use.	Small, 29.0	24.0
files	Boolean	Specifies whether files can act as resources for Chatter REST API	Small, 29.0	24.0
filesOnComments	Boolean	Specifies whether files can be attached to comments	Small, 29.0	24.0
groupsCanFollow	Boolean	Reserved for future use	Small, 29.0	28.0–29.0
ideas	Boolean	Specifies whether Ideas is enabled	Small, 29.0	29.0
managedTopicsEnabled	Boolean	Indicates access to the community home feed and the managed topic feed	Small, 32.0	32.0
mobileNotificationsEnabled	Boolean	Reserved for future use	Small, 29.0	29.0
multiCurrency	Boolean	Indicates whether the user's organization uses multiple currencies (true) or not (false). When false, the defaultCurrencyIsoCode indicates the ISO code of the default currency.	Small, 29.0	28.0
publisherActions	Boolean	Indicates whether actions in the publisher are enabled	Small, 29.0	28.0
storeDataOnDevicesEnabled	Boolean	Indicates whether the Salesforce1 downloadable apps can use secure, persistent storage on mobile devices to cache data.	Small, 30.0	30.0
trendingTopics	Boolean	Specifies whether trending topics are enabled	Small, 29.0	24.0
viralInvitesAllowed	Boolean	Specifies whether existing Chatter users can invite people in their company to use Chatter	Small, 29.0	24.0

Feed

Property	Type	Description	Filter Group and Version	Available Version
`feedElementPostUrl`	String	URL for posting feed elements to this subject When returned by a request that includes the HTTP header `X-Connect-Bearer-Urls: true`, the URL includes a bearer token. Use this bearer token URL to make POST requests.	Small, 31.0	31.0
`feedElementsUrl`	String	URL of feed elements	Small, 31.0	31.0
`feedItemsUrl`	String	URL of feed items	Small, 29.0	23.0–31.0
`isModifiedUrl`	String	Reserved for future use.	Small, 29.0	23.0

Feed Directory

Property	Type	Description	Filter Group and Version	Available Version
`favorites`	Favorite[]	Collection of favorites that the context user has saved.	Small, 29.0	24.0
`feeds`	Feed Directory Item[]	Collection of feed directory items that the context user is allowed to view	Small, 29.0	23.0

Feed Directory Item

The definition of a feed.

Property	Type	Description	Filter Group and Version	Available Version
`feedElementsUrl`	String	URL to the resource of the feed elements for a specific feed. For example: `/chatter/feeds/files/me/feed-elements`	Small, 31.0	31.0
`feedItemsUrl`	String	URL to the resource of the feed items for a specific feed. For example: `/chatter/feeds/files/me/feed-elements`	Small, 29.0	23.0–31.0
`feedType`	String	The feed type. One of these values: • `Bookmarks`—Contains all feed items saved as bookmarks by the context user.	Big, 30.0	30.0

Property	Type	Description	Filter Group and Version	Available Version
		• `Company`—Contains all feed items except feed items of type `TrackedChange`. To see the feed item, the user must have sharing access to its parent. • `Files`—Contains all feed items that contain files posted by people or groups that the context user follows. • `Filter`—Contains the news feed filtered to contain feed items whose parent is a specified object type. • `Groups`—Contains all feed items from all groups the context user either owns or is a member of. • `Home`—Contains all feed items associated with any managed topic in a community. • `Moderation`—Contains all feed items that have been flagged for moderation. The Communities Moderation feed is available only to users with "Moderate Community Feeds" permissions. • `News`—Contains all updates for people the context user follows, groups the user is a member of, and files and records the user is following. Also contains all updates for records whose parent is the context user and every feed item and comment that mentions the context user or that mentions a group the context user is a member of. • `People`—Contains all feed items posted by all people the context user follows. • `Record`—Contains all feed items whose parent is a specified record, which could be a group, user, object, file, or any other standard or custom object. When the record is a group, the feed also contains feed items that mention the group. When the record is a user, the feed contains only feed items on that user. • `To`—Contains all feed items with mentions of the context user, feed items the context user commented on, and feed items created by the context user that are commented on. • `Topics`—Contains all feed items that include the specified topic. • `UserProfile`—Contains feed items created when a user changes records that can be tracked in a feed, feed items whose parent is the user, and feed items that @mention the user. This feed is		

Property	Type	Description	Filter Group and Version	Available Version
		different than the news feed, which returns more feed items, including group updates.		
`feedUrl`	String	URL to the resource of a specific feed. For example: `/services/data/v34.0/chatter/feeds/files/me`	Small, 29.0	23.0
`keyPrefix`	String	For filter feeds, this value is the key prefix associated with the entity type used to filter this feed. All feed items in this feed have a parent whose entity type matches this key prefix value. For non-filter feeds, this value is `null`. A *key prefix* is the first three characters of a record ID, which specifies the entity type.	Big, 30.0	30.0
`label`	String	Name of the feed, such as, Groups, People, Company, and so on	Small, 29.0	23.0

Feed Element Capabilities

A container for all capabilities that can be included with a feed element.

In API versions 30.0 and earlier, every feed item can have comments, likes, topics, and so on. In versions 31.0 and later, every feed item (and feed element) can have a unique set of *capabilities*. If a capability property exists on a feed element, that capability is available, even if the capability property doesn't have a value. For example, if the `ChatterLikes` capability property exists on a feed element (with or without a value), the context user can like that feed element. If the capability property doesn't exist, it isn't possible to like that feed element. A capability can also contain associated data. For example, the `Moderation` capability contains data about moderation flags.

Property Name	Type	Description	Filter Group and Version	Available Version
`approval`	Approval Capability	If a feed element has this capability, it includes information about an approval.	Small, 32.0	32.0
`associatedActions`	Associated Actions Capability	If a feed element has this capability, it has platform actions associated with it.	Small, 31.0	31.0
`banner`	Banner Capability	If this feed element has this capability, it has a banner motif and style.	Small, 31.0	31.0
`bookmarks`	Bookmarks Capability	If a feed element has this capability, the context user can bookmark it.	Small, 31.0	31.0

Property Name	Type	Description	Filter Group and Version	Available Version
`bundle`	Generic Bundle Capability or Tracked Change Bundle Capability	If a feed element has this capability, the feed element may be a group of other feed elements condensed into one.	Small, 31.0	31.0
`canvas`	Canvas Capability	If a feed element has this capability, it renders a canvas app.	Small, 32.0	32.0
`caseComment`	Case Comment Capability	If a feed element has this capability, it has a case comment on the case feed.	Small, 32.0	32.0
`chatterLikes`	Chatter Likes Capability	If a feed element has this capability, the context user can like it. Exposes information about existing likes.	Small, 31.0	31.0
`comments`	Comments Capability	If a feed element has this capability, the context user can add a comment.	Small, 31.0	31.0
`content`	Content Capability	If a feed element has this capability, it has a file attachment. Most Content Capability properties are null if the content has been deleted from the feed element or if the access has changed to private.	Small, 32.0	32.0
`dashboardComponentSnapshot`	Dashboard Component Snapshot Capability	If a feed element has this capability, it has a dashboard component snapshot. A snapshot is a static image of a dashboard component at a specific point in time.	Small, 32.0	32.0
`edit`	Edit Capability	If a feed element has this capability, users who have permission can edit it.	Small, 34.0	34.0
`emailMessage`	Email Message Capability	If a feed element has this capability, it has an email message from a case.	Small, 32.0	32.0
`enhancedLink`	Enhanced Link Capability	If a feed element has this capability, it has a link that may contain supplemental information like an icon, a title, and a description.	Small, 32.0	32.0
`link`	Link Capability	If a feed element has this capability, it has a link.	Small, 32.0	32.0
`moderation`	Moderation Capability	If a feed element has this capability, users in the community can flag the feed element for moderation. This response	Small, 31.0	31.0

Property Name	Type	Description	Filter Group and Version	Available Version
		body also includes information about the flagged content.		
`poll`	Poll Capability	If a feed element has this capability, the feed element can include a poll for voting.	Small, 31.0	31.0
`origin`	Origin Capability	If a feed element has this capability, it was created by a feed action.	Small, 33.0	33.0
`questionAndAnswers`	Question and Answers Capability	If a feed element has this capability, it has a question and comments on the feed element are answers to the question.	Small, 31.0	31.0
`recommendations`	Recommendations Capability	If a feed element has this capability, it has a recommendation.	Small, 32.0	32.0
`recordSnapshot`	Record Snapshot Capability	If a feed element has this capability, it contains all the snapshotted fields of a record for a single create record event.	Small, 32.0	32.0
`topics`	Topics Capability	If a feed element has this capability, the context user can add topics to it. Topics help users organize and discover conversations.	Small, 31.0	31.0
`trackedChanges`	Tracked Changes Capability	If a feed element has this capability, it contains all changes to a record for a single tracked change event.	Small, 32.0	32.0

Feed Element Page

Container of feed elements with paginated results.

Property Name	Type	Description	Filter Group and Version	Available Version
`currentPageUrl`	String	Chatter REST API URL identifying the current page.	Medium, 31.0	31.0
`elements`		Collection of feed elements. A feed element can be any of these: • Generic Feed Element • Feed Item	Small, 31.0	31.0
`isModifiedToken`	String	Reserved for future use.	Medium, 31.0	31.0
`isModifiedUrl`	String	Reserved for future use.	Medium, 31.0	31.0

Property Name	Type	Description	Filter Group and Version	Available Version
`nextPageToken`	String	Token identifying the next page or `null` if there isn't a next page.	Small, 31.0	31.0
`nextPageUrl`	String	Chatter REST API URL identifying the next page or `null` if there isn't a next page. Check whether this value is `null` before getting another page. If a page doesn't exist, a `ConnectApi.NotFoundException` error is returned.	Small, 31.0	31.0
`updatesToken`	String	Token to use in an *`updatedSince`* parameter, or `null` if not available. An opaque polling token that describes when a news feed was last modified. Use this as the value in the *`updatedSince`* request parameter of the `/chatter/feeds/news/me/feed-elements` resource.	Big, 31.0	31.0
`updatesUrl`	String	A `/chatter/feeds/news/me/feed-elements` resource with the feed elements updated since the feed was refreshed. Value is `null` if the feed doesn't support updates polling.	Big, 31.0	31.0

Feed Entity Is Editable

Indicates whether the context user can edit a feed element or comment.

Property Name	Type	Description	Filter Group and Version	Available Version
`feedEntityUrl`	String	URL of the feed element or comment.	Small, 34.0	34.0
`isEditableByMe`	Boolean	`true` if the context user can edit the feed element or comment, `false` otherwise.	Small, 34.0	34.0

Feed Item

A feed is made up of feed elements. A feed item is a type of feed element.

Property	Type	Description	Filter Group and Version	Available Version
`actor`	Depends on parent	The entity that created the feed item. Values are: • Record Summary—if a record (23.0) • User Summary—if a user (23.0) • Unauthenticated User—if a Chatter customer (24.0) Note: If the actor is a user, a User Summary response body is returned, not a Record Summary of type user.	Small, 29.0	
`attachment`	Depends on type	Depends on the `type` value of the attachment. Values include: • `Approval`—Feed Item Attachment: Approval (v27.0–31.0) • `BasicTemplate`—Feed Item Attachment: Basic Template (v27.0–31.0) • `Canvas`—Feed Item Attachment: Canvas Template (v29.0–31.0) • `CaseComment`—Feed Item Attachment: Case Comment (v24.0–31.0) • `Content`—Feed Item Attachment: Content or `null` if the content is unavailable (v23.0–31.0) • `DashboardComponent`—Feed Item Attachment: Dashboard (v24.0–31.0) • `EmailMessage`—Feed Item Attachment: Email Message (v29.0–31.0) • `Link`—Feed Item Attachment: Link (v23.0–31.0) • `Poll`—Feed Item Attachment: Poll (v26.0–31.0) • `RecordSnapshot`—Feed Item Attachment: Record Snapshot (v29.0–31.0) • `TrackedChange`—Feed Item Attachment: Tracked Change (v29.0–31.0) Important: As of API version 32.0, use the `capabilities` property.	Medium, 29.0	
`body`	Feed Item Body	Information about the feed item. In API versions to 28.0, use the `/body/text` property as the default case for rendering text. In API versions 29.0–30.0, use the `/preamble/text` property as the default case for rendering text. In API versions 31.0 and later, use the `/header/text` property as the default case for rendering text.	Small, 29.0	23.0

Property	Type	Description	Filter Group and Version	Available Version
`capabilities`	Feed Element Capabilities	Auxiliary information for the feed element. The feed element's characteristics are made up of these capabilities.	Medium, 31.0	31.0
`canShare`	Boolean	`true` if feed item can be shared, `false` otherwise	Medium, 29.0	27.0
`clientInfo`	ClientInfo	Information about the connected app used to authenticate the connection	Medium, 29.0	23.0
`comments`	Comment Page	Comments for this feed item. **Important:** As of API version 32.0, use the `capabilities.comments.page` property.	Small, 29.0	23.0–31.0
`createdDate`	String	ISO8601 date string, for example, 2011-02-25T18:24:31.000Z	Small, 29.0	23.0
`currentUserLike`	Reference	**Note:** This property is no longer available as of version 24.0. Use the `myLike` property instead. A reference to a like object if applicable, `null` otherwise.		23.0–24.0
`event`	Boolean	`true` if feed item created due to an event change, `false` otherwise	Medium, 29.0	23.0
`feedElementType`	String	Feed elements are the top-level objects that a feed contains. The feed element type describes the characteristics of that feed element. One of these values: • `Bundle`—A container of feed elements. A bundle also has a body made up of message segments that can always be gracefully degraded to text-only values. • `FeedItem`—A feed item has a single parent and is scoped to one community or across all communities. A feed item can have capabilities such as bookmarks, canvas, content, comment, link, poll. Feed items have a body made up of message segments that can always be gracefully degraded to text-only values. • `Recommendation`—A recommendation is a feed element with a recommendations capability. A recommendation suggests records to follow, groups to join, or applications that are helpful to the context user.	Small, 31.0	31.0
`header`	Feed Item Body	The header is the title of the post. This property contains renderable plain text for all the segments of the	Medium, 31.0	31.0

Property	Type	Description	Filter Group and Version	Available Version
		message. If a client doesn't know how to render a feed element type, it should render this text.		
`id`	String	18-character ID of the feed item	Small, 29.0	23.0
`isBookmarkedByCurrentUser`	Boolean	`true` if the current user has bookmarked this feed item, `false` otherwise. **Important:** As of API version 32.0, use the `capabilities.bookmarks.isBookmarkedByCurrentUser` property.	Medium, 29.0	24.0–31.0
`isDeleteRestricted`	Boolean	Returns `true` if a feed item cannot be deleted by the context user. If it returns `false`, possibly the context users can delete the feed item.	Medium, 29.0	25.0
`isLikedByCurrentUser`	Boolean	`true` if the current user has liked this feed item, `false` otherwise. **Important:** As of API version 32.0, use the `capabilities.chatterLikes.isLikedByCurrentUser` property.	Medium, 29.0	23.0–31.0
`likes`	Like Page	Likes for this feed item, **Important:** As of API version 32.0, use the `capabilities.chatterLikes.page` property.	Medium, 29.0	23.0–31.0
`likesMessage`	Like Message Body	A message body that describes who likes the feed item. **Important:** As of API version 32.0, use the `capabilities.chatterLikes.likesMessage` property.	Small, 29.0	27.0–31.0
`modifiedDate`	String	ISO8601 date string, for example, 2011-02-25T18:24:31.000Z	Small, 29.0	23.0
`moderationFlags`	Moderation Flags	Flags for this feed item. Community moderators can view and take action on flagged items. If the `communityModeration` property of the Features response body is `false`, this property is `null`. **Important:** As of API version 32.0, use the `capabilities.moderation.moderationFlags` property.	Big, 29.0	29.0–31.0

Property	Type	Description	Filter Group and Version	Available Version
`myLike`	Reference	If the context user has liked the comment, this property is a reference to the specific like, `null` otherwise. **Important:** As of API version 32.0, use the `capabilities.chatterLikes.myLike` property.	Medium, 29.0	23.0–31.0
`originalFeedItem`	Reference	A reference to the original feed item if this feed item is a shared feed item, `null` otherwise	Small, 29.0	24.0
`originalFeedItemActor`	Depends on type	Value depends on type of original feed item. Values are: • Record Summary–if original feed item is a record • User Summary–if original feed item is a user • Unauthenticated User—if original feed item is from a Chatter customer • `null` if this feed item is not a shared feed item	Medium, 29.0	24.0
`parent`	Depends on type	Value depends on type of parent. Values are: • File Summary–if parent is a file • Group–if parent is a group • Record Summary–if parent is a record • User Summary–if parent is a user **Note:** If the parent is a user, a User Summary response body is returned, not a Record Summary of type user. The same applies for both groups and files.	Medium, 29.0	23.0
`photoUrl`	String	URL of the photo associated with the feed item	Medium, 29.0	23.0
`preamble`	Feed Item Preamble Message Body	The preamble is the title of the post. This property contains renderable plain text for all the segments of the message, for example, “App Builders - Pam Jones”. In API versions to 28.0, use the `/body/text` property as the default case for rendering text. In API versions 29.0–30.0, use the `/preamble/text` property as the default case for rendering text. In API versions 31.0 and later, use the `/header/text` property as the default case for rendering text.	Medium, 29.0	27.0–30.0
`relativeCreatedDate`	String	ISO8601 date string, for example, 2011-02-25T18:24:31.000Z	Small, 29.0	28.0

Property	Type	Description	Filter Group and Version	Available Version
`topics`	Feed Item Topics	Topics for this feed item. **Important:** As of API version 32.0, use the `capabilities.topics.items` property.	Medium, 29.0	28.0–31.0
`type`	String	The `type` property always returns all types, regardless of API version. However, not all types are valid with all API versions. **Important:** As of API version 32.0, use the `capabilities` property to determine what can be done with a feed item. See Capabilities. Values are: • `ActivityEvent`—Feed item generated in Case Feed when an event or task associated with a parent record with a feed enabled is created or updated. • `AdvancedTextPost`—A feed item with advanced text formatting, such as a group announcement post. • `ApprovalPost`—Feed item with an approval capability. Approvers can act on the feed item parent. • `AttachArticleEvent`—Feed item generated when an article is attached to a case in Case Feed. • `BasicTemplateFeedItem`—Feed item with an enhanced link capability. • `CallLogPost`—Feed item generated when a call log is saved to a case in Case Feed. • `CanvasPost`—Feed item generated by a canvas app in the publisher or from Chatter REST API or Chatter in Apex. The post itself is a link to a canvas app. • `CaseCommentPost`—Feed item generated when a case comment is saved in Case Feed. • `ChangeStatusPost`—Feed item generated when the status of a case is changed in Case Feed. • `ChatTranscriptionPost`—Feed item generated in Case Feed when a Live Agent chat transcript is saved to a case. • `CollaborationGroupCreated`—Feed item generated when a new public group is created. Contains a link to the new group.	Small, 29.0	23.0

Property	Type	Description	Filter Group and Version	Available Version
		• `CollaborationGroupUnarchived`—Deprecated. Feed item generated when an archived group is activated. • `ContentPost`—Feed item with a content capability. • `CreateRecordEvent`—Feed item that describes a record created in the publisher. • `DashboardComponentAlert`—Feed item with a dashboard alert. • `DashboardComponentSnapshot`—Feed item with a dashboard component snapshot capability. • `EmailMessageEvent`—Feed item generated when an email is sent from a case in Case Feed. • `FacebookPost`—Deprecated. Feed item generated when a Facebook post is created from a case in Case Feed. • `LinkPost`—Feed item with a link capability. • `MilestoneEvent`—Feed item generated when a case milestone is either completed or reaches a violation status. Contains a link to the case milestone. • `PollPost`—Feed item with a poll capability. Viewers of the feed item are allowed to vote on the options in the poll. • `ProfileSkillPost`—Feed item generated when a skill is added to a user's profile. • `QuestionPost`—Feed item generated when a question is asked. As of API version 33.0, a feed item of this type can have a content capability and a link capability. • `ReplyPost`—Feed item generated by a Chatter Answers reply. • `RypplePost`—Feed item generated when a user posts thanks. • `SocialPost`—Feed item generated when a social post is created from a case in Case Feed. • `TextPost`—Feed item containing text only. • `TrackedChange`—Feed item created when one or more fields on a record have been changed.		

Property	Type	Description	Filter Group and Version	Available Version
		• `UserStatus`—Deprecated. A user's post to their own profile.		
`url`	String	URL to this feed item	Small, 29.0	23.0
`visibility`	String	Who can see this feed item. Values are: • `InternalUsers`—If Communities is enabled, only members of the default, internal community can see the feed item. • `AllUsers`—If Communities is enabled, all members of the community can see the feed item.	Small, 29.0	26.0

Feed Item Attachment: Approval

An approval attachment.

Important: This response body isn't available in version 32.0 and later. In version 32.0 and later, Approval Capability is used.

Property	Type	Description	Filter Group and Version	Available Version
`id`	String	The work item ID	Small, 29.0	27.0–31.0
`processInstanceStepId`	String	The process instance step ID. The associated record represents one step in an approval process.	Small, 30.0	30.0–31.0
`status`	String	The status of the approval. Values are: • `Approved` • `Fault` • `Held` • `NoResponse` • `Pending` • `Reassigned` • `Rejected` • `Removed` • `Started`	Small, 29.0	27.0–31.0
`postTemplateFields`	Approval Post Template Field[]	Collection of approval post template fields	Small, 29.0	28.0–31.0
`type`	String	`Approval`	Small, 29.0	27.0–31.0

Feed Item Attachment: Basic Template

Important: This response body isn't available in version 32.0 and later. In version 32.0 and later, Enhanced Link Capability is used.

Property	Type	Description	Filter Type and Version	Available Version
`description`	String	An optional description with a 500 character limit	Small, 29.0	27.0–31.0
`icon`	Icon	An optional icon	Small, 29.0	27.0–31.0
`linkRecordId`	String	If `linkURL` refers to a Salesforce record, `linkRecordId` contains the ID of the record.	Small, 29.0	28.0–31.0
`linkUrl`	String	An optional link URL to a detail page if available content can't display inline	Small, 29.0	27.0–31.0
`subtype`	String	The subtype associated with the basic template feed item. Value is: • `CollaborationGroupCreated`	Big, 29.0	27.0 only This property is deprecated in 28.0 and always returns `None` for clients with older versions.
`title`	String	An optional title to a detail page	Small, 29.0	27.0–31.0
`type`	String	`BasicTemplate`	Small, 29.0	27.0–31.0

Feed Item Attachment: Canvas Template

Important: This response body isn't available in version 32.0 and later. In version 32.0 and later, Canvas Capability is used.

Property	Type	Description	Filter Type and Version	Available Version
`description`	String	Optional. Description of the canvas app. The maximum length of this field is 500 characters.	Small, 29.0	29.0–31.0
`developerName`	String	The developer name (API name) of the canvas app	Small, 29.0	29.0–31.0
`height`	String	Optional. The height of the canvas app in pixels. Default height is 200 pixels.	Small, 29.0	29.0–31.0
`icon`	Icon	The canvas app icon	Small, 29.0	29.0–31.0
`namespacePrefix`	String	Optional. The namespace prefix of the Developer Edition organization in which the canvas app was created.	Small, 29.0	29.0–31.0

Property	Type	Description	Filter Type and Version	Available Version
`parameters`	String	Optional. Parameters passed to the canvas app in JSON format. Example: `{'isUpdated'='true'}`	Small, 29.0	29.0–31.0
`thumbnailUrl`	String	Optional. The URL to a thumbnail image for the canvas app. Maximum dimensions are 120x120 pixels.	Small, 29.0	29.0–31.0
`title`	String	The title of the link used to call the canvas app	Small, 29.0	29.0–31.0
`type`	String	`Canvas`	Small, 29.0	29.0–31.0

Feed Item Attachment: Case Comment

Important: This response body isn't available in version 32.0 and later. In version 32.0 and later, Case Comment Capability is used.

Property	Type	Description	Filter Group and Version	Available Version
`actorType`	String	Specifies the type of user who made the comment. Valid values are: • `Customer`—if a Chattercustomer made the comment • `CustomerService`—if a service representative made the comment	Small, 29.0	24.0–31.0
`createdBy`	User Summary	Information about the user who created the comment	Small, 29.0	24.0–31.0
`createdDate`	String	ISO8601 date string, for example, 2011-02-25T18:24:31.000Z	Small, 29.0	24.0–31.0
`eventType`	String	Specifies the event type of a comment in Case Feed One of these values: • `NewInternal`—A case comment that has newly been marked Internal Only. • `NewPublished`—A newly published case comment. • `NewPublishedByCustomer`—A case comment by a customer that was newly published. • `PublishExisting`—An existing case comment that was republished. • `PublishExistingByCustomer`—An existing case comment by a customer that was republished.	Small, 29.0	29.0–31.0

Property	Type	Description	Filter Group and Version	Available Version
		• `UnpublishExistingByCustomer`—An existing case comment by a customer that was unpublished. • `UnpublishExsiting`—An existing case comment that was unpublished. Note: Unfortunately, this typo is in the code, not the documentation. Use this spelling in your code.		
`id`	String	18-character ID of case comment	Small, 29.0	24.0–31.0
`published`	Boolean	Specifies whether the comment has been published	Small, 29.0	24.0–31.0
`text`	String	Text of the case comment	Small, 29.0	24.0–31.0
`type`	String	`CaseComment`	Small, 29.0	27.0–31.0

Feed Item Attachment: Content

Description of a feed item content attachment.

Important: This response body isn't available in version 32.0 and later. In version 32.0 and later, Content Capability is used.

Property	Type	Description	Filter Group and Version	Available Version
`checkSum`	String	MD5 checksum for the file	Small, 29.0	26.0–31.0
`contentUrl`	String	URL for link files and Google Docs; otherwise the value is `null`.	Small, 31.0	31.0–31.0
`description`	String	Description of the attachment	Small, 29.0	23.0–31.0
`downloadUrl`	String	File's URL. This value is `null` if the content is a link or a Google Doc.	Small, 29.0	23.0–31.0
`fileExtension`	String	Extension of the file	Small, 29.0	26.0–31.0
`fileSize`	String	Size of the file in bytes. If size cannot be determined, returns `unknown`.	Small, 29.0	23.0–31.0
`fileType`	String	Type of file	Small, 29.0	23.0–31.0
`hasImagePreview`	Boolean	`true` if the file has a preview image available, `false` otherwise	Small, 29.0	23.0–29.0
`hasPdfPreview`	Boolean	`true` if the file has a PDF preview available, `false` otherwise	Small, 29.0	23.0–31.0

Property	Type	Description	Filter Group and Version	Available Version
`id`	String	18-character ID of the content	Small, 29.0	23.0–31.0
`isInMyFileSync`	Boolean	`true` if the file is synced withSalesforce Files Sync; `false` otherwise.	Small, 29.0	26.0–31.0
`mimeType`	String	MIME type content of the file	Small, 29.0	23.0–31.0
`renditionUrl`	String	URL to the rendition resource for the file. For shared files, renditions process asynchronously after upload. For private files, renditions process when the first file preview is requested, and aren't available immediately after the file is uploaded.	Small, 29.0	26.0–31.0
`renditionUrl 240By180`	String	URL to the 240 x 180 rendition resource for the file. For shared files, renditions process asynchronously after upload. For private files, renditions process when the first file preview is requested, and aren't available immediately after the file is uploaded.	Small, 30.0	30.0–31.0
`renditionUrl 720By480`	String	URL to the 720 x 480 rendition resource for the file. For shared files, renditions process asynchronously after upload. For private files, renditions process when the first file preview is requested, and aren't available immediately after the file is uploaded.	Small, 30.0	30.0–31.0
`textPreview`	String	Text preview of the file if available, `null` otherwise.	Big, 30.0	30.0–31.0
`thumb120By90 RenditionStatus`	String	Specifies the rendering status of the 120 x 90 preview image of the file. One of these values: • `Processing`—Image is being rendered. • `Failed`—Rendering process failed. • `Success`—Rendering process was successful. • `Na`—Rendering is not available for this image.	Medium, 30.0	30.0–31.0
`thumb240By180 RenditionStatus`	String	Specifies the rendering status of the 240 x 180 preview image of the file. One of these values: • `Processing`—Image is being rendered. • `Failed`—Rendering process failed. • `Success`—Rendering process was successful. • `Na`—Rendering is not available for this image.	Medium, 30.0	30.0–31.0
`thumb720By480 RenditionStatus`	String	Specifies the rendering status of the 720 x 480 preview image of the file. One of these values: • `Processing`—Image is being rendered. • `Failed`—Rendering process failed. • `Success`—Rendering process was successful.	Medium, 30.0	30.0–31.0

Property	Type	Description	Filter Group and Version	Available Version
		• `Na`—Rendering is not available for this image.		
`title`	String	Title of the file	Small, 29.0	23.0–31.0
`type`	String	`Content`	Small, 29.0	27.0–31.0
`versionId`	String	18-character ID of this version of the content	Medium, 29.0	23.0–31.0

Feed Item Attachment: Dashboard

Important: This response body isn't available in version 32.0 and later. In version 32.0 and later, Dashboard Component Snapshot Capability is used.

Property	Type	Description	Filter Group and Version	Available Version
`componentId`	String	18-character ID of the component	Small, 29.0	24.0–31.0
`componentName`	String	Name of the component. If no name is saved with the component, returns the localized string, "Untitled Component."	Small, 29.0	24.0–31.0
`dashboardBodyText`	String	Text that should be displayed next to the actor in the body of a feed item. This is used instead of the default body text. If no text is specified, and there is no default body text, returns null.	Small, 29.0	24.0–31.0
`dashboardId`	String	18-character ID of the dashboard	Small, 29.0	24.0–31.0
`dashboardName`	String	Name of the dashboard	Small, 29.0	24.0–31.0
`fullSizeImageUrl`	String	URL of the full-sized dashboard image	Small, 29.0	24.0–31.0
`lastRefreshDate`	String	ISO8601 date string, for example, 2011-02-25T18:24:31.000Z of when this dashboard was last refreshed	Small, 29.0	24.0–31.0
`lastRefreshDateDisplayText`	String	The text of the last refresh date to be displayed, such as, "Last refreshed on October 31, 2011."	Small, 29.0	24.0–31.0
`runningUser`	User Summary	The user running the dashboard	Small, 29.0	24.0–31.0
`thumbnailUrl`	String	URL of the thumbnail-sized dashboard image	Small, 29.0	24.0–31.0
`type`	String	`DashboardComponent`	Small, 29.0	27.0–31.0

Feed Item Attachment: Email Message

An email message sent from a case record in Case Feed.

Important: This response body isn't available in version 32.0 and later. In version 32.0 and later, Email Message Capability is used.

Property	Type	Description	Filter Group and Version	Available Version
`direction`	String	The direction of the email message. One of these values: • `Inbound`—An inbound message (sent by a customer). • `Outbound`—An outbound message (sent to a customer by a support agent).	Small, 29.0	29.0–31.0
`emailMessageId`	String	ID of the email message	Small, 29.0	29.0–31.0
`subject`	String	Subject of the email message	Small, 29.0	29.0–31.0
`textBody`	String	Body of the email message	Big, 29.0	29.0–31.0
`toAddresses`	Email Address[]	Collection of addresses to receive the email message	Small, 29.0	29.0–31.0
`type`	String	`Emailmessage`	Small, 29.0	27.0–31.0

Feed Item Attachment: Link

Important: This response body isn't available in version 32.0 and later. In version 32.0 and later, Link Capability is used.

Property	Type	Description	Filter Group and Version	Available Version
`title`	String	Description of the link	Small, 29.0	23.0–31.0
`type`	String	`Link`	Small, 29.0	27.0–31.0
`url`	String	URL attached to the feed item (could be external)	Small, 29.0	23.0–31.0

Feed Item Attachment: Poll

Important: This response body isn't available in version 32.0 and later. In version 32.0 and later, Poll Capability is used.

Property	Type	Description	Filter Group and Version	Available Version
`choices`	Feed Poll Choice[]	Collection of feed poll choices that make up the poll	Small, 29.0	26.0–31.0

Property	Type	Description	Filter Group and Version	Available Version
`myChoiceId`	String	18-character ID of the poll choice that the current user has voted for in this poll. Returns `null` if the current user hasn't voted.	Small, 29.0	26.0–31.0
`totalVoteCount`	Integer	Total number of votes cast on the feed poll item	Small, 29.0	26.0–31.0
`type`	String	`Poll`	Small, 29.0	27.0–31.0

Feed Item Attachment: Record Snapshot

Important: This response body isn't available in version 32.0 and later. In version 32.0 and later, Record Snapshot Capability is used.

Property	Type	Description	Filter Group and Version	Available Version
`recordView`	Record View	A representation of a record.	Small, 29.0	29.0–31.0
`type`	String	`RecordSnapshot`	Small, 29.0	29.0–31.0

Feed Item Attachment: Tracked Change

Important: This response body isn't available in version 32.0 and later. In version 32.0 and later, Tracked Changes Capability is used.

Property	Type	Description	Filter Group and Version	Available Version
`changes`	Feed Tracked Change[]	Collection of feed tracked changes.	Small, 29.0	29.0–31.0
`type`	String	`TrackedChange`	Small, 29.0	29.0–31.0

Feed Item Body

Property	Type	Description	Filter Group and Version	Available Version
`messageSegments`	Segment[]	Collection of message segments, based on type. Segments can be one or more of the following: • Message Segment: Entity Link • Message Segment: Field Change • Message Segment: Field Change Name	Medium, 29.0	23.0

Property	Type	Description	Filter Group and Version	Available Version
		• Message Segment: Field Change Value • Message Segment: Hashtag • Message Segment: Link • Message Segment: Mention • Message Segment: More Changes • Message Segment: Text		
`text`	String	Display-ready text of the message. Use this text if you don't want to process the message segments.	Small, 29.0	23.0

Feed Item Page

Important: This response body isn't available in version 32.0 and later. In version 32.0 and later, Feed Element Page is used.

Property	Type	Description	Filter Group and Version	Available Version
`currentPageUrl`	String	URL to the current page of feed items	Medium, 29.0	23.0–31.0
`isModifiedToken`	String	Reserved for future use.	Medium, 29.0	27.0–31.0
`isModifiedUrl`	String	Reserved for future use.	Medium, 29.0	23.0–31.0
`items`	Feed Item[]	Collection of one or more feed items	Small, 29.0	23.0–31.0
`nextPageUrl`	String	URL to the next page of feed items	Small, 29.0	23.0–31.0
`updatesToken`	String	Token to use in an `updatedSince` parameter, or `null` if not available.	Big, 30.0	30.0–31.0
`updatesUrl`	String	A Chatter REST API resource with a query string containing the value of the `updatesToken` property. The resource returns the feed items that have been updated since the last request. Use the URL as it is—do not modify it. Property is `null` if not available.	Big, 30.0	30.0–31.0

Feed Item Preamble Message Body

Property	Type	Description	Filter Group and Version	Available Version
`messageSegments`	Segments[]	Collection of message segments, based on type. Segments can be one or more of the following: • Message Segment: Entity Link	Medium, 29.0	27.0

Property	Type	Description	Filter Group and Version	Available Version
		• Message Segment: Text		
`text`	String	The preamble is the title of the post. This property contains renderable plain text for all the segments of the message, for example, “App Builders - Pam Jones”.	Small, 29.0	27.0

Feed Item Topics

Important: This response body isn’t available in version 32.0 and later. In version 32.0 and later, Topics Capability is used.

Property	Type	Description	Filter Group and Version	Available Version
`canAssignTopics`	Boolean	`true` if a topic can be assigned to the feed item, `false` otherwise	Small, 29.0	28.0–31.0
`topics`	Topic[]	Collection of topics	Small, 29.0	28.0–31.0

Feed Poll Choice

Property	Type	Description	Filter Group and Version	Available Version
`id`	String	18-character ID of the poll choice	Small, 29.0	26.0
`position`	Integer	The location in the poll where this poll choice exists. The first poll choice starts at 1.	Small, 29.0	26.0
`text`	String	Label text associated with the poll choice	Small, 29.0	26.0
`voteCount`	Integer	Total number of votes for this poll choice	Small, 29.0	26.0
`voteCountRatio`	Double	The ratio of total number of votes for this poll choice to all votes cast in the poll. Multiply the ratio by 100 to get the percentage of votes cast for this poll choice.	Small, 29.0	26.0

Feed Tracked Change

Property	Type	Description	Filter Group and Version	Available Version
`fieldName`	String	The name of the field that was updated.	Small, 29.0	29.0

Property	Type	Description	Filter Group and Version	Available Version
`newValue`	String	The new value of the field or `null` if the field length is long or empty.	Small, 29.0	29.0
`oldValue`	String	The old value of the field or `null` if the field length is long or empty.	Small, 29.0	29.0

File Detail

Includes all of the File Summary response body with the additional item:

Property	Type	Description	Filter Group and Version	Available Version
`pageCount`	Integer	Number of pages for documents that support flash renditions, or the number of slides in a presentation. Returns 0 if the flash rendition fails, or if it isn't supported for the file type.	Medium, 29.0	24.0

SEE ALSO:

File Summary

File Share Link

Description of a file shared as a link.

Property	Type	Description	Filter Group and Version	Available Version
`fileViewUrl`	String	URL where the file can be viewed and downloaded. This value is also called the file share link.	Small, 30.0	30.0
`sharingType`	String	The permission level of sharing. Values are: • `C`—Collaborator • `I`—Sharing permission is determined by location. This generally occurs when the sharing permission is not set on the file, but on the library. This value is also used when the context user is the owner of the file. • `N`—No access • `V`—Viewer	Small, 30.0	30.0

File Shares Page

Property	Type	Description	Filter Group and Version	Available Version
`currentPageUrl`	String	URL for the current page of file shares	Medium, 29.0	24.0
`linkShare`	File Share Link	URL of the file share link	Small, 30.0	30.0
`shares`	Shares[]	Collection of one or more file shares	Small, 29.0	24.0
`nextPageUrl`	String	URL for the next page of file shares	Small, 29.0	24.0
`previousPageUrl`	String	URL for the previous page of file shares	Medium, 29.0	24.0

File Summary

Description of a file.

For all rendition properties, note the following: For shared files, renditions process asynchronously after upload. For private files, renditions process when the first file preview is requested, and aren't available immediately after the file is uploaded.

Property	Type	Description	Filter Group and Version	Available Version
`checkSum`	String	MD5 checksum for the file	Big, 29.0	26.0
`contentSize`	Integer	Size of the file in bytes	Small, 29.0	24.0
`contentHubRepository`	Files Connect Repository	The external data source associated with this file, or `null` if it doesn't point to an external file.	Small, 32.0	32.0
`contentModifiedDate`	Date	An ISO 8601 format date string, for example, 2011-02-25T18:24:31.000Z. File-specific modified date, which is updated only for direct file operations, such as rename. Modifications to the file from outside of Salesforce can update this date.	Small, 32.0	32.0
`contentUrl`	String	If the file is a link, returns the URL, otherwise the string "null"	Small, 29.0	24.0
`description`	String	Description of the file	Medium, 29.0	24.0
`downloadUrl`	String	URL to the file	Small, 29.0	26.0
`externalDocumentUrl`	String	The external URL this file points to, or `null` if it doesn't point to an external file.	Small, 32.0	32.0
`flashRenditionStatus`	String	Specifies if a flash preview version of the file has been rendered	Big, 29.0	24.0
`fileExtension`	String	Extension of the file	Small, 29.0	26.0
`fileType`	String	Type of file, such as PDF, PowerPoint, and so on	Medium, 29.0	24.0

Property	Type	Description	Filter Group and Version	Available Version
`id`	String	18-character ID of the file	Small, 29.0	24.0
`isInMyFileSync`	Boolean	`true` if the file is synced withSalesforce Files Sync; `false` otherwise.	Medium, 29.0	26.0
`isMajorVersion`	Boolean	`true` if the file is a major version; `false` if the file is a minor version. Major versions can't be replaced.	Medium, 31.0	31.0
`mimeType`	String	MIME type of the file	Medium, 29.0	26.0
`moderationFlags`	Moderation Flags	Flags for this file. Community moderators can view and take action on flagged items. If the `communityModeration` property of the Features response body is `false`, this property is `null`.	Big, 30.0	30.0
`modifiedDate`	Date	An ISO 8601 format date string, for example, 2011-02-25T18:24:31.000Z. Modifications to the file from within Salesforce update this date.	Small, 29.0	24.0
`motif`	Motif	Small, medium, and large icons indicating that the object is a file. The motif can also contain the object's base color. The icon is not a preview of the file.	Small, 29.0	27.0
`mySubscription`	Reference	If the context user is following this file, this contains information about the subscription, otherwise returns `null`	Medium, 29.0	23.0
`name`	String	Name of the file	Small, 29.0	24.0
`origin`	String	Specifies the file source. Valid values are: • `Chatter`—file came from Chatter • `Content`—file came from content • `FileField`—file came from a record or other form	Big, 29.0	24.0
`owner`	User Summary	Owner of the file	Small, 29.0	24.0
`parentFolder`	Reference	A reference to the parent folder.	Small, 29.0	28.0
`pdfRenditionStatus`	String	Specifies if a PDF preview version of the file has been rendered	Big, 29.0	24.0
`publishStatus`	String	The publish status of the file. One of these values: • `PendingAccess`—File is pending publishing. • `PrivateAccess`—File is private. • `PublicAccess`—File is public.	Small, 29.0	29.0

Property	Type	Description	Filter Group and Version	Available Version
`renditionUrl`	String	URL to the rendition resource for the file. For shared files, renditions process asynchronously after upload. For private files, renditions process when the first file preview is requested, and aren't available immediately after the file is uploaded.	Medium, 29.0	26.0
`renditionUrl 240By180`	String	URL to the 240px by 180px rendition for the file, if available. For shared files, renditions process asynchronously after upload. For private files, renditions process when the first file preview is requested, and aren't available immediately after the file is uploaded.	Medium, 29.0	29.0
`renditionUrl 720By480`	String	URL to the 720px by 480px rendition for the file, if available. For shared files, renditions process asynchronously after upload. For private files, renditions process when the first file preview is requested, and aren't available immediately after the file is uploaded.	Medium, 29.0	29.0
`repositoryFile Url`	String	The URL of the Chatter REST API resource giving access to the external repository file properties, or `null` if this file doesn't point to an external file.	Small, 32.0	32.0
`sharingRole`	String	The sharing role of the file. One of these values: • `Admin`—Owner permission, but doesn't own the file. • `Collaborator`—Viewer permission, and can edit, change permissions, and upload a new version of a file. • `Owner`—Collaborator permission, and can make a file private, and delete a file. • `Viewer`—Can view, download, and share a file. • `WorkspaceManaged`—Permission controlled by the library.	Medium, 29.0	29.0
`textPreview`	String	Text preview of the file if available, `null` otherwise.	Big, 30.0	30.0
`thumb120By90 RenditionStatus`	String	Specifies the rendering status of the 120 x 90 preview image of the file. One of these values: • `Processing`—Image is being rendered. • `Failed`—Rendering process failed. • `Success`—Rendering process was successful. • `Na`—Rendering is not available for this image.	Small, 29.0	24.0
`thumb240By180 RenditionStatus`	String	Specifies the rendering status of the 240 x 180 preview image of the file. One of these values: • `Processing`—Image is being rendered.	Small, 29.0	24.0

Property	Type	Description	Filter Group and Version	Available Version
		• `Failed`—Rendering process failed. • `Success`—Rendering process was successful. • `Na`—Rendering is not available for this image.		
`thumb720By480RenditionStatus`	String	Specifies the rendering status of the 720 x 480 preview image of the file. One of these values: • `Processing`—Image is being rendered. • `Failed`—Rendering process failed. • `Success`—Rendering process was successful. • `Na`—Rendering is not available for this image.	Small, 29.0	24.0
`title`	String	Title of the file	Small, 29.0	24.0
`type`	String	`ContentDocument`	Small, 29.0	24.0
`url`	String	URL for the file	Small, 29.0	24.0
`versionNumber`	String	Version number of the file	Medium, 29.0	24.0

File Summary Page

Property	Type	Description	Filter Group and Version	Available Version
`currentPageUrl`	String	URL for the current page of files	Medium, 29.0	24.0
`files`	File Summary[]	Collection of one or more files	Small, 29.0	24.0
`nextPageUrl`	String	URL for the next page of files	Small, 29.0	24.0
`previousPageUrl`	String	URL for the previous page of files	Medium, 29.0	24.0

Files Connect Provider Type

Information about the files connect provider.

Property	Type	Description	Filter Group and Version	Available Version
`label`	String	The localized label of the provider type.	Small, 32.0	32.0
`type`	String	The provider type. One of these values: • `ContentHubCmis` • `ContentHubGoogleDrive`	Small, 32.0	32.0

Property	Type	Description	Filter Group and Version	Available Version
		• `ContentHubSharepoint` • `ContentHubSharepointOffice365` • `SimpleUrl`		

Files Connect Repository

Information about a files connect repository.

Property	Type	Description	Filter Group and Version	Available Version
`features`	Files Connect Repository Features	The repository features.	Small, 32.0	32.0
`id`	String	18-character ID of the external data source	Small, 32.0	32.0
`label`	String	The repository label.	Small, 32.0	32.0
`motif`	Motif	The motif properties contain URLs for small, medium, and large icons that indicate the Salesforce record type. Common record types are files, users, and groups, but all record types have a set of motif icons. Custom object records use their tab style icon. All icons are available to unauthenticated users so that, for example, you can display the motif icons in an email. The motif can also contain the record type's base color.	Small, 32.0	32.0
`mySubscription`	Reference	If the context user is following this record, this contains information about the subscription; otherwise returns `null`.	Medium, 32.0	32.0
`name`	String	The repository name.	Small, 32.0	32.0
`providerType`	Files Connect Provider Type	The repository provider type.	Small, 32.0	32.0
`rootFolderItemsUrl`	String	URL to the repository root folder items, or `null` if the repository does not support browsing.	Small, 32.0	32.0
`type`	String	Type of record, `ContentHubRepository`.	Small, 32.0	32.0
`url`	String	Chatter REST API resource URL for the files connect repository	Small, 32.0	32.0

Files Connect Repository Collection

A collection of files connect repositories.

Property Name	Type	Description	Filter Group and Version	Available Version
`currentPageUrl`	String	Chatter REST API URL identifying the current page.	Medium, 32.0	32.0
`nextPageUrl`	String	Chatter REST API URL identifying the next page or `null` if there isn't a next page. Check whether this value is `null` before getting another page. If a page doesn't exist, a `ConnectApi.NotFoundException` error is returned.	Small, 32.0	32.0
`previousPageUrl`	String	Chatter REST API URL identifying the previous page or `null` if there isn't a previous page.	Medium, 32.0	32.0
`repositories`	Files Connect Repository[]	Collection of repositories.	Small, 32.0	32.0

Files Connect Repository Features

Information about the features of a files connect repository.

Property	Type	Description	Filter Group and Version	Available Version
`canBrowse`	Boolean	Specifies whether the repository's folder hierarchy can be browsed (`true`) or not (`false`).	Small, 32.0	32.0
`canSearch`	Boolean	Specifies whether the repository can be searched (`true`) or not (`false`).	Small, 32.0	32.0

Flat Feed

Property	Type	Description	Filter Group and Version	Available Version
`currentPageUrl`	String	URL for the current page of items	Medium, 29.0	25.0–26.0
`items`	Flat Feed Item[]	Collection of one or more flat feed items	Small, 29.0	25.0–26.0
`nextPageUrl`	String	URL for the next page of items	Small, 29.0	25.0–26.0

Flat Feed Item

Property	Type	Description	Filter Group and Version	Available Version
`item`	Depends on `type`	The flat feed item, depending on the value of `type`. Valid values are: • If `type` is `Comment`—Comment • If `type` is `FeedItem`—Feed Item	Small, 29.0	25.0–26.0
`type`	String	The type of item. Valid values are: • `Comment` • `FeedItem`	Small, 29.0	25.0–26.0

Folder

Description of a folder.

Property	Type	Description	Filter Group and Version	Available Version
`createdDate`	Date	An ISO-8601 formatted date string specifying the creation date of the folder.	Small, 30.0	30.0
`id`	String	18-character ID of the folder.	Small, 30.0	30.0
`isInMyFileSync`	Boolean	`true` if the folder and all folders in this folder are synced with Salesforce Files Sync; `false` otherwise.	Small, 33.0	33.0
`modifiedDate`	Date	An ISO-8601 formatted date string specifying the last modified date of the folder.	Small, 30.0	30.0
`motif`	Motif	A small, medium, and large icon indicating the record's type. The motif can also contain the record type's base color.	Small, 30.0	30.0
`mySubscription`	Reference	If the context user is following this record, this property contains information about the subscription; otherwise this property is `null`.	Medium, 30.0	30.0
`name`	String	The folder name.	Small, 30.0	30.0
`parentFolder`	Reference	A reference to the parent folder.	Small, 30.0	30.0
`path`	String	The path to the folder.	Small, 30.0	30.0
`type`	String	`Folder`	Small, 30.0	30.0
`url`	String	A Chatter REST API URL to this resource.	Small, 30.0	30.0

Folder Item

Description of an item (file or folder) in a folder.

Property	Type	Description	Filter Group and Version	Available Version
`file`	File Summary	A file representation	Small, 30.0	30.0
`folder`	Folder	A folder representation	Small, 30.0	30.0
`type`	String	The type of item. Valid values are: • `File` • `Folder`	Small, 30.0	30.0

Folder Item Page

A paged collection of Folder Item response bodies.

Property	Type	Description	Filter Group and Version	Available Version
`currentPageUrl`	String	Chatter REST API URL identifying the current page.	Medium, 30.0	30.0
`items`	Folder Item[]	A collection of folder items.	Small, 30.0	30.0
`nextPageUrl`	String	Chatter REST API URL identifying the next page or `null` if there isn't a next page. Check whether this value is `null` before getting another page. If a page doesn't exist, a `ConnectApi.NotFoundException` error is returned.	Small, 30.0	30.0
`previousPageUrl`	String	Chatter REST API URL identifying the previous page or `null` if there isn't a previous page.	Medium, 30.0	30.0

Follower Page

Property	Type	Description	Filter Group and Version	Available Version
`currentPageUrl`	String	URL of the current page	Medium, 29.0	23.0
`followers`	Subscription[]	Collection of one or more subscriptions of those that follow a record	Small, 29.0	23.0
`mySubscription`	Reference	Note: This field is no longer available as of version 24.0.		23.0–24.0

Property	Type	Description	Filter Group and Version	Available Version
		If the context user is following this user, this contains information about the subscription, else returns `null`		
`nextPageUrl`	String	Chatter REST API URL identifying the next page or `null` if there isn't a next page. Check whether this value is `null` before getting another page. If a page doesn't exist, a `ConnectApi.NotFoundException` error is returned.	Small, 29.0	23.0
`previousPageUrl`	String	Chatter REST API URL identifying the previous page or `null` if there isn't a previous page.	Medium, 29.0	23.0
`total`	Integer	Total number of followers	Small, 29.0	23.0

Following Counts

Property	Type	Description	Filter Group and Version	Available Version
`people`	Integer	Number of people the user is following	Small, 29.0	23.0
`records`	Integer	Number of records the user is following Topics are a type of record that can be followed as of version 29.0.	Small, 29.0	23.0
`total`	Integer	Total number of items the user is following	Small, 29.0	23.0

Following Page

Property	Type	Description	Filter Group and Version	Available Version
`currentPageUrl`	String	URL to the current page	Medium, 29.0	23.0
`following`	Subscription[]	Collection of one or more subscriptions	Small, 29.0	23.0
`nextPageUrl`	String	URL of the next page	Small, 29.0	23.0
`previousPageUrl`	String	URL of the previous page	Medium, 29.0	23.0
`total`	Integer	Total number of records being followed	Small, 29.0	23.0

Generic Bundle Capability

If a feed element has this capability, the feed element may be a group of other feed elements condensed into one.

Note: Bundled posts contain feed-tracked changes. In Salesforce1 downloadable apps, bundled posts are in record feeds only.

Property Name	Type	Description	Filter Group and Version	Available Version
`bundleType`	String	Specifies the type of bundle. The value is `GenericBundle`.	Small, 31.0	31.0
`page`	Feed Element Page	A collection of feed elements.	Small, 31.0	31.0
`totalElements`	Integer	The total number of feed elements that this bundle aggregates.	Small, 31.0	31.0

Generic Feed Element

A generic feed element.

Property Name	Type	Description	Filter and Version	Available Version
`body`	Feed Item Body	Information about the feed element.	Small, 22.0	22.0
`capabilities`	Feed Element Capabilities	A container for all capabilities that can be included with a feed element.	Medium, 31.0	31.0
`createdDate`	Date	An ISO 8601 format date string, for example, 2011-02-25T18:24:31.000Z.	Small, 29.0	31.0
`feedElementType`	String	Feed elements are the top-level objects that a feed contains. The feed element type describes the characteristics of that feed element. One of these values: • `Bundle`—A container of feed elements. A bundle also has a body made up of message segments that can always be gracefully degraded to text-only values. • `FeedItem`—A feed item has a single parent and is scoped to one community or across all communities. A feed item can have capabilities such as bookmarks, canvas, content, comment, link, poll. Feed items have a body made up of message segments that can always be gracefully degraded to text-only values.	Small, 31.0	31.0

Property Name	Type	Description	Filter and Version	Available Version
		• `Recommendation`—A recommendation is a feed element with a recommendations capability. A recommendation suggests records to follow, groups to join, or applications that are helpful to the context user.		
`header`	Feed Item Body	The header is the title of the post. This property contains renderable plain text for all the segments of the message. If a client doesn't know how to render a feed element type, it should render this text.	Medium, 31.0	31.0
`id`	String	The ID for a feed element.	Small, 22.0	22.0
`modifiedDate`	Date	An ISO 8601 format date string, for example, 2011-02-25T18:24:31.000Z.	Small, 29.0	31.0
`parent`	One of these response bodies: • File Detail • File Summary • Group • Group Detail • Record Summary • Record View • User Detail • User Summary	The parent of the feed element.	Medium, 29.0	28.0
`relativeCreatedDate`	Date	The created date formatted as a relative, localized string, for example, "17m ago" or "Yesterday."	Small, 29.0	31.0
`url`	String	Chatter REST API URL to this feed element.	Small, 29.0	22.0

Group

Property	Type	Description	Filter Group and Version	Available Version
`additionalLabel`	String	An additional label for the group, for example, "Archived," "Private," or "Private With Customers." If there isn't an additional label, the value is `null`.	Small, 30.0	30.0
`announcement`	Announcement	An announcement for the group.	Big, 31.0	31.0

Property	Type	Description	Filter Group and Version	Available Version
`canHaveChatterGuests`	Boolean	`true` if this group allows Chatter customers, `false` otherwise	Small, 29.0	23.0
`community`	Reference	Information about the community. Returns `null` if group exists in the default community or communities aren't enabled.	Medium, 29.0	26.0
`description`	String	Text-only description of the group	Big, 29.0	23.0
`emailToChatterAddress`	String	Group's email address for posting to this group by email. Returns `null` if Chatter emails and posting to Chatter by email aren't both enabled in your organization.	Medium, 30.0	30.0
`isArchived`	Boolean	Specifies whether the group is archived (`true`) or not (`false`).	Big, 29.0	29.0
`isAutoArchiveDisabled`	Boolean	Specifies whether automatic archiving is disabled for the group (`true`) or not (`false`).	Big, 29.0	29.0
`fileCount`	Integer	Number of files posted to this group	Big, 29.0	24.0–27.0 The `fileCount` property is in the Group Detail response body.
`id`	String	18-character ID of the group	Small, 29.0	23.0
`lastFeedElementPostDate`	Date	Date of the most recent feed element posted to the group	Big, 31.0	31.0
`lastFeedItemPostDate`	Date	Date of the most recent feed item posted to the group. After API version 30.0, use `lastFeedElementPostDate`.	Big, 29.0	24.0–30.0
`memberCount`	Integer	Total number of group members	Small, 29.0	23.0
`motif`	Motif	Small, medium, and large icons indicating that the object is a group. The motif can also contain the record type's base color. The icon is not the group photo.	Small, 29.0	27.0
`myRole`	String	Values are: • `GroupOwner` • `GroupManager` • `NotAMember` • `NotAMemberPrivateRequested`	Small, 29.0	23.0

Property	Type	Description	Filter Group and Version	Available Version
		• `StandardMember`		
`mySubscription`	Reference	If the context user is a member of this group, this contains information about that subscription, or returns `null` otherwise	Medium, 29.0	23.0
`name`	String	Name of the group	Small, 29.0	23.0
`owner`	User Summary	Information about the owner of the group	Medium, 29.0	24.0
`photo`	Photo	Information about the group photo	Small, 29.0	23.0
`type`	String	`CollaborationGroup`	Small, 29.0	23.0
`url`	String	Relative URL string to the group	Small, 29.0	23.0
`visibility`	String	Values are: • `PrivateAccess`—Only members of the group can see posts to this group. • `PublicAccess`—All users within the community can see posts to this group. • `Unlisted`—Reserved for future use.	Small, 29.0	23.0

Group Chatter Settings

A user's Chatter settings for a specific group.

Property	Type	Description	Filter Group and Version	Available Version
`emailFrequency`	String	The frequency with which a group member receives email from a group. One of the following values: • `EachPost` • `DailyDigest` • `WeeklyDigest` • `Never` • `UseDefault` The value `UseDefault` uses the user's default global setting defined in User Chatter Settings. In Communities, the Email on Each Post option is disabled once a group exceeds 10,000 members. All members who had this option selected are automatically switched to Daily Digest.	Small, 29.0	27.0

Group Detail

Property	Type	Description	Filter Group and Version	Available Version
`additionalLabel`	String	An additional label for the group, for example, "Archived," "Private," or "Private With Customers." If there isn't an additional label, the value is `null`.	Small, 30.0	30.0
`announcement`	Announcement	The current announcement for this group.	Big, 31.0	31.0
`canHaveChatterGuests`	Boolean	`true` if this group allows Chatter customers, `false` otherwise	Small, 29.0	23.0
`community`	Reference	Information about the community. Returns `null` if group exists in the default community or communities aren't enabled.	Medium, 29.0	26.0
`description`	String	Text-only description of the group	Big, 29.0	23.0
`emailToChatterAddress`	String	Group's email address for posting to this group by email. Returns `null` if Chatter emails and posting to Chatter by email aren't both enabled in your organization.	Medium, 30.0	30.0
`fileCount`	Integer	Number of files posted to this group	Big, 29.0	28.0
`id`	String	18-character ID of the group	Small, 29.0	23.0
`information`	Group Information	The "Information" section of the group. In the Web UI, this section is above the "Description" section. If the group is private, this section is visible only to members. If the context user is not a member of the group or does not have "Modify All Data" or "View All Data" permission, the value is `null`.	Big, 29.0	28.0
`lastFeedElementPostDate`	String	Date of the most recent feed element posted to the group	Big, 31.0	31.0
`lastFeedItemPostDate`	String	Date of the most recent feed item posted to the group. After API version 30.0, use `lastFeedElementPostDate`.	Big, 29.0	24.0–30.0
`memberCount`	Integer	Total number of group members	Small, 29.0	23.0
`motif`	Motif	Small, medium, and large icons indicating that the object is a group. The motif can also contain the record type's base color. The icon is not the group photo.	Small, 29.0	27.0
`myRole`	String	Values are: • `GroupOwner`	Small, 29.0	23.0

Property	Type	Description	Filter Group and Version	Available Version
		• `GroupManager` • `NotAMember` • `NotAMemberPrivateRequested` • `StandardMember`		
`mySubscription`	Reference	If the context user is a member of this group, this contains information about that subscription, or returns `null` otherwise	Medium, 29.0	23.0
`name`	String	Name of the group	Small, 29.0	23.0
`owner`	User Summary	Information about the owner of the group	Medium, 29.0	24.0
`pendingRequests`	Integer	The number of pending requests to join a private group. For public groups, the value is `null`.	Medium, 29.0	29.0
`photo`	Photo	Information about the group photo	Small, 29.0	23.0
`type`	String	`CollaborationGroup`	Small, 29.0	23.0
`url`	String	Relative URL string to the group	Small, 29.0	23.0
`visibility`	String	Values are: • `PrivateAccess`—Only members of the group can see posts to this group. • `PublicAccess`—All users within the community can see posts to this group. • `Unlisted`—Reserved for future use.	Small, 29.0	23.0

Group Information

Describes the "Information" section of a group. Information about groups, such as the group's members, photo, and the groups in the organization. Create and delete a group, add members to a group, and change the group photo.

Property	Type	Description	Filter Group and Version	Available Version
`text`	String	The text of the "Information" section of a group	Small, 29.0	28.0
`title`	String	The title of the "Information" section of a group	Small, 29.0	28.0

Group Member

Property	Type	Description	Filter Group and Version	Available Version
`id`	String	18-character ID of the group membership	Small, 29.0	23.0
`lastFeedAccessDate`	Date	The date and time at which the group member last accessed the group feed.	Small, 31.0	31.0
`role`	String	Values are: • `GroupOwner` • `GroupManager` • `NotAMember` • `NotAMemberPrivateRequested` • `StandardMember`	Small, 29.0	23.0
`url`	String	URL to this membership	Small, 29.0	23.0
`user`	User Summary	Information about the user who is subscribed to this group	Small, 29.0	23.0

Group Member Page

Property	Type	Description	Filter Group and Version	Available Version
`currentPageUrl`	String	URL to this page of members	Medium, 29.0	23.0
`members`	Group Member[]	Collection of group members	Small, 29.0	23.0
`myMembership`	Reference	If the context user is a member of this group, returns information about that membership, or returns `null` otherwise	Medium, 29.0	23.0
`nextPageUrl`	String	URL to the next page of members	Small, 29.0	23.0
`previousPageUrl`	String	URL to the previous page of members	Medium, 29.0	23.0
`totalMemberCount`	Integer	Total number of group members	Small, 29.0	23.0

Group Membership Request

Property	Type	Description	Filter Group and Version	Available Version
`createdDate`	String	An ISO formatted date string specifying the date of the membership request	Big, 29.0	27.0
`id`	String	ID for the group membership request object	Small, 29.0	27.0
`lastUpdateDate`	String	An ISO formatted date string specifying the last time the request was updated	Small, 29.0	27.0
`requestedGroup`	Reference	Information about the group the user is requesting to join	Big, 29.0	27.0
`responseMessage`	String	A message to display to the user if their membership request is declined. The value of this property is used only when the value of the `status` property is `Declined`. The maximum length is 756 characters.	Big, 29.0	27.0
`status`	String	The status of the request to join a private group. Values are: • `Accepted` • `Declined` • `Pending`	Small, 29.0	27.0
`url`	String	URL to this group membership request object	Small, 29.0	27.0
`user`	User Summary	Information about the user requesting membership in a group	Small, 29.0	27.0

Group Membership Request Collection

Property	Type	Description	Filter Group and Version	Available Version
`requests`	Group Membership Request	Information about all requests to join a private group	Small, 29.0	27.0
`total`	Integer	The total number of requests to join a private group	Small, 29.0	27.0

Group Page

Property	Type	Description	Filter Group and Version	Available Version
`currentPageUrl`	String	URL to this page of groups	Medium, 29.0	23.0
`groups`	Group Detail[]	Collection of group details	Small, 29.0	23.0
`nextPageUrl`	String	URL to the next page of groups	Small, 29.0	23.0
`previousPageUrl`	String	URL to the previous page of groups	Medium, 29.0	23.0

Group Record

A record associated with a group.

Property	Type	Description	Filter Group and Version	Available Version
`id`	String	18-character ID of the record associated with the group	Small, 33.0	33.0
`record`	Record Summary User Summary	Information about the record associated with the group	Small, 33.0	33.0
`url`	String	URL to the record associated with the group	Small, 33.0	33.0

Group Record Page

A paginated list of records associated with a group.

Property	Type	Description	Filter Group and Version	Available Version
`currentPageUrl`	String	URL to the current page	Medium, 33.0	33.0
`nextPageUrl`	String	URL to the next page. If there is no next page, returns `null`	Small, 3.0	33.0
`previousPageUrl`	String	URL to the previous page. If there is no previous page, returns `null`	Medium, 3.0	33.0
`records`	Group Record[]	Collection of records associated with a group	Small, 33.0	33.0
`totalRecordCount`	Integer	Total number of records associated with a group	Small, 33.0	33.0

Icon

Property	Type	Description	Filter Group and Version	Available Version
`height`	Integer	The height of the icon, in pixels	Small, 29.0	27.0
`width`	Integer	The width of the icon, in pixels	Small, 29.0	27.0
`url`	String	The URL of the icon. This URL is available to unauthenticated users. The URL doesn't expire.	Small, 29.0	27.0

Influence Thresholds

Property	Type	Description	Filter Group and Version	Available Version
`name`	String	Description of this influencer category. For example: `Active Influencer`	Small, 29.0	24.0
`threshold`	Double	Number expressed as a raw percentile that defines the bottom threshold for this influencer category. For example: `0.25` defines the cutoff for the `Active Influencer` category, which means it includes users with an influence rank equal to or greater than the 25th percentile (25%) and less than the next highest category threshold.	Small, 29.0	24.0

Knowledgeable People

Represents a user that is knowledgeable about a specified topic.

Property	Type	Description	Filter Group and Version	Available Version
`isEndorsedByMe`	Boolean	Specifies if the knowledgeable user is endorsed for the topic by the context user, `true` if so, `false` otherwise	Small, 30.0	30.0
`knowledgeableId`	String	ID of the knowledgeable record	Small, 30.0	30.0
`numEndorsements`	Integer	Number of endorsements received by the knowledgeable user for the topic	Medium, 30.0	30.0

Property	Type	Description	Filter Group and Version	Available Version
`reasons`	String	Specifies the reasons why the user is considered knowledgeable about the topic. Valid values are: • `Influential`—User was mentioned in posts or in comments on posts with the topic or received likes on comments on posts with the topic. • `Endorsed`—User received endorsement from another user for the topic.	Medium, 30.0	30.0
`user`	User Summary	User that is knowledgeable about the topic	Small, 30.0	30.0

Knowledgeable People Collection

A collection of knowledgeable people response bodies.

Property	Type	Description	Filter Group and Version	Available Version
`currentPageUrl`	String	Chatter REST API URL identifying the current page.	Medium, 30.0	30.0
`knowledgeableUsers`	Knowledgeable People[]	Collection of knowledgeable people response bodies	Small, 30.0	30.0
`nextPageUrl`	String	Chatter REST API URL identifying the next page or `null` if there isn't a next page. Check whether this value is `null` before getting another page. If a page doesn't exist, a `ConnectApi.NotFoundException` error is returned.	Small, 30.0	30.0
`previousPageUrl`	String	Chatter REST API URL identifying the previous page or `null` if there isn't a previous page.	Medium, 30.0	30.0

Like

Property	Type	Description	Filter Group or Version	Available Version
`id`	String	18-character ID of the like	Small, 29.0	23.0
`likedItem`	Reference	A reference to the liked comment or feed element	Medium, 29.0	27.0
`url`	String	URL to this like	Small, 29.0	23.0
`user`	User Summary	Information about the user who liked this comment or feed element	Small, 29.0	23.0

Like Message Body

Property	Type	Description	Filter Group and Version	Available Version
`messageSegments`	Segments[]	Collection of message segments, based on type. Segments can be one or more of the following: • Message Segment: Entity Link • Message Segment: Resource Link • Message Segment: Text	Medium, 29.0	27.0
`text`	String	Text summary of the users who like the feed item or comment. For example, "You like this." or "Renee Moreau, Gordon Johnson, and Pam Jones like this."	Small, 29.0	27.0

Like Page

Property	Type	Description	Filter Group and Version	Available Version
`currentPageUrl`	String	URL to the current page of likes	Medium, 29.0	23.0
`items`	Like[]	Collection of likes	Small, 32.0	32.0
`likes`	Like[]	Collection of likes Important: As of API version 32.0, use the `items` property.	Small, 29.0	23.0–31.0
`myLike`	Reference	Note: This field is no longer available as of version 24.0. Either returns a reference to the specific like or returns `null`		23.0–24.0
`nextPageUrl`	String	URL to the next page of likes	Small, 29.0	23.0
`previousPageUrl`	String	URL to the previous page of likes	Medium, 29.0	23.0
`total`	Integer	Total number of likes for the item	Small, 29.0	23.0

Link Capability

If a feed element has this capability, it has a link.

Property Name	Type	Description	Filter Group and Version	Available Version
`url`	String	Link URL. The URL can be to an external site.	Small, 32.0	32.0
`urlName`	String	Description of the link.	Small, 32.0	32.0

List View Column

Describes a single column of a record's related list. Describes how to lay out the column; doesn't contain the column's data.

Property	Type	Description	Filter Group and Version	Available Version
`dataType`	String	The object type for this column's field. This is the object type used in REST API and SOAP API.	Medium, 29.0	29.0
`fieldId`	String	The fully qualified API field ID, for example, `Contact.Name`.	Big, 29.0	29.0
`label`	String	The localized label of this column.	Small, 29.0	29.0
`name`	String	The short name of the API field, for example, `Name`.	Medium, 29.0	29.0

List View Definition

Defines a related list structure or a search list structure on a record view. There is no data in the list.

Property	Type	Description	Filter Group and Version	Available Version
`columns`	List View Column[]	The columns of fields shown in this list view.	Big, 29.0	29.0
`label`	String	The localized label of this list view.	Small, 29.0	29.0
`objectType`	List View Object	The definition of the type of the objects in this list view.	Medium, 29.0	29.0
`rowLimit`	Integer	The maximum number of objects per page in this list view. To view all objects in this related list, page through the data in pages no larger than this size. To build a SOQL statement from this describe, the SOQL statement must include a LIMIT value equal to this value or an error is thrown on execution.	Medium, 29.0	29.0
`sortColumns`	List View Sort[]	The columns in which to sort this list view. The sort columns are given in the order of their priority (for example, primary sort, secondary sort, and so on).	Medium, 29.0	29.0

List View Object

Defines the type of record displayed in the rows of a related list.

Property	Type	Description	Filter Group and Version	Available Version
`fieldId`	String	The API field ID that refers to the parent record. For example, if an Account record has a related list of Contact records, then in the Contact related list definition on the Account this value might be `"Account"`.	Big, 29.0	29.0
`type`	String	The object type name for this object. This is the object type used in REST API and SOAP API.	Small, 29.0	29.0

List View Record

The list view of a record.

Property	Type	Description	Filter Group and Version	Available Version
`fields`		The fields and values for the record displayed in this list. A collection of any of these: • Record Field • Record Field: Blank • Record Field: Compound • Record Field: Currency • Record Field: Date • Record Field: Percent • Record Field: Picklist • Record Field: Reference • Record Field: Reference with Date	Medium, 30.0	30.0
`id`	String	18-character ID of the object	Small, 29.0	23.0
`motif`	Motif	Small, medium, and large icons indicating the type of the object. The motif can also contain the object's base color. The icon is not a preview of the file.	Small, 29.0	27.0
`mySubscription`	Reference	If the context user is following this file, this contains information about the subscription; otherwise returns `null`.	Medium, 29.0	23.0
`name`	String	The localized name of this record.	Small, 29.0	23.0

Property	Type	Description	Filter Group and Version	Available Version
`type`	String	A fixed type name for this object. Check this value to determine the shape of a response body.	Small, 29.0	23.0
`url`	String	The Chatter REST API URL resource for this object.	Small, 29.0	23.0

List View Sort

Defines an ordinal of a sort order in a related list.

Property	Type	Description	Filter Group and Version	Available Version
`column`	String	The field ID on the related list column to sort.	Small, 29.0	29.0
`order`	String	The order in which to sort the column. One of these values: • `Ascending`—Ascending order (A-Z). • `Descending`—Descending order (Z-A).	Small, 29.0	29.0

Maintenance Information

Information about the upcoming scheduled maintenance for the organization.

Property Name	Type	Description	Filter Group and Version	Available Version
`description`	String	Description of the maintenance.	Small, 34.0	34.0
`maintenanceTitle`	String	Title of the maintenance.	Small, 34.0	34.0
`maintenanceType`	String	Specifies the type of maintenance. One of the following: • `Downtime`—Downtime maintenance. • `GenerallyAvailable`—Maintenance with generally available mode. • `MaintenanceWithDowntime`—Scheduled maintenance with downtime. • `ReadOnly`—Maintenance with read-only mode.	Small, 34.0	34.0
`messageEffectiveTime`	Date	Effective time when users start seeing the maintenance message.	Small, 34.0	34.0

Property Name	Type	Description	Filter Group and Version	Available Version
`messageExpirationTime`	Date	Expiration time of the maintenance message.	Small, 34.0	34.0
`scheduledEndDowntime`	Date	Scheduled end of downtime. `null` for `GenerallyAvailable` and `ReadOnly` maintenance types.	Small, 34.0	34.0
`scheduledEndMaintenanceTime`	Date	Scheduled end of maintenance. `null` for `Downtime` maintenance type.	Small, 34.0	34.0
`scheduledStartDowntime`	Date	Scheduled start of downtime. `null` for `GenerallyAvailable` and `ReadOnly` maintenance types.	Small, 34.0	34.0
`scheduledStartMaintenanceTime`	Date	Scheduled start time of maintenance. `null` for `Downtime` maintenance type.	Small, 34.0	34.0

Managed Topic

Represents a managed topic in a community.

Property Name	Type	Description	Filter Group and Version	Available Version
`id`	String	ID of managed topic.	Small, 32.0	32.0
`managedTopicType`	String	Type of managed topic. • `Featured`—Topics that are featured, for example, on the community home page, but don't provide overall navigation. • `Navigational`—Topics that display in a navigational menu in the community.	Small, 32.0	32.0
`topic`	Topic	Information about the topic.	Small, 32.0	32.0
`url`	String	Chatter REST API URL to the managed topic.	Small, 32.0	32.0

Managed Topic Collection

A collection of managed topics in a community.

Property Name	Type	Description	Filter Group and Version	Available Version
`currentPageUrl`	String	Chatter REST API URL identifying the current page.	Medium, 32.0	32.0
`managedTopics`	Managed Topic[]	Collection of managed topics.	Small, 32.0	32.0

Mention Completion

Information about a record that could be used to @mention a user or group.

Property	Type	Description	Filter Group and Version	Available Version
`additionalLabel`	String	An additional label (if one exists) for the record represented by this completion, for example, "(Customer)" or "(Acme Corporation)".	Big, 29.0	29.0
`description`	String	Description for the record represented by this completion	Medium, 29.0	29.0
`name`	String	Name for the record represented by this completion	Small, 29.0	29.0
`photoUrl`	String	URL to the photo or icon for the record represented by this completion	Medium, 29.0	29.0
`recordId`	String	18-character ID for the record represented by this completion	Small, 29.0	29.0
`userType`	String	If the record represented by this completion is a user, this value is the user type associated with that user; otherwise the value is `null`. One of these values: • `ChatterGuest`—User is an external user in a private group. • `ChatterOnly`—User is a Chatter Free customer. • `Guest`—User is unauthenticated. • `Internal`—User is a standard organization member. • `Portal`—User is an external user in a customer portal, partner portal, or community. • `System`—User is Chatter Expert or a system user. • `Undefined`—User is a user type that is a custom object.	Medium, 30.0	30.0

Mention Completion Page

A paginated list of Mention Completion response bodies.

Property	Type	Description	Filter Group and Version	Available Version
`currentPageUrl`	String	URL of the current page	Medium, 29.0	29.0
`mentionCompletions`	Mention Completion[]	Collection of Mention Completion response bodies to use in a mention.	Small, 29.0	29.0
`nextPageUrl`	String	Chatter REST API URL identifying the next page or `null` if there isn't a next page. Check whether this value is `null` before getting another page. If a page doesn't exist, a `ConnectApi.NotFoundException` error is returned.	Small, 29.0	29.0
`previousPageUrl`	String	Chatter REST API URL identifying the previous page or `null` if there isn't a previous page.	Medium, 29.0	29.0

Mention Validation

Information about whether a proposed mention is valid for the context user.

Property	Type	Description	Filter Group and Version	Available Version
`recordId`	String	The ID of the mentioned record.	Small, 29.0	29.0
`validationStatus`	String	Specifies the type of validation error for a proposed mention, if any. One of these values: • `Disallowed`—The proposed mention is invalid and is rejected because the context user is trying to mention something that is not allowed. For example, a user who is not a member of a private group is trying to mention the private group. • `Inaccessible`—The proposed mention is allowed but the user or record being mentioned isn't notified because they don't have access to the parent record being discussed. • `Ok`—There is no validation error for this proposed mention.	Small, 29.0	29.0

Mention Validations

Information about whether a set of mentions is valid for the context user.

Property	Type	Description	Filter Group and Version	Available Version
`hasErrors`	Boolean	Indicates whether at least one of the proposed mentions has an error (`true`), or not (`false`). For example, context users can't mention private groups they don't belong to. If such a group is included in the list of mention validations, `hasErrors` is `true` and the group has a `validationStatus` of `Disallowed` in its mention validation.	Small, 29.0	29.0
`mentionValidations`	Mention Validation[]	Information about whether a set of mentions is valid for the context user.	Medium, 29.0	29.0

Message

Property	Type	Description	Filter Group and Version	Available Version
`body`	Feed Item Body	Text of this message	Small, 29.0	23.0
`conversationId`	String	18-character ID of the conversation this message is part of	Big, 29.0	23.0
`conversationUrl`	String	URL to the conversation this message is part of	Small, 29.0	23.0
`id`	String	18-character ID of the message	Small, 29.0	23.0
`recipients`	User Summary[]	Collection of one or more message recipients Note: The collection of recipients is empty for the following resources because the conversation participants are already identified in the conversation's `members` property in Conversation Detail and Conversation Summary. • `/chatter/users/me/conversations` • `/chatter/users/me/conversations/conversationId`	Small, 29.0	23.0
`sentDate`	String	ISO8601 date string, for example, 2011-02-25T18:24:31.000Z	Small, 29.0	23.0
`sender`	User Summary	Information about the user who sent this message	Small, 29.0	23.0

Property	Type	Description	Filter Group and Version	Available Version
`sendingCommunity`	Reference	Information about the community from which the message was sent Returns `null` for the default community or if communities aren't enabled.	Medium, 32.0	32.0
`url`	String	URL to this message	Small, 29.0	23.0

Message Collection

Property	Type	Description	Filter Group and Version	Available Version
`currentPageUrl`	String	Chatter REST API URL identifying the current page.	Medium, 29.0	23.0
`messages`	Message[]	Collection of one or more messages	Small, 29.0	23.0
`nextPageUrl`	String	Chatter REST API URL identifying the next page or `null` if there isn't a next page. Check whether this value is `null` before getting another page. If a page doesn't exist, a `ConnectApi.NotFoundException` error is returned.	Small, 29.0	23.0

Message Segment: Entity Link

Represents an entity, such as a user or a group.

Property	Type	Description	Filter Group and Version	Available Version
`motif`	Motif	A set of small, medium, and large icons that correspond to the entity type. The motif can also contain the entity type's base color.	Small, 29.0	27.0
`reference`	Reference	Information about the entity	Small, 29.0	25.0
`text`	String	Plain text output for this segment	Small, 29.0	25.0
`type`	String	`EntityLink`	Small, 29.0	25.0

Message Segment: Field Change

Represents an inline field change.

Example: An example of an inline field change is `"First Name from Bob to Joe"`. This example segment consists of multiple segments.

```
[
   Text("Owner from "),
   FieldChangeValue("Bob Smith"),
   Text(" to "),
   FieldChangeValue("Dave Jones")
]
```

Property	Type	Description	Filter Group and Version	Available Version
`segments`	Segments[]	Collection of message segments, based on type. Segments can be one or more of the following: • Message Segment: Entity Link • Message Segment: Field Change • Message Segment: Field Change Name • Message Segment: Field Change Value • Message Segment: Hashtag • Message Segment: Link • Message Segment: Mention • Message Segment: More Changes • Message Segment: Text	Medium, 29.0	23.0
`text`	String	Formatted text of the change. For example: `"City from San Francisco to New Orleans"`	Small, 29.0	23.0
`type`	String	Value is `FieldChange`	Small, 29.0	23.0

Message Segment: Field Change Name

Represents an inline name of a field change, like “Name” or “Description.” Making this a separate segment makes it possible to give special user interface treatments to field names, for example, making a name field bold.

Property	Type	Description	Filter Group and Version	Available Version
`text`	String	Formatted text of the change. Possible values include: • Old name of the field • New name of the field	Small, 29.0	23.0
`type`	String	Value is `FieldChangeName`	Small, 29.0	23.0

Message Segment: Field Change Value

Represents an inline value of a field change, like "1" or "Bob Smith." Making this a separate segment makes it possible to add special coding, for example, to wrap values in anchor tags.

Property	Type	Description	Filter Group and Version	Available Version
`valueType`	String	The value type. One of these values: • `NewValue`—The new value of the field change. • `OldValue`—The old value of the field change.	Big, 29.0	29.0
`text`	String	Formatted text of the change. Possible values include: • the words `a blank value` Note: This text is also localized. • the old value of the field • the new value of the field	Small, 29.0	23.0
`type`	String	Value is `FieldChangeValue`	Small, 29.0	23.0
`url`	String	URL value if the field change is to a URL field (such as a web address)	Small, 29.0	23.0

Message Segment: Hashtag

Represents an inline hashtag.

Property	Type	Description	Filter Group and Version	Available Version
`tag`	String	Text of the hashtag without the hash symbol (#)	Small, 29.0	23.0
`text`	String	Text of the hashtag with the hash symbol (#)	Small, 29.0	23.0
`topicUrl`	String	URL for the topic in this format: `/services/data/.../topics?exactMatch=true&q=tag`	Small, 29.0	28.0
`type`	String	Value is `Hashtag`	Small, 29.0	23.0
`url`	String	URL search for the tag in this format: `services/data/.../feed-elements?q=tag`	Small, 29.0	23.0

Message Segment: Link

Represents an inline link to a URL

Property	Type	Description	Filter Group and Version	Available Version
`text`	String	Formatted text of the feed item of this segment	Small, 29.0	23.0
`type`	String	Value is `Link`	Small, 29.0	23.0
`url`	String	URL value of the link	Small, 29.0	23.0

Message Segment: Mention

Represents an inline @mention.

Property	Type	Description	Filter Group and Version	Available Version
`accessible`	Boolean	Specifies whether the mentioned user or group can see the mention (`true`) or not (`false`).	Small, 29.0	23.0
`name`	String	Name of the mentioned user or group	Small, 29.0	23.0
`text`	String	Formatted text of the feed item of this segment	Small, 29.0	23.0
`type`	String	Value is `Mention`	Small, 29.0	23.0
`record`		Information about the mentioned user or group. One of these response bodies: • Group • User Detail • User Summary	Small, 29.0	29.0
`user`	User Summary	Information about the mentioned user, `null` if the context user doesn't have access. **Important:** In versions 29.0 and later, use the `record` property.	None	23.0–28.0 In versions before 29.0, if the mention is not a user, the mention is in a Message Segment: Text response body.

Message Segment: More Changes

Property	Type	Description	Filter Group and Version	Available Version
`moreChanges`	Message Segment: Field Change[]	The complete collection of tracked changes for a feed item.	Big, 29.0	29.0
`moreChanges Count`	Integer	Number of changes	Small, 29.0	23.0
`text`	String	Formatted text of the number of changes. For example: `"2 other changes"` Note: This value is localized.	Small, 29.0	23.0
`type`	String	Value is `MoreChanges`	Small, 29.0	23.0

Message Segment: Resource Link

Property	Type	Description	Filter Group and Version	Available Version
`text`	String	Formatted text of this segment	Small, 29.0	23.0
`type`	String	Value is `"ResourceLink"`	Small, 29.0	23.0
`url`	String	URL value of the resource link	Small, 29.0	27.0

Message Segment: Text

Represents plain inline text.

Property	Type	Description	Filter Group and Version	Available Version
`text`	String	Formatted text associated with the item. Possible values include: • formatted text of a comment • the name of the field that has been changed with the word `from`	Small, 29.0	23.0
`type`	String	Value is `Text`	Small, 29.0	23.0

Moderation Capability

If a feed element has this capability, users in the community can flag the feed element for moderation. This response body also includes information about the flagged content.

Property Name	Type	Description	Filter Group and Version	Available Version
`moderationFlags`	Moderation Flags	Flags for this feed element. Community moderators can view and take action on flagged items.	Big, 31.0	31.0

Moderation Flags

Moderation flags are available if Salesforce Communities is enabled in an organization and the community administrator selected "Allow members to flag content" during community setup.

Property	Type	Description	Filter Group and Version	Available Version
`flagCount`	Integer	If the context user is a community moderator, shows the total number of flags on the post, comment, or file. If the context user isn't a community moderator, value is `null`. Flags on files is available starting in version 30.0.	Small, 29.0	29.0
`flaggedByMe`	Boolean	Indicates whether the post, comment, or file was flagged by the context user (`true`) or not (`false`). Flags on files is available starting in version 30.0.	Small, 29.0	29.0

Motif

The motif properties contain URLs for small, medium, and large icons that indicate the Salesforce record type. Common record types are files, users, and groups, but all record types have a set of motif icons. Custom object records use their tab style icon. All icons are available to unauthenticated users so that, for example, you can display the motif icons in an email. The motif can also contain the record type's base color.

Example:

Note: The motif images are icons, not user uploaded images or photos. For example, every user has the same set of motif icons.

Custom object records use their tab style icon, for example, the following custom object uses the "boat" tab style:

```
"motif": {
   "color": "8C004C",
   "largeIconUrl": "/img/icon/custom51_100/boat64.png",
```

```
    "mediumIconUrl": "/img/icon/custom51_100/boat32.png",
    "smallIconUrl": "/img/icon/custom51_100/boat16.png",
    "svgIconUrl": null
  },
```

Users use the following icons:

```
  "motif": {
    "color": "1797C0",
    "largeIconUrl": "/img/icon/profile64.png",
    "mediumIconUrl": "/img/icon/profile32.png",
    "smallIconUrl": "/img/icon/profile16.png",
    "svgIconUrl": null
  },
```

Groups use the following icons:

```
  "motif": {
    "color": "1797C0",
    "largeIconUrl": "/img/icon/groups64.png",
    "mediumIconUrl": "/img/icon/groups32.png",
    "smallIconUrl": "/img/icon/groups16.png"
  },
```

Files use the following icons:

```
 "motif": {
    "color": "1797C0",
    "largeIconUrl": "/img/content/content64.png",
    "mediumIconUrl": "/img/content/content32.png",
    "smallIconUrl": "/img/icon/files16.png",
    "svgIconUrl": null
  },
```

Note: To view the icons in the previous examples, preface the URL with `https://instance_name`. For example, `https://instance_name/img/icon/profile64.png`.

Property	Type	Description	Filter Group and Version	Available Version
`color`	String	A hex value representing the base color of the record type, or `null`.	Small, 29.0	29.0
`largeIconUrl`	String	A large icon indicating the record type.	Small, 29.0	27.0
`mediumIconUrl`	String	A medium icon indicating the record type.	Small, 29.0	27.0
`smallIconUrl`	String	A small icon indicating the record type.	Small, 29.0	27.0
`svgIconUrl`	String	An icon in SVG format indicating the record type, or `null` if the icon doesn't exist.	Small, 34.0	34.0

Non-Entity Recommendation

Represents a recommendation for a non-Salesforce entity, such as an application.

This response body isn't available in version 34.0 and later. In version 34.0 and later, the Recommendation response body is used for all recommendations.

Property Name	Type	Description	Filter Group and Version	Available Version
`displayLabel`	String	Localized label of the non-entity object.	Small, 32.0	32.0
`explanation`	Explanation Summary	The recommendation explanation.	Big, 29.0	23.0
`motif`	Motif	Motif for the non-entity object.	Small, 32.0	32.0
`recommendationType`	String	Specifies the type of record being recommended. • `apps` • `files` • `groups` • `records` • `users`	Small, 32.0	32.0

Organization

Property	Type	Description	Filter Group and Version	Available Version
`accessTimeout`	Integer	Amount of time after which the system prompts users who have been inactive to log out or continue working	Small, 29.0	24.0
`features`	Features	Information about features available in the organization or community	Small, 29.0	24.0
`maintenanceInfo`	Maintenance Information[]	Information about a list of upcoming scheduled maintenances for the organization.	Medium, 34.0	34.0
`name`	String	Organization name	Small, 29.0	26.0
`orgId`	String	18-character ID for the organization	Small, 29.0	24.0
`userSettings`	User Settings	Information about the organization permissions for the user	Small, 29.0	24.0

Origin Capability

If a feed element has this capability, it was created by a feed action.

Property Name	Type	Description	Filter Group and Version	Available Version
`actor`	User Summary	The user who executed the feed action.	Small, 33.0	33.0
`originRecord`	Reference	A reference to the feed element containing the feed action.	Small, 33.0	33.0

Phone Number

A phone number.

Property	Type	Description	Filter Group and Name	Available Version
`label`	String	Localized string indicating the phone type	Small, 30.0	30.0
`number`	String	Note: This property is not available after version 26.0. Use the `phoneNumber` property instead. Phone number	NA	23.0–26.0
`phoneNumber`	String	Phone number	Small, 29.0	27.0
`phoneType`	String	Phone type. One of these values: • `Fax` • `Mobile` • `Work` These values are not localized.	Small, 30.0	30.0
`type`	String	Note: This property is not available after version 29.0. Use the `phoneType` property instead. Values are: • `Fax` • `Mobile` • `Work`	Small, 29.0	23.0–29.0

Photo

Property	Type	Description	Filter Group and Version	Available Version
`fullEmailPhotoUrl`	String	A temporary URL to the large profile picture that expires after 30 days. This URL is available to unauthenticated users.	Medium, 29.0	27.0
`largePhotoUrl`	String	URL to the large profile picture. The default width is 200 pixels, while the length is scaled so the original image proportions are maintained.	Medium, 29.0	23.0
`photoVersionId`	String	18-character ID to that version of the photo.	Small, 29.0	24.0
`smallPhotoUrl`	String	URL to the small profile picture. The default size is 64x64 pixels.	Small, 29.0	23.0
`standardEmailPhotoUrl`	String	A temporary URL to the small profile picture that expires after 30 days. This URL is available to unauthenticated users.	Small, 29.0	27.0
`url`	String	URL to a resource that returns a Photo response body: for example, `/services/data/v34.0/chatter/users/005D0000001LL8OIAW/photo`.	Small, 29.0	27.0

Platform Action

A platform action instance with state information for the context user.

Action links are one type of platform action and are therefore represented as platform action response bodies.

Property Name	Type	Description	Filter Group and Version	Available Version
`actionUrl`	String	For action links of `subtype` `Ui` or `Download`, direct the user to download or visit the UI from this link. Salesforce issues a Javascript redirect for the link in this format: `/action-link-redirect/`***`communityId`***`/`***`actionLinkId`***`?_bearer=`***`bearerToken`*** For `Api` and `ApiAsync` action links and for all platform actions, this value is `null`. Salesforce handles the callout to `Api` or and `ApiAsync` action links. To call an `Api` or an `ApiAsync` action link, you can also make a PATCH request to the	Medium, 33.0	33.0

Property Name	Type	Description	Filter Group and Version	Available Version
		/connect/action-links/*actionLinkId* resource and set `status` to `PendingStatus`.		
`apiName`	String	The API name. This value may be `null`.	Medium, 33.0	33.0
`confirmationMessage`	String	If this action requires a confirmation and has a status of `NewStatus`, this is a default localized message that should be shown to an end user prior to invoking the action. Otherwise, this is `null`.	Medium, 33.0	33.0
`createdRecords`	Record Summary[]	The collection of records that this action created after it was invoked by the context user. This collection is empty if the action hasn't been performed yet by the context user or if the action doesn't create a record. Only records created by the context user are shown.	Medium, 33.0	33.0
`executingUser`	User Summary	User who initiated execution of the platform action.	Medium, 33.0	33.0
`groupDefault`	Boolean	`true` if this platform action is the default or primary platform action in the platform action group; `false` otherwise. There can be only one default platform action per platform action group.	Small, 33.0	33.0
`iconUrl`	String	The URL of the icon for the platform action. This value may be `null`.	Medium, 33.0	33.0
`id`	String	The ID of the platform action. If the `type` is `QuickAction` and the `subtype` is `Create`, this value is `null`.	Small, 33.0	33.0
`label`	String	The localized label for this platform action.	Small, 33.0	33.0
`modifiedDate`	Date	ISO8601 date string, for example, 2011-02-25T18:24:31.000Z	Small, 33.0	33.0
`platformActionGroup`	Reference	A reference to the platform action group containing this platform action.	Small, 33.0	33.0
`status`	String	The execution status of the platform action. Values are: • `FailedStatus`—The action link execution failed.	Small, 33.0	33.0

Property Name	Type	Description	Filter Group and Version	Available Version
		• `NewStatus`—The action link is ready to be executed. Available for `Download` and `Ui` action links only. • `PendingStatus`—The action link is executing. Choosing this value triggers the API call for `Api` and `ApiAsync` action links. • `SuccessfulStatus`—The action link executed successfully. For action links only, you can update the `status` by making a PATCH request to the `/connect/action-links/actionLinksId` resource.		
`subtype`	String	The subtype of a platform action or `null`. If the `type` property is `ActionLink`, possible values are: • `Api`—The action link calls a synchronous API at the action URL. Salesforce sets the status to `SuccessfulStatus` or `FailedStatus` based on the HTTP status code returned by your server. • `ApiAsync`—The action link calls an asynchronous API at the action URL. The action remains in a `PendingStatus` state until a third party makes a request to `/connect/action-links/actionLinkId` to set the status to `SuccessfulStatus` or `FailedStatus` when the asynchronous operation is complete. • `Download`—The action link downloads a file from the action URL. • `Ui`—The action link takes the user to a web page at the action URL.	Small, 33.0	33.0
`type`	String	The type of platform action. Values are: • `ActionLink`—An indicator on a feed element that targets an API, a web page, or a file and is represented by a button in the Salesforce Chatter feed UI.	Small, 33.0	33.0

Property Name	Type	Description	Filter Group and Version	Available Version
		• `ProductivityAction`—Productivity actions are predefined by Salesforce and are attached to a limited set of objects. You can't edit or delete productivity actions.. • `CustomButton`—When clicked, opens a URL or a Visualforce page in a window or executes JavaScript. • `QuickAction`—A global or object-specific action. • `StandardButton`—A predefined Salesforce button such as New, Edit, and Delete.		
`url`	String	Chatter REST API resource URL for the platform action. If the `type` is `QuickAction` and the `subtype` is `Create`, this value is `null`.	Small, 33.0	33.0

Platform Action Group

A platform action group instance with state appropriate for the context user.

Action link groups are one type of platform action group and are therefore represented as platform action group response bodies.

Property Name	Type	Description	Filter Group and Version	Available Version
`category`	String	Indicates the priority and relative locations of platform actions. Values are: • `Primary`—The action link group is displayed in the body of the feed element. • `Overflow`—The action link group is displayed in the overflow menu of the feed element.	Medium, 33.0	33.0
`id`	String	An 18-character ID or an opaque string ID for the platform action group. If the Platform Action `type` is `QuickAction` and the `subtype` is `Create`, this value is `null`.	Small, 33.0	33.0

Property Name	Type	Description	Filter Group and Version	Available Version
`modifiedDate`	Date	ISO8601 date string, for example, 2011-02-25T18:24:31.000Z.	Small, 33.0	33.0
`platformActions`	Platform Action[]	The platform action instances for this group. Within an action link group, action links are displayed in the order listed in the `actionLinks` property of the Action Link Group Definition Input request body. Within a feed item, action link groups are displayed in the order specified in the `actionLinkGroupIds` property of the Associated Actions Capability Input request body.	Medium, 33.0	33.0
`url`	String	Chatter REST API resource URL for the platform action group. If the Platform Action `type` is `QuickAction` and the `subtype` is `Create`, this value is `null`.	Small, 33.0	33.0

Poll Capability

If a feed element has this capability, the feed element can include a poll for voting.

Property Name	Type	Description	Filter and Version	Available Version
`choices`	Feed Poll Choice[]	Collection of poll choices that make up the poll.	Small, 32.0	32.0
`myChoiceId`	String	18-character ID of the poll choice that the current user has voted for in this poll. Returns `null` if the current user has not voted.	Small, 32.0	32.0
`totalVoteCount`	String	Total number of votes cast on the feed poll element.	Small, 32.0	32.0

Question and Answers Capability

If a feed element has this capability, it has a question and comments on the feed element are answers to the question.

Property Name	Type	Description	Filter and Version	Available Version
`bestAnswer`	Comment	The comment selected as the best answer for the question.	Small, 32.0	32.0
`bestAnswerSelectedBy`	User Summary	The user who selected the best answer for the question.	Small, 32.0	32.0
`canCurrentUserSelectOrRemoveBestAnswer`	Boolean	Indicates whether current user can select or remove a best answer (`true`) or not (`false`).	Small, 32.0	32.0
`escalatedCase`	Reference	If a question post is escalated, this is the case to which it was escalated.	Medium, 33.0	33.0
`questionTitle`	String	Title for the question.	Small, 32.0	32.0

Question and Answers Suggestion Collection

Question and answers suggestions search result.

Property Name	Type	Description	Filter Group and Version	Available Version
`articles`	Article Item[]	Collection of articles.	Small, 32.0	32.0
`questions`		Collection of questions, which can be any of these: • Generic Feed Element • Feed Item	Small, 32.0	32.0

Recommendation

Represents a recommendation.

Property	Type	Description	Filter Group and Version	Available Version
`actOnUrl`	String	For user, file, group, and record `entity` types, use this Chatter REST URL with a POST request to follow the recommended user, file, or record, or join the recommended group. For example: `/services/data/v34.0/chatter/users/005D0000001AzOAIA0/following?subjectId=005D0000001B4CvIAK` For recommended object `entity` types, such as custom recommendations, use the `actionUrl`	Small, 29.0	23.0

Property	Type	Description	Filter Group and Version	Available Version
		property of the Platform Action response body to take action on the recommendation.		
`action`	String	Specifies the action to take on the recommendation. Values are: • `follow` (V23.0) • `join` (V23.0) • `view` (V25.0)	Small, 29.0	
`entity`	File Summary Group Recommended Object (version 34.0 and later) Record Summary User Summary	The entity with which the receiver is recommended to take action.	Small, 32.0	32.0
`explanation`	Explanation Summary	The recommendation explanation.	Big, 29.0	23.0
`object`	File Summary Group Record Summary User Summary	The object with which the receiver is recommended to take action. **Important:** In version 32.0 and later, use the `entity` property.	Small, 29.0	23.0-31.0
`platformActionGroup`	Platform Action Group	A platform action group instance with state appropriate for the context user.	Small, 34.0	34.0
`recommendationType`	String	Specifies the type of record being recommended. • `apps` • `files` • `groups` • `records` • `users`	Small, 32.0	32.0
`type`	String	Not available after version 23.0.		23.0 -23.0
`url`	String	URL to the recommendation.	Small, 34.0	34.0

Recommendation Definition

Represents a custom recommendation definition.

Property Name	Type	Description	Filter Group and Version	Available Version
`actionUrl`	String	The URL for acting on this recommendation.	Small, 33.0	33.0
`actionUrlName`	String	The text label for the action URL in the user interface.	Small, 34.0	34.0
`explanation`	String	Explanation of the recommendation definition.	Medium, 33.0	33.0
`id`	String	18-character ID of the recommendation definition.	Small, 33.0	33.0
`name`	String	Name of the recommendation definition. The name is displayed in Setup.	Medium, 33.0	33.0
`photo`	Photo	Photo of the recommendation definition.	Medium, 33.0	33.0
`title`	String	Title of the recommendation definition.	Small, 33.0	33.0
`url`	String	URL to the Chatter REST API resource for the recommendation definition.	Small, 33.0	33.0

Recommendation Definition Collection

Represents a list of recommendation definitions.

Property Name	Type	Description	Filter Group and Version	Available Version
`recommendationDefinitions`	Recommendation Definition[]	A list of recommendation definitions.	Small, 33.0	33.0
`url`	String	URL to the Chatter REST API resource for the recommendation definition collection.	Small, 33.0	33.0

Recommendations

A list of recommendations.

Property Name	Type	Description	Filter Group and Version	Available Version
`recommendations`	Recommendation[]	Collection of recommendations.	Small, 29.0	23.0

Recommendations Capability

If a feed element has this capability, it has a recommendation.

Property Name	Type	Description	Filter Group and Version	Available Version
`items`	A collection of these response bodies: • Non-Entity Recommendation This response body isn't available in version 34.0 and later. In version 34.0 and later, the Recommendation response body is used for all recommendations. • Recommendation	A collection of recommendations.	Small, 32.0	32.0

Recommended Object

An object that is recommended, such as a custom or static recommendation.

Property Name	Type	Description	Filter Group and Version	Available Version
`idOrEnum`	String	ID of a recommendation definition for a custom recommendation.	Small, 34.0	34.0
`motif`	Motif	Motif of the recommended object.	Small, 34.0	34.0
`name`	String	Name of the recommended object.	Small, 29.0	22.0
`type`	String	`RecommendedObject`	Small, 29.0	22.0

Record Field

A generic record field containing a label and text value.

Property	Type	Description	Filter Group and Version	Available Version
`label`	String	A localized string describing the record field.	Small, 29.0	29.0
`text`	String	The text value of the record field. All record fields have a text value. To ensure that all clients can consume new	Small, 29.0	29.0

Property	Type	Description	Filter Group and Version	Available Version
		content, inspect the record field's `type` property. If it isn't recognized, render the text value as the default case.		
`type`	String	Valid values are: • `Boolean` • `Email` • `Number` • `Phone` • `Text`	Small, 29.0	29.0

Record Field: Blank

A record field displayed as a place holder in a grid of fields.

Property	Type	Description	Filter Group and Version	Available Version
`type`	String	`Blank`	Small, 29.0	29.0

Record Field: Compound

A record field that is a composite of subfields.

Property	Type	Description	Filter Group and Version	Available Version
`fields`		A collection of subfields that make up the compound field. A collection of these response bodies: • Record Field • Record Field: Blank • Record Field: Compound • Record Field: Currency • Record Field: Date • Record Field: Percent • Record Field: Picklist • Record Field: Reference • Record Field: Reference with Date	Big, 29.0	29.0
`label`	String	A localized string describing the record field.	Small, 29.0	29.0

Property	Type	Description	Filter Group and Version	Available Version
`text`	String	The text value of the record field. All record fields have a text value. To ensure that all clients can consume new content, inspect the record field's `type` property. If it isn't recognized, render the text value as the default case.	Small, 29.0	29.0
`type`	String	Valid values are: • `Address` • `Compound` • `Location` • `Name`	Small, 29.0	29.0

Record Field: Currency

A record field containing a currency value.

Property	Type	Description	Filter Group and Version	Available Version
`label`	String	A localized string describing the record field.	Small, 29.0	29.0
`text`	String	The text value of the record field. All record fields have a text value. To ensure that all clients can consume new content, inspect the record field's `type` property. If it isn't recognized, render the text value as the default case. Currency is formatted in the default currency of the organization.	Small, 29.0	29.0
`type`	String	`Currency`	Small, 29.0	29.0

Record Field: Date

A record field containing a date.

Property	Type	Description	Filter Group and Version	Available Version
`dateValue`	Date	A date that a machine can read.	Medium, 29.0	29.0
`label`	String	A localized string describing the record field.	Small, 29.0	29.0
`text`	String	The text value of the record field. All record fields have a text value. To ensure that all clients can consume new	Small, 29.0	29.0

Property	Type	Description	Filter Group and Version	Available Version
		content, inspect the record field's `type` property. If it isn't recognized, render the text value as the default case.		
`type`	String	Valid values are: • `Date` • `DateTime` • `Time`	Small, 29.0	29.0

Record Field: Percent

A record field containing a percentage value.

Property	Type	Description	Filter Group and Version	Available Version
`value`	Double	The value of the percentage.	Medium, 29.0	29.0
`label`	String	A localized string describing the record field.	Small, 29.0	29.0
`text`	String	The text value of the record field. All record fields have a text value. To ensure that all clients can consume new content, inspect the record field's `type` property. If it isn't recognized, render the text value as the default case.	Small, 29.0	29.0
`type`	String	`Percent`	Small, 29.0	29.0

Record Field: Picklist

A record field containing an enumerated value.

Property	Type	Description	Filter Group and Version	Available Version
`label`	String	A localized string describing the record field.	Small, 29.0	29.0
`text`	String	The text value of the record field. All record fields have a text value. To ensure that all clients can consume new content, inspect the record field's `type` property. If it isn't recognized, render the text value as the default case.	Small, 29.0	29.0
`type`	String	`Picklist`	Small, 29.0	29.0

Record Field: Reference

A record field containing a reference to another object.

Property	Type	Description	Filter Group and Version	Available Version
`label`	String	A localized string describing the record field.	Small, 29.0	29.0
`reference`	Reference	The object referenced by the record field.	Medium, 29.0	29.0
`text`	String	The text value of the record field. All record fields have a text value. To ensure that all clients can consume new content, inspect the record field's `type` property. If it isn't recognized, render the text value as the default case.	Small, 29.0	29.0
`type`	String	`Reference`	Small, 29.0	29.0

Record Field: Reference with Date

A record field containing a referenced object that acted at a specific time, for example, "Created By...".

Property	Type	Description	Filter Group and Version	Available Version
`dateValue`	Date	A time at which the referenced object acted.	Medium, 29.0	29.0
`label`	String	A localized string describing the record field.	Small, 29.0	29.0
`reference`	Reference	The object referenced by the record field.	Medium, 29.0	29.0
`text`	String	The text value of the record field. All record fields have a text value. To ensure that all clients can consume new content, inspect the record field's `type` property. If it isn't recognized, render the text value as the default case.	Small, 29.0	29.0
`type`	String	Valid values are: • `CreatedBy` • `LastModifiedBy`	Small, 29.0	29.0

Record Snapshot Capability

If a feed element has this capability, it contains all the snapshotted fields of a record for a single create record event.

Property Name	Type	Description	Filter Group and Version	Available Version
`recordView`	Record View	A record representation that includes metadata and data so you can display the record easily.	Small, 32.0	32.0

Record Summary

Property	Type	Description	Filter Group and Version	Available Version
`id`	String	18-character ID of the record	Small, 29.0	23.0
`motif`	Motif	Small, medium, and large icons indicating the tab style of the record. The motif can also contain the record type's base color.	Small, 29.0	27.0
`mySubscription`	Reference	If the context user is following this record, this contains information about the subscription; otherwise returns `null`.	Medium, 29.0	24.0
`name`	String	The localized name	Small, 29.0	23.0
`type`	String	Type of record, such as `CustomObject` or `ContentDocument`	Small, 29.0	23.0
`url`	String	Relative URL to this record	Small, 29.0	23.0

Record Summary List

Summary information about a collection of records in the organization, including custom objects.

Property	Type	Description	Filter Group and Version	Available Version
`records`		A collection of records. The collection can contain any of these response bodies: • File Summary • Group • Record Summary • User Summary	Small, 30.0	30.0
`url`	String	The URL to this collection of records.	Small, 30.0	30.0

Record View

A view of any record in the organization, including a custom object record. This object is used if a specialized object, such as User or ChatterGroup, is not available for the record type. Contains data and metadata so you can render a record with one response.

Property	Type	Description	Filter Group and Version	Available Version
`id`	String	18-character ID of the object	Small, 29.0	23.0
`motif`	Motif	Small, medium, and large icons indicating the type of the object. The motif can also contain the object's base color. The icon is not a preview of the file.	Small, 29.0	27.0
`mySubscription`	Reference	If the context user is following this file, this contains information about the subscription, otherwise returns `null`.	Medium, 29.0	23.0
`name`	String	The localized name	Small, 29.0	29.0
`relatedList Definitions`	List View Definition[]	The related list definitions (metadata) for this object. These list definitions do not include the related list data. .	Medium, 29.0	29.0
`sections`	Record View Section[]	A list of record view sections.	Small, 29.0	29.0
`type`	String	A fixed type name for this object. Check this value to determine the shape of a response body.	Small, 29.0	23.0
`url`	String	The Chatter REST API URL resource for this object.	Small, 29.0	23.0

Record View Section

A section of record fields and values on a record detail.

Property	Type	Description	Filter Group and Version	Available Version
`columnCount`	String	The number of columns to use to lay out the fields in a record section.	Small, 29.0	29.0
`columnOrder`	String	The order of the fields to use in the `fields` property to lay out the fields in a record section. • `LeftRight`—Fields are rendered from left to right. • `TopDown`—Fields are rendered from the top down.	Small, 29.0	29.0

Property	Type	Description	Filter Group and Version	Available Version
`fields`		The fields and values for the record contained in this section. A collection of these response bodies: • Record Field • Record Field: Blank • Record Field: Compound • Record Field: Currency • Record Field: Date • Record Field: Percent • Record Field: Picklist • Record Field: Reference • Record Field: Reference with Date	Small, 29.0	29.0
`heading`	String	A localized label to display when rendering this section of fields.	Small, 29.0	29.0
`isCollapsible`	Boolean	Indicates whether the section can be collapsed to hide all the fields (`true`) or not (`false`).	Small, 29.0	29.0

Reference

A reference to a record.

Property	Type	Description	Filter Group and Version	Available Version
`id`	String	The ID of the record being referenced, which could be an 18-character ID or some other string identifier.	Small, 29.0	23.0
`url`	String	The URL to the resource endpoint.	Small, 29.0	23.0

Repository File Detail

Detailed description of a repository file.

The repository file detail response body includes all of the of the Repository File Summary response body.

Repository File Summary

Description of a repository file.

Property Name	Type	Description	Filter Group and Version	Available Version
`checkinComment`	String	The checkin comment of the file.	Small, 32.0	32.0
`contentSize`	Integer	Length in bytes of the content of the file.	Small, 32.0	32.0
`createdBy`	String	Name of user who created the file.	Small, 32.0	32.0
`createdDate`	Date	Created date of the file.	Small, 32.0	32.0
`description`	String	File description.	Small, 32.0	32.0
`downloadUrl`	String	URL to the repository file content.	Small, 32.0	32.0
`external ContentUrl`	String	URL of this file's content in the external system.	Small, 32.0	32.0
`external DocumentUrl`	String	URL of this file in the external system.	Small, 32.0	32.0
`id`	String	ID of the file.	Small, 32.0	32.0
`mimeType`	String	Mime type of the file.	Small, 32.0	32.0
`modifiedBy`	String	Name of user who last modified the file.	Small, 32.0	32.0
`modifiedDate`	Date	Modified date of the file.	Small, 32.0	32.0
`motif`	Motif	Motif of the file.	Small, 32.0	32.0
`name`	String	Name of the file.	Small, 32.0	32.0
`repository`	Reference	The file external repository.	Small, 32.0	32.0
`title`	String	Title of the file.	Small, 32.0	32.0
`type`	String	Record type, `file`.	Small, 32.0	32.0
`url`	String	Chatter REST API URL for this file.	Small, 32.0	32.0
`versionId`	String	ID of the file version in the external system.	Small, 32.0	32.0

Repository Folder Item

Description of a folder item.

Property Name	Type	Description	Filter Group and Version	Available Version
`file`	Repository File Summary	If this folder item represents a file, the file summary. If this folder item represents a folder, `null`.	Small, 32.0	32.0

Property Name	Type	Description	Filter Group and Version	Available Version
`folder`	Repository Folder Summary	If this folder item represents a folder, the folder summary. If this folder item represents a file, `null`.	Small, 32.0	32.0
`type`	String	The type of folder item: `folder` or `file`.	Small, 32.0	32.0

Repository Folder Items Collection

A collection of repository folder items.

Property Name	Type	Description	Filter Group and Version	Available Version
`currentPageUrl`	String	Chatter REST API URL identifying the current page.	Medium, 32.0	32.0
`items`	Repository Folder Item[]	Collection of items in a repository folder.	Small, 32.0	32.0
`nextPageUrl`	String	Chatter REST API URL identifying the next page or `null` if there isn't a next page. Check whether this value is `null` before getting another page. If a page doesn't exist, a `ConnectApi.NotFoundException` error is returned.	Small, 32.0	32.0
`previousPageUrl`	String	Chatter REST API URL identifying the previous page or `null` if there isn't a previous page.	Medium, 32.0	32.0

Repository Folder Summary

Description of a repository folder.

Property Name	Type	Description	Filter Group and Version	Available Version
`createdBy`	String	Name of user who created the folder.	Small, 32.0	32.0
`createdDate`	Date	Created date of the folder.	Small, 32.0	32.0
`description`	String	Folder description.	Small, 32.0	32.0
`externalFolderUrl`	String	Chatter REST API URL for this folder in the external system.	Small, 32.0	32.0

Property Name	Type	Description	Filter Group and Version	Available Version
`folderItemsUrl`	String	Chatter REST API URL that lists the files and folders in this folder.	Small, 32.0	32.0
`id`	String	ID of the folder.	Small, 32.0	32.0
`modifiedBy`	String	Name of user who last modified the folder.	Small, 32.0	32.0
`modifiedDate`	Date	Modified date of the folder.	Small, 32.0	32.0
`motif`	Motif	Motif of the folder.	Small, 32.0	32.0
`name`	String	Name of the folder.	Small, 32.0	32.0
`path`	String	The absolute path of the folder in the external system.	Small, 32.0	32.0
`repository`	Reference	The folder external repository.	Small, 32.0	32.0
`type`	String	Record type, `folder`.	Small, 32.0	32.0
`url`	String	Chatter REST API URL for this folder.	Small, 32.0	32.0

Reputation

Reputation for a user.

Property Name	Type	Description	Filter Group and Version	Available Version
`reputationLevel`	Reputation Level	User's reputation level.	Small, 32.0	32.0
`reputationPoints`	Double	User's reputation points, which can be earned by performing different activities in the community.	Small, 32.0	32.0
`url`	String	A Chatter REST API URL to the reputation.	Small, 32.0	32.0

Reputation Level

Reputation level for a user.

Property Name	Type	Description	Filter Group and Version	Available Version
`levelImageUrl`	String	URL to the reputation level image.	Small, 32.0	32.0
`levelName`	String	Name of the reputation level.	Small, 32.0	32.0

Property Name	Type	Description	Filter Group and Version	Available Version
`levelNumber`	Integer	Reputation level number, which is the numerical rank of the level, with the lowest level at 1. Administrators define the reputation level point ranges.	Small, 32.0	32.0

Request Header

An HTTP request header name and value pair.

Property Name	Type	Description	Filter Group and Version	Available Version
`name`	String	The name of the request header.	Small, 33.0	33.0
`value`	String	The value of the request header.	Small, 33.0	33.0

Scheduled Recommendation

Represents a scheduled recommendation.

Property Name	Type	Description	Filter Group and Version	Available Version
`enabled`	Boolean	Indicates whether scheduling is enabled. If `true`, the recommendation is enabled and appears in communities. If `false`, recommendations in feeds in the Salesforce1 mobile browser app aren't removed, but no new recommendations appear. In communities using the Summer '15 or later version of the Napili template, disabled recommendations no longer appear.	Small, 34.0	34.0
`id`	String	18-character ID of the scheduled recommendation.	Small, 33.0	33.0
`rank`	Integer	The rank determining the order of this scheduled recommendation.	Small, 33.0	33.0
`recommendationDefinitionRepresentation`	Recommendation Definition	Recommendation definition that this scheduled recommendation schedules.	Small, 33.0	33.0
`url`	String	URL to the Chatter REST API resource for the scheduled recommendation.	Small, 33.0	33.0

Scheduled Recommendation Collection

Represents a list of scheduled recommendations.

Property Name	Type	Description	Filter Group and Version	Available Version
`scheduledRecommendations`	Scheduled Recommendation[]	A list of scheduled recommendations.	Small, 33.0	33.0
`url`	String	URL to the Chatter REST API resource for the scheduled recommendation collection.	Small, 33.0	33.0

Shares

Property	Type	Description	Filter Group and Version	Available Version
`entity`	Depends on sharing	Returns information about the user, record, or group the file is shared with. Value is one of the following: • Group • Record Summary • User Summary	Small, 29.0	24.0
`sharingType`	String	The permission level of sharing. Values are: • `C`—collaborator • `I`—sharing permission is determined by location. This generally occurs when the sharing permission is not set on the file, but on the library instead. This value is also used when the context user is the same as owner of the file. • `N`—no access • `V`—viewer	Small, 29.0	24.0

Subscription

Property	Type	Description	Filter Group and Version	Available Version
`community`	Reference	Information about the community. Returns `null` if subscription exists in default community or communities aren't enabled.	Medium, 29.0	26.0
`id`	String	18-character ID of the subscription	Small, 29.0	23.0

Property	Type	Description	Filter Group and Version	Available Version
`subject`	Depends	Information about the parent, that is, the thing or person being followed. Valid values are: • Group (23.0) • Record Summary (23.0) • Topic (29.0) • User Summary (23.0)	Small, 29.0	
`subscriber`	Depends	Information about the subscriber, that is, the person following this item. Valid values are: • Group • User Summary	Small, 29.0	23.0
`url`	String	Relative URL to this specific subscription	Small, 29.0	23.0

Subscription Page

Property	Type	Description	Filter Group and Version	Available Version
`currentPageUrl`	String	Returns the current page of subscriptions	Medium, 29.0	23.0
`nextPageUrl`	String	Returns the next page of subscriptions. If there is no next page, returns the string "null"	Small, 29.0	23.0
`mySubscription`	Reference	Returns information about the subscription, or `null` the context user is not subscribed.	Medium, 29.0	23.0
`previousPageUrl`	String	Returns the previous page of subscriptions. If there is no previous page, returns the string "null"	Medium, 29.0	23.0
`subscriptions`	Subscription[]	Collection of subscriptions	Small, 29.0	23.0
`total`	Integer	Total number of subscriptions	Small, 29.0	23.0

Time Zone

The user's time zone as selected in My Settings in Salesforce. This value does not reflect a device's current location.

Property	Type	Description	Filter Group and Version	Available Version
`gmtOffset`	Double	Signed offset, in hours, from GMT	Small, 30.0	30.0
`name`	String	Display name of this time zone	Small, 30.0	30.0

Topic

Property	Type	Description	Filter Group and Version	Available Version
`createdDate`	String	ISO8601 date string, for example, 2011-02-25T18:24:31.000Z	Big, 29.0	28.0
`description`	String	Description of the topic	Big, 29.0	28.0
`id`	String	18-character ID of the topic	Small, 29.0	28.0
`images`	Topic Images	Images associated with the topic	Medium, 32.0	32.0
`isBeingDeleted`	Boolean	`true` if the topic is currently being deleted; `false` otherwise. After the topic is deleted, when attempting to retrieve the topic, the response is 404: Not Found.	Small, 33.0	33.0
`name`	String	Name of the topic	Small, 29.0	24.0
`talkingAbout`	Integer	Number of people talking about this topic over the last two months, based on factors such as topic additions and comments on posts with the topic	Medium, 29.0	28.0
`url`	String	URL to the topic detail page	Small, 29.0	29.0

Topic Collection

Property	Type	Description	Filter Group and Version	Available Version
`currentPageUrl`	String	Returns the current page of topics	Medium, 32.0 Small, 29.0–31.0	29.0
`nextPageUrl`	String	Returns the next page of topics. If there is no next page, returns the string "null"	Small, 29.0	29.0
`topics`	Topic[]	Collection of topics	Small, 29.0	24.0

Topic Endorsement

Represents one user endorsing another user for a single topic.

Property	Type	Description	Filter Group and Version	Available Version
`endorsee`	User Summary	User being endorsed	Small, 30.0	30.0
`endorsementId`	String	ID of the endorsement record	Small, 30.0	30.0
`endorser`	User Summary	User performing the endorsement	Small, 30.0	30.0
`topic`	Topic	Topic the user is being endorsed for	Small, 30.0	30.0
`url`	String	Chatter REST API resource URL for the endorsement record	Small, 30.0	30.0

Topic Endorsement Collection

A collection of topic endorsement response bodies.

Property	Type	Description	Filter Group and Version	Available Version
`currentPageUrl`	String	Chatter REST API URL identifying the current page.	Medium, 30.0	30.0
`topicEndorsements`	Topic Endorsement[]	Collection of topic endorsement response bodies	Small, 30.0	30.0
`nextPageUrl`	String	Chatter REST API URL identifying the next page or `null` if there isn't a next page. Check whether this value is `null` before getting another page. If a page doesn't exist, a `ConnectApi.NotFoundException` error is returned.	Small, 30.0	30.0
`previousPageUrl`	String	Chatter REST API URL identifying the previous page or `null` if there isn't a previous page.	Medium, 30.0	30.0

Topic Images

Images associated with a topic.

Property Name	Type	Description	Filter Group and Version	Available Version
`coverImageUrl`	String	URL to a topic's cover image, which appears on the topic page. Both topics and managed topics can have cover images.	Small, 32.0	32.0
`featuredImageUrl`	String	URL to a managed topic's featured image, which appears wherever you feature it, for example, on the communities home page.	Small, 32.0	32.0

Topic Opt Out

An "opt out" from a topic. Indicates that a user doesn't want to be identified as knowledgeable about a topic.

Property	Type	Description	Filter Group and Version	Available Version
`id`	String	ID of the topic opt out record	Small, 30.0	30.0
`reason`	String	The reason the user is opting out of the topic. The only valid value is `topicInfluenceHide`.	Small, 30.0	30.0
`topic`	Topic	Topic the user opted out of	Small, 30.0	30.0
`url`	String	Chatter REST API resource URL for the topic opt out	Small, 30.0	30.0
`user`	User Summary	User that opted out of the topic	Small, 30.0	30.0

Topic Opt Out Collection

A collection of topic opt out response bodies.

Property	Type	Description	Filter Group and Version	Available Version
`currentPageUrl`	String	Chatter REST API URL identifying the current page.	Medium, 30.0	30.0
`topicOptOuts`	Topic Opt Out[]	Collection of topic opt out response bodies.	Small, 30.0	30.0

Topic Suggestion

Property	Type	Description	Filter Group and Version	Available Version
`existingTopic`	Topic	Topic that already exists or `null` for a new topic	Small, 29.0	28.0
`name`	String	Name of the topic	Small, 29.0	28.0

Topic Suggestion Collection

Property	Type	Description	Filter Group and Version	Available Version
`topicSuggestions`	Topic Suggestion[]	Collection of topic suggestion response bodies	Small, 29.0	28.0

Topics Capability

If a feed element has this capability, the context user can add topics to it. Topics help users organize and discover conversations.

Property Name	Type	Description	Filter and Version	Available Version
`canAssignTopics`	Boolean	`true` if a topic can be assigned to the feed element, `false` otherwise.	Small, 32.0	32.0
`items`	Topic[]	A collection of topics associated with this feed element.	Small, 32.0	32.0

Topics People Are Knowledgeable About

Represents a topic that a specified user is knowledgeable about.

Property	Type	Description	Filter Group and Version	Available Version
`isEndorsedByMe`	Boolean	Specifies if the knowledgeable user is endorsed for the topic by the context user, `true` if so, `false` otherwise	Small, 30.0	30.0
`knowledgeableId`	String	ID of the knowledgeable record	Small, 30.0	30.0
`numEndorsements`	Integer	Number of endorsements received by the knowledgeable user for the topic	Medium, 30.0	30.0
`reasons`	String	Specifies the reasons why the user is considered knowledgeable about the topic. Valid values are: • `Influential`—User was mentioned in posts or in comments on posts with the topic or received likes on comments on posts with the topic. • `Endorsed`—User received endorsement from another user for the topic.	Medium, 30.0	30.0
`topic`	Topic	Topic that the user is knowledgeable about	Small, 30.0	30.0

Topics People Are Knowledgeable About Collection

A collection of topics people are knowledgeable about response bodies.

Property	Type	Description	Filter Group and Version	Available Version
`currentPageUrl`	String	Chatter REST API URL identifying the current page.	Medium, 30.0	30.0

Property	Type	Description	Filter Group and Version	Available Version
`knowledgeableAboutTopic`	Topics People Are Knowledgeable About[]	Collection of topics people are knowledgeable about response bodies	Small, 30.0	30.0
`nextPageUrl`	String	Chatter REST API URL identifying the next page or `null` if there isn't a next page. Check whether this value is `null` before getting another page. If a page doesn't exist, a `ConnectApi.NotFoundException` error is returned.	Small, 30.0	30.0
`previousPageUrl`	String	Chatter REST API URL identifying the previous page or `null` if there isn't a previous page.	Medium, 30.0	30.0

Tracked Change Bundle Capability

If a feed element has this capability, it has a group of other feed elements aggregated into one feed element called a *bundle*. This type of bundle aggregates feed tracked changes.

Note: Bundled posts contain feed-tracked changes. In Salesforce1 downloadable apps, bundled posts are in record feeds only.

Property Name	Type	Description	Filter Group and Version	Available Version
`changes`	Feed Tracked Change[]	Collection of feed tracked changes.	Small, 31.0	31.0
`bundleType`	String	Specifies the type of bundle. The value is `TrackedChanges`.	Small, 31.0	31.0
`page`	Feed Element Page	A collection of feed elements.	Small, 31.0	31.0
`totalElements`	Integer	The total number of feed elements that this bundle aggregates.	Small, 31.0	31.0

Tracked Changes Capability

If a feed element has this capability, it contains all changes to a record for a single tracked change event.

Property Name	Type	Description	Filter Group and Version	Available Version
`changes`	Feed Tracked Change[]	Collection of feed tracked changes.	Small, 32.0	32.0

Unauthenticated User

Property	Type	Description	Filter Group and Version	Available Version
`name`	String	Either the name in the `From` field of the source email, or the email address from the source email.	Small, 29.0	24.0
`type`	String	`UnauthenticatedUser`	Small, 29.0	24.0

User Capabilities

Property	Type	Description	Filter Group and Version	Available Version
`canChat`	Boolean	Specifies if the context user can use Chatter Messenger with the subject user (`true`) or not (`false`)	Small, 29.0	29.0
`canDirectMessage`	Boolean	Specifies if the context user can direct message the subject user (`true`) or not (`false`)	Small, 29.0	29.0
`canEdit`	Boolean	Specifies if the context user can edit the subject user's account (`true`) or not (`false`)	Small, 29.0	29.0
`canFollow`	Boolean	Specifies if the context user can follow the subject user's feed (`true`) or not (`false`)	Small, 29.0	29.0
`canViewFeed`	Boolean	Specifies if the context user can view the feed of the subject user (`true`) or not (`false`)	Small, 29.0	29.0
`canView FullProfile`	Boolean	Specifies if the context user can view the full profile of the subject user (`true`) or only the limited profile (`false`)	Small, 29.0	29.0
`isModerator`	Boolean	Specifies if the subject user is a Chatter moderator or admin (`true`) or not (`false`)	Small, 29.0	29.0

User Chatter Settings

A user's global Chatter settings.

Property	Type	Description	Filter Group and Version	Available Since
`defaultGroup EmailFrequency`	String	The default frequency with which a user receives email from a group when they join it. Values are: • `EachPost` • `DailyDigest`	Small, 29.0	27.0

Property	Type	Description	Filter Group and Version	Available Since
		• `WeeklyDigest` • `Never` In Communities, the Email on Each Post option is disabled once a group exceeds 10,000 members. All members who had this option selected are automatically switched to Daily Digest.		

User Detail

If a context user doesn't have permission to see a property, the property is set to `null`.

Property	Type	Description	Filter Group and Version	Available Version
`aboutMe`	String	Text from user's profile	Big, 29.0	23.0
`additionalLabel`	String	An additional label for the user, for example, "Customer," "Partner," or "Acme Corporation." If the user doesn't have an additional label, the value is `null`.	Small, 30.0	30.0
`address`	Address	Address	Big, 29.0	23.0
`chatterActivity`	Chatter Activity	Chatter activity statistics	Big, 29.0	23.0
`chatterInfluence`	Chatter Influence	User's influence rank	Big, 29.0	23.0
`communityNickname`	String	User's nickname in the community	Medium, 32.0	32.0
`companyName`	String	Company name	Small, 29.0	23.0
`currentStatus`	User Status	Note: This property is no longer available as of version 25.0. User's current status		23.0–25.0
`displayName`	String	User's name that is displayed in the community. If nicknames are enabled, the nickname is displayed. If nicknames aren't enabled, the full name is displayed.	Small, 32.0	32.0
`email`	String	User's email address	Medium, 29.0	23.0
`firstName`	String	User's first name	Small, 29.0	23.0
`followersCount`	Integer	Number of users following this user	Big, 29.0	23.0
`followingCounts`	Following Counts	Information about items the user is following	Big, 29.0	23.0
`groupCount`	Integer	Number of groups user is following	Big, 29.0	23.0

Property	Type	Description	Filter Group and Version	Available Version
`hasChatter`	Boolean	`true` if user has access to Chatter; `false` otherwise	Small, 31.0	31.0
`id`	String	18-character ID of the user	Small, 29.0	23.0
`isActive`	Boolean	`true` if user is active, `false` otherwise	Small, 29.0	23.0
`isChatterGuest`	Boolean	Note: This property is no longer available as of version 26.0. Use the `userType` property instead. `true` if user is a Chatter customer, `false` otherwise		23.0–25.0
`isInThis Community`	Boolean	`true` if user is in the same community as the context user; `false` otherwise. Can be `false` for record feed @mentions on posts with the scope "All Users."	Small, 29.0	28.0
`lastName`	String	User's last name	Small, 29.0	23.0
`managerId`	String	18-character ID of the user's manager	Big, 29.0	23.0
`managerName`	String	Locale-based concatenation of manager's first and last names	Big, 29.0	23.0
`motif`	Motif	Small, medium, and large icons indicating that the object is a user. The motif can also contain the object's base color. The motif icon is not the user's profile picture.	Small, 29.0	27.0
`mySubscription`	Reference	If the context user is following this user, this contains information about the subscription, else returns `null`	Medium, 29.0	23.0
`name`	String	Locale-based concatenation of user's first and last names	Small, 29.0	23.0
`phoneNumbers`	Phone Number[]	Collection of user's phone numbers	Big, 29.0	23.0
`photo`	Photo	Information about the user's photos	Small, 29.0	23.0
`reputation`	Reputation	Reputation of the user in the community, which is determined by the number of points earned from events and actions performed in the community	Big, 32.0	32.0
`thanksReceived`	Integer	Number of thanks the user has received	Big, 29.0	29.0
`title`	String	Title of the user	Small, 29.0	23.0
`type`	String	`User`	Small, 29.0	23.0
`url`	String	URL to the user's Chatter profile	Small, 29.0	23.0
`username`	String	User name of the user, such as *`Admin@mycompany.com`*	Small, 29.0	24.0

Property	Type	Description	Filter Group and Version	Available Version
`userType`	String	Specifies the type of user. Valid values are: • `ChatterGuest`—User is an external user in a private group. • `ChatterOnly`—User is a Chatter Free customer. • `Guest`—User is unauthenticated. • `Internal`—User is a standard organization member. • `Portal`—User is an external user in a customer portal, partner portal, or community. • `System`—User is Chatter Expert or a system user. • `Undefined`—User is a user type that is a custom object.	Medium, 29.0	26.0

SEE ALSO:

User Summary

User Group Page

A paginated collection of groups a user is a member of.

Property	Type	Description	Filter Group and Version	Available Version
`currentPageUrl`	String	URL to the current page	Medium, 29.0	23.0
`groups`	Group[]	Collection of groups for a user	Small, 29.0	23.0
`nextPageUrl`	String	URL to the next page, if there is no next page, returns the string “null”	Small, 29.0	23.0
`previousPageUrl`	String	URL to the previous page, if there is no previous page, returns the string “null”	Medium, 29.0	23.0
`total`	Integer	Total number of groups a user is a member of. This number is always the total number of groups a user is a member of, even when the request is a search and the response contains a subset of that total.	Small, 29.0	23.0

User Page

Property	Type	Description	Filter Group and Version	Available Version
`currentPageUrl`	String	URL to the current page	Medium, 29.0	23.0
`nextPageUrl`	String	URL to the next page, if there is no next page, returns `null`	Small, 29.0	23.0
`previousPageUrl`	String	URL to the previous page, if there is no previous page, returns `null`	Medium, 29.0	23.0
`users`	User Detail[]	Collection of user detail information.	Small, 29.0	24.0

User Profile

Property	Type	Description	Filter Group and Version	Available Version
`capabilities`	User Capabilities	The context user's capabilities specific to the subject user's profile	Medium, 29.0	29.0
`id`	String	The ID of the user attached to the profile	Small, 29.0	29.0
`tabs`	User Profile Tab[]	The tabs visible to the context user for that user type	Big, 29.0	29.0
`url`	String	The URL of the user's profile	Small, 29.0	29.0
`userDetail`	User Detail	The details about the user attached to the profile	Small, 29.0	29.0

User Profile Tab

Property	Type	Description	Filter Group and Version	Available Version
`id`	String	The tab's unique identifier or 18–character Id	Small, 29.0	29.0
`isDefault`	Boolean	Specifies if the tab appears first when clicking the user profile (`true`) or not (`false`)	Small, 29.0	29.0
`tabType`	String	Specifies the type of user profile tab: • `CustomVisualForce`—Tab that displays data from a Visualforce page. • `CustomWeb`—Tab that displays data from any external web-based application or web page. • `Element`—Tab that displays generic content inline.	Small, 29.0	29.0

Property	Type	Description	Filter Group and Version	Available Version
		• `Feed`—Tab that displays the Chatter feed. • `Overview`—Tab that displays user details.		
`tabUrl`	String	The current tab's content URL (for non built-in tab types)	Small, 29.0	29.0

User Settings

Settings specific to a user.

Property	Type	Description	Filter Group and Version	Available Version
`approvalPosts`	Boolean	User can approve workflows from Chatter posts.	Small, 29.0	27.0
`canFollow`	Boolean	The context user is allowed to follow things.	Small, 29.0	24.0
`canModifyAllData`	Boolean	User has "Modify all Data" permission	Small, 29.0	24.0
`canOwnGroups`	Boolean	User can own groups	Small, 29.0	24.0
`canViewAllData`	Boolean	User has "View all Data" permission	Small, 29.0	24.0
`canViewAllGroups`	Boolean	User has "View all Groups" permission	Small, 29.0	24.0
`canViewAllUsers`	Boolean	User has "View all Users" permission	Small, 29.0	24.0
`canViewCommunitySwitcher`	Boolean	User can see the community switcher menu.	Small, 34.0	34.0
`canViewFullUserProfile`	Boolean	User can see of other user's Chatter profiles	Small, 29.0	24.0
`canViewPublicFiles`	Boolean	User can see all files marked as public	Small, 29.0	24.0
`currencySymbol`	String	Currency symbol to use for displaying currency values. Applicable only when the `multiCurrency` property of the Features response body is `false`.	Small, 29.0	28.0
`externalUser`	Boolean	User is a Chattercustomer	Small, 29.0	24.0
`fileSyncLimit`	Integer	Maximum number of files user can sync	Small, 32.0	32.0
`fileSyncStorageLimit`	Integer	Maximum storage for synced files, in megabytes (MB)	Small, 29.0	29.0
`folderSyncLimit`	Integer	Maximum number of folders user can sync	Small, 32.0	32.0

Property	Type	Description	Filter Group and Version	Available Version
`hasAccessToInternalOrg`	Boolean	User is a member of the internal organization	Small, 29.0	26.0
`hasChatter`	Boolean	User has access to Chatter	Small, 31.0	31.0
`hasFileSync`	Boolean	User has "Sync Files" permission	Small, 29.0	26.0
`hasFileSyncManagedClientAutoUpdate`	Boolean	Administrator for the user's organization allows file sync clients to update automatically.	Small, 34.0	34.0
`hasRestDataApiAccess`	Boolean	User has access to REST API.	Small, 29.0	29.0
`timeZone`	Time Zone	The user's time zone as selected in My Settings in Salesforce. This value does not reflect a device's current location.	Small, 30.0	30.0
`userDefaultCurrencyIsoCode`	String	The ISO code for the default currency. Applicable only when the `multiCurrency` property of the Features response body is `true`.	Small, 29.0	28.0
`userId`	String	18-character ID of the user	Small, 29.0	24.0
`userLocale`	String	Locale of user	Small, 29.0	24.0

User Status

Property	Type	Description	Filter Group and Version	Available Version
`body`	Feed Item Body[]	Body of the post	Small, 29.0	23.0–25.0
`parentId`	String	18-character ID	Small, 29.0	23.0–25.0
`url`	String	URL to this user status post	Small, 29.0	23.0–25.0

User Summary

The User Summary response body is a subset of the User Detail response body.

Property	Type	Description	Filter Group and Version	Available Version
`additionalLabel`	String	An additional label for the user, for example, "Customer," "Partner," or "Acme Corporation." If the user doesn't have an additional label, the value is `null`.	Small, 30.0	30.0

Property	Type	Description	Filter Group and Version	Available Version
`communityNickname`	String	User's nickname in the community	Medium, 32.0	32.0
`companyName`	String	Name of the company	Small, 29.0	23.0
`displayName`	String	User's name that is displayed in the community. If nicknames are enabled, the nickname is displayed. If nicknames aren't enabled, the full name is displayed.	Small, 32.0	32.0
`firstName`	String	User's first name	Small, 29.0	23.0
`id`	String	18-character ID	Small, 29.0	23.0
`isActive`	Boolean	`true` if user is active, `false` otherwise.	Small, 29.0	25.0
`isChatterGuest`	Boolean	Note: This property is no longer available as of version 26.0. Use the `userType` property instead. `true` if user is a Chatter customer, `false` otherwise		23.0–26.0
`isInThisCommunity`	Boolean	`true` if user is in the same community as the context user; `false` otherwise. Can be `false` for record feed @mentions on posts with the scope "All Users."	Small, 29.0	28.0
`lastName`	String	User's last name	Small, 29.0	23.0
`motif`	Motif	Small, medium, and large icons indicating that the object is a user. The motif can also contain the object's base color. The motif icon is not the user's profile photo.	Small, 29.0	27.0
`mySubscription`	Reference	If the context user is following this user, this contains information about the subscription, else returns `null`	Medium, 29.0	23.0
`name`	String	Locale-based concatenation of first and last names	Small, 29.0	23.0
`photo`	Photo	Information about the user's photo	Small, 29.0	23.0
`reputation`	Reputation	Reputation of the user in the community, which is determined by the number of points earned from events and actions performed in the community	Big, 32.0	32.0
`title`	String	Title of the user	Small, 29.0	23.0
`type`	String	`User`	Small, 29.0	23.0
`url`	String	URL to the user's Chatter profile	Small, 29.0	23.0
`userType`	String	Specifies the type of user. Valid values are: • `ChatterGuest`—User is an external user in a private group.	Medium, 29.0	26.0

Property	Type	Description	Filter Group and Version	Available Version
		• `ChatterOnly`—User is a Chatter Free customer. • `Guest`—User is unauthenticated. • `Internal`—User is a standard organization member. • `Portal`—User is an external user in a customer portal, partner portal, or community. • `System`—User is Chatter Expert or a system user. • `Undefined`—User is a user type that is a custom object.		

Example:

Note: If the user's name is Chatter Expert and the return contains mostly null values, it indicates that the Chatter Expert user is either deactivated or taken over by an administrator, or the organization was created before Salesforce introduced the Chatter Expert user. This is a special user that offers tips, tricks, and sample posts to help users get started with Chatter.

SEE ALSO:

User Detail

Zone

A Chatter Answers zone.

Property	Type	Description	Filter Group and Version	Available Version
`description`	String	The description of the zone	Big, 29.0	29.0
`id`	String	The zone ID	Small, 29.0	29.0
`isActive`	Boolean	Indicates that the zone is active	Small, 29.0	29.0
`isChatterAnswers`	Boolean	Indicates that the zone is available for Chatter Answers	Small, 29.0	29.0
`name`	String	Name of the zone	Small, 29.0	29.0
`url`	String	Chatter REST API resource URL for the zone	Small, 30.0	30.0
`visibility`	String	Zone visibility type. One of these values: • `Community`—Available in a community. • `Internal`—Available internally only. • `Portal`—Available in a portal.	Small, 29.0	29.0

Property	Type	Description	Filter Group and Version	Available Version
`visibilityId`	String	If the zone is available in a portal or a community, this property contains the ID of the portal or community. If the zone is available to all portals, this property contains the value `All`.	Small, 29.0	29.0

Zone Page

A page of Chatter Answers zones.

Property	Type	Description	Filter Group and Version	Available Version
`zones`	Zone[]	Collection of one or more zones	Small, 29.0	29.0
`currentPageUrl`	String	Chatter REST API URL identifying the current page.	Medium, 29.0	29.0
`nextPageUrl`	String	Chatter REST API URL identifying the next page or `null` if there isn't a next page. Check whether this value is `null` before getting another page. If a page doesn't exist, a `ConnectApi.NotFoundException` error is returned.	Small, 29.0	29.0

Zone Search Page

A page of results from a search within a Chatter Answers zone.

Property	Type	Description	Filter Group and Version	Available Version
`currentPageToken`	String	Token identifying the current page.	Medium, 29.0	29.0
`currentPageUrl`	String	Chatter REST API URL identifying the current page.	Medium, 29.0	29.0
`items`	Zone Search Result[]	Collection of search results	Small, 29.0	29.0
`currentPageToken`	String	Token identifying the next page or `null` if there isn't a next page.	Small, 29.0	29.0
`nextPageUrl`	String	Chatter REST API URL identifying the next page or `null` if there isn't a next page. Check whether this value is `null` before getting another page. If a page doesn't exist, a `ConnectApi.NotFoundException` error is returned.	Small, 29.0	29.0

Zone Search Result

The result of a search within a Chatter Answers zone.

Property	Type	Description	Filter Group and Version	Available Version
`hasBestAnswer`	Boolean	Indicates whether or not the search result has a best answer	Small, 29.0	29.0
`id`	String	ID of the search result. The search result can be a question or an article.	Small, 29.0	29.0
`title`	String	Title of the search result	Small, 29.0	29.0
`type`	String	Type of the search result. One of these values: • `Article`—Search results contain only articles. • `Question`—Search results contain only questions.	Small, 29.0	29.0
`voteCount`	String	Number of votes given to the search result	Small, 29.0	29.0

CHAPTER 8 Chatter REST API Appendices

This chapter contains appendix material.

IN THIS SECTION:

Action Links Labels
Use these labels for action link buttons.

Action Links Labels

Use these labels for action link buttons.

An action link is a button on a feed element. Clicking an action link can take a user to a Web page, initiate a file download, or invoke an API call to Salesforce or to an external server. An action link includes a URL and an HTTP method, and can include a request body and header information, such as an OAuth token for authentication. Use action links to integrate Salesforce and third-party services into the feed so that users can take action to drive productivity and accelerate innovation.

Specify the key in the `labelKey` property of the Action Link Definition Input request body. When the action link is rendered, the UI uses labels for the "New," "Pending," "Success," and "Failed" states as needed.

Key	New	Pending	Success	Failed
Accept	Accept	Acceptance Pending	Accepted	Acceptance Failed
Activate	Activate	Activation Pending	Activated	Activation Failed
Add	Add	Add Pending	Added	Add Failed
Add to Calendar	Add to Calendar	Add to Calendar Pending	Added to Calendar	Add to Calendar Failed
Add to Cart	Add to Cart	Add Pending	Added	Add Failed
Agree	Agree	Agree Pending	Agree	Agree Failed
Alert	Alert	Alert Pending	Alerted	Alert Failed
Answer	Answer	Answer Pending	Answered	Answer Failed
Approve	Approve	Approval Pending	Approved	Approval Failed
Assign	Assign	Assign Pending	Assigned	Assign Failed
Assist	Assist	Assistance Pending	Assisted	Assistance Failed
Attach	Attach	Attach Pending	Attached	Attach Failed
Authorize	Authorize	Authorization Pending	Authorized	Authorization Failed
Begin	Begin	Begin Pending	Started	Begin Failed
Book	Book	Book Pending	Booked	Book Failed

Key	New	Pending	Success	Failed
Buy	Buy	Buy Pending	Bought	Buy Failed
Call	Call	Call Pending	Called	Call Failed
Call Me	Call Me	Call Pending	Call Succeeded	Call Failed
Certify	Certify	Certifcation Pending	Certified	Certification Failed
Change	Change	Change Pending	Changed	Change Failed
Chat	Chat	Chat Pending	Chat Completed	Chat Failed
Check	Check	Check Pending	Checked	Check Failed
Clear	Clear	Clear Pending	Clear	Clear Failed
Clone	Clone	Clone Pending	Cloned	Clone Failed
Close	Close	Close Pending	Closed	Close Failed
Confirm	Confirm	Confirmation Pending	Confirmed	Confirmation Failed
Convert	Convert	Convert Pending	Converted	Convert Failed
Convert a Lead	Convert a Lead	Lead Conversion Pending	Lead Converted	Lead Conversion Failed
Create	Create	Create Pending	Created	Create Failed
Deactivate	Deactivate	Deactivation Pending	Deactivated	Deactivation Failed
Decline	Decline	Decline Pending	Declined	Decline Failed
Delete	Delete	Delete Pending	Deleted	Delete Failed
Deny	Deny	Denial Pending	Denied	Denial Failed
Detach	Detach	Detach Pending	Detached	Detach Failed
Disagree	Disagree	Disagree Pending	Disagree	Disagree Failed
Dislike	Dislike	Dislike Pending	Disliked	Dislike Failed
Dismiss	Dismiss	Dismissal Pending	Dismissed	Dismissal Failed
Do	Do	Do Response Pending	Do	Do Response Failed
Donate	Donate	Donation Pending	Donated	Donation Failed
Down	Down	Down Response Pending	Down	Down Response Failed
Download	Download	Download Pending	Downloaded	Download Failed
Edit	Edit	Edit Pending	Edited	Edit Failed
End	End	End Pending	Ended	End Failed
Endorse	Endorse	Endorsement Pending	Endorsed	Endorsement Failed
Enter	Enter	Enter Pending	Entered	Enter Failed

Key	New	Pending	Success	Failed
Escalate	Escalate	Escalation Pending	Escalated	Escalation Failed
Estimate	Estimate	Estimate Pending	Estimate	Estimate Failed
Exclude	Exclude	Exclude Pending	Excluded	Exclude Failed
Exit	Exit	Exit Pending	Exited	Exit Failed
Export	Export	Export Pending	Exported	Export Failed
File	File	File Pending	Filed	File Failed
Fill	Fill	Fill Pending	Filled	Fill Failed
Finish	Finish	Finish Pending	Finished	Finish Failed
Flag	Flag	Flag Pending	Flagged	Flag Failed
Flip	Flip	Flip Pending	Flipped	Flip Failed
Follow	Follow	Follow Pending	Followed	Follow Failed
Generate	Generate	Generate Pending	Generated	Generate Failed
Give	Give	Give Pending	Given	Give Failed
Help	Help	Help Pending	Helped	Help Failed
Hide	Hide	Hide Pending	Hidden	Hide Failed
High	High	High Response Pending	High	High Response Failed
Hold	Hold	Hold Pending	Hold Succeeded	Hold Failed
Import	Import	Import Pending	Imported	Import Failed
Include	Include	Include Pending	Included	Include Failed
Join	Join	Join Pending	Joined	Join Failed
Launch	Launch	Launch Pending	Launched	Launch Failed
Leave	Leave	Leave Pending	Left	Leave Failed
Like	Like	Like Pending	Liked	Like Failed
List	List	List Pending	Listed	List Failed
Log	Log	Log Pending	Logged	Log Failed
Log a Call	Log a Call	Log a Call Pending	Logged a Call	Log a Call Failed
Low	Low	Low Response Pending	Low	Low Response Failed
Mark	Mark	Mark Pending	Marked	Mark Failed
Maybe	Maybe	Maybe Response Pending	Maybe	Maybe Response Failed
Medium	Medium	Medium Response Pending	Medium	Medium Response Failed

Key	New	Pending	Success	Failed
Meet	Meet	Meet Pending	Meet	Meet Failed
Message	Message	Message Pending	Message	Message Failed
Move	Move	Move Pending	Moved	Move Failed
Negative	Negative	Negative Response Pending	Negative	Negative Response Failed
New	New	New Pending	New	New Failed
No	No	No Response Pending	No	No Response Failed
OK	OK	OK Response Pending	OK	OK Response Failed
Open	Open	Open Pending	Opened	Open Failed
Order	Order	Order Pending	Ordered	Order Failed
Positive	Positive	Positive Response Pending	Positive	Positive Response Failed
Post	Post	Post Pending	Posted	Post Failed
Post Review	Post Review	Post Pending	Posted	Post Failed
Process	Process	Process Pending	Processed	Process Failed
Provide	Provide	Provide Pending	Provided	Provide Failed
Purchase	Purchase	Purchase Pending	Purchased	Purchase Failed
Quote	Quote	Quote Pending	Quoted	Quote Failed
Receive	Receive	Receive Pending	Received	Receive Failed
Recommend	Recommend	Recommend Pending	Recommended	Recommend Failed
Redo	Redo	Redo Response Pending	Redo	Redo Response Failed
Refresh	Refresh	Refresh Pending	Refreshed	Refresh Failed
Reject	Reject	Rejection Pending	Rejected	Rejection Failed
Release	Release	Release Pending	Released	Release Failed
Remind	Remind	Reminder Pending	Reminded	Reminder Failed
Remove	Remove	Removal Pending	Removed	Removal Failed
Repeat	Repeat	Repeat Pending	Repeated	Repeat Failed
Report	Report	Report Pending	Reported	Report Failed
Request	Request	Request Pending	Requested	Request Failed
Reserve	Reserve	Reservation Pending	Reserved	Reservation Failed
Resolve	Resolve	Resolve Pending	Resolved	Resolve Failed

Key	New	Pending	Success	Failed
Respond	Respond	Response Pending	Responded	Response Failed
Restore	Restore	Restore Pending	Restored	Restore Failed
Review	Review	Review Pending	Reviewed	Review Failed
Revise	Revise	Revision Pending	Revised	Revision Failed
Save	Save	Save Pending	Saved	Save Failed
Schedule	Schedule	Schedule Pending	Scheduled	Schedule Failed
Sell	Sell	Sell Pending	Sold	Sell Failed
Send	Send	Send Pending	Sent	Send Failed
Send Email	Send Email	Send Email Pending	Email Sent	Send Email Failed
Share	Share	Share Pending	Shared	Share Failed
Ship	Ship	Shipment Pending	Shipped	Shipment Failed
Show	Show	Show Pending	Shown	Show Failed
Start	Start	Start Pending	Started	Start Failed
Stop	Stop	Stop Pending	Stopped	Stop Failed
Submit	Submit	Submit Pending	Submitted	Submit Failed
Subscribe	Subscribe	Subscribe Pending	Subscribed	Subscribe Failed
Test	Test	Test Pending	Tested	Test Failed
Thank	Thank	Thanks Pending	Thanked	Thanks Failed
Unauthorize	Unauthorize	Unauthorization Pending	Unauthorized	Unauthorization Failed
Uncheck	Uncheck	Uncheck Pending	Unchecked	Uncheck Failed
Undo	Undo	Undo Response Pending	Undo	Undo Response Failed
Unflag	Unflag	Unflag Pending	Unflagged	Unflag Failed
Unfollow	Unfollow	Unfollow Pending	Unfollowed	Unfollow Failed
Unlike	Unlike	Unlike Pending	Unliked	Unlike Failed
Unmark	Unmark	Unmark Pending	Unmarked	Unmark Failed
Unsubscribe	Unsubscribe	Unsubscribe Pending	Unsubscribed	Unsubscribe Failed
Up	Up	Up Response Pending	Up	Up Response Failed
Update	Update	Update Pending	Updated	Update Failed
Validate	Validate	Validate Pending	Validated	Validate Failed
Verify	Verify	Verify Pending	Verified	Verify Failed

Key	New	Pending	Success	Failed
View	View	View Pending	Viewed	View Failed
Visit	Visit	Visit Pending	Visit Successful	Visit Failed
Yes	Yes	Yes Response Pending	Yes	Yes Response Failed

INDEX

@ me resources 258

A

Access tokens 37
Action Link
    inputs 415
Action Link Definition
    inputs 407
Action Link Definition response body 465
Action Link Diagnostic 468
Action Link Group Definition
    inputs 412
Action link group definition resource 113
Action Link Group Definition response body 468
Action link group definitions resource 110
Action link group templates
    deleting 68
    editing 67
    packaging 68
Action Link Group Templates
    design 56
Action link groups resource 113
Action link resource 114
Action Link Template Binding Input 415
Action link templates
    creating 64
Action links
    invoking example 93
    updating status example 93
Action Links
    authentication 51
    create template example 90
    define example 87, 90
    labels 597
    overview 51
    post example 87, 90
    security 51
    templates 56
    use cases 49
    versioning 51
    working with 47
Action Links Diagnostic Information 117
Action links resources 109
Adding member to a group 307
Adding photo
    group 316
Adding photo *(continued)*
    user 389
Address response body 470
Announcement 470
Announcement input 416
Announcement Page 470
Announcements
    group 304
Announcements resource 117
Appendices 597
Approval Capability 471
Approval Post Template Field response body 472
Article Item 472
Associated Actions Capability
    inputs 417
Associated Actions Capability response body 472
Attach input
    canvas app 417
    existing content 418
    feed poll 419
    link 419
    new file upload 420
Authentication 2

B

Banner Capability Response Body 473
Batch Collection Input 421
Batch feed items 170
Batch information
    group-memberships resource 313
Batch Input 422
Batch requests
    batch input 440
    batch request input 441
Batch resource 119
Batch Result Item response body 473
Batch Results response body 474
Batching information
    groups 303
    users 369
Bearer token 38
Bookmarks
    feed items 182
    feed URL 180
    flat feed 184
    resources 179

Bookmarks Capability Input 423
Bookmarks Capability Response Body 474

C

Canvas apps
  attachment input 417
Canvas Capability 476
Canvas Capability Input 424
Case Comment Capability 475
Chatter activity
  get statistics for user example 100
Chatter Activity response body 476
Chatter Directory response body 477
Chatter feed elements 70
Chatter feed items 70
Chatter feeds 70
Chatter in Apex
  wildcards 19
Chatter Influence response body 477
Chatter Likes Capability Response Body 477
Chatter resource 124
Chatter REST API
  cURL 5
  Differences with other APIs 2
  examples 79
  named resources 4
  resources 6
  rich input 10
  tasks 79
  URLs 4
  using 2
ClientInfo response body 478
Comment Capabilities
  input 426
Comment Capabilities Response Body 479
Comment Page response body 480
Comment response body 478
Comments
  access 125
  delete 125
  editability 126
  editing example 95
  input 425
  likes 127
  on feed items 173
  posting example 83
  posting example with existing contnet 83
  resources 125
Comments Capability Response Body 480
Comments Capability, Editability 126
Communities
  building URL 4
  list of 128
  OAuth 44
  request example 81
  specific community information 129
  working with 68
Communities Home Feed
  resources 185
Communities moderation feed
  feed elements 187
  feed items 188
Communities Moderation Feed
  resources 186
Communities Moderation resources 129
Communities resources 128
Community Page response body 482
Community response body 481
Company feed
  feed elements 191
  feed items 192
  getting updated feed elements example 97
  resources 189
  URL 190
Connect
  resource 136
Connect Directory response body 482
Content Capability 482
Content Capability Input 426
Conversation Detail response body 484
Conversation Summary Collection response body 484
Conversation Summary response body 484
Conversation Unread Count response body 485
Conversations
  change status 372
  input 464
  unread count 373
CORS 40

D

Dashboard component snapshot
  input 428
  resource 137
Dashboard Component Snapshot 485
Dashboard Component Snapshot Capability 486
Data format 2
Datacloud Company 486
Datacloud Company Collection 488

Datacloud Company Information 139
Datacloud Company Order Informatoin 140
Datacloud Contact 489
Datacloud Contact Collection 489
Datacloud Contact Information 139
Datacloud Contact Order Information 141
Datacloud Order 490
Datacloud OrderId Information 140
Datacloud Orders 142
Datacloud Purchase Usage Collection 490
Datacloud Resources 139
Datacloud Usage 143
Deleting photo
group 316
user 389

E

Edit Capability 491
Email Address response body 492
Email Message Capability 492
Encoding 2
Enhanced Link Capability 492
Error Message response body 493
Error responses 26
Error with Output response body 493
Examples
approve private group request 102
create group 100
define action links 87, 90
edit comment 95
edit feed post 94
edit question post 95
edit question title 95
filter response body 80
follow record 103
following list 99
get @mention proposals 87
get Chatter activity statistics for user 100
get feed with internal feed items only 99
get following recommendations 100
get group feed 98
get my news feed 98
get record feed 104
get updated feed elements 97
get user information, batch 106
invoke action link 93
Java 234
join group 101
like feed item 96
Examples *(continued)*
like feed item, batch 96
link post 234
make communities request 81
post @mention 84
post a batch of feed elements with binary file 84
post action links 87, 90
post binary file to File list 84
post comment 83
post comment with binary file 84
post comment with existing content 83
post feed element with binary file 84
post feed element with existing content 82
post feed item 81
post feed item with binary file 84
post group photo 84
post multiple @mentions 85
post to a group 103
post user photo 84
reject private group request 102
request to join private group 102
search feed 99
send private message 106
share feed item 97
stop following ecord 104
update action link status 93
update group photo 101
update user photo 105
update user profile 104
uploading binary files 10
Explanation Details response body 493
Explanation Summary response body 495

F

Favorite
input 430
Favorite response body 496
Favorites feed
feed elements 199
feed items 197
list of favorites 194
resources 194
specific favorite 196
Favorites Page response body 497
Features
common 46
Features response body 497
Feed Directory Item response body 499
Feed Directory response body 499

Feed Element 151
Feed Element Capabilities Input 429
Feed Element Capabilities Response Body 501
Feed Element Page Response Body 503
Feed elements
  about 70
  company feed 187, 191
  favorites feed 199
  files feed 201
  groups feed 214
  key prefix filtered feed 211
  layout 70
  news feed 220
  people feed 237
  posting 70
  posting example with existing contnet 82
  record feed 243
  rendering 70
  to feed 259
  user profile feed 268
Feed Elements 145
Feed Elements Batch 148, 151
Feed Elements Bookmarks Capability 154
Feed Elements Capabilities 152
Feed Elements Capability 163
Feed Elements Capability Bundle 156
Feed Elements Capability Chatter Likes 158
Feed Elements Capability, Approval 153, 165
Feed Elements Capability, Associated Actions 153
Feed Elements Capability, Banner 154
Feed Elements Capability, Bundle 155
Feed Elements Capability, Canvas 157
Feed Elements Capability, Case Comment 157
Feed Elements Capability, Content 161
Feed Elements Capability, Dashboard Component 161
Feed Elements Capability, Editability 162
Feed Elements Capability, Email Message 162
Feed Elements Capability, Enhanced Link 163
Feed Elements Capability, Link 163
Feed Elements Capability, Poll 165
Feed Elements Capability, Recommendations 167
Feed Elements Capability, Record Snapshot 168
Feed Elements Capability, Tracked Changes 170
Feed Elements Comments Capability 159
Feed Elements Comments Capability Items 159
Feed Elements Likes Capability 158
Feed Elements Question and Answers Capability 166
Feed elements resources
  resources 143
Feed Elements Topics Capability 168
Feed Entity Is Editable 504
Feed Item Attachment: Approval response body 511
Feed Item Attachment: Basic Template response body 512
Feed Item Attachment: Canvas Template response body 512
Feed Item Attachment: Case Comment response body 513
Feed Item Attachment: Content response body 514
Feed Item Attachment: Dashboard response body 516
Feed Item Attachment: Email Message response body 517
Feed Item Attachment: Link response body 517
Feed Item Attachment: Poll response body 517
Feed Item Attachment: Record Snapshot response body 518
Feed Item Attachment: Tracked Change response body 518
Feed Item Body response body 518
Feed Item Page response body 519
Feed Item Preamble Message Body response body 519
Feed Item response body 504
Feed Item Topic
  input 436
Feed Item Topics response body 520
Feed items
  about 70
  accessing 172
  bookmark feed 182
  bookmarking 172
  company feed 188, 192
  deleting 172
  favorites feed 197
  files feed 203
  groups feed 216
  input 431
  key prefix filtered feed 208
  layout 70
  like, batch 96
  liking example 96
  news feed 222
  people feed 239
  posting 70
  posting example 81
  record feed 245
  rendering 70
  sharing example 97
  to feed 261
  topics feed 266
  user profile feed 269
Feed Poll Choice response body 520
Feed response body 499
Feed Tracked Change response body 520

Feed-items resources
accessing specific feed item 172
batch 170
comments 173
likes 175
polls 176
query 171
resources 170
topics 177
Feeds
about 70
bookmarks resources 179
communities home feed 185
communities moderation 186
company resources 189
editing post example 94
favorites resources 194
files resources 200
filter resources 206
groups resources 213
list resource 200
news resource 219
people resources 236
record resource 242
to resources 258
topics resources 264
user-profile resources 267
users resource 364
File Detail response body 521
File Share Link response body 521
File Shares
input 437
File Shares Page response body 522
File Summary Page response body 525
File Summary response body 522
Files
group files 306
input 436
posting binary example 84
sync 76
topic files 359
uploading binary example 10
working with 76
Files Connect Provider Type Response Body 525
Files Connect Repository
resource 283
Files Connect Repository Collection 527
Files connect repository features response body 527
Files Connect Repository File 283
Files Connect Repository File Content 284
Files Connect Repository Folder Items 284
Files Connect Repository List 282
Files Connect Repository Resources 282
Files connect repository response body 526
Files feed
feed elements 201
feed items 203
flat feed 205
getting updated feed elements example 97
resources 200
URL 201
Files resources
content 288
file information in batch 292
rendition 291
shares 288
shares link 290
specific file information 285
user files, filtered by groups 376
user files, filtered by sharing 377
user files, general 373
Filter feed
flat feed 210
getting updated feed elements example 97
key prefix 207
key prefix feed elements 211
key prefix feed items 208
list of filters 207
resources 206
Filter response bodies 20
Filters
exclude 22
groups 20
include 24
Flat feed
bookmarks feed 184
files feed 205
filtered feed 210
groups feed 218
news feed 235
people feed 241
record feed 257
to feed 263
user-profile feed 280
Flat Feed Item response body 528
Flat Feed response body 527
Folder
input 438

Folder Item
input 438
Folder Item Page response body 529
Folder Item response body 529
Folder response body 528
Follower Page response body 529
Followers
on records 342
users resources 378
Following
input 439
records 378
stop following record example 104
users 378
Following Counts response body 530
Following Page response body 530

G

Generic Bundle Capability Response Body 531
Generic Feed Element 531
Group
input 443
Group chatter settings
input 442
Group Chatter Settings response body 534
Group Detail response body 535
Group Information response body 536
Group member
input 445
Group Member Page response body 537
Group Member response body 537
Group Membership Request Collection response body 538
Group Membership Request response body 538
Group Membership Request Update
input 445
Group Page response body 539
Group Record Input 446
Group Record Page response body 539
Group Record response body 539
Group Records Resources 320
Group response body 532
Group-memberships batch resource 313
Group-memberships resource 313
Groups
approve private group request example 102
create group example 100
join example 101
posting example 103
reject private group request example 102
Groups *(continued)*
requesting membership 312
requesting membership in a private group 309
requesting status for membership requests 309
requesting to join private group example 102
topic groups 361
updating group photo example 101
Groups feed
feed elements 214
feed items 216
flat feed 218
getting updated feed elements example 97
resources 213
URL 214
Groups Feed
get group feed example 98
Groups resources
announcements information 304
batching information 303
files information 306
groups in organization 297
information 300
members 307
members requests for private groups 309
membership request 312
photo 316
records 319
settings 320
topics 322

H

Hash tags
message segment input 451
HTML encoding 2

I

Icon response body 540
Influence
resource 322
Influence Thresholds response body 540
Inputs
Action Link 415
Action Link Definition 407
Action Link Group Definition 412
Action Link Template Binding Input 415
Announcement input 416
Associated Actions Capability 417
attachment: existing content 418
attachment: feed poll 419

Inputs *(continued)*
attachment: link 419
attachment: new file upload 420
batch 440
Batch Collection Input 421
Batch Input 422
batch request 441
Bookmarks Capability Input 423
canvas app 417
Canvas Capability Input 424
comment capabilities 426
comments 425
Content Capability Input 426
conversation 464
dashboard component snapshot 428
Datacloud Order 427
favorite 430
Feed Element Capabilities 429
feed item 431
feed item topic 436
feed poll input 436
file input 436
file shares 437
folder 438
folder item 438
following 439
group 443
group chatter settings 442
group member 445
group membership request update 445
Group Record Input 446
Link Capability Input 446
managed topic 447
managed topic position collection 448
managed topic position input 450
message body 450
message segment: hash tag 451
message segment: link 451
message segment: mention 451
message segment: text 452
messages 462
moderation flag 452
PATCH 10
photo 452
Poll Capability Input 453
POST 10
Push Notification 454
PUT 10
Question and Answers Capability Input 455
Inputs *(continued)*
Recommendation Definition Input 455
Request Header 456
Scheduled Recommendation Input 457
share 459
topic 460
topic assignment 459
topic endorsement 460
user 461
user chatter settings 461
user message: recipient 463
user message: recipient list 463

J

Java example 234
JavaScript 40
JSON 2

K

Knowledgeable People Collection response body 541
Knowledgeable People response body 540

L

Like Message Body response body 542
Like Page response body 542
Like response body 541
Likes
access 323
delete 323
on comments 127
on feed item 175
resource 323
Limits 2–3
Link Capability 542
Link Capability Input 446
Link post examples 234
Links
attachment input 419
message segment input 451
List View Definition response body 543
List View Record response body 544
List View response body 544
Localization 2

M

Maintenance Information 545
Managed topic collection 546
Managed topic input 447
Managed Topic Position Collection Input 448

Managed Topic Position Input 450
Managed topic resource 328
Managed topic response body 546
Managed topics resource 323–324
Members of a group 307
Mention Completion Page response body 548
Mention Completion response body 547
Mention Validation response body 548
Mention Validations response body 549
Mentions
  get proposals example 87
  message segment input 451
  posting example 84
  posting multiple @mentions example 85
  resources 331
Mentions completions
  resources 328–329
Message body input 450
Message Collection response body 550
Message response body 549
Message segment input
  hash tag 451
  link 451
  mention 451
  text 452
Message Segment: Entity Link response body 550
Message Segment: Field Change Name response body 551
Message Segment: Field Change response body 550
Message Segment: Field Change Value response body 552
Message Segment: Hashtag response body 552
Message Segment: Link response body 553
Message Segment: Mention response body 553
Message Segment: More Changes response body 554
Message Segment: Resource Link response body 554
Message Segment: Text response body 554
Messages
  input 462
  sending private message example 106
  user messages, specific 389
Moderation Capability Response Body 555
Moderation Flag
  input 452
Moderation Flags response body 555
Motif response body 555
Motifs
  batch motifs resource 333
  resource 332
Motifs resources 332

N

New file upload 420
News feed
  feed elements 220
  feed items 222
  flat feed 235
  get my news feed example 98
  getting updated feed elements example 97
  resources 219
  URL 219
NonEntityRecommendationRepresentation 557

O

OAuth
  authentication 43
  user-agent flow 35
  Web server flow 30
Organization
  resource 137
Organization response body 557
Origin Capability 558
Outputs
  Announcement 470
  Announcement Page 470

P

PATCH input 10
People feed
  feed elements 237
  feed items 239
  flat feed 241
  getting updated feed elements example 97
  resources 236
  URL 237
Phone Number response body 558
Photo
  group 316
  input 452
  user 389
Photo response body 559
Platform Action Group response body 562
Platform Action response body 559
Poll Capability Input 453
Poll Capability Response Body 563
Polls
  attachment input 419
  feed poll input 436
  on feed items 176
POST input 10

Push Notification Input 454
Push Notifications Resource 334
PUT input 10

Q

Query feed items 171
Question And Answers Capability 563
Question and Answers Capability Input 455
Question and Answers Suggestion Collection 564
Question and Answers Suggestions Resource 335
Questions
    editing post example 95
    editing title example 95
Quickstart
    access token 42
    Developer Edition 42
    OAuth 42
    prerequisites 42

R

Rate Limits 3
Recipient
    input 463
    list input 463
Recommendation Definition 338, 565
Recommendation Definition Collection 566
Recommendation Definition Input 455
Recommendation Definition Photo 339
Recommendation Definition Resources 336
Recommendation Definitions 336
Recommendation response body 564
RecommendationCollectionRepresentation 566
Recommendations
    get following recommendations example 100
    specific action 393
    specific action and key prefix 398
    specific action and object category 395
    specific action and object ID 400
    specific action and object type 400
Recommendations Capability 566
Recommended Object 567
Record feed
    feed elements 243
    feed items 245
    flat feed 257
    get internal feed items only example 99
    getting updated feed elements example 97
    resources 242
    URL 242
Record Field response body 567
Record Field: Blank response body 568
Record Field: Compound response body 568
Record Field: Currency response body 569
Record Field: Date response body 569
Record Field: Percent response body 570
Record Field: Picklist response body 570
Record Field: Reference response body 571
Record Field: Reference with Date response body 571
Record Snapshot Capability 571
Record Summary List response body 572
Record Summary response body 572
Record View response body 573
Record View Section response body 573
Records
    Chatter group records 319
    followers 342
    following 378
    following example 103
    get record feed 104
    resources 342
Reference response body 574
Refresh tokens 37
Related List Column response body 543
Related List Sort response body 545
Repository file detail representation 574
Repository File Summary 574
Repository Folder Item 575
Repository Folder Items Collection 576
Repository folder summary 576
Reputation Level response body 577
Reputation response body 577
Request bodies
    Action Link 415
    Action Link Definition 407
    Action Link Group Definition 412
    Action Link Template Binding Input 415
    Associated Actions Capability 417
    attachment: canvas app 417
    attachment: existing content 418
    attachment: feed poll 419
    attachment: link 419
    attachment: new file upload 420
    batch 440
    Batch Collection Input 421
    Batch Input 422
    batch request 441
    Bookmarks Capability Input 423
    Canvas Capability Input 424

Request bodies *(continued)*
comment capabilities 426
comments 425
Content Capability Input 426
conversation 464
dashboard component snapshot 428
datacloud order 427
favorite 430
Feed Element Capabilities 429
feed item 431
feed item topic 436
feed poll input 436
file input 436
file shares 437
folder 438
folder item 438
following 439
group 443
group chatter settings 442
group member 445
group membership request update 445
Group Record Input 446
Link Capability Input 446
managed topic 447
Managed topic position collection 448
managed topic position input 450
message body 450
message segment: hash tag 451
message segment: link 451
message segment: mention 451
message segment: text 452
messages 462
moderation flag 452
photo 452
Poll Capability Input 453
Push Notification 454
Question and Answers Capability Input 455
Recommendation Definition Input 455
Request Header 456
Scheduled Recommendation Input 457
share 459
topic 460
topic assignment 459
topic endorsement 460
user chatter settings 461
user input 461
user message: recipient 463
user message: recipient list 463
Request Header 578
Request Header Input
inputs 456
Requesting membership in a group 312
Requesting membership in a private group 309
Resources
action links 109
action links diagnostic information 117
announcements 117
batch 119
batch motifs 333
bookmarks 179
chatter 124
comments 125
communities 128
communities home feed 185
Communities Moderation 129
communities moderation feed 186
community information 129
company feed 189
connect 136
dashboard component snapshot 137
datacloud 139
datacloud company information 139
datacloud company order information 140
datacloud contact information 139
datacloud contact order information 141
datacloud orderId information 140
datacloud orders 142
datacloud usage 143
error responses 26
favorites 194
feed element 151
feed elements 143, 145
feed elements batch 148, 151
feed elements bookmarks capability 154
feed elements capabilities 152
feed elements capability 163
feed elements capability bundle elements 156
Feed elements capability chatter likes 158
feed elements capability, approval 153
feed elements capability, associated actions 153
feed elements capability, banner 154
feed elements capability, bundle 155
feed elements capability, canvas 157
feed elements capability, case comment 157
feed elements capability, content 161
feed elements capability, dashboard component 161
feed elements capability, editability 162
feed elements capability, email message 162

Resources *(continued)*
feed elements capability, enhanced link 163
feed elements capability, link 163
feed elements capability, origin 165
feed elements capability, poll 165
feed elements capability, recommendations 167
feed elements capability, record snapshot 168
feed elements capability, tracked changes 170
feed elements comments capability 159
feed elements comments capability items 159
feed elements linkes capability 158
feed elements question and answers capability 166
feed elements topics capability 168
feed-items 170
feeds 200
files 292
files connect repository 283
files connect repository file 283
files connect repository file content 284
files connect repository folder items 284
files connect repository list 282
files feed 200
filter feed 206
folders 292
followers on records 342
group records 320
groups feed 213
influence 322
likes 323
list of communities 128
managed topic 328
managed topics 324
mentions completions 328–329
mentions validations 331
motifs 332
organization 136
people feed 236
push notifications 334
question and answers suggestions 335
recommendation definition 338
recommendation definition photo 339
recommendation definitions 336
records 342
scheduled recommendation 347
scheduled recommendations 345
subscriptions 350
topics feed 264
topics on records 343
user profile 364

Resources *(continued)*
user reputation 401
zone search results 405

Response bodies
Action Link Definition 465
Action Link Diagnostic 468
Action Link Group Definition 468
Address 470
Approval Capability 471
Approval Post Template Field 472
Article Item 472
Associated Actions Capability 472
Banner Capability 473
Batch Result Item 473
Batch Results 474
Bookmarks Capability 474
Canvas Capability 476
Case Comment Capability 475
Chatter Activity 476
Chatter Directory 477
Chatter Influence 477
Chatter Likes Capability 477
ClientInfo 478
Comment 478
Comment Page 480
Comments Capability 480
CommentsCapabilities 479
Community 481
Community Page 482
Connect Directory 482
Content Capability 482
Conversation Detail 484
Conversation Summary 484
Conversation Summary Collection 484
Conversation Unread Count 485
Dashboard Component Snapshot 485
Dashboard Component Snapshot Capability 486
Datacloud Company 486
Datacloud Company Collection 488
Datacloud Contact 489
Datacloud Contact Collection 489
Datacloud Order 490
Datacloud Purchase Usage Collection 490
Edit Capability 491
Email Address 492
Email Message Capability 492
Enhanced Link Capability 492
Error Message 493
Error with Output 493

Response bodies *(continued)*
excluding properties 22
Explanation Details 493
Explanation Summary 495
Favorite 496
Favorites Page 497
Features 497
Feed 499
Feed Directory 499
Feed Directory Item 499
Feed Element Capabilities 501
Feed Element Page 503
Feed Entity Is Editable 504
Feed Item 504
Feed Item Attachment: Approval 511
Feed Item Attachment: Basic Template 512
Feed Item Attachment: Canvas Template 512
Feed Item Attachment: Case Comment 513
Feed Item Attachment: Content 514
Feed Item Attachment: Dashboard 516
Feed Item Attachment: Email Message 517
Feed Item Attachment: Link 517
Feed Item Attachment: Poll 517
Feed Item Attachment: Record Snapshot 518
Feed Item Attachment: Tracked Change 518
Feed Item Body 518
Feed Item Page 519
Feed Item Preamble Message Body 519
Feed Item Topics 520
Feed Poll Choice 520
Feed Tracked Change 520
File Detail 521
File Share Link 521
File Shares Page 522
File Summary 522
File Summary Page 525
files connect repository 526
files connect repository collection 527
files connect repository features 527
filtering 20
filtering example 80
filtering properties 20, 22, 24
Flat Feed 527
Flat Feed Item 528
Folder 528
Folder Item 529
Folder Item Page 529
Follower Page 529
Following Counts 530

Response bodies *(continued)*
Following Page 530
Generic Bundle Capability 531
Generic Feed Element 531
Group 532
Group Chatter Settings 534
Group Detail 535
Group Information 536
Group Member 537
Group Member Page 537
Group Membership Request 538
Group Membership Request Collection 538
Group Page 539
Group Record 539
Group Record Page 539
group records list 539
Icon 540
including properties 24
Influence Thresholds 540
Knowledgeable People 540
Knowledgeable People Collection 541
Like 541
Like Message Body 542
Like Page 542
Link Capability 542
List View Definition 543
List View Object 544
List View Record 544
Maintenance Information 545
managed topic 546
managed topic collection 546
Mention Completion 547
Mention Completion Page 548
Mention Validation 548
Mention Validations 549
Message 549
Message Collection 550
Message Segment: Entity Link 550
Message Segment: Field Change 550
Message Segment: Field Change Name 551
Message Segment: Field Change Value 552
Message Segment: Hashtag 552
Message Segment: Link 553
Message Segment: Mention 553
Message Segment: More Changes 554
Message Segment: Resource Link 554
Message Segment: Text 554
Moderation Capability 555
Moderation Flags 555

Response bodies *(continued)*
Motif 555
NonEntityRecommendationRepresentation 557
Organization 557
Origin Capability 558
Phone Number 558
Photo 559
Platform Action 559
Platform Action Group 562
Poll Capability 563
Question And Answers Capability 563
Question and Answers Suggestion Collection 564
Recommendation 564
Recommendation Definition 565
Recommendation Definition Collection 566
RecommendationCollectionRepresentation 566
Recommendations Capability 566
Recommended Object 567
Record Field 567
Record Field: Blank 568
Record Field: Compound 568
Record Field: Currency 569
Record Field: Date 569
Record Field: Percent 570
Record Field: Picklist 570
Record Field: Reference 571
Record Field: Reference with Date 571
Record Snapshot Capability 571
Record Summary 572
Record Summary List 572
Record View 573
Record View Section 573
Reference 574
Related List Column 543
Related List Sort 545
repository file detail representation 574
repository file summary 574
repository folder item 575
repository folder items collection 576
repository folder summary 576
Reputation 577
Reputation Level 577
Request Header 578
Scheduled Recommendation 578
Scheduled Recommendation Collection 579
Shares 579
Subscription 579
Subscription Page 580
Time Zone 580

Response bodies *(continued)*
Topic 581
Topic Collection 581
Topic Endorsement 581
Topic Endorsement Collection 582
Topic images 582
Topic Opt Out 583
Topic Opt Out Collection 583
Topic Suggestion 583
Topic Suggestion Collection 583
Topics Capability 584
Topics People Are Knowledgeable About 584
Topics People Are Knowledgeable About Collection 584
Tracked Change Bundle Capability 585
Tracked Changes Capability 585
Unauthenticated User 586
User Capabilities 586
User Chatter Settings 586
User Detail 587
User Group Page 589
User Page 590
User Profile 590
User Profile Tab 590
User Settings 591
User Status 592
User Summary 592
Zone 594
Zone Page 595
Zone Search Page 595
Zone Search Result 596
Response Bodies
files connect provider type 525
Rich input 10

S

Scheduled Recommendation 347, 578
Scheduled Recommendation Collection 579
Scheduled Recommendation Input 457
Scheduled Recommendation Resources 344
Scheduled Recommendations 345
Search feed items 171
Settings
group 320
users 401
Share
input 459
Shares response body 579
Subscribing to a group 307
Subscription Page response body 580

Subscription response body 579
Subscriptions
  resource 350

T

Tasks
  approve private group request 102
  create group 100
  define action link templates 90
  define action links 87, 90
  edit comment 95
  edit feed post 94
  edit question post 95
  edit question title 95
  filter response body 80
  follow record 103
  get @mention proposals 87
  get Chatter activity statistics for user 100
  get feed with internal feed items only 99
  get following recommendations 100
  get group feed 98
  get list of records user follows 99
  get my news feed 98
  get record feed 104
  get updated feed elements 97
  get user information, batch 106
  invoke action link 93
  join group 101
  like feed item 96
  like feed item, batch 96
  make communities request 81
  post @mention 84
  post a batch of feed elements with binary file 84
  post action links 87, 90
  post binary file to File list 84
  post comment 83
  post comment with binary file 84
  post comment with existing content 83
  post feed element with binary file 84
  post feed element with existing content 82
  post feed item 81
  post feed item with binary file 84
  post group photo 84
  post multiple @mentions 85
  post to a group 103
  post user photo 84
  reject private group request 102
  request to join private group 102
  search feed 99
Tasks *(continued)*
  send private message 106
  share feed item 97
  stop following record 104
  update action link status 93
  update group photo 101
  update user photo 105
  update user profile 104
Text
  message segment input 452
Time Zone response body 580
To feed
  feed elements 259
  feed items 261
  flat feed 263
  resources 258
  URL 259
Tokens 37–38
Topic
  input 460
Topic Assignment
  input 459
Topic Collection response body 581
Topic endorsement
  input 460
Topic Endorsement Collection response body 582
Topic Endorsement response body 581
Topic images response body 582
Topic Opt Out Collection response body 583
Topic Opt Out response body 583
Topic resource 355
Topic response body 581
Topic Suggestion Collection response body 583
Topic Suggestion response body 583
Topics
  on feed items 177
Topics Capability Response Body 584
Topics feed
  feed items 266
  resources 264
Topics Feed
  URL 264
Topics on records
  resource 343
Topics People Are Knowledgeable About Collection response body 584
Topics People Are Knowledgeable About response body 584
Topics resource
  endorse people 357

Topics resource *(continued)*
endorsements 358
files information 359
groups information 361
knowledgeable people 361
knowledgeable people opt out 363
knowledgeable people opt outs 362
list of topics 352
related topics 362
topic suggestions 353
trending topics 354
Tracked Change Bundle Capability Response Body 585
Tracked Changes Capability 585

U

Unauthenticated User response body 586
Updating photo
group 316
user 389
Uploading binary files 10
URLs 4
Usage Limits 3
User Capabilities response body 586
User chatter settings
input 461
User Chatter Settings response body 586
User Detail response body 587
User Group Page response body 589
User Page response body 590
User profile
resources 364
User profile feed
feed elements 268
feed items 269
User Profile response body 590
User Profile Tab response body 590
User Reputation Resource 401
User Settings response body 591
User Status response body 592
User Summary response body 592
User-agent OAuth flow 35
User-profile feed
flat feed 280
resources 267
URL 267
Users
updating photo example 105
updating profile example 104
Users resources
batching information 369
conversations 369
conversations, change status 372
conversations, specific 371
conversations, unread count 373
files, filtered by groups 376
files, filtered by sharing 377
files, general 373
followers 378
following 378
groups 380
knowledgeable about topics 381
messages 382
messages, specific 389
photo 389
recommendations 391
recommendations for specific action 393
recommendations for specific action and object ID 400
recommendations for specific action and object type 400
recommendations, specific action and key prefix 398
recommendations, specific action and object category 395
topics 404
user information 366
user profile information 367
user settings 401
user status 403

W

Web server OAuth flow 30
Wildcards 19

Z

Zone Page response body 595
Zone response body 594
Zone Search Page response body 595
Zone Search Result response body 596
Zones 404
Zones resources
list of zones in organization 404
specific zonezone information 405
zone search results 405