Version 25.0: Summer '12

Database.com Chatter REST API Developer's Guide

Last updated: September 2 2012

Table of Contents

GETTING STARTED

Chapter 1

Introducing Chatter REST API

In this chapter ...

- Using Chatter API
- Understanding Chatter API Resources
- Send HTTP Requests with cURL
- Understanding OAuth and Chatter API

Chatter API is a REST API that provides programmatic access to Chatter feeds and social data such as users, groups, followers, and files. It's used by developers who want to integrate Chatter into a variety of applications such as mobile applications, intranet sites, and third-party Web applications. Chatter API is similar to APIs offered by other companies with feeds, such as Facebook and Twitter. Its advantages include ease of integration and development.

To use this documentation, you should have a basic familiarity with software development, Web services, OAuth, and Chatter.

The following sections include:

- The key characteristics and architecture of Chatter REST API. This will help you understand how your applications can best use Chatter API resources.
- How to set up your development environment so you can start working with Chatter API.
- How to configure OAuth for your client application.
- A tutorial that leads you step by step through a typical use case.

Using Chatter API

Use Chatter API to:

- Build a mobile client that displays a Chatter feed.
- Integrate a third-party Web application with Chatter so it can notify groups of users about events.
- Display the Chatter 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. Games might include things like sales incentive competitions.
- Creating a custom, branded skin for Chatter for your organization.

How is Chatter API Different from other APIs?

Chatter API complements the SOAP API by making it easy to interact with Chatter data, such as feeds, group information, and Chatter user-profile details.

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

- Returned information is automatically localized to the user's time zone and language.
- Feed items are structured in a way that makes it easy to render on Web sites and mobile devices.
- Changed values that are tracked in a feed are returned as value-pair representations.
- Relationships between objects can easily be traversed without having to resolve foreign keys. For example, using Chatter API you can get all of a user's followers or groups with one request per page.

Selecting the Right API for Your Application

Database.com provides many types of APIs for various use cases, including both SOAP API and REST API. These APIs provide access to the majority of Database.com data, including most of the Chatter data.

So when should you use which API?

Situations where you would want to use SOAP API or REST API include:

- Migrating or synchronizing user profiles and their associated user photos from one system to another.
- Creating analytic applications for Chatter.
- Queries where you want to select specifically which records or fields are returned.

SOAP API and REST API’s rate limiting and powerful query language more suited to these use cases and contains most of the applicable Chatter data.

However, Chatter API is the only API that supports some of Chatter’s more complex data structures, for example posting photos or using @mention rich text in feed items. If necessary, you can use SOAP API or REST API for the bulk of your queries, then use the feed item or comment Ids you’ve extracted to selectively work with Chatter API.

Understanding Chatter API Resources

Each resource in Chatter 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.

The primary resources for Chatter API are:

Resource	Description
Bookmarks Resources	Feed made up of Chatter feed items posted to bookmarks saved by the logged-in user. Available in versions 24.0 and later. Available resources are: Bookmarks Feed URL Bookmarks Feed Items Bookmarks Feed Items Flat—Available in versions 25.0 and higher.
Chatter Resource	Directory of the general resources available. Available in versions 24.0 and later. Contains a single resource.
Dashboard Component Snapshot Resource	Posts a new dashboard component snapshot to the specified location. Available in versions 25.0 and higher. Contains a single resource.
Comments Resource	Information about the specified comment. Also used to delete a comment, or to like a comment. Available resources are: Comment Comment Likes
Company Feed Resources	A feed that returns all updates for people the current user follows, groups the user is a member of, and files and records the user is following. Available resources are: Company Feed URL Company Feed Items
Favorites Feed Resources	Feeds made up of Chatter favorites saved by the logged-in user. Available in versions 24.0 and later. Available resources are: List of Favorites Favorite Feed Favorites Feed Items
Feeds Resource	A list of all of the feeds the logged in user is able to view. Contains a single resource.
Feed-Items Resources	Represents a specific feed item. Also used to query all feed items, post comments to feed-items, as well as delete a specific feed item. Available resources are: Feed Item Comments Feed Item Likes

Resource	Description
	Feed Items Query Specific Feed Item
Files Resources	Information about the specified file, including content, rendered version, and sharing. Available in versions 24.0 and later. Available resources are: File Information File Content File Rendition File Shares
Files Feed Resources	All feed items that contain files posted by people or groups that the current user follows. Available resources are: Files Feed URL Files Feed Items Files Feed Items Flat Available in versions 25.0 and higher.
Filter Feed Resources	Feeds filtered by the specified criteria for the logged-in user. Available resources are: List of Filters Keyprefix Filter Feed Keyprefix Filtered Feed Items Filter Feed Items Flat—Available in versions 25.0 and higher.
Groups Feed Resources	All feed items from all groups the current user either owns or is a member of. **Note:** To get the feed for a specific group, use the Record Feed Resource resource with a `groupId`. Available resources are: Groups Feed URL Groups Feed Items Groups Feed Items Flat—Available in versions 25.0 and higher.
Group-Memberships Resource	Access information about a member of a group, or remove a member from a group. Contains a single resource.
Groups Resources	Information about groups, such as the group's members, photo, and the groups the current user is following. Also used to add members to a group and change the group

Resource	Description
	photo. This is not a feed. To get the feed for a specific group, use the Record Feed Resource resource with a `groupId`. Available resources are: List of Groups Batch Group Information Group Files Group Information Group Members Group Photo
Influence Resource	List of the percentile thresholds used to categorize an organization's users by their influence in Chatter. Available in versions 24.0 and later. Contains a single resource.
Likes Resource	Information about the specified like. Also used to delete a like. Contains a single resource.
News Feed Resources	The news feed of the current user. Includes feed items from all groups the current user either owns or is a member of, as well as all files, records, and all users the current user follows. You can also use this resource to add feed-items. Available resources are: News Feed URL News Feed Items News Feed Items Flat—Available in versions 25.0 and higher.
Organization Resource	Information about the current user's organization and settings. Available in versions 24.0 and later. Contains a single resource.
People Feed Resources	All feed items posted by all of the people the current user follows. Available resources are: People Feed URL People Feed Items People Feed Items Flat—Available in versions 25.0 and higher.
Record Feed Resource	The feed of the specified record, which could be a group, person, object or file. Available resources are: Record Feed URL Record Feed Items Record Feed Items Flat—Available in versions 25.0 and higher.

Resource	Description
Records Resources	Information about the specified record, including followers. This is not a feed. To access a record feed, use the Record Feed Resource. Available resources are: Record Specifics Record Followers
Subscriptions Resource	Information about the specified subscription. Also used to delete a subscription, for example, to unfollow a record. Contains a single resource.
To Feed Resources	Feed of all @mentions of the current user and posts others make to the logged-in user's feed. Also used to post to the user's feed. Available resources are: To Feed URL To Feed Items To Feed Items Flat—Available in versions 25.0 and higher.
Topics Resource	List of the trending topics for the organization. Available in versions 24.0 and later. Contains a single resource.
User-Profile Feed Resources	Feed of all actions by the current, logged-in user on all records that can be tracked in a feed. This is different than`/chatter/feeds/news`, which returns everything in a user's feed, including posts, group updates, and record updates. Also used to add feed items. Available resources are: User–Profile Feed URL User–Profile Feed Items User-Profile Feed Items Flat—Available in versions 25.0 and higher.
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. This is not a feed. To access the user profile feed, use User-Profile Feed Resources. Available resources are: Batch User Information Change Status of Conversations Users Files, General User Files, Filtered by Group User Files, Filtered by Sharing Following Users User Information

Resource	Description
	User Messages, General User Messages, Specific User Recommendations, General Get Unread Count for Conversations User Conversations, General User Conversations, Specific User Followers User Groups User Photos User Profile Information User Recommendations for a Specific Action User Recommendations for a Specific Action and Object Category User Recommendations for a Specific Action and Object ID User Status—No longer available as of version 25.0.

Many of the resources use either a specified userId or the keyword `me` to indicate the current, logged-in user. For example, to return the user-profile information for the current, logged-in 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 either XML or JSON. 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:

Name	Type	Description
`aboutMe`	String	Text from user's profile
`address`	Address	Address
`chatterActivity`	Chatter Activity	Chatter activity statistics
`chatterInfluence`	Chatter Influence	User’s influence rank
`companyName`	String	Company name
`currentStatus`	User Status	**Note:** This property is no longer available as of version 25.0.

Name	Type	Description
		User's current status
`email`	String	User's email address
`firstName`	String	User's first name
`followersCount`	Integer	Number of users following this user
`followingCounts`	Following Counts	Information about items the user is following
`groupCount`	Integer	Number of groups user is following
`id`	String	18-character `Id` of the user
`isActive`	Boolean	`true` if user is active, `false` otherwise
`isChatterGuest`	Boolean	`true` if user is a Chatter customer, `false` otherwise
`lastName`	String	User's last name
`managerId`	String	18-character `Id` of the user’s manager
`managerName`	String	Locale-based concatenation of manager's first and last names
`mySubscription`	Reference	If the logged-in user is following this user, this contains information about the subscription, else returns `null`
`name`	String	Locale-based concatenation of user's first and last names
`phoneNumbers`	Phone Number[]	Collection of user's phone numbers
`photo`	Photo	Information about the user's photos
`title`	String	Title of the user
`type`	String	`User`
`url`	String	URL to the user's profile
`username`	String	User name of the user, such as *`Admin@mycompany.com`*. Available in versions 24.0 and later.

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.",
   "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
   },
   "companyName": "Seattle Apps",
   "email": "mraven@seattleapps.com",
   "firstName": "Marion",
   "followersCount": 1,
   "followingCounts": {
      "people": 5,
      "records": 0,
      "total": 5
   },
   "groupCount": 3,
   "id": "005D0000001Kl6xIAC",
   "isActive": true,
   "isChatterGuest": false,
   "lastName": "Raven",
   "managerId": null,
   "managerName": null,
   "mySubscription": null,
   "name": "Marion Raven",
   "phoneNumbers": [{
      "number": "(206) 999-2846",
      "type": "Work"
   }],
   "photo": {
      "largePhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/F",
      "photoVersionId": "729D00000000BDV",
      "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/T"
   },
   "title": null,
   "type": "User",
   "url": "/services/data/v25.0/chatter/users/005D0000001Kl6xIAC",
   "username": "mraven@seattleapps.com"
}
```

Note: Database.com 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 `na3.database.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. The instance given to the client app when they are authenticated should be prepended to the URLs.

The following are important characteristics of Chatter API resources and architecture:

Stateless

Each request from client to server must contain all the information necessary to understand the request, and not use any stored context on the server. However, the representations of the resources are interconnected using URLs, which allow the client to progress between states.

Uniform interface

All resources are accessed with a generic interface over HTTP.

Named resources

Access almost all resources by using the URI for your company's instance, combined with version information and the URI for the resource. For example, if your instance is:

```
https://na1.database.com/
```

And the version information is:

```
/services/data/v25.0/chatter/
```

And the resource is :

```
feeds/news/me/feed-items
```

Put together, the full URL is:

```
https://na1.database.com/services/data/v25.0/chatter/
feeds/news/me/feed-items
```

Note: The Organization resource has the base URL of `/services/v25.0/connect`.

Authentication

Chatter API uses OAuth 2.0.

Support for JSON and XML

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.

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-items.xml`.

Support for Method Overriding

If you use an HTTP library that doesn't allow overriding or setting an arbitrary HTTP method name, you can send a POST request and provide an override to the HTTP method with the request parameter `_HttpMethod`. For example:

```
https://instance_name/services/data/v25.0/chatter/
/chatter/users/me/conversations/03MD0000000008KMAQ
?_HttpMethod=PATCH&read=true
```

Note: The `_HttpMethod` parameter is case sensitive. Be sure to use the correct case for all values.

Localized names and times

Chatter API localizes both the names and datetimes to the language and time zone settings of the current user. This means mobile devices don't need their own language or time zone settings. Users set their locale in their personal information.

Chatter API Request Limits

Chatter API request limit is 200 requests per OAuth consumer per user per hour. For example, if you have created three remote access applications (each of which represents a different OAuth consumer with its own `Consumer Key` and `Consumer Secret`), each user of those apps has 200 requests per hour. If a user issues more than 200 requests per hour, Chatter API returns a 503 Service Unavailable error code.

If an application uses a regular session ID for authentication instead of an OAuth token, that application's requests count against a general allocation of 200 requests per user per hour where there is no associated OAuth consumer key. However, any other application the user accesses, that is also authenticating with session IDs, is contending for the same allocation of 200 hourly requests per user. For example, your app connects a user to Database.com using a regular session ID. This user now only has 200 hourly requests regardless of the app they are using. They no longer have a separate request limit per application: all applications are now using the same limit. Because of this, we recommend clients use OAuth tokens whenever feasible.

Note: Stated limits aren't a promise that the specified resource is available at its limit in all circumstances. For example, load, performance and other system issues might prevent some limits from being reached. The stated limit may change without notice. Applications should make efficient use of available requests and gracefully handle the 503 error code.

Default HTML Entity Encoding

Chatter API strings are minimally HTML entity encoded by default, which is suitable in most cases for display between HTML tags, but not necessarily in other HTML contexts.

For non-HTML contexts, such as native mobile applications, Chatter API clients may request raw (unencoded) output by setting the `X-Chatter-Entity-Encoding` HTTP header in your request to `false`.

For more information, see Response Body Encoding on page 192.

Send HTTP Requests with cURL

Chatter 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. We recommend a command-line tool called cURL to simplify sending and receiving HTTP requests and responses.

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.

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

- An HTTP method (such as HEAD, GET, POST, or DELETE).
- An OAuth 2.0 access token used to authenticate the request. For information on how to retrieve the token, see Step One: Set Up Authorization on page 24.
- The Chatter API resource.
- Any JSON or XML files containing information needed for requests, such as updating a record with new information.

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 Chatter API item, such as a comment or like, or subscribe to a group.
- PATCH and PUT are used to update an item, for example, marking a conversation as read.
- DELETE is used to delete a Chatter API item, such as a feed-item, or unsubscribe from a group.
- HEAD is used to retrieve resource metadata. It's similar to using GET but without the response body. You can use HEAD for testing the availability of a resource.

To access a resource, submit an HTTP request containing a header, method, and resource name.

Understanding OAuth and Chatter 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 API uses OAuth to securely identify your application before connecting to Database.com.

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 Database.com page, which returns an access token to your application. Your application can use this access token to access Chatter API Web services. This approach to authentication can be used in both mobile apps and from a Web page.

Chatter API uses OAuth 2.0.

Understanding Remote Access Applications

To use Chatter API, you must create a REST entry point in your organization. Database.com defines this entry point as a remote access application definition.

For an example in creating a remote access application definition, see Step One: Set Up Authorization on page 24.

Considerations when creating and using a remote access application:

- The following maps the labels used in the remote access application definition to OAuth terms:

Remote Access Application Label	OAuth Term	Description
`Consumer Key`	`client_id`	A unique identifier that identifies your application to Database.com.
`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 actually used; however, between your client application and the server (the remote access application definition) the value must be same. For example, you may want to use a value that identifies the application, such as `http://MyCompany.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 remote access application does not have to reside in the same organization as your users. The remote access application you create can be used to sign into any organization.
- We recommend creating more than one remote access application 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 remote access application 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 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 remote access application) 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 Database.com. 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 Database.com. On successful authorization, the client application is provided with access and refresh tokens.

Database.com supports the following flows for use with the Chatter 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.

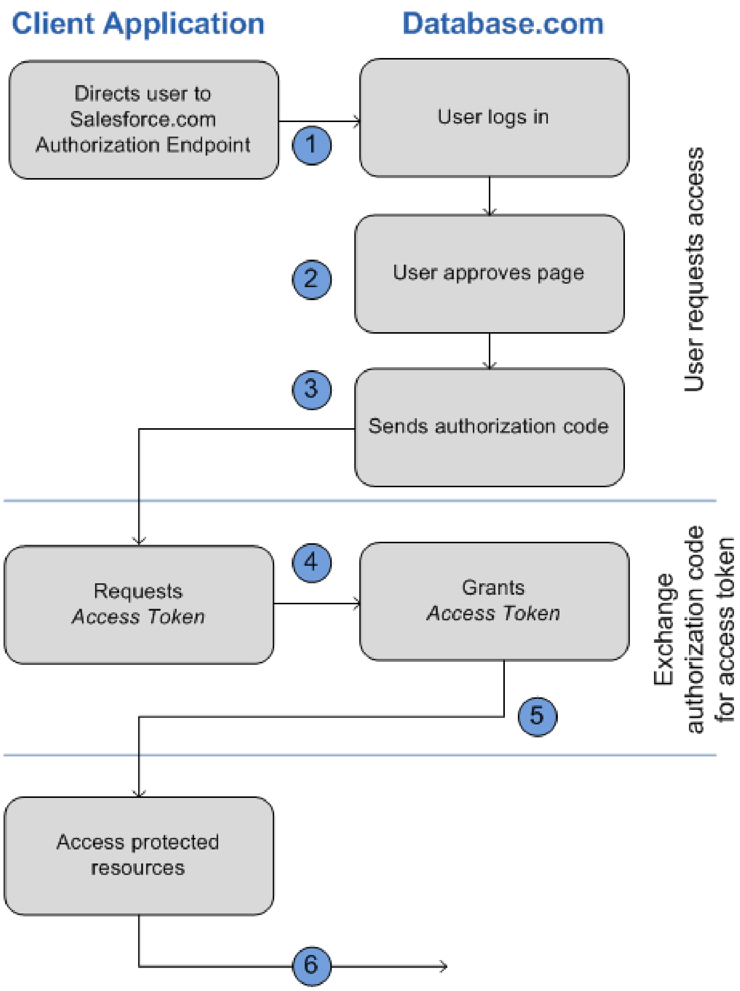
Revoking Access

After a user has granted access to a client application, they can revoke access by clicking **My Personal Information** > **Personal Information**, then in the Remote Access related section, clicking Revoke.

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 Database.com log in page.
2. The end-user logs into Database.com to authenticate themselves. Since this web page is hosted by the resource owner (Database.com) 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. Database.com 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, Database.com 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 API endpoints.

Using the Web Server Flow with Chatter API and Database.com

The following provides specific details for the OAuth Web-server flow when used with Database.com and Chatter 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 remote access application defined for this application
`redirect_url`	The `Callback URL` value from the remote access application 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. • `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 Database.com 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 Database.com 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 remote access application definition.
`client_secret`	Consumer secret from the remote access application definition.
`redirect_uri`	URI to redirect the user to after approval. This must match the value in the `Callback URL` field in the remote access application 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 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). **Caution:** 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.database.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

Parameters	Description
	and `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.

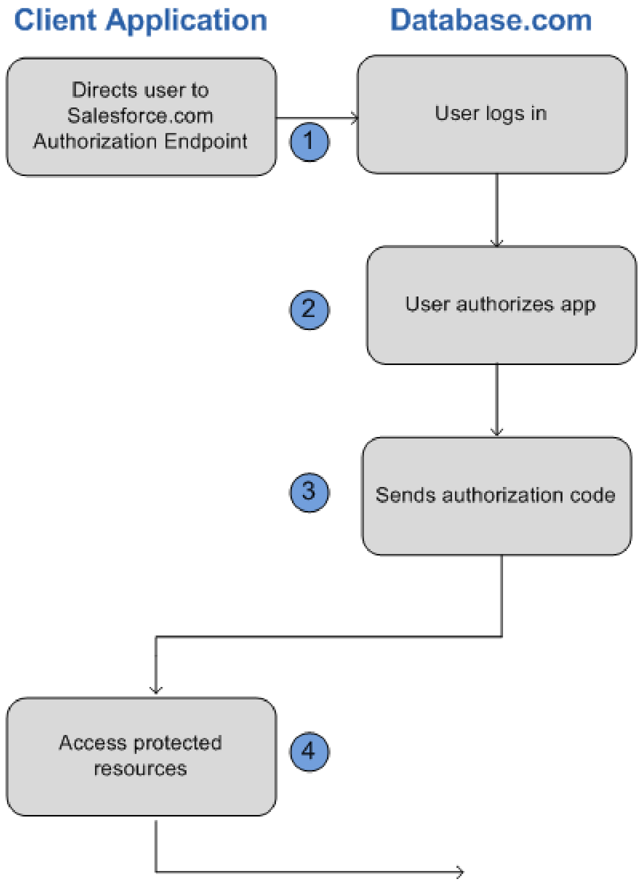
Understanding the User-Agent OAuth Authentication Flow

Typically this flow is used by mobile apps or applications that directly access theChatter 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.

Caution: 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 Database.com login page.

2. The end-user logs into Database.com to authenticate themselves. Since this Web page is hosted by the resource owner (Database.com) 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 API resources.

Using the User-Agent Flow with Chatter API and Database.com

The following provides specific details for the OAuth user-agent flow when used with Database.com and Chatter 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 remote access application defined for this application
`redirect_url`	The `Callback URL` value from the remote access application 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 API If you specify any values, you must include the `chatter_api` value. For more information, see "Scope Parameter Values" in the online help.
`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 Database.com 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 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). ⚠ **Caution:** 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 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 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 API request, you must include a valid access token in the request. This can be done by using the HTTP Authorization header:

```
Authorization: Bearer <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 remote access application definition.

You can also include the following optional parameters:

Parameters	Description
`client_secret`	Consumer secret from the remote access application 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 API:

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

Chapter 2

Quick Start

In this chapter ...

Create a sample application in your development environment to see the power and flexibility of the Chatter REST API.

Prerequisites

Completing the prerequisites make it easier to build and use the quick-start sample:

- Install your development platform according to its product documentation
- Become familiar with cURL, the tool used to execute REST requests in this quick start. If you use another tool, you should be familiar enough with it to translate the example code.
- Become familiar with JavaScript Object Notation (JSON), which is used in this quick start, or be able to translate samples from JSON to the standard you use.
- Enable an SSL endpoint in your application server
- Become familiar with OAuth 2.0, which requires some setup. We provide the steps, but it will help if you are familiarize yourself with terms and concepts.

Step One: Set Up Authorization

Setting up OAuth 2.0 requires that you take some steps within your development environment and in other locations. If any of the steps are unfamiliar, you can consult the online help or the OAuth 2.0 documentation.

1. Decide where to create your remote access application.

   Your remote access application does not have to reside in the same organization as your users. The remote access application you create can be used to sign into any organization.
2. In the appropriate organization, navigate to **Develop** > **Remote Access**, and click **New** to create a new remote access application.
3. Enter a `Callback URL`. It must be the same as your Web application's callback URL. It must be secure: `http://` does not work, only `https://`.
   For development environments, the callback URL is similar to
   `https://`***`instance_name`***`/ConnectTest/oauth/_callback`. If you work with Java it's generally a servlet.
4. Enter a URL for the `Info URL`. This is the name that is returned in the client info response, in the `applicationName` field.
5. Enter the contact email, as well as any other information appropriate to your application.
6. Click Save.
   The `Consumer Key` is created and displayed, and a `Consumer Secret` is created (click the link to reveal it).

Use this remote access application to connect to the servers.

Step Two: Connect to Chatter API Using OAuth

You must create a remote access application (such as the example in Step One: Set Up Authorization) before you can do this step.

Note: The OAuth 2.0 specification uses "client" instead of "consumer."

The following maps the terms used in the remote access application you just created to the OAuth values used in the examples.

Remote Access Application Label	Value in Example
`Consumer Key`	`client_id`
`Consumer Secret`	`client_secret`
`Callback URL`	`redirect_uri`

In addition, the value of `grant_type` depends on the OAuth authentication flow you are using. For this flow, the value is `password`.

Important: The OAuth flow being used in this example is not suitable for production client apps because it involves passing in the client secret. Use of the username-password flow is not recommended for most applications.

This example also uses the following values:

Name	Value
Instance	na1.database.com
username	admin@seattleapps.com
Consumer key	3MVG9PhR6g6B7ps4xDycwGrI4PvjVZvK9
Consumer secret	8870355475032095511
Callback URL	https://www.seattleapps.com
Grant type	password
Password	1Lsfdc!

1. Generate the access token.

   The following is an example of the cURL command to generate 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://na1.database.com/services/oauth2/token
```

   The following is the response that includes the access token:

```
{
"id":"https://login.na1.database.com/id/00DD0000000FJ6TMAW/
     005D0000001B5T4IAK",
"issued_at":"1302907727777",
"instance_url":"https://na1.database.com",
"signature":"5jcevY5fUai0lWntuSxkwBzWcvRjd01RCOkIBZpyGv0=",
"access_token":"00DD0000000FJ6T!AQkAQPde_DMF2vGzddfZmBRS95GojDbtA
     rKkgukAgZP0OVFYY5KkAqhLw9ejeKIlpJ3FgwGAWeRlBiWRt8mfXEuAZGbZNosk"
}
```

2. Access Chatter API using the access token.

```
curl -X GET https://na1.database.com/services/data/v25.0/chatter/users/me
     -H 'Authorization: Bearer 00DD0000000FJ6T!AQkAQPde_DMF2vGzddfZmBRS95Goj
     DbtArKkgukAgZP0OVFYY5KkAqhLw9ejeKIlpJ3FgwGAWeRlBiWRt8mfXEuAZGbZNosk'
```

REFERENCE

Chapter 3

Chatter API Resources Overview

This chapter contains information about the relationships between the resources as well as a tip sheet with the most common Chatter API uses.

Chatter API Resources Relationships

The following shows the relationships between the various resources. Please also note:

- Many of the resources use either a specified userId or the keyword `me` to indicate the current, logged-in user.
- The Organization resource has the base URL of `/services/v25.0/connect`.

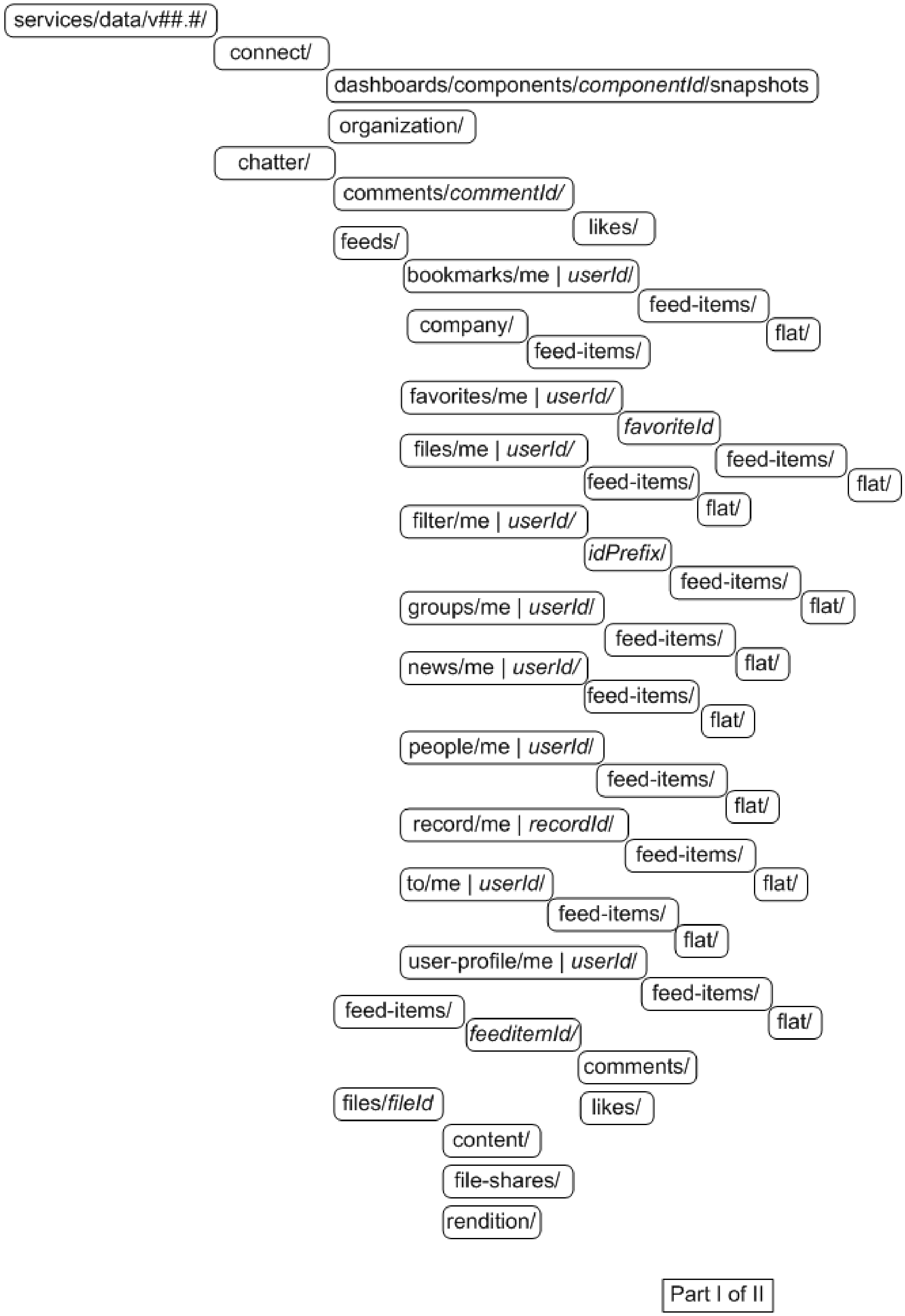


Figure 1: Chatter API Resources Relationships Part I

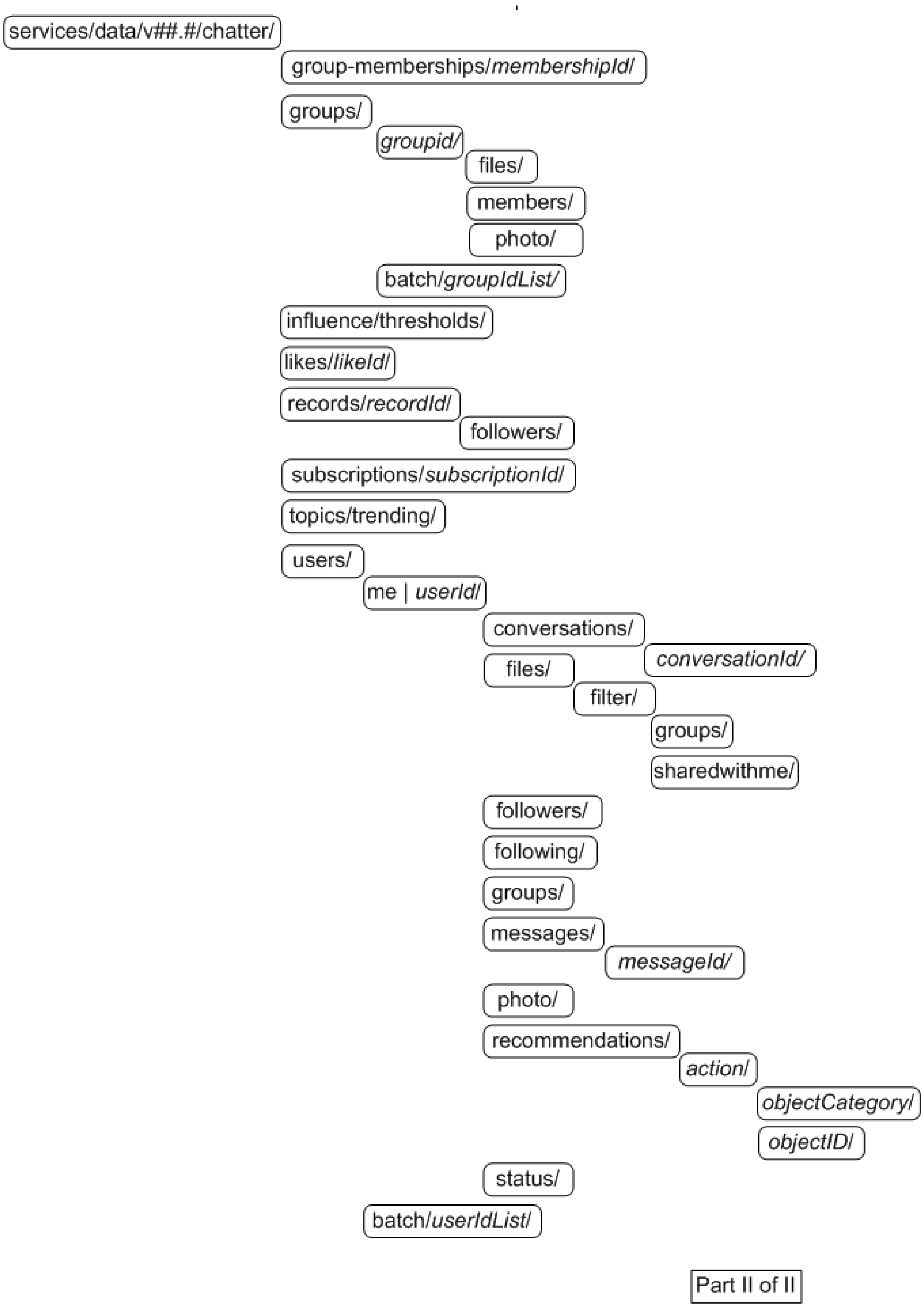


Figure 2: Chatter API Resources Relationships Part II

Chatter API: How do I. . .

- Post a feed item?
- Add a comment to a record or user?
- Post a comment or feed item with an @mention?
- Like a feed item?
- Share a feed item?
- Get my news feed?
- Get feed items posted to a record?
- Get the list of records a specific user follows?
- Get Chatter activity statistics?
- Get recommendations of people to follow?
- Join a group?
- Post to a group?
- Follow a record?
- Unfollow a record?
- Send a private message?
- How to create autocomplete for @mentions?
- Retrieve all the information for a group of users?

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, see Using POST, PATCH, or PUT for Input on page 44.

Post feed item

Resource:	Use one of the feed resources, such as `/chatter/feeds/news/`***`userId`***`/feed-items`.
HTTP method:	POST
Request body:	Feed Item Input
Request parameter:	`text`

Request body example:	`{ "body" :`
	`{`
	`"messageSegments" : [`
	`{`
	`"type": "Text",`
	`"text" : "New post"`
	`}`
	`]`
	`}`
	`}`

Request parameter example:	`/services/data/v25.0/chatter/feeds/news/me/feed-items?text=New+post`
Returns:	Feed Item

Add comments

Resource:	`/chatter/feed-items/`***`feedItemId`***`/comments`
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/v25.0/chatter/feed-items/0D5D0000000DaSbKAK/comments?text=New+comment`
Returns:	Comment

Post @mentions

<table>
<tr><td>Resource:</td><td>To post a feed-item with an @mention, use one of the feed resources, such as <code>/chatter/feeds/user-profile/<b><i>userId</i></b>/feed-items</code>. To post a comment, use the Feed-Items resource: <code>/chatter/feed-items/<b><i>feedItemId</i></b>/comments</code></td></tr>
<tr><td>HTTP method:</td><td>POST</td></tr>
<tr><td>Request body:</td><td>The appropriate container input, such as comment input or feed-item input, with the message segment input: mention</td></tr>
<tr><td>Request body example:</td><td>

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

Note: No spaces are added either before or after the @mention, so the text message segment includes a space.
</td></tr>
<tr><td>Returns</td><td>Feed Item</td></tr>
</table>

Like feed items

Resource:	`/chatter/feed-items/`***`feedItemId`***`/likes`
HTTP method:	POST
Request body:	NA
Request parameter:	NA
Example:	`/services/data/v25.0/chatter/feed-items/0D5D0000000DaZBKA0` `/likes`
Returns:	Like

Share a feed item

Resource:	Use one of the feed resources, such as `/chatter/feeds/user-profile/`***`userId`***`/feed-items/`***`feedItemId`***.
HTTP method:	POST
Request body:	`originalFeedItem`
Request parameter:	`originalFeedItem`

Request body example:	`{ "body" :`
	`{`
	`"feedItem" : [`
	`{`
	`"originalFeedItem": "0D5D0000000DaZBKA0",`
	`"text" : "Great post!"`
	`},`
	`]`
	`}`
	`}`

Returns:	Feed Item

Get news feeds

The following gets the news feed of the current user.

Resource:	`/chatter/feeds/news/me/feed-items`
HTTP method:	GET
Request body:	NA
Request parameter:	NA
Example:	`/services/data/v25.0/chatter/feeds/news/me/feed-items`
Returns:	Feed Item Page

Get feed items for a record

Resource:	`/chatter/feeds/record/`***`recordId`***`/feed-items`
HTTP method:	GET
Request body:	NA

Request parameter:	`page` (optional) **Note:** Page tokens are returned as part of the response body, in one of the URL parameters, such as `nextPageURL`.
Request parameter example:	`/services/data/v25.0/chatter/feeds/record /005x0000001oXx9AAE/feed-items?page=2011-03-31T20:57:24Z,0D5x0000001C2XPCA0`
Returns:	Feed Item Page

Get list of what user is following

Resource:	`/chatter/users/`***`userId`***`/following`
HTTP method:	GET
Request body:	NA
Request parameter:	`page` (optional)
Request parameter example:	`/services/data/v25.0/chatter/users /005D0000001GLowIAB/following?page=3`
Returns:	Subscription Page

Get Chatter activity statistics

Resource:	`/chatter/users/`***`userId`***
HTTP method:	GET
Request body:	NA
Request parameter:	NA
Example:	`/services/data/v25.0/chatter/users /005D0000001GLowIAB`
Returns:	User Detail

Get recommendations

Resource:	`/chatter/users/me/recommendations/follow/users`

HTTP method:	GET
Request body:	NA
Request parameter:	NA
Example:	`/services/data/v25.0/chatter/users` `/me/recommendations/follow/users`
Returns:	Recommendation

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/v25.0/chatter/groups/0F9D000000006bOKAQ`
	`/members?userId=005D0000001GpHp`

Returns:	Group Member

Post to a group

Resource:	`/chatter/feeds/record/`***`groupId`***`/feed-items`
HTTP method:	POST
Request body:	Feed item
Request parameter:	`Text`

Request body example:	`{ "body" :`
	`{`
	`"messageSegments" : [`
	`{`
	`"type": "Text",`
	`"text" : "Has everyone seen the latest building proposal?"`
	`}`
	`]`
	`}`
	`}`

Request parameter example:	`/feeds/record/0F9D000000006bO/feed-items?text=Has+everyone+seen+the+latest+building+proposal?`
Returns:	Feed Item

Follow a record

The `subjectId` is the ID of the item you want to follow.

Resource:	`/chatter/users/me/following`
HTTP method:	POST
Request body:	Following input
Request parameter:	`subjectId`
Request body example:	`{ "subjectId" : "001D000000Iyu2p" }`
Request parameter example:	`/services/data/v25.0/chatter/users/me/following?subjectId=001D000000Iyu2p`
Returns:	Subscription

Unfollow records

Note that a complete structured resource is not returned, just a return code.

Resource:	`/chatter/subscriptions/`***`subscriptionId`***
HTTP method:	DELETE
Request body:	NA
Request parameter:	NA

Example:	`/services/data/v25.0/chatter/subscriptions/0E8D00000001JkFKAU`
Returns:	204: Successful Delete

Send a private message

Resource:	`/chatter/users/me/messages/`
HTTP method:	POST
Request body:	Messaging input

Request parameter:	`recipients` OR `inReplyTo`
	`text`

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

Request parameters example:	`/services/data/v25.0/chatter/users`
	`/me/messages?recipients=005D0000001GOYpIAO,`
	`005D0000001GbdHIAS&Are+we+ready+for+next+week's+customer+meeting?`

Returns:	Message

Build a user name autocomplete for @mentions

Resource:	`/chatter/users?q=`***searchtext*** where *searchtext* includes wildcards. For every two or three characters the user types, you can add a wildcard and get the list of matching user names. For example, `/chatter/users?q=cha*` finds all users whose user names begin with "cha".
HTTP method:	GET
Request parameter:	`q=`***searchtext***
Request parameters example:	`/services/data/v25.0/chatter/users?q=cha*`
Returns:	User Page

Return all 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 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
Request parameter:	NA

Request parameters example:	`/services/data/v25.0/chatter/users`
	`/batch/005D0000001Kl6xIAC,005D0000005J3D4HVB, . . .`

Returns:	Batch Results

Chapter 4

Chatter API Resources

The following are Chatter API resources:

Resource	Description
Bookmarks Resources	Feed made up of Chatter feed items posted to bookmarks saved by the logged-in user. Available in versions 24.0 and later. Available resources are: Bookmarks Feed URL Bookmarks Feed Items Bookmarks Feed Items Flat—Available in versions 25.0 and higher.
Chatter Resource	Directory of the general resources available. Available in versions 24.0 and later. Contains a single resource.
Dashboard Component Snapshot Resource	Posts a new dashboard component snapshot to the specified location. Available in versions 25.0 and higher. Contains a single resource.
Comments Resource	Information about the specified comment. Also used to delete a comment, or to like a comment. Available resources are: Comment Comment Likes
Company Feed Resources	A feed that returns all updates for people the current user follows, groups the user is a member of, and files and records the user is following. Available resources are: Company Feed URL Company Feed Items
Favorites Feed Resources	Feeds made up of Chatter favorites saved by the logged-in user. Available in versions 24.0 and later. Available resources are: List of Favorites Favorite Feed Favorites Feed Items

Resource	Description
Feeds Resource	A list of all of the feeds the logged in user is able to view. Contains a single resource.
Feed-Items Resources	Represents a specific feed item. Also used to query all feed items, post comments to feed-items, as well as delete a specific feed item. Available resources are: Feed Item Comments Feed Item Likes Feed Items Query Specific Feed Item
Files Resources	Information about the specified file, including content, rendered version, and sharing. Available in versions 24.0 and later. Available resources are: File Information File Content File Rendition File Shares
Files Feed Resources	All feed items that contain files posted by people or groups that the current user follows. Available resources are: Files Feed URL Files Feed Items Files Feed Items Flat Available in versions 25.0 and higher.
Filter Feed Resources	Feeds filtered by the specified criteria for the logged-in user. Available resources are: List of Filters Keyprefix Filter Feed Keyprefix Filtered Feed Items Filter Feed Items Flat—Available in versions 25.0 and higher.
Groups Feed Resources	All feed items from all groups the current user either owns or is a member of. **Note:** To get the feed for a specific group, use the Record Feed Resource resource with a `groupId`. Available resources are: Groups Feed URL Groups Feed Items Groups Feed Items Flat—Available in versions 25.0 and higher.

Resource	Description
Group-Memberships Resource	Access information about a member of a group, or remove a member from a group. Contains a single resource.
Groups Resources	Information about groups, such as the group's members, photo, and the groups the current user is following. Also used to add members to a group and change the group photo. This is not a feed. To get the feed for a specific group, use the Record Feed Resource resource with a `groupId`. Available resources are: List of Groups Batch Group Information Group Files Group Information Group Members Group Photo
Influence Resource	List of the percentile thresholds used to categorize an organization's users by their influence in Chatter. Available in versions 24.0 and later. Contains a single resource.
Likes Resource	Information about the specified like. Also used to delete a like. Contains a single resource.
News Feed Resources	The news feed of the current user. Includes feed items from all groups the current user either owns or is a member of, as well as all files, records, and all users the current user follows. You can also use this resource to add feed-items. Available resources are: News Feed URL News Feed Items News Feed Items Flat—Available in versions 25.0 and higher.
Organization Resource	Information about the current user's organization and settings. Available in versions 24.0 and later. Contains a single resource.
People Feed Resources	All feed items posted by all of the people the current user follows. Available resources are: People Feed URL People Feed Items People Feed Items Flat—Available in versions 25.0 and higher.
Record Feed Resource	The feed of the specified record, which could be a group, person, object or file. Available resources are: Record Feed URL Record Feed Items Record Feed Items Flat—Available in versions 25.0 and higher.

Resource	Description
Records Resources	Information about the specified record, including followers. This is not a feed. To access a record feed, use the Record Feed Resource. Available resources are: Record Specifics Record Followers
Subscriptions Resource	Information about the specified subscription. Also used to delete a subscription, for example, to unfollow a record. Contains a single resource.
To Feed Resources	Feed of all @mentions of the current user and posts others make to the logged-in user's feed. Also used to post to the user's feed. Available resources are: To Feed URL To Feed Items To Feed Items Flat—Available in versions 25.0 and higher.
Topics Resource	List of the trending topics for the organization. Available in versions 24.0 and later. Contains a single resource.
User-Profile Feed Resources	Feed of all actions by the current, logged-in user on all records that can be tracked in a feed. This is different than`/chatter/feeds/news`, which returns everything in a user's feed, including posts, group updates, and record updates. Also used to add feed items. Available resources are: User–Profile Feed URL User–Profile Feed Items User-Profile Feed Items Flat—Available in versions 25.0 and higher.
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. This is not a feed. To access the user profile feed, use User-Profile Feed Resources. Available resources are: Batch User Information Change Status of Conversations Users Files, General User Files, Filtered by Group User Files, Filtered by Sharing Following Users User Information User Messages, General User Messages, Specific User Recommendations, General

Resource	Description
	Get Unread Count for Conversations User Conversations, General User Conversations, Specific User Followers User Groups User Photos User Profile Information User Recommendations for a Specific Action User Recommendations for a Specific Action and Object Category User Recommendations for a Specific Action and Object ID User Status—No longer available as of version 25.0.

Wildcards

Some of the resource request parameters support wildcards.

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

Wildcard	Description
*	Use an asterisk (*) to match one or more characters at the middle or end of your search term. Do not use the asterisk at the beginning of a search term. If you are searching for a literal asterisk in a word or phrase, then escape the asterisk (precede it with the \ character). For example, a search for `john*` finds items that start with variations on the term `john`, such as, `johnson` or `johnny`. A search for `ma*` finds items with `mary` or `marty`.
?	Use a question mark (?) to match one character at the middle or end of your search term. Do not use the question mark wildcard at the beginning of a search term. For example, a search for `jo?n` finds items with the word `john` or `joan`.

When using wildcards, consider the following issues:

- 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.

Process Flow

In general, the process flow for using Chatter API is as follows:

1. Send an HTTP request to Database.com.

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

Using POST, PATCH, or PUT for Input

HTTP POST, PATCH, and PUT methods take either request parameters or a JSON/XML request body that can include rich input. If you pass in both a request body and request parameters, sometimes a 400 error is thrown, however, generally the request parameters will be ignored.

To submit a request using request parameters, you must use a `Content-Type` header parameter with `application/x-www-form-urlencoded`.

For the request body, you must use a `Content-Type` header parameter with one of the following values:

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

Uploading Binary Files

When uploading a new file either in a comment or in a feed post using a request body and Attachment Input: New File Upload, the content type of the request must be `multipart/form-data` with one binary upload part named `feedItemFileUpload` parameter and one part holding the serialized input as either `application/json` or `application/xml`. You must specify the `accept` type of XML if you want XML back. JSON is default.

The following is an example of posting a new feed item that includes a binary file. The following specifies the `content-disposition` as `application/json` as well as shows rich JSON input. You might post a request like this using the `/chatter/feeds/news/me/feed-items` or `/chatter/feed-items/`***`feedItemId`***`/comments` resources.

```
POST /services/data/v25.0/chatter/feeds/user-profile/005x0000001T9PwAAK/feed-items HTTP/1.1
Authorization: Bearer 00Dx00000001hIF!AQ8AQKy6K1Tmko19Au4U29YanMFt53ccip
1969CAg7t.vlXqyFyMd8cEh8axN1u8o86yAKBlcuyr7YOdNSnW52qNDppsduxI
User-Agent: Jakarta Commons-HttpClient/3.0.1
Host: instance_name
Content-Length: 817
Content-Type: multipart/form-data; boundary=a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq

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

{ "body":
   {
      "messageSegments" : [
      {
         "type" : "Text",
         "text" : "High priority content "
      }, {
         "type" : "Hashtag",
         "tag" : "important"
      }, {
         "type" : "Text",
         "text" : "Please review this as soon as possible."
      }
      ]
   },
   "attachment":
   {
      "desc": "Quarterly review",
      "filename": "2012_q1"
   }
}
--a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq

Content-Disposition: form-data; name="feedItemFileUpload"; filename="2012_q1_review.ppt"
Content-Type: application/octet-stream; charset=ISO-8859-1
Content-Transfer-Encoding: binary

binary stream of file
--a7V4kRcFA8E79pivMuV2tukQ85cmNKeoEgJgq--
```

Spacing and carriage returns or line feeds (CRLF) are important.

For example, in the following line, if you have a CRLF instead of spaces, you receive an error:

`Content-Disposition: form-data; name="feedItemFileUpload"; filename="2012_q1_review.ppt"`

Also see Form-based File Upload in HTML and User–Profile Feed Items on page 126.

The following is also an example of posting a new feed item, however, this example uses XML:

```
 POST /services/data/v25.0/chatter/feeds/user-profile/005x0000001TBMVAA4/feed-items HTTP/1.1
Authorization: Bearer 00Dx00000001hQs!ASAAQClqZ7lCeGp01tC1ap0DG3eTWXxYw
_5YV9l3fSBRqdfshcE7yZa8weqxMR2DIMjLnUhIetykeydyLphphQYeG5vJFQlt
Accept: application/xml
User-Agent: Jakarta Commons-HttpClient/3.0.1
Host: instance_name
Content-Length: 897
Content-Type: multipart/form-data; boundary=0HWq8x4y4DSQ4fqjXt6MinVMyqbf1r

--0HWq8x4y4DSQ4fqjXt6MinVMyqbf1r
Content-Disposition: form-data; name="xml"
Content-Type: application/xml; charset=UTF-8
Content-Transfer-Encoding: 8bit

<feedItem>
   <body>
      <messageSegments>
         <segment>
            <text>High priority content </text>
            <type>Text</type>
         </segment>
         <segment>
             <tag>important</tag>
            <type>Hashtag</type>
         </segment>
         <segment>
            <text>Please review this as soon as possible</text>
            <type>Text</type>
         </segment>
      </messageSegments>
   </body>
   <attachment>
      <desc>Quarterly review</desc>
      <filename>2012_q1</filename>
   </attachment>
</feedItem>
--0HWq8x4y4DSQ4fqjXt6MinVMyqbf1r
Content-Disposition: form-data; name="feedItemFileUpload"; filename="2012_q1_review.ppt"
Content-Type: application/octet-stream; charset=ISO-8859-1
Content-Transfer-Encoding: binary

binary stream of file

--0HWq8x4y4DSQ4fqjXt6MinVMyqbf1r--
```

For more information on the resource, see User–Profile Feed Items on page 126.

The following is the XML version of posting a comment.

```
POST /services/data/v25.0/chatter/feed-items/0D5x00000000RryCAE/comments HTTP/1.1
Authorization: Bearer 00Dx00000001hQs!ASAAQGmoa_8P5XX7G6LP92kXX7IkPm0vCa
E2q23jvofDMWT9VUlkAYD2iuyCT2269Z_Gfd6Tdu08asLYsfSargwhYyHzC3RR
Accept: application/xml
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="xml"
Content-Type: application/xml; charset=UTF-8
Content-Transfer-Encoding: 8bit

<comment>
   <body>
      <messageSegments>
         <segment>
            <text>Please review this file</text>
            <type>Text</type>
         </segment>
         <segment>
            <url>http://www.SeattleApps.com/QuarterlyReviews</url>
            <type>Link
         </type>
         </segment>
         <segment>
            <tag>2012_q1</tag>
            <type>Hashtag</type>
         </segment>
         <segment>
            <type>Mention</type>
            <id>005x0000001TBMT</id>
         </segment>
      </messageSegments>
   </body>
   <attachment>
      <desc>2012 Q1 Review</desc>
      <fileName>2012_q1_review</fileName>
   </attachment>
</comment>
--F9jBDELnfBLAVmLNbnLIYibT5Icp0h3VJ7mkI
Content-Disposition: form-data; name="feedItemFileUpload"; filename="2012_q1_review.ppt"
Content-Type: application/octet-stream; charset=ISO-8859-1
Content-Transfer-Encoding: binary

binary stream of file

--F9jBDELnfBLAVmLNbnLIYibT5Icp0h3VJ7mkI--
```

For more information on the resource, see Comments Resource on page 53.

Bookmarks Resources

Feed made up of Chatter feed items posted to bookmarks saved by the logged-in user. Available in versions 24.0 and later.

To bookmark an item, use PATCH with the Feed Item resource and the `isBookmarkedByCurrentUser` parameter. For example:

```
/feed-item/0D5D0000000Ir6zKAC?isBookmarkedByCurrentUser=true
```

Available resources are:

Resource	Description
`/chatter/feeds/bookmarks/me`	Returns the URL for the bookmarks feed for the logged-in user.
`/chatter/feeds/bookmarks/me/feed-items`	Returns all feed items for all bookmarks for the logged-in user.
`/chatter/feeds/bookmarks/me/feed-items/flat`	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.

Bookmarks Feed URL

Returns the URL for the bookmarks feed for the logged-in user.

Resource

```
/chatter/feeds/bookmarks/me
```

OR

```
/chatter/feeds/bookmarks/userId
```

Note: *`userId`* must be the same as the logged-in user.

HTTP methods

GET or HEAD

Request parameter

Parameter Name	Type	Description
`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 `CreatedDateDesc`.

Response body

Feed

Bookmarks Feed Items

Returns all feed items for all bookmarks for the logged-in user.

Resource

```
/chatter/feeds/bookmarks/me/feed-items
```

OR

```
/chatter/feeds/bookmarks/userId/feed-items
```

Note: *`userId`* must be the same as the logged-in user.

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
`page`	Token	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/v25.0/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 between 1 and 100. If you don't specify a size, the default is 25.
`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 `CreatedDateDesc`.

Response body

Feed Item Page

Example response body

```
{
   "currentPageUrl":
"/services/data/v25.0/chatter/feeds/bookmarks/005D0000001Kl6xIAC/feed-items",
   "isModifiedUrl": null,
   "items": [
      {
         "actor": {
            "companyName": "Seattle Apps",
            "firstName": "Marion",
            "id": "005D0000001Kl6xIAC",
            "isChatterGuest": false,
            "lastName": "Raven",
            "mySubscription": null,
            "name": "Marion Raven",
            "photo": {
              "largePhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/F",

               "photoVersionId": "729D00000000BDV",
               "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/T"

            },
            "title": null,
            "type": "User",
            "url": "/services/data/v25.0/chatter/users/005D0000001Kl6xIAC"
         },
         "attachment": {
            "description": "Sharing feed items policy.",
            "downloadUrl":
"/services/data/v25.0/chatter/files/069D00000000xNXIAY/content?versionNumber=2",
            "fileSize": "371033",
            "fileType": "Adobe PDF",
            "hasImagePreview": true,
            "hasPdfPreview": true,
            "id": "069D00000000xNXIAY",
            "mimeType": "application/pdf",
            "title": "ReviewForJohn",
            "versionId": "068D00000000uFE"
         },
         "body": {
            "messageSegments": [{
               "text": "More testing of files",
               "type": "Text"
            }],
            "text": "More testing of files"
         },
         "clientInfo": null,
         "comments": {
            "comments": [],
            "currentPageUrl":
"/services/data/v25.0/chatter/feed-items/0D5D0000000Gh5VKAS/comments",
            "nextPageUrl": null,
            "total": 0
         },
         "createdDate": "2011-11-10T20:02:10.000Z",
         "event": false,
         "id": "0D5D0000000Gh5VKAS",
         "isBookmarkedByCurrentUser": true,
         "isLikedByCurrentUser": false,
         "likes": {
            "currentPageUrl":
"/services/data/v25.0/chatter/feed-items/0D5D0000000Gh5VKAS/likes",
            "likes": [],
            "nextPageUrl": null,
            "previousPageUrl": null,
            "total": 0
         },
         "modifiedDate": "2011-11-10T20:02:10.000Z",
         "myLike": null,
         "originalFeedItem": null,
         "originalFeedItemActor": null,
         "parent": {
            "companyName": "Seattle Apps",
            "firstName": "Marion",
            "id": "005D0000001Kl6xIAC",
            "isChatterGuest": false,
            "lastName": "Raven",
            "mySubscription": null,
            "name": "Marion Raven",
            "photo": {
              "largePhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/F",

               "photoVersionId": "729D00000000BDV",
               "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/T"

            },
            "title": null,
            "type": "User",
            "url": "/services/data/v25.0/chatter/users/005D0000001Kl6xIAC"
         },
         "photoUrl": "https://instance_name/profilephoto/729D00000000BDV/T",
         "type": "ContentPost",
         "url": "/services/data/v25.0/chatter/feed-items/0D5D0000000Gh5VKAS"
      },
      {
         "actor": {
            "companyName": "Seattle Apps",
            "firstName": "Marion",
            "id": "005D0000001Kl6xIAC",
            "isChatterGuest": false,
            "lastName": "Raven",
            "mySubscription": null,
            "name": "Marion Raven",
            "photo": {
              "largePhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/F",

               "photoVersionId": "729D00000000BDV",
               "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/T"

            },
            "title": null,
            "type": "User",
            "url": "/services/data/v25.0/chatter/users/005D0000001Kl6xIAC"
         },
         "attachment": null,
         "body": {
            "messageSegments": [{
               "text": "created this group.",
               "type": "Text"
            }],
            "text": "created this group."
         },
         "clientInfo": null,
         "comments": {
            "comments": [],
            "currentPageUrl":
"/services/data/v25.0/chatter/feed-items/0D5D0000000F4JnKAK/comments",
            "nextPageUrl": null,
            "total": 0
         },
         "createdDate": "2011-09-21T20:23:46.000Z",
         "event": true,
         "id": "0D5D0000000F4JnKAK",
         "isBookmarkedByCurrentUser": true,
         "isLikedByCurrentUser": false,
         "likes": {
            "currentPageUrl":
"/services/data/v25.0/chatter/feed-items/0D5D0000000F4JnKAK/likes",
            "likes": [],
            "nextPageUrl": null,
            "previousPageUrl": null,
            "total": 0
         },
         "modifiedDate": "2011-09-21T20:23:46.000Z",
         "myLike": null,
         "originalFeedItem": null,
         "originalFeedItemActor": null,
         "parent": {
            "canHaveChatterGuests": true,
            "description": "Each week we test different apps with live customers!
This group is for invites and comments.",
            "fileCount": 0,
            "id": "0F9D00000000FG5KAM",
            "lastFeedItemPostDate": "2011-09-21T20:23:46.000Z",
            "memberCount": 3,
            "myRole": "GroupOwner",
            "mySubscription": {
               "id": "0FBD00000002oJqOAI",
             "url": "/services/data/v25.0/chatter/group-memberships/0FBD00000002oJqOAI"

            },
            "name": "Testing User Experience",
            "owner": {
               "companyName": "Seattle Apps",
               "firstName": "Marion",
               "id": "005D0000001Kl6xIAC",
               "isChatterGuest": false,
               "lastName": "Raven",
               "mySubscription": null,
               "name": "Marion Raven",
               "photo": {
                "largePhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/F",

                  "photoVersionId": "729D00000000BDV",
                "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/T"

               },
               "title": null,
               "type": "User",
               "url": "/services/data/v25.0/chatter/users/005D0000001Kl6xIAC"
            },
            "photo": {
               "largePhotoUrl": "https://instance_name/profilephoto/0F9/F",
               "photoVersionId": null,
               "smallPhotoUrl": "https://instance_name/profilephoto/0F9/T"
            },
            "type": "CollaborationGroup",
            "url": "/services/data/v25.0/chatter/groups/0F9D00000000FG5KAM",
            "visibility": "PrivateAccess"
         },
         "photoUrl": "https://instance_name/profilephoto/729D00000000BDV/T",
         "type": "TrackedChange",
         "url": "/services/data/v25.0/chatter/feed-items/0D5D0000000F4JnKAK"
      }
   ],
   "nextPageUrl": null
}
```

Bookmarks Feed Items Flat

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. Available in versions 25.0 and higher.

Resource

```
/chatter/feeds/bookmarks/me/feed-items/flat
```

or

```
/chatter/feeds/bookmarks/userId/feed-items/flat
```

Note: *userId* must be the same as the logged-in user.

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
numFeedItems	Integer	Specifies the number of feed items returned. Valid values are between 0 and 100. Default value is 25. Because feed items may 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	Token	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 between 0 and 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

Chatter Resource

Directory of the general resources available. Available in versions 24.0 and later.

Resource

```
/chatter
```

HTTP methods

GET or HEAD

Response body

Chatter Directory

Example response body

```
{
   "groups" : "/services/data/v25.0/chatter/groups",
   "organization" : "/services/data/v25.0/connect/organization",
   "users" : "/services/data/v25.0/chatter/users",
   "feeds" : "/services/data/v25.0/chatter/feeds"
}
```

Comments Resource

Information about the specified comment. Also used to delete a comment, or to like a comment.

Available resources:

Resource	Description
/chatter/comments/***commentId***	Returns information about the specified comment. Also used to delete a comment.
/chatter/comments/***commentId***/likes	Returns information about the likes for the specified comment. Also used to add a like to a comment.

Comment

Returns information about the specified comment. Also used to delete a comment.

Resource

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

HTTP methods

GET, DELETE or HEAD

Response body for GET or HEAD

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/0D7D00000000LdQKAU
```

Which returns the following:

```
{
   "attachment": null,
   "body": {
      "messageSegments": [{
         "text": "Has everyone reviewed this?",
         "type": "Text"
      }],
      "text": "Has everyone reviewed this?"
   },
   "clientInfo": null,
   "createdDate": "2011-12-12T21:53:12.000+0000",
   "deletable": true,
   "feedItem": {
      "id": "0D5D0000000Gh5VKAS",
      "url": "/services/data/v25.0/chatter/feed-items/0D5D0000000Gh5VKAS"
   },
   "id": "0D7D00000000MKbKAM",
   "likes": {
     "currentPageUrl": "/services/data/v25.0/chatter/comments/0D7D00000000MKbKAM/likes",

      "likes": [],
      "nextPageUrl": null,
      "previousPageUrl": null,
      "total": 0
   },
   "myLike": null,
   "parent": {
      "id": "005D0000001Kl6xIAC",
      "url": "/services/data/v25.0/chatter/users/005D0000001Kl6xIAC"
   },
   "type": "TextComment",
   "url": "/services/data/v25.0/chatter/comments/0D7D00000000MKbKAM",
   "user": {
      "companyName": "Seattle Apps",
      "firstName": "Marion",
      "id": "005D0000001Kl6xIAC",
      "isChatterGuest": false,
      "lastName": "Raven",
      "mySubscription": null,
      "name": "Marion Raven",
      "photo": {
         "largePhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/F",
         "photoVersionId": "729D00000000BDV",
         "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/T"
      },
      "title": null,
      "type": "User",
      "url": "/services/data/v25.0/chatter/users/005D0000001Kl6xIAC"
   }
}
```

Comment Likes

Returns information about the likes for the specified comment. Also used to add a like to a comment.

Resource

```
/chatter/comments/commentId/likes
```

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 between 1 and 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

Company Feed Resources

A feed that returns all updates for people the current user follows, groups the user is a member of, and files and records the user is following.

Available resources are:

Resource	Description
`/chatter/feeds/company`	Returns a URL to the company feed.
`/chatter/feeds/company/feed-items`	Returns the feed items of a company feed.

Company Feed URL

Returns a URL to the company feed.

Resource

```
/chatter/feeds/company
```

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
`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 `CreatedDateDesc`.

Response body

Feed

Example response body

```
{
   "feedItemsUrl": "/services/data/v25.0/chatter/feeds/company/feed-items",
}
```

Company Feed Items

Returns the feed items of a company feed.

Resource

```
/chatter/feeds/company/feed-items
```

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
`page`	Token	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

Parameter Name	Type	Description
		parameters, such as `nextPageURL`, for example: `"nextPageUrl": "/services/data/v25.0/ 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 between 1 and 100. If you don't specify a size, the default is 25.
`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 `CreatedDateDesc`.

Response body

Feed Item Page

Example response body

```
{
   "currentPageUrl": "/services/data/v25.0/chatter/feeds/company/feed-items",
   "items": [
      {
         "actor": {
            "companyName": "Seattle Apps",
            "firstName": "Adam",
            "id": "005D0000001Gl4rIAC",
            "isChatterGuest": false,
            "lastName": "Grant",
            "mySubscription": null,
            "name": "Adam Grant",
            "photo": {
              "largePhotoUrl": "https://instance_name/profilephoto/729D000000007ux/F",

               "smallPhotoUrl": "https://instance_name/profilephoto/729D000000007ux/T"

            },
            "title": "Service Engineer",
            "type": "User",
            "url": "/services/data/v25.0/chatter/users/005D0000001Gl4rIAC"
         },
         "attachment": null,
         "body": {
            "messageSegments": [{
               "text": "Working today on Dreamforce presentation.",
               "type": "Text"
            }],
            "text": "Working today on Dreamforce presentation."
         },
         "clientInfo": null,
         "comments": {
            "comments": [],
            "currentPageUrl":
"/services/data/v25.0/chatter/feed-items/0D5D0000000IqJqKAK/comments",
            "nextPageUrl": null,
            "total": 0
         },
         "createdDate": "2011-08-22T22:04:18.000Z",
         "currentUserLike": null,
         "event": false,
         "id": "0D5D0000000IqJqKAK",
         "isLikedByCurrentUser": false,
         "likes": {
            "currentPageUrl":
"/services/data/v25.0/chatter/feed-items/0D5D0000000IqJqKAK/likes",
            "likes": [],
            "myLike": null,
            "nextPageUrl": null,
            "previousPageUrl": null,
            "total": 0
         },
         "modifiedDate": "2011-08-22T22:04:18.000Z",
         "parent": {
            "companyName": "Seattle Apps",
            "firstName": "Adam",
            "id": "005D0000001Gl4rIAC",
            "isChatterGuest": false,
            "lastName": "Grant",
            "mySubscription": null,
            "name": "Adam Grant",
            "photo": {
              "largePhotoUrl": "https://instance_name/profilephoto/729D000000007ux/F",

               "smallPhotoUrl": "https://instance_name/profilephoto/729D000000007ux/T"

            },
            "title": "Service Engineer",
            "type": "User",
            "url": "/services/data/v25.0/chatter/users/005D0000001Gl4rIAC"
         },
         "photoUrl": "https://instance_name/profilephoto/729D000000007ux/T",
         "type": "UserStatus",
         "url": "/services/data/v25.0/chatter/feed-items/0D5D0000000IqJqKAK"
      },
      {
         "actor": {
          . . .
         }
        . . .
      }
   "nextPageUrl": null
}
```

Favorites Feed Resources

Feeds made up of Chatter favorites saved by the logged-in user. Available in versions 24.0 and later. Favorites are searches, list views, feed items, or topics that the logged-in user has starred.

Available resources are:

Resource	Description
/chatter/feeds/favorites/me	Returns a list of all the favorites for the logged-in user. Can also be used to create a favorite.
/chatter/feeds/favorites/me/***favoriteId***	Returns the feed for the specified favorite. Can also be used to delete the specified favorite.
/chatter/feeds/favorites/me/***favoriteId***/feed-items	Returns a filtered feed, only showing those feed items for the specified favorite.

List of Favorites

Returns a list of all the favorites for the logged-in user. Can also be used to create a favorite.

Resource

```
/chatter/feeds/favorites/me
```

or

```
/chatter/feeds/favorites/userId
```

Note: *userId* must be the same as the logged-in user.

HTTP methods

GET, HEAD, or POST

Request body for POST

Available in versions 24.0 and later. In XML form, the root XML tag is expected to be `<favorite>`.

Name	Type	Description
searchText	String	Text of the search that comprises the favorite. This parameter can only be used with saved searches and not list views.

Request parameters for POST

Name	Type	Description
searchText	String	Text of the search that comprises the favorite. This parameter can only be used with saved searches and not list views.

Response body for GET or HEAD

Favorites Page

Response body for POST

Favorite

Sample response body for GET

```
{"favorites": [{
   "createdBy": {
      "companyName": "Seattle Apps",
      "firstName": "Marion",
      "id": "005D0000001Kl6xIAC",
      "isChatterGuest": false,
      "lastName": "Raven",
      "mySubscription": null,
      "name": "Marion Raven",
      "photo": {
         "largePhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/F",
         "photoVersionId": "729D00000000BDV",
         "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/T"
      },
      "title": null,
      "type": "User",
      "url": "/services/data/v25.0/chatter/users/005D0000001Kl6xIAC"
   },
   "feedUrl":
"/services/data/v25.0/chatter/feeds/favorites/me/07FD00000000Ghm/feed-items",
   "id": "07FD00000000GhmMAE",
   "lastViewDate": "2011-10-13T21:52:50.000Z",
   "name": "dreamforce",
   "searchText": "dreamforce",
   "type": "Search",
   "url": "/services/data/v25.0/chatter/feeds/favorites/me/07FD00000000GhmMAE",
   "user": {
      "companyName": "Seattle Apps",
      "firstName": "Marion",
      "id": "005D0000001Kl6xIAC",
      "isChatterGuest": false,
      "lastName": "Raven",
      "mySubscription": null,
      "name": "Marion Raven",
      "photo": {
         "largePhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/F",
         "photoVersionId": "729D00000000BDV",
         "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/T"
      },
      "title": null,
      "type": "User",
      "url": "/services/data/v25.0/chatter/users/005D0000001Kl6xIAC"
   }
}]}
```

Favorite Feed

Returns the feed for the specified favorite. Can also be used to delete the specified favorite.

Resource

```
/chatter/feeds/favorites/me/favoriteId
```

or

```
/chatter/feeds/favorites/userId/favoriteId
```

Note: `userId` must be the same as the logged-in user.

HTTP methods

GET, HEAD, PATCH, or DELETE

Request parameter for GET or HEAD

Parameter Name	Type	Description
`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 `CreatedDateDesc`.

Request body for PATCH

Available in versions 24.0 and later. In XML form, the root XML tag is expected to be `<favorite>`.

Name	Type	Description
`updateLastViewDate`	Boolean	Sets the last view date of the specified feed favorite to the 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 or HEAD

Feed

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 same as the logged-in user.

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
`page`	Token	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/v25.0/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 between 1 and 100. If you don't specify a size, the default is 25.
`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 `CreatedDateDesc`.

Response body

Feed Item Page

Feeds Resource

A list of all of the feeds the logged in 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/
```

HTTP methods

GET or HEAD

Response body

Feed Directory

Example response body

```
{
   "feeds" : [ {
      "label" : "Bookmarks",
      "feedItemsUrl" : "/services/data/v25.0/chatter/feeds/bookmarks/me/feed-items",
      "feedUrl" : "/services/data/v25.0/chatter/feeds/bookmarks/me"
}, {
      "label" : "Files",
      "feedItemsUrl" : "/services/data/v25.0/chatter/feeds/files/me/feed-items",
      "feedUrl" : "/services/data/v25.0/chatter/feeds/files/me"
}, {
      "label" : "Groups",
      "feedItemsUrl" : "/services/data/v25.0/chatter/feeds/groups/me/feed-items",
      "feedUrl" : "/services/data/v25.0/chatter/feeds/groups/me"
}, {
      "label" : "My Chatter",
      "feedItemsUrl" : "/services/data/v25.0/chatter/feeds/news/me/feed-items",
      "feedUrl" : "/services/data/v25.0/chatter/feeds/news/me"
}, {
      "label" : "People",
      "feedItemsUrl" : "/services/data/v25.0/chatter/feeds/people/me/feed-items",
      "feedUrl" : "/services/data/v25.0/chatter/feeds/people/me"
}, {
      "label" : "Record",
      "feedItemsUrl" : "/services/data/v25.0/chatter/feeds/record/me/feed-items",
      "feedUrl" : "/services/data/v25.0/chatter/feeds/record/me"
}, {
      "label" : "Me",
      "feedItemsUrl" : "/services/data/v25.0/chatter/feeds/to/me/feed-items",
      "feedUrl" : "/services/data/v25.0/chatter/feeds/to/me"
}, {
      "label" : "User Profile",
     "feedItemsUrl" : "/services/data/v25.0/chatter/feeds/user-profile/me/feed-items",

      "feedUrl" : "/services/data/v25.0/chatter/feeds/user-profile/me" } ],
   "favorites" : [ {
      "name" : "dreamforce",
      "id" : "07FD00000000GhmMAE",
      "type" : "Search",
      "user" : {
         "name" : "Marion Raven",
         "title" : null,
         "firstName" : "Marion",
         "lastName" : "Raven",
         "companyName" : "Seattle Apps",
         "mySubscription" : null,
         "photo" : {
            "largePhotoUrl" : "https://instance_name/profilephoto/729D00000000BDV/F",

         "photoVersionId" : "729D00000000BDV",
         "smallPhotoUrl" : "https://instance_name/profilephoto/729D00000000BDV/T"
      },
      "isChatterGuest" : false,
      "id" : "005D0000001Kl6xIAC",
      "url" : "/services/data/v25.0/chatter/users/005D0000001Kl6xIAC",
      "type" : "User"
   },
   "url" : "/services/data/v25.0/chatter/feeds/favorites/me/07FD00000000GhmMAE",
   "createdBy" : {
      "name" : "Marion Raven",
      "title" : null,
      "firstName" : "Marion",
      "lastName" : "Raven",
      "companyName" : "Seattle Apps",
      "mySubscription" : null,
      "photo" : {
         "largePhotoUrl" : "https://instance_name/profilephoto/729D00000000BDV/F",
         "photoVersionId" : "729D00000000BDV",
         "smallPhotoUrl" : "https://instance_name/profilephoto/729D00000000BDV/T"
      },
      "isChatterGuest" : false,
      "id" : "005D0000001Kl6xIAC",
      "url" : "/services/data/v25.0/chatter/users/005D0000001Kl6xIAC",
      "type" : "User"
   },
   "searchText" : "dreamforce",
   "feedUrl" :
"/services/data/v25.0/chatter/feeds/favorites/me/07FD00000000Ghm/feed-items",
   "lastViewDate" : "2011-10-13T21:52:50.000Z"
   } ]
}
```

Feed-Items Resources

Represents a specific feed item. Also used to query all feed items, post comments to feed-items, as well as delete a specific feed item.

Available resources:

Resource	Description
`/chatter/feed-items?q=`***`searchTerm`***	Used to search for specific feed items. The query parameter (`q`) is required with this resource. 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

Resource	Description
	items that have comments that contain the search term.
`/chatter/feed-items/`***`feedItemId`***	Used to access or delete a specific feed item. Also used to mark a feed item as bookmarked.
`/chatter/feed-items/`***`feedItemId`***`/comments`	Used to access comments for a specific feed item, or to post a comment.
`/chatter/feed-items/`***`feedItemId`***`/likes`	Used to access likes for a specific feed item, or to add a like to a specific feed item.

Feed Items Query

Used to search for specific feed items. The query parameter (`q`) is required with this resource. 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
```

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
`page`	Token	A generated token that indicates the view of feed items in the feed.
`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.
`q`	String	Specify the string to search. The search string must contain at least two characters, not including any wildcards. For more information about wildcards, see Wildcards on page 43.

Note: 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.

Response body

Feed Item Page

Specific Feed Item

Used to access or delete a specific feed item. Also used to mark a feed item as bookmarked.

Resource

```
/chatter/feed-items/feedItemId
```

HTTP methods

GET, DELETE, PATCH, or HEAD

Request body for PATCH

In XML form, the root XML tag is expected to be `<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 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 response body

```
{
   "parent" : {
      "name" : "Pierre Delacruix",
      "title" : null,
      "firstName" : "Pierre",
      "lastName" : "Delacruix",
      "companyName" : "Seattle Apps",
      "isActive" : true,
      "mySubscription" : null,
      "photo" : {
         "largePhotoUrl" : "https://instance_name/profilephoto/729D00000000Fj1/F",
         "photoVersionId" : "729D00000000Fj1",
         "smallPhotoUrl" : "https://instance_name/profilephoto/729D00000000Fj1/T" },
      "isChatterGuest" : false,
      "id" : "005D0000001L3zVIAS",
      "url" : "/services/data/v25.0/chatter/users/005D0000001L3zVIAS",
      "type" : "User" },
   "id" : "0D5D0000000HvPwKAK",
   "type" : "TextPost",
   "clientInfo" : null,
   "url" : "/services/data/v25.0/chatter/feed-items/0D5D0000000HvPwKAK", "body" : {
      "text" : "Reviewed the Racer 5 proposal",
      "messageSegments" : [ {
         "type" : "Text",
         "text" : "Reviewed the Racer 5 proposal"
      } ]
   },
   "createdDate" : "2012-04-20T22:49:34.000Z",
   "modifiedDate" : "2012-04-20T22:49:34.000Z",
   "photoUrl" : "https://instance_name/profilephoto/729D00000000Fj1/T", "comments" : {

      "total" : 0,
      "comments" : [ ],
      "nextPageUrl" : null,
      "currentPageUrl" :
"/services/data/v25.0/chatter/feed-items/0D5D0000000HvPwKAK/comments" },
   "likes" : {
      "total" : 0,
      "likes" : [ ],
      "nextPageUrl" : null,
      "currentPageUrl" :
"/services/data/v25.0/chatter/feed-items/0D5D0000000HvPwKAK/likes",
      "previousPageUrl" : null },
   "isBookmarkedByCurrentUser" : false,
   "isDeleteRestricted" : false,
   "isLikedByCurrentUser" : false,
   "myLike" : null,
   "actor" : {
      "name" : "Pierre Delacruix",
      "title" : null,
      "firstName" : "Pierre",
      "lastName" : "Delacruix",
      "companyName" : "Seattle Apps",
      "isActive" : true,
      "mySubscription" : null,
      "photo" : {
         "largePhotoUrl" : "https://instance_name/profilephoto/729D00000000Fj1/F",
         "photoVersionId" : "729D00000000Fj1",
         "smallPhotoUrl" : "https://instance_name/profilephoto/729D00000000Fj1/T" },
      "isChatterGuest" : false,
      "id" : "005D0000001L3zVIAS",
      "url" : "/services/data/v25.0/chatter/users/005D0000001L3zVIAS",
      "type" : "User" },
   "event" : false,
    "attachment" : null,
   "originalFeedItem" : null,
   "originalFeedItemActor" : null
}
```

Feed Item Comments

Used to access comments for a specific feed item, or to post a comment. See Using POST, PATCH, or PUT for Input on page 44.

Resource

```
/chatter/feed-items/feedItemId/comments
```

HTTP methods

GET, POST or HEAD

Request body for POST

Available in versions 24.0 and later.

In XML form, the root XML tag is expected to be `<comment>`.

Name	Type	Description
`attachment`	Depends on type.	Optional Values are: • Attachment Input: Existing Content • Attachment Input: Link • Attachment Input: New File Upload **Note:** If you're uploading a new file, you must include a binary file in the multipart request.
`body`	Message Body Input	Description of message body

Request parameters for POST

For file posts:

Parameter Name	Type	Description	Required or Optional
`contentDocumentId`	String	ID of an existing file to be posted.	Either `contentDocumentId` or `feedItemFileUpload` is required for file posts.
`desc`	String	Description of file being posted. Only used with `fileName`.	Optional.
`feedItemFileUpload`	String	Contents of file to be uploaded. Must be an absolute path.	Either `contentDocumentId` or

Parameter Name	Type	Description	Required or Optional
			`feedItemFileUpload` is required for file posts.
`fileName`	String	File name of a file to be uploaded and posted.	Required. **Attention:** This parameter should not be used as of version 24.0. Use `title` instead.
`text`	String	Text of the feed item.	Optional.
`title`	String	Title of the file.	Optional. Available in versions 24.0 and later.

For text posts:

Parameter Name	Type	Description	Required or Optional
`text`	String	Text of the feed item.	Required

Request parameters for GET or HEAD

Parameter Name	Type	Description

`page`	Token	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/v25.0/`
		`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 between 1 and 100. If you don't specify a size, the default is 25.
`sort`	String. One of the following: • `CreatedDateDesc`	Sorts the returned feed either by created date in ascending order, or by last modified date, also in ascending

Parameter Name	Type	Description
	• `LastModifiedDateDesc`	order. The default is `CreatedDateDesc`.

Response body for GET or HEAD

Comment Page

Response body for POST

Comment

Example response body using POST

```
{
   "parent" : {
      "id" : "005D0000001Gl4rIAC",
      "url" : "/services/data/v25.0/chatter/users/005D0000001Gl4rIAC"
   },
   "id" : "0D7D00000000LflKAE",
   "user" : {
      "name" : "Adam Grant",
      "title" : "Service Engineer",
      "firstName" : "Adam",
      "lastName" : "Grant",
      "companyName" : "Seattle Apps",
      "mySubscription" : null,
      "photo" : {
         "smallPhotoUrl" : "https://instance_name/profilephoto/729D000000007ux/T",
         "largePhotoUrl" : "https://instance_name/profilephoto/729D000000007ux/F"
      },
      "isChatterGuest" : false,
      "id" : "005D0000001Gl4rIAC",
      "type" : "User",
      "url" : "/services/data/v25.0/chatter/users/005D0000001Gl4rIAC"
   },
   "url" : "/services/data/v25.0/chatter/comments/0D7D00000000LflKAE",
   "body" : {
      "text" : "What are the new dates?",
      "messageSegments" : [ {
         "type" : "Text",
         "text" : "What are the new dates?"
      } ]
   },
   "clientInfo" : null,
   "createdDate" : "2011-08-24T18:06:30.000+0000",
   "deletable" : true,
   "feedItem" : {
      "id" : "0D5D0000000Ir6zKAC",
      "url" : "/services/data/v25.0/chatter/feed-items/0D5D0000000Ir6zKAC"
   }
}
```

Feed Item 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
```

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 between 1 and 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

Example response body with GET

```
{
   "id": "0I0D000000002XiKAI",
   "url": "/services/data/v25.0/chatter/likes/0I0D000000002XiKAI",
   "user": {
      "companyName": "Seattle Apps",
      "firstName": "Adam",
      "id": "005D0000001Gl4rIAC",
      "isChatterGuest": false,
      "lastName": "Grant",
      "mySubscription": null,
      "name": "Adam Grant",
      "photo": {
         "largePhotoUrl": "https://instance_name/profilephoto/729D000000007ux/F",
         "smallPhotoUrl": "https://instance_name/profilephoto/729D000000007ux/T"
      },
      "title": "Service Engineer",
      "type": "User",
      "url": "/services/data/v25.0/chatter/users/005D0000001Gl4rIAC"
   }
}
```

Files Resources

Information about the specified file, including content, rendered version, and sharing. Available in versions 24.0 and later.

Available resources are:

Resource	Description
/chatter/files/***fileId***	Return details about the specified file.
/chatter/files/***fileId***/content	Return the content of the file. The content is streamed as the body of the response.
/chatter/files/***fileId***/file-shares	Returns information about the entities with which the specified file has been shared. These can be users, groups, or records.
/chatter/files/***fileId***/rendition	Returns 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.

File Information

Return details about the specified file.

Resource

```
/chatter/files/fileId
```

HTTP methods

GET or HEAD

Request parameters

Name	Type	Description
versionNumber	Integer	Specify an existing version number for the file. If not specified, the latest version is returned.

Response body

File Details

Example response body

```
{
   "pageCount" : 3,
   "id" : "069D00000000xNXIAY",
   "owner" : {
      "name" : "Marion Raven",
      "title" : "Head SE",
      "firstName" : "Marion",
      "lastName" : "Raven",
      "companyName" : "Seattle Apps",
      "mySubscription" : null,
      "photo" : { "largePhotoUrl" :
"https://instance_name/profilephoto/729D00000000BDV/F",
      "photoVersionId" : "729D00000000BDV",
      "smallPhotoUrl" : "https://instance_name/profilephoto/729D00000000BDV/T" },
      "isChatterGuest" : false,
      "id" : "005D0000001Kl6xIAC",
      "type" : "User",
      "url" : "/services/data/v25.0/chatter/users/005D0000001Kl6xIAC"
   },
   "description" : "Sharing feed items policy.",
   "origin" : "Content",
   "url" : "/services/data/v25.0/chatter/files/069D00000000xNXIAY",
   "title" : "ReviewForJohn",
   "fileType" : "PDF",
   "flashRenditionStatus" : "Success",
   "versionNumber" : "2",
   "modifiedDate" : "2011-11-10T19:53:04.000Z",
   "contentSize" : 371033,
   "contentUrl" : null,
   "pdfRenditionStatus" : "NA",
   "thumb240By180RenditionStatus" : "Success",
   "thumb720By480RenditionStatus" : "Success",
   "thumb120By90RenditionStatus" : "Success",
   "mySubscription" : {
      "id" : "0E8D0000000SipsKAC",
      "url" : "/services/data/v25.0/chatter/subscriptions/0E8D0000000SipsKAC"
   }
   "type" : "ContentDocument"
}
```

File Content

Return the content of the file. The content is streamed as the body of the response.

Resource

```
/chatter/files/fileId/content
```

HTTP methods

GET

Request parameters

Name	Type	Description
versionNumber	Integer	Specify an existing version number for the file. If not specified, the latest version is returned.

Response body

Binary stream of the file content.

File Shares

Returns information about the entities with which the specified file has been shared. These can be users, groups, or records.

Resource

```
/chatter/files/fileId/file-shares
```

HTTP methods

GET or HEAD

Response body for GET or HEAD

File Shares Page

Example response body for GET or HEAD

```
{
   "currentPageUrl": "/services/data/v/chatter/files/069D00000000yO6IAI/file-shares",
   "nextPageUrl": null,
   "previousPageUrl": null,
   "shares": [
      {
         "entity": {
            "companyName": "Seattle Apps",
            "firstName": "Marion",
            "id": "005D0000001Kl6xIAC",
            "isChatterGuest": false,
            "lastName": "Raven",
            "mySubscription": null,
            "name": "Marion Raven",
            "photo": {
              "largePhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/F",

               "photoVersionId": "729D00000000BDV",
               "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/T"

            },
            "title": null,
            "type": "User",
            "url": "/services/data/v25.0/chatter/users/005D0000001Kl6xIAC"
         },
         "sharingType": "I"
      },
      {
         "entity": {
            "id": "058D00000000Rz7IAE",
            "name": "Review library",
            "type": "ContentWorkspace",
            "url": "/services/data/v25.0/chatter/records/058D00000000Rz7IAE"
         },
         "sharingType": "I"
      }
   ]
}
```

File Rendition

Returns 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.

Resource

```
/chatter/files/fileId/rendition
```

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` • `SLIDE` • `THUMB120BY90` • `THUMB240BY180` • `THUMB720BY480`

Response body

Binary stream of the rendition.

Files Feed Resources

All feed items that contain files posted by people or groups that the current user follows.

Available resources are:

Resource	Description
`/chatter/feeds/files/me`	Returns the URL to the feed-items that contain files that are posted by users or groups the current user is following.
`/chatter/feeds/files/me/feed-items`	Returns all feed items posted with files by people or groups that the logged-in user follows.

Resource	Description
/chatter/feeds/files/me/feed-items/flat	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.

Files Feed URL

Returns the URL to the feed-items that contain files that are posted by users or groups the current user is following.

Resource

```
/chatter/feeds/files/me
```

OR

```
/chatter/feeds/files/userId
```

Note: *userId* must be the same as the logged-in user.

HTTP methods

GET or HEAD

Request parameter

Parameter Name	Type	Description
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 CreatedDateDesc.

Response body

Feed

Files Feed Items

Returns all feed items posted with files by people or groups that the logged-in user follows.

Resource

```
/chatter/feeds/files/me/feed-items
```

or

```
/chatter/feeds/files/userId/feed-items
```

Note: *`userId`* must be the same as the logged-in user.

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description

`page`	Token	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/v25.0/`
		`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 between 1 and 100. If you don't specify a size, the default is 25.
`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 `CreatedDateDesc`.

Response body

Feed Item Page

Response body example

```
{ "items" : [ ],
   "nextPageUrl" : null,
   "currentPageUrl" :
"/services/data/v25.0/chatter/feeds/files/005D0000001L3zVIAS/feed-items?page=2012-04-17T18%3A13%3A57Z",

   "isModifiedUrl" : null
   }
```

Files Feed Items Flat

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. Available in versions 25.0 and higher.

Resource

```
/chatter/feeds/files/me/feed-items/flat
```

or

```
/chatter/feeds/files/userId/feed-items/flat
```

Note: `userId` must be the same as the logged-in user.

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
numFeedItems	Integer	Specifies the number of feed items returned. Valid values are between 0 and 100. Default value is 25. Because feed items may 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	Token	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 between 0 and 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 Resources

Feeds filtered by the specified criteria for the logged-in user.

Available resources are:

Resource	Description
`/chatter/feeds/filter/me`	List of Filters
`/chatter/feeds/filter/me/`***`idPrefix`***	Use this resource to return the URL for the specified key prefix feed.
`/chatter/feeds/filter/me/`***`idPrefix`***`/feed-items`	Use this resource to return the feed items for the specified key prefix, such as the feed items from all the groups the logged-in user is a member of.
`/chatter/feeds/filter/me/`***`idPrefix`***`/feed-items/flat`	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.

List of Filters

Use this resource to return a list of the possible filters for a user. You should get this list, then construct additional resource URLs using the keyprefixes that are returned.

Resource

```
/chatter/feeds/filter/me
```

or

```
/chatter/feeds/filter/userId
```

`userId` must be the same as the logged-in user.

HTTP methods

GET or HEAD

Response body

Feed Directory

Keyprefix Filter Feed

Use this resource to return the URL for the specified key prefix feed. The key prefix (*`idPrefix`*) 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 feeds to see the available key prefixes.

Resource

```
/chatter/feeds/filter/me/idPrefix
```

or

```
/chatter/feeds/filter/userId/idPrefix
```

userId must be the same as the logged-in user.

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
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 CreatedDateDesc.

Response body

Feed

Example response body

The following is an example of a URL that's filtered by user Id, that is prefix 005.

```
{
   "feedItemsUrl":
"/services/data/v25.0/chatter/feeds/filter/005D0000001Gl4rIAC/005/feed-items",
}
```

Keyprefix Filtered Feed Items

Use this resource to return the feed items for the specified key prefix, such as the feed items from all the groups the logged-in user is a member of.

Resource

```
/chatter/feeds/filter/me/idPrefix/feed-items
```

or

```
/chatter/feeds/filter/userId/idPrefix/feed-items
```

userId must be the same as the logged-in user.

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description

`page`	Token	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/v25.0/`
		`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 between 1 and 100. If you don't specify a size, the default is 25.
`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 `CreatedDateDesc`.

Response body

Feed Item Page

Filter Feed Items Flat

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. Available in versions 25.0 and higher.

Resource

```
/chatter/feeds/filter/me/feed-items/flat
```

or

```
/chatter/feeds/filter/userId/feed-items/flat
```

Note: *`userId`* must be the same as the logged-in user.

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
`numFeedItems`	Integer	Specifies the number of feed items returned. Valid values are between 0 and 100. Default value is 25. Because feed items may 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`	Token	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 between 0 and 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

Groups Feed Resources

All feed items 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-items for all the groups the current, signed-in user is a member of.
`/chatter/feeds/groups/me/feed-items`	Returns all feed items from all groups the current user either owns or is a member of.

Resource	Description
`/chatter/feeds/groups/me/feed-items/flat`	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.

Groups Feed URL

Returns a URL to the feed-items for all the groups the current, signed-in user is a member of.

Resource

```
/chatter/feeds/groups/me
```

or

```
/chatter/feeds/groups/userId
```

Note: *`userId`* must be the same as the logged-in user.

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
`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 `CreatedDateDesc`.

Response body

Feed

Groups Feed Items

Returns all feed items from all groups the current user either owns or is a member of.

Resource

```
/chatter/feeds/groups/me/feed-items
```

or

```
/chatter/feeds/groups/userId/feed-items
```

Note: *userId* must be the same as the logged-in user.

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
page	Token	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/v25.0/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 between 1 and 100. If you don't specify a size, the default is 25.
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 CreatedDateDesc.

Response body

Feed Item Page

Groups Feed Items Flat

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. Available in versions 25.0 and higher.

Resource

```
/chatter/feeds/groups/me/feed-items/flat
```

or

```
/chatter/feeds/groups/userId/feed-items/flat
```

Note: `userId` must be the same as the logged-in user.

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
`numFeedItems`	Integer	Specifies the number of feed items returned. Valid values are between 0 and 100. Default value is 25. Because feed items may 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`	Token	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 between 0 and 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

Group-Memberships Resource

Access information about a member of a group, or remove a member from a group.

Membership IDs are returned in the `mySubscription` value of the response body from the groups resource. The following resource returns the membership ID for the logged-in user.

```
/groups/0F9D000000006bOKA
```

Resource

```
/chatter/group-memberships/membershipId
```

HTTP methods

GET, DELETE or HEAD

Response body for GET or HEAD

Group Member

Response body example for GET

```
{
   "id": "0FBD00000002kYFOAY",
   "role": "GroupOwner",
   "url": "/services/data/v25.0/chatter/group-memberships/0FBD00000002kYFOAY",
   "user": {
      "companyName": "Seattle Apps",
      "firstName": "Adam",
      "id": "005D0000001Gl4rIAC",
      "isChatterGuest": false,
      "lastName": "Grant",
      "mySubscription": null,
      "name": "Adam Grant",
      "photo": {
         "largePhotoUrl": "https://instance_name/profilephoto/729D000000007ux/F",
         "smallPhotoUrl": "https://instance_name/profilephoto/729D000000007ux/T"
      },
      "title": "Service Engineer",
      "type": "User",
      "url": "/services/data/v25.0/chatter/users/005D0000001Gl4rIAC"
   }
}
```

Groups Resources

Information about groups, such as the group's members, photo, and the groups the current user is following. Also used to 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/groups/`	Returns a list of groups the logged–in user is a member of.
`/chatter/groups/`***`groupId`***	Returns information about the specified group.
`/chatter/groups/batch/`***`group_list`***	Returns information about more than one group at the same time.

Resource	Description
/chatter/groups/***groupID***/files	Returns information about the files posted to the specified group. Available in versions 24.0 and later.
/chatter/groups/***groupId***/members	Returns a list of members for the specified group. Also use this resource and POST to add members to a group.
/chatter/groups/***groupId***/photo	Use this resource to access the photo for the specified group, add, update, or delete the photo.

List of Groups

Returns a list of groups the logged–in user is a member of.

Resource

```
/chatter/groups/
```

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
page	Token	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/v25.0/ 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 between 1 and 250. If you don't specify a size, the default is 25.
q	String	Specify the string to search. The search string must contain at least two characters, not including any wildcards. For more information about wildcards, see Wildcards on page 43.

Response body

Group Page

Example response body

```
 "currentPageUrl": "/services/data/v25.0/chatter/groups",
   "groups": [
      {
         "canHaveChatterGuests": false,
         "description": "Get company announcements and other important updates",
         "fileCount": 0,
         "id": "0F9D00000000FG0KAM",
         "lastFeedItemPostDate": "2011-09-21T19:58:58.000Z",
         "memberCount": 2,
         "myRole": "GroupOwner",
         "mySubscription": {
            "id": "0FBD00000002oJVOAY",
            "url": "/services/data/v25.0/chatter/group-memberships/0FBD00000002oJVOAY"

         },
         "name": "All Seattle Apps",
         "owner": {
            "companyName": "Seattle Apps",
            "firstName": "Marion",
            "id": "005D0000001Kl6xIAC",
            "isChatterGuest": false,
            "lastName": "Raven",
            "mySubscription": null,
            "name": "Marion Raven",
            "photo": {
              "largePhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/F",

               "photoVersionId": "729D00000000BDV",
               "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/T"

            },
            "title": null,
            "type": "User",
            "url": "/services/data/v25.0/chatter/users/005D0000001Kl6xIAC"
         },
         "photo": {
            "largePhotoUrl": "https://instance_name/profilephoto/729D00000000BDQ/F",
            "photoVersionId": "729D00000000BDQ",
            "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000BDQ/T"
         },
         "type": "CollaborationGroup",
         "url": "/services/data/v25.0/chatter/groups/0F9D00000000FG0KAM",
         "visibility": "PublicAccess"
      },
      {
         "canHaveChatterGuests": true,
         "description": "Each week we test different apps with live customers!
This group is for invites and comments.",
         "fileCount": 0,
         "id": "0F9D00000000FG5KAM",
         "lastFeedItemPostDate": "2011-09-21T20:23:46.000Z",
         "memberCount": 3,
         "myRole": "GroupOwner",
         "mySubscription": {
            "id": "0FBD00000002oJqOAI",
            "url": "/services/data/v25.0/chatter/group-memberships/0FBD00000002oJqOAI"

         },
         "name": "Testing User Experience",
         "owner": {
            "companyName": "Seattle Apps",
            "firstName": "Marion",
            "id": "005D0000001Kl6xIAC",
            "isChatterGuest": false,
            "lastName": "Raven",
            "mySubscription": null,
            "name": "Marion Raven",
            "photo": {
              "largePhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/F",

               "photoVersionId": "729D00000000BDV",
               "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/T"

            },
            "title": null,
            "type": "User",
            "url": "/services/data/v25.0/chatter/users/005D0000001Kl6xIAC"
         },
         "photo": {
            "largePhotoUrl": "https://instance_name/profilephoto/0F9/F",
            "photoVersionId": null,
            "smallPhotoUrl": "https://instance_name/profilephoto/0F9/T"
         },
         "type": "CollaborationGroup",
         "url": "/services/data/v25.0/chatter/groups/0F9D00000000FG5KAM",
         "visibility": "PrivateAccess"
      },
      {
         "canHaveChatterGuests": false,
         "description": "This group contains information about our monthly volunteer
opportunity.
Please join so you can also volunteer!
List any additional volunteer opportunities as well.",
         "fileCount": 0,
         "id": "0F9D00000000FG1KAM",
         "lastFeedItemPostDate": "2011-09-21T20:06:03.000Z",
         "memberCount": 2,
         "myRole": "GroupOwner",
         "mySubscription": {
            "id": "0FBD00000002oJWOAY",
            "url": "/services/data/v25.0/chatter/group-memberships/0FBD00000002oJWOAY"

         },
         "name": "Volunteer Opportunities",
         "owner": {
            "companyName": "Seattle Apps",
            "firstName": "Marion",
            "id": "005D0000001Kl6xIAC",
            "isChatterGuest": false,
            "lastName": "Raven",
            "mySubscription": null,
            "name": "Marion Raven",
            "photo": {
              "largePhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/F",

               "photoVersionId": "729D00000000BDV",
               "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/T"

            },
            "title": null,
            "type": "User",
            "url": "/services/data/v25.0/chatter/users/005D0000001Kl6xIAC"
         },
         "photo": {
            "largePhotoUrl": "https://instance_name/profilephoto/0F9/F",
            "photoVersionId": null,
            "smallPhotoUrl": "https://instance_name/profilephoto/0F9/T"
         },
         "type": "CollaborationGroup",
         "url": "/services/data/v25.0/chatter/groups/0F9D00000000FG1KAM",
         "visibility": "PublicAccess"
      }
   ],
   "nextPageUrl": null,
   "previousPageUrl": null
}
```

Group Information

Returns information about the specified group.

Resource

```
/chatter/groups/groupId
```

HTTP methods

GET or HEAD

Response body

Group

Example response body

```
{
   "canHaveChatterGuests": false,
   "description": "Find out everything that&#39;s going on with the product! Join this
 group!",
   "id": "0F9D00000000BsTKAU",
   "memberCount": 23,
   "myRole": "GroupOwner",
   "mySubscription": {
      "id": "0FBD00000002kYFOAY",
      "url": "/services/data/v25.0/chatter/group-memberships/0FBD00000002kYFOAY"
   },
   "name": "Seattle Prime",
   "photo": {
      "largePhotoUrl": "https://instance_name/profilephoto/0F9/F",
      "smallPhotoUrl": "https://instance_name/profilephoto/0F9/T"
   },
   "type": "CollaborationGroup",
   "url": "/services/data/v25.0/chatter/groups/0F9D00000000BsTKAU",
   "visibility": "PublicAccess"
}
```

Batch Group Information

Returns information about more than one group at the same time.

Resource

```
/chatter/groups/batch/group_list
```

Where *group_list* is a comma separated list of group IDs.

HTTP methods

GET or HEAD

Response body

Batch Results

Group Files

Returns information about the files posted to the specified group. Available in versions 24.0 and later.

Resource

```
/chatter/groups/groupID/files
```

HTTP methods

GET or HEAD

Response body

File Page

Example response body

```
{
   "currentPageUrl": "/services/data/v25.0/chatter/users/005D0000001Kl6xIAC/files",
   "files": [
      {
         "contentSize": 7193018,
         "contentUrl": null,
         "description": null,
         "fileType": "PDF",
         "flashRenditionStatus": "SUCCESS",
         "id": "069D00000000yO6IAI",
         "modifiedDate": "2011-11-18T22:14:23.000Z",
         "origin": "C",
         "owner": {
            "companyName": "Seattle Apps",
            "firstName": "Marion",
            "id": "005D0000001Kl6xIAC",
            "isChatterGuest": false,
            "lastName": "Raven",
            "mySubscription": null,
            "name": "Marion Raven",
            "photo": {
              "largePhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/F",

               "photoVersionId": "729D00000000BDV",
               "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/T"

            },
            "title": null,
            "type": "User",
            "url": "/services/data/v25.0/chatter/users/005D0000001Kl6xIAC"
         },
         "pdfRenditionStatus": "NA",
         "thumb120By90RenditionStatus": "SUCCESS",
         "thumb240By180RenditionStatus": "SUCCESS",
         "thumb720By480RenditionStatus": "SUCCESS",
         "title": "Web Services API",
         "type": "ContentDocument",
         "url": "/services/data/v25.0/chatter/files/069D00000000yO6IAI",
         "versionNumber": "1"
      },
      {
         "contentSize": 371033,
         "contentUrl": null,
         "description": "Sharing feed items policy.",
         "fileType": "PDF",
         "flashRenditionStatus": "SUCCESS",
         "id": "069D00000000xNXIAY",
         "modifiedDate": "2011-11-10T19:53:04.000Z",
         "origin": "H",
         "owner": {
            "companyName": "Seattle Apps",
            "firstName": "Marion",
            "id": "005D0000001Kl6xIAC",
            "isChatterGuest": false,
            "lastName": "Raven",
            "mySubscription": null,
            "name": "Marion Raven",
            "photo": {
              "largePhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/F",

               "photoVersionId": "729D00000000BDV",
               "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/T"

            },
            "title": null,
            "type": "User",
            "url": "/services/data/v25.0/chatter/users/005D0000001Kl6xIAC"
         },
         "pdfRenditionStatus": "NA",
         "thumb120By90RenditionStatus": "SUCCESS",
         "thumb240By180RenditionStatus": "SUCCESS",
         "thumb720By480RenditionStatus": "SUCCESS",
         "title": "ReviewForJohn",
         "type": "ContentDocument",
         "url": "/services/data/v25.0/chatter/files/069D00000000xNXIAY",
         "versionNumber": "2"
      }
   ],
   "nextPageUrl": null,
   "previousPageUrl": null
}
```

Group Members

Returns a list of members for the specified group. Also use this resource and POST to add members to a group.

Resource

```
/chatter/groups/groupId/members
```

HTTP methods

GET, POST or HEAD

Request body for POST

In XML form, the root XML tag is expected to be `<member>`.

Name	Type	Description
`userId`	String	The user ID of the member to be added to the group

Request parameter for POST

Parameter Name	Type	Description
`userId`	String	ID of the user to add to the group.

Request parameters for GET or HEAD

Parameter Name	Type	Description
`page`	Token	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/v25.0/ 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 between 1 and 1000. If you don't specify a size, the default is 25.

Response body for GET or HEAD

Group Member Page

Response body for POST

Group Member

Example response body for GET

```
{
   "currentPageUrl": "/services/data/v25.0/chatter/groups/0F9D00000000BsTKAU/members",

   "members": [
      {
         "id": "0FBD00000002kYFOAY",
         "role": "GroupOwner",
         "url": "/services/data/v25.0/chatter/group-memberships/0FBD00000002kYFOAY",
         "user": {
            "companyName": "Seattle Apps",
            "firstName": "Adam",
            "id": "005D0000001Gl4rIAC",
            "isChatterGuest": false,
            "lastName": "Grant",
            "mySubscription": null,
            "name": "Adam Grant",
            "photo": {
              "largePhotoUrl": "https://instance_name/profilephoto/729D000000007ux/F",

               "smallPhotoUrl": "https://instance_name/profilephoto/729D000000007ux/T"

            },
            "title": "Service Engineer",
            "type": "User",
            "url": "/services/data/v25.0/chatter/users/005D0000001Gl4rIAC"
         }
      },
      {
         "id": "0FBD00000002zClOAI",
         "role": "StandardMember",
         "url": "/services/data/v25.0/chatter/group-memberships/0FBD00000002zClOAI",
         "user": {
            "companyName": "Seattle Chatter",
            "firstName": "Devi",
            "id": "005D0000001GpHpIAK",
            "isChatterGuest": false,
            "lastName": "Tapash",
            "mySubscription": {
               "id": "005D0000001GpHpIAK",
               "url": "/services/data/v25.0/chatter/subscriptions/0E8D00000001VQCKA2"
            },
            "name": "Devi Tapash",
            "photo": {
              "largePhotoUrl": "https://instance_name/profilephoto/729D00000000CSQ/F",

               "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000CSQ/T"

            },
            "title": "Service Engineer",
            "type": "User",
            "url": "/services/data/v25.0/chatter/users/005D0000001GpHpIAK"
         }
      }
   ],
   "myMembership": {
      "id": "0FBD00000002kYFOAY",
      "url": "/services/data/v25.0/chatter/group-memberships/0FBD00000002kYFOAY"
   },
   "nextPageUrl": null,
   "previousPageUrl": null,
   "totalMemberCount": 2
}
```

Group Photo

Use this resource to access the photo for the specified group, add, update, or delete the photo.

Resource

```
/chatter/groups/groupId/photo
```

HTTP methods

GET, POST, DELETE, or HEAD

Request body for POST

In XML form, the root XML tag is expected to be `<photo>`. Available in versions 25.0 and higher.

Name	Type	Description
`fileId`	String	18–character Id of an already existing file to be used as the photo
`version`	Integer	Version number of the already existing file

Request parameters for POST

You can specify either `fileUpload` or `fileId`. You'll receive an error if you specify both.

Parameter Name	Type	Description
`fileUpload`	String *`multipart/form-data for a file`*	Contents of file to be uploaded
`fileId`	String *`ID`*	ID of a file already uploaded. The file must have a content type of image, and be smaller than 2 MB.

Note: Photos are processed asynchronously and may not be visible right away.

Response body for GET or HEAD

Photo

Response body for POST

Photo

Influence Resource

List of the percentile thresholds used to categorize an organization's users by their influence in Chatter. Available in versions 24.0 and later. 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
```

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.

Like

Resource

```
/chatter/likes/likeId
```

HTTP methods

GET, DELETE or HEAD

Response body for GET or HEAD

Like

Example response body

You can find the ID for a specific like from any feed, such as a news feed or a record feed.

```
{
   "id": "0I0D000000002ZtKAI",
   "url": "/services/data/v25.0.0/chatter/likes/0I0D000000002ZtKAI",
   "user": {
      "companyName": "Seattle Apps",
      "firstName": "Adam",
      "id": "005D0000001Gl4rIAC",
      "isChatterGuest": false,
      "lastName": "Grant",
      "mySubscription": null,
      "name": "Adam Grant",
      "photo": {
         "largePhotoUrl": "https://instance_name/profilephoto/729D000000007ux/F",
         "smallPhotoUrl": "https://instance_name/profilephoto/729D000000007ux/T"
      },
      "title": "Service Engineer",
      "type": "User",
      "url": "/services/data/v25.0/chatter/users/005D0000001Gl4rIAC"
   }
}
```

News Feed Resources

The news feed of the current user. Includes feed items from all groups the current user either owns or is a member of, as well as all files, records, and all users the current user follows. You can also use this resource to add feed-items.

Available resources are:

Resource	Description
`/chatter/feeds/news/me`	Returns a URL to the feed-items for all feeds for the logged–in user.
`/chatter/feeds/news/me/feed-items`	Returns all feed items from all groups the logged-in user either owns or is a member of, as well as all files, records, and all users the current user follows.
`/chatter/feeds/news/me/feed-items/flat`	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.

News Feed URL

Returns a URL to the feed-items for all feeds for the logged–in user.

Resource

```
/chatter/feeds/news/me
```

or

```
/chatter/feeds/news/userId
```

Note: `userId` must be the same as the logged-in user.

HTTP methods

GET or HEAD

Request parameter

Parameter Name	Type	Description
`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 `CreatedDateDesc`.

Response body

Feed

News Feed Items

Returns all feed items from all groups the logged-in user either owns or is a member of, as well as all files, records, and all users the current user follows. You can also use this resource to add feed-items. See Using POST, PATCH, or PUT for Input on page 44.

Resource

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

or

```
/chatter/feeds/news/userId/feed-items
```

Note: `userId` must be the same as the logged-in user.

HTTP methods

GET, POST, or HEAD

Request body for POST

In XML form, the root XML tag is expected to be `<feedItem>`.

Name	Type	Description
`attachment`	Depends on type	Optional. You can specify only one of these values in the attachment parameter. Values are: • Attachment Input: Existing Content • Attachment Input: Link • Attachment Input: New File Upload **Note:** If you're uploading a new file, you must include a binary file in the multipart request.
`body`	Message Body Input	Description of message body
`isBookmarkedByCurrentUser`	Boolean	Specifies if the new feed item should be bookmarked for the user (`true`) or not (`false`. Default value is `false`. This parameter is only available for PATCH methods.
`originalFeedItemId`	String	If this feed item is shared, this is the original feed item 18-character `Id` **Note:** This parameter takes precedence over all other parameters. For example, if you share a post and also try to upload a file, the generated feed-item only contains the shared post. You should only share a post or upload a file, not both in the same post.

Request parameters for POST

Use only one of these sets of parameters at a time. For example, you can't make both a link post and a file post at the same time. Sharing posts take precedence over all other types of posts. If you post using the sharing parameter `originalFeedItemId` and any other parameters, such as a file post parameter `contentDocumentId`, the file post parameters are ignored.

Note: If you're uploading a new file, you must include a binary file in the multipart request. See Using POST, PATCH, or PUT for Input on page 44.

File post parameters:

Parameter Name	Type	Description	Required or Optional
`contentDocumentId`	String	ID of an existing file to be posted.	Either `contentDocumentId` or `feedItemFileUpload` is required for file posts.
`desc`	String	Description of file being posted. Only used with `fileName`.	Optional.
`feedItemFileUpload`	String	Contents of file to be uploaded. Must be an absolute path.	Either `contentDocumentId` or `feedItemFileUpload` is required for file posts.
`fileName`	String	File name of a file to be uploaded and posted.	Required. ⚠ **Attention:** This parameter should not be used as of version 24.0. Use `title` instead.
`text`	String	Text of the feed item.	Optional.
`title`	String	Title of the file.	Optional. Available in versions 24.0 and later.

Link post parameters:

Parameter Name	Type	Description	Required or Optional
`text`	String	Text of the feed item.	Optional
`url`	String	Href of the URL.	Required
`urlName`	String	Description name of the URL.	Optional

Share post parameters:

Parameter Name	Type	Description	Required or Optional
`originalFeedItemId`	String	ID of the existing feed item to share.	Required. Available in versions 24.0 and later.
`text`	String	Text of the feed item.	Optional.

Text post parameters:

Parameter Name	Type	Description	Required or Optional
`text`	String	Text of the feed item.	Required

Request parameters for GET or HEAD

Parameter Name	Type	Description
`page`	Token	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/v25.0/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 between 1 and 100. If you don't specify a size, the default is 25.
`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 `CreatedDateDesc`.

Response body for GET or HEAD

Feed Item Page

Response body for POST

Feed Item

Example of file post

The following is an example of updating a news feed with a file post.

```
curl --form feedItemFileUpload=@/home/ujones/Desktop/FY12_results.ppt
     --form fileName=FY12_results
     --form "desc=FY 12 SE Division Sales Results"
     --form "text=Please review"
     --header 'Authorization: Bearer 00Dx0000000M5Li!ARIAQCZ4DkPlHzGMaYKgVubVN.
              K3ztqnibMTXDGDs8zEjbB7wVC6xA0pcfOMFWwJkjmNhNIH5RbVZ722hBeN0VE_c8jNFrAv'

     https://instance_name/services/data/v25.0/chatter/feeds/news/me/feed-items
```

The following is an example in Java of updating 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/v25.0/chatter/" +
              "feeds/user-profile/me/feed-items";
String text = "I love posting files to Chatter!";
File contentFile = getFile();
String desc = "This is a test file that I'm posting.";
String fileName = "contentFile";
Part[] parts = {
        new StringPart("desc", desc),
        new StringPart("fileName", 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", "x-www-form-urlencoded")
    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

The following creates a link post using the request parameters:

```
/feeds/news/me/feed-items?text=Did+you+see+this?&url=http://www.chatter.com
```

The following is the request body for creating a link in a feed-item post:

```
{ "body" :
   {
      "messageSegments" : [
         {
           "type": "Link",
           "url" : "http://www.chatter.com"
         },
         {
           "type": "Text",
           "text" : "Is everyone following this?"
         }
      ]
    }
}
```

The following is the request body for including 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

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. Available in versions 25.0 and higher.

Resource

```
/chatter/feeds/news/me/feed-items/flat
```

or

```
/chatter/feeds/news/userId/feed-items/flat
```

Note: *userId* must be the same as the logged-in user.

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
numFeedItems	Integer	Specifies the number of feed items returned. Valid values are between 0 and 100. Default value is 25. Because feed items may include comments, specifying 25 flat feed items may

Parameter Name	Type	Description
		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`	Token	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 between 0 and 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

Example response

```
{
 "items" : [ {
   "type" : "Comment",
   "item" : {
      "parent" : {
         "id" : "005D0000001L3zVIAS",
         "url" : "/services/data/v25.0/chatter/users/005D0000001L3zVIAS"
      },
      "id" : "0D7D00000000N5rKAE",
      "type" : "TextComment",
      "user" : {
         "name" : "Pierre Delacruix",
         "title" : null,
         "firstName" : "Pierre",
         "lastName" : "Delacruix",
         "companyName" : "Seattle Apps",
         "mySubscription" : null,
         "photo" : {
            "largePhotoUrl" : "https://instance_name/profilephoto/729D00000000Fj1/F",

            "photoVersionId" : "729D00000000Fj1IAE",
            "smallPhotoUrl" : "https://instance_name/profilephoto/729D00000000Fj1/T"
         },
         "isChatterGuest" : false,
         "id" : "005D0000001L3zVIAS",
         "url" : "/services/data/v25.0/chatter/users/005D0000001L3zVIAS",
         "type" : "User"
      },
      "url" : "/services/data/v25.0/chatter/comments/0D7D00000000N5rKAE",
      "body" : {
         "text" : "Time to get working",
         "messageSegments" : [ {
            "type" : "Text",
            "text" : "Time to get working"
            } ]
      },
      "clientInfo" : null,
      "createdDate" : "2012-03-18T00:03:20.000Z",
      "likes" : {
         "total" : 0,
         "likes" : [ ],
         "nextPageUrl" : null,
         "currentPageUrl" : null,
         "previousPageUrl" : null
      },
      "isDeleteRestricted" : false,
      "myLike" : null,
      "attachment" : null,
      "feedItem" : {
         "id" : "0D5D0000000HWC2KAO",
         "url" : "/services/data/v25.0/chatter/feed-items/0D5D0000000HWC2KAO"
      }
   }
}
```

Organization Resource

Information about the current user's organization and settings. Available in versions 24.0 and later.

Note: This resource is under the base URL of `/services/data/v25.0/connect`, not `/services/data/v25.0/chatter`.

Organization

Resource

```
/connect/organization
```

HTTP methods

GET or HEAD

Response body

Organization

Example response body

```
{
   "orgId" : "00DD0000000JW3yMAG",
   "features" : {
      "files" : false,
      "trendingTopics" : true,
      "chatterActivity" : true,
      "chatterGlobalInfluence" : true,
      "chatterMessages" : true,
      "feedPolling" : false,
      "filesOnComments" : true,
      "viralInvitesAllowed" : true,
      "dashboardComponentSnapshots" : false
   },
   "userSettings" : {
      "userId" : "005D0000001Kl6xIAC",
      "canFollow" : true,
      "canOwnGroups" : true,
      "canModifyAllData" : true,
      "canViewAllData" : true,
      "canViewAllUsers" : true,
      "canViewAllGroups" : true,
      "canViewFullUserProfile" : true,
      "canViewPublicFiles" : true,
      "externalUser" : false
      "userLocale" : "en_US",
   },
   "accessTimeout" : 0
}
```

People Feed Resources

All feed items posted by all of the people the current user follows.

Available resources are:

Resource	Description
/chatter/feeds/people/me	Returns a URL to the feed-items for all the people the current user is following.
/chatter/feeds/people/me/feed-items	Returns all the feed items from all the people the current user follows.
/chatter/feeds/people/me/feed-items/flat	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.

People Feed URL

Returns a URL to the feed-items for all the people the current user is following.

Resource

```
/chatter/feeds/people/me
```

or

```
/chatter/feeds/people/userId
```

Note: *userId* must be the same as the logged-in user.

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
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 CreatedDateDesc.

Response body

Feed

People Feed Items

Returns all the feed items from all the people the current user follows.

Resource

```
/chatter/feeds/people/me/feed-items
```

or

```
/chatter/feeds/people/userId/feed-items
```

Note: *userId* must be the same as the logged-in user.

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
page	Token	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

Parameter Name	Type	Description
		parameters, such as `nextPageURL`, for example: `"nextPageUrl": "/services/data/v25.0/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 between 1 and 100. If you don't specify a size, the default is 25.
`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 `CreatedDateDesc`.

Response body

Feed Item Page

People Feed Items Flat

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. Available in versions 25.0 and higher.

Resource

```
/chatter/feeds/people/me/feed-items/flat
```

or

```
/chatter/feeds/people/userId/feed-items/flat
```

Note: *`userId`* must be the same as the logged-in user.

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
numFeedItems	Integer	Specifies the number of feed items returned. Valid values are between 0 and 100. Default value is 25. Because feed items may 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	Token	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 between 0 and 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

Record Feed Resource

The feed of the specified record, which could be a group, person, object or file.

Available resources are:

Resource	Description
/chatter/feeds/record/me	Returns a URL to the feed-items for all the records the current user is following, or a URL to all the feed-items of the specified *recordId*.
/chatter/feeds/record/me/feed-items	Returns the feed-items for all the records the current, signed-in user is following, or all the feed-items of the specified *recordId*.
/chatter/feeds/record/me/feed-items/flat	Returns a flat feed, that is, a feed where there is no hierarchy between feed items and comments: Feed items and comments

Resource	Description
	are at the same level. All items are returned in chronological order.

Record Feed URL

Returns a URL to the feed-items for all the records the current user is following, or a URL to all the feed-items of the specified *recordId*.

Resource

```
/chatter/feeds/record/me
```

or

```
/chatter/feeds/record/recordId
```

Note: If *recordId* is a user ID, it must be the same as the current user.

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
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 CreatedDateDesc.

Response body

Feed

Record Feed Items

Returns the feed-items for all the records the current, signed-in user is following, or all the feed-items of the specified *recordId*. You can also use this resource to POST a new feed item. See Using POST, PATCH, or PUT for Input on page 44.

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.

HTTP methods

GET, POST or HEAD

Request body for POST

In XML form, the root XML tag is expected to be `<feedItem>`.

Name	Type	Description
`attachment`	Depends on type	Optional. You can specify only one of these values in the attachment parameter. Values are: • Attachment Input: Existing Content • Attachment Input: Link • Attachment Input: New File Upload **Note:** If you're uploading a new file, you must include a binary file in the multipart request.
`body`	Message Body Input	Description of message body
`isBookmarkedByCurrentUser`	Boolean	Specifies if the new feed item should be bookmarked for the user (`true`) or not (`false`. Default value is `false`. This parameter is only available for PATCH methods.
`originalFeedItemId`	String	If this feed item is shared, this is the original feed item 18-character `Id` **Note:** This parameter takes precedence over all other parameters. For example, if you share a post and also try to upload a file, the generated feed-item only contains the shared post. You should only share a post or upload a file, not both in the same post.

Request parameters for POST

Use only one of these sets of parameters at a time. For example, you can't make both a link post and a file post at the same time. Sharing posts take precedence over all other types of posts. If you post using the sharing parameter `originalFeedItemId` and any other parameters, such as a file post parameter `contentDocumentId`, the file post parameters are ignored.

Note: If you're uploading a new file, you must include a binary file in the multipart request. See Using POST, PATCH, or PUT for Input on page 44.

File post parameters:

Parameter Name	Type	Description	Required or Optional
`contentDocumentId`	String	ID of an existing file to be posted.	Either `contentDocumentId` or `feedItemFileUpload` is required for file posts.
`desc`	String	Description of file being posted. Only used with `fileName`.	Optional.
`feedItemFileUpload`	String	Contents of file to be uploaded. Must be an absolute path.	Either `contentDocumentId` or `feedItemFileUpload` is required for file posts.
`fileName`	String	File name of a file to be uploaded and posted.	Required. **Attention:** This parameter should not be used as of version 24.0. Use `title` instead.
`text`	String	Text of the feed item.	Optional.
`title`	String	Title of the file.	Optional. Available in versions 24.0 and later.

Link post parameters:

Parameter Name	Type	Description	Required or Optional
`text`	String	Text of the feed item.	Optional
`url`	String	Href of the URL.	Required
`urlName`	String	Description name of the URL.	Optional

Share post parameters:

Parameter Name	Type	Description	Required or Optional
`originalFeedItemId`	String	ID of the existing feed item to share.	Required. Available in versions 24.0 and later.
`text`	String	Text of the feed item.	Optional.

Text post parameters:

Parameter Name	Type	Description	Required or Optional
`text`	String	Text of the feed item.	Required

Request parameters for GET or HEAD

Parameter Name	Type	Description

`page`	Token	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/v25.0/`
		`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 between 1 and 100. If you don't specify a size, the default is 25.
`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 `CreatedDateDesc`.

Response body for GET or HEAD

Feed Item Page

Response body for POST

Feed Item

Example response body for POST

```
{
   "parent" : {
      "name" : "Seattle App Land",
      "description" : "Find out everything that&#39;s going on with the product!",
      "visibility" : "PublicAccess",
      "memberCount" : 203,
      "canHaveChatterGuests" : false,
      "mySubscription" : {
         "id" : "0FBD00000002kYFOAY",
         "url" : "/services/data/v25.0/chatter/group-memberships/0FBD00000002kYFOAY"
      },
      "myRole" : "GroupOwner",
      "photo" : { "smallPhotoUrl" : "https://instance_name/profilephoto/0F9/T",
      "largePhotoUrl" : "https://instance_name/profilephoto/0F9/F" },
      "id" : "0F9D00000000BsTKAU",
      "type" : "CollaborationGroup",
      "url" : "/services/data/v25.0/chatter/groups/0F9D00000000BsTKAU"
   },
   "id" : "0D5D0000000ItYkKAK",
   "type" : "TextPost",
   "url" : "/services/data/v25.0/chatter/feed-items/0D5D0000000ItYkKAK",
   "body" : {
      "text" : "Has everyone seen the latest building proposal?",
      "messageSegments" : [ {
         "type" : "Text",
         "text" : "Has everyone seen the latest building proposal?"
      } ]
   },
   "clientInfo" : null,
   "createdDate" : "2011-08-25T19:03:49.000Z",
   "modifiedDate" : "2011-08-25T19:03:49.000Z",
   "photoUrl" : "https://instance_name/profilephoto/729D000000007ux/T",
   "comments" : {
      "total" : 0,
      "comments" : [ ],
      "nextPageUrl" : null,
      "currentPageUrl" :
"/services/data/v25.0/chatter/feed-items/0D5D0000000ItYkKAK/comments"
   },
   "likes" : {
      "total" : 0,
      "likes" : [ ],
      "previousPageUrl" : null,
      "nextPageUrl" : null,
      "currentPageUrl" :
"/services/data/v25.0/chatter/feed-items/0D5D0000000ItYkKAK/likes",
      "myLike" : null
   },
   "isLikedByCurrentUser" : false,
   "currentUserLike" : null,
   "actor" : {
      "name" : "Adam Grant",
      "title" : "Service Engineer",
      "firstName" : "Adam",
      "lastName" : "Grant",
      "companyName" : "Seattle Apps",
      "mySubscription" : null,
      "photo" : {
         "smallPhotoUrl" : "https://instance_name/profilephoto/729D000000007ux/T",
         "largePhotoUrl" : "https://instance_name/profilephoto/729D000000007ux/F"
         },
      "isChatterGuest" : false,
      "id" : "005D0000001Gl4rIAC",
      "type" : "User",
      "url" : "/services/data/v25.0/chatter/users/005D0000001Gl4rIAC"
   },
   "event" : false,
   "attachment" : null
}
```

Record Feed Items Flat

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. Available in versions 25.0 and higher.

Resource

```
/chatter/feeds/record/me/feed-items/flat
```

or

```
/chatter/feeds/record/userId/feed-items/flat
```

Note: *`userId`* must be the same as the logged-in user.

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
`numFeedItems`	Integer	Specifies the number of feed items returned. Valid values are between 0 and 100. Default value is 25. Because feed items may 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`	Token	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 between 0 and 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

Parameter Name	Type	Description
	• `LastModifiedDateDesc`	last modified date, also in ascending order. The default is `LastModifiedDateDesc`

Response body

Flat Feed

Records Resources

Information about the specified record, including followers.

Note: This is not a feed. To obtain a feed for a record, use the Record Feed Resource.

Available resources are:

Resource	Description
`/chatter/records/`***`recordId`***	Returns information about the specific record, such as the name of the record, the record type, a URL to the feed-items for the record, and so on.
`/chatter/records/`***`recordId`***`/followers`	Returns information about the followers of the specified record.

Record Specifics

Returns information about the specific record, such as the name of the record, the record type, a URL to the feed-items for the record, and so on.

Resource

```
/chatter/records/recordId
```

HTTP methods

GET or HEAD

Response body

Record Summary

Example response body

```
{
   "id": "0F9D00000000BsTKAU",
   "name": "Seattle Apps",
   "type": "CollaborationGroup",
   "url": "/services/data/v25.0/chatter/groups/0F9D00000000BsTKAU"
}
```

Record Followers

Returns information about the followers of the specified record.

Resource

```
/chatter/records/recordId/followers
```

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description

`page`	Token	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/v25.0/`
		`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 between 1 and 1000. If you don't specify a size, the default is 25.

Response body

Subscription Page

Subscriptions Resource

Information about the specified subscription. Also used to delete a subscription, for example, to unfollow a record. 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
```

HTTP methods

GET, DELETE, or HEAD

Response body for GET or HEAD

Subscription

Example response body for GET

```
{
   "id": "0E8D0000000SV5ZKAW",
   "subject": {
      "companyName": "Seattle Apps",
      "firstName": "Joe",
      "id": "005D0000001KlmFIAS",
      "isChatterGuest": false,
      "lastName": "Blogs",
      "mySubscription": {
         "id": "0E8D0000000SV5ZKAW",
         "url": "/services/data/v25.0/chatter/subscriptions/0E8D0000000SV5ZKAW"
      },
      "name": "Joe Blogs",
      "photo": {
         "largePhotoUrl": "https://instance_name/profilephoto/729D00000000BDf/F",
         "photoVersionId": "729D00000000BDf",
         "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000BDf/T"
      },
      "title": "Senior Blogger",
      "type": "User",
      "url": "/services/data/v25.0/chatter/users/005D0000001KlmFIAS"
   },
   "subscriber": {
      "companyName": "Seattle Apps",
      "firstName": "Marion",
      "id": "005D0000001Kl6xIAC",
      "isChatterGuest": false,
      "lastName": "Raven",
      "mySubscription": null,
      "name": "Marion Raven",
      "photo": {
         "largePhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/F",
         "photoVersionId": "729D00000000BDV",
         "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/T"
      },
      "title": null,
      "type": "User",
      "url": "/services/data/v25.0/chatter/users/005D0000001Kl6xIAC"
   },
   "url": "/services/data/v25.0/chatter/subscriptions/0E8D0000000SV5ZKAW"
}
```

To Feed Resources

Feed of all @mentions of the current user and posts others make to the logged-in user's feed. Also used to post to the user's feed.

Available resources are:

Resource	Description
/chatter/feeds/to/me	Returns a URL to the feed-items for the feed of the logged-in user.
/chatter/feeds/to/me/feed-items	Returns the feed-items for all @mentions of the logged-in user, as well as all posts other users make to the user's feed.
/chatter/feeds/to/me/feed-items/flat	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.

To Feed URL

Returns a URL to the feed-items for the feed of the logged-in user.

Resource

```
/chatter/feeds/to/me
```

OR

```
/chatter/feeds/to/userId
```

Note: *userId* must be the same as the logged-in user.

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
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 CreatedDateDesc.

Response body

Feed

To Feed Items

Returns the feed-items for all @mentions of the logged-in user, as well as all posts other users make to the user's feed. You can also use this resource to POST a new feed item. See Using POST, PATCH, or PUT for Input on page 44.

Resource

```
/chatter/feeds/to/me/feed-items
```

OR

```
/chatter/feeds/to/userId/feed-items
```

Note: *`userId`* must be the same as the logged-in user.

HTTP methods

GET, POST or HEAD

Request body for POST

In XML form, the root XML tag is expected to be `<feedItem>`.

Name	Type	Description
`attachment`	Depends on type	Optional. You can specify only one of these values in the attachment parameter. Values are: • Attachment Input: Existing Content • Attachment Input: Link • Attachment Input: New File Upload **Note:** If you're uploading a new file, you must include a binary file in the multipart request.
`body`	Message Body Input	Description of message body
`isBookmarkedByCurrentUser`	Boolean	Specifies if the new feed item should be bookmarked for the user (`true`) or not (`false`. Default value is `false`. This parameter is only available for PATCH methods.

Name	Type	Description
originalFeedItemId	String	If this feed item is shared, this is the original feed item 18-character Id **Note:** This parameter takes precedence over all other parameters. For example, if you share a post and also try to upload a file, the generated feed-item only contains the shared post. You should only share a post or upload a file, not both in the same post.

Request parameters for POST

Use only one of these sets of parameters at a time. For example, you can't make both a link post and a file post at the same time. Sharing posts take precedence over all other types of posts. If you post using the sharing parameter originalFeedItemId and any other parameters, such as a file post parameter contentDocumentId, the file post parameters are ignored.

Note: If you're uploading a new file, you must include a binary file in the multipart request. See Using POST, PATCH, or PUT for Input on page 44.

File post parameters:

Parameter Name	Type	Description	Required or Optional
contentDocumentId	String	ID of an existing file to be posted.	Either contentDocumentId or feedItemFileUpload is required for file posts.
desc	String	Description of file being posted. Only used with fileName.	Optional.
feedItemFileUpload	String	Contents of file to be uploaded. Must be an absolute path.	Either contentDocumentId or feedItemFileUpload is required for file posts.
fileName	String	File name of a file to be uploaded and posted.	Required. **Attention:** This parameter should not be used as of version 24.0. Use title instead.
text	String	Text of the feed item.	Optional.

Parameter Name	Type	Description	Required or Optional
`title`	String	Title of the file.	Optional. Available in versions 24.0 and later.

Link post parameters:

Parameter Name	Type	Description	Required or Optional
`text`	String	Text of the feed item.	Optional
`url`	String	Href of the URL.	Required
`urlName`	String	Description name of the URL.	Optional

Share post parameters:

Parameter Name	Type	Description	Required or Optional
`originalFeedItemId`	String	ID of the existing feed item to share.	Required. Available in versions 24.0 and later.
`text`	String	Text of the feed item.	Optional.

Text post parameters:

Parameter Name	Type	Description	Required or Optional
`text`	String	Text of the feed item.	Required

Request parameters for GET or HEAD

Parameter Name	Type	Description
`page`	Token	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/v25.0/chatter/feeds/news/005D0000001GLowIAN/feed-items?page=2011-03-30T17:34:50Z, 0D5D0000000DSv4KAG"`

Parameter Name	Type	Description
`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.
`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 `CreatedDateDesc`.

Response body for GET or HEAD

Feed Item Page

Response body for POST

Feed Item

Example request body for an @mention POST

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

Note: No spaces are added either before or after the @mention, so the text message segment includes a space.

To Feed Items Flat

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. Available in versions 25.0 and higher.

Resource

```
/chatter/feeds/to/me/feed-items/flat
```

or

```
/chatter/feeds/to/userId/feed-items/flat
```

Note: `userId` must be the same as the logged-in user.

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
`numFeedItems`	Integer	Specifies the number of feed items returned. Valid values are between 0 and 100. Default value is 25. Because feed items may 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`	Token	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 between 0 and 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

Topics Resource

Trending Topics

List of the trending topics for the organization. Available in versions 24.0 and later.

Resource

```
/chatter/topics/trending
```

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

Topics

Example response body

```
{
   "name": "#UniversalPaper"
}
{
   "name": "#caseUpdates"
}
{
   "name": "#salesTips"
}
{
   "name": "#companypicnic"
}
{
   "name": "#AccountExecs"
}
```

User-Profile Feed Resources

Feed of all actions by the current, logged-in user on all records that can be tracked in a feed. This is different than`/chatter/feeds/news`, which returns everything in a user's feed, including posts, group updates, and record updates. Also used to add feed items.

Available resources are:

Resource	Description
`/chatter/feeds/user-profile/me`	Returns a URL to the feed-items for the user-profile feed of the signed-in user.
`/chatter/feeds/user-profile/me/feed-items`	Returns the feed-items for all actions by the logged-in user on all records that can be tracked in a feed.
`/chatter/feeds/user-profile/me/feed-items/flat`	Returns a flat feed, that is, a feed where there is no hierarchy between feed items and comments: Feed items and comments

Resource	Description
	are at the same level. All items are returned in chronological order.

User–Profile Feed URL

Returns a URL to the feed-items for the user-profile feed of the signed-in user.

Resource

```
/chatter/feeds/user-profile/me
```

or

```
/chatter/feeds/user-profile/userId
```

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
`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 `CreatedDateDesc`.

Response body

Feed

User–Profile Feed Items

Returns the feed-items for all actions by the logged-in user on all records that can be tracked in a feed. You can also use this resource to POST a new feed item. See Using POST, PATCH, or PUT for Input on page 44.

Resource

```
/chatter/feeds/user-profile/me/feed-items
```

or

```
/chatter/feeds/user-profile/userId/feed-items
```

HTTP methods

GET, POST or HEAD

Request body for POST

In XML form, the root XML tag is expected to be `<feedItem>`.

Name	Type	Description
`attachment`	Depends on type	Optional. You can specify only one of these values in the attachment parameter. Values are: • Attachment Input: Existing Content • Attachment Input: Link • Attachment Input: New File Upload **Note:** If you're uploading a new file, you must include a binary file in the multipart request.
`body`	Message Body Input	Description of message body
`isBookmarkedByCurrentUser`	Boolean	Specifies if the new feed item should be bookmarked for the user (`true`) or not (`false`. Default value is `false`. This parameter is only available for PATCH methods.
`originalFeedItemId`	String	If this feed item is shared, this is the original feed item 18-character `Id` **Note:** This parameter takes precedence over all other parameters. For example, if you share a post and also try to upload a file, the generated feed-item only contains the shared post. You should only share a post or upload a file, not both in the same post.

Request parameters for POST

Use only one of these sets of parameters at a time. For example, you can't make both a link post and a file post at the same time. Sharing posts take precedence over all other types of posts. If you post using the sharing parameter `originalFeedItemId` and any other parameters, such as a file post parameter `contentDocumentId`, the file post parameters are ignored.

Note: If you're uploading a new file, you must include a binary file in the multipart request. See Using POST, PATCH, or PUT for Input on page 44.

File post parameters:

Parameter Name	Type	Description	Required or Optional
contentDocumentId	String	ID of an existing file to be posted.	Either contentDocumentId or feedItemFileUpload is required for file posts.
desc	String	Description of file being posted. Only used with fileName.	Optional.
feedItemFileUpload	String	Contents of file to be uploaded. Must be an absolute path.	Either contentDocumentId or feedItemFileUpload is required for file posts.
fileName	String	File name of a file to be uploaded and posted.	Required. ⚠ **Attention:** This parameter should not be used as of version 24.0. Use title instead.
text	String	Text of the feed item.	Optional.
title	String	Title of the file.	Optional. Available in versions 24.0 and later.

Link post parameters:

Parameter Name	Type	Description	Required or Optional
text	String	Text of the feed item.	Optional
url	String	Href of the URL.	Required
urlName	String	Description name of the URL.	Optional

Share post parameters:

Parameter Name	Type	Description	Required or Optional
originalFeedItemId	String	ID of the existing feed item to share.	Required. Available in versions 24.0 and later.
text	String	Text of the feed item.	Optional.

Text post parameters:

Parameter Name	Type	Description	Required or Optional
`text`	String	Text of the feed item.	Required

Request parameters for GET or HEAD

Parameter Name	Type	Description
`page`	Token	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/v25.0/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 between 1 and 100. If you don't specify a size, the default is 25.
`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 `CreatedDateDesc`.

Response body for GET or HEAD

Feed Item Page

Response body for POST

Feed Item

User-Profile Feed Items Flat

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. Available in versions 25.0 and higher.

Resource

```
/chatter/feeds/user-profile/me/feed-items/flat
```

or

```
/chatter/feeds/user-profile/userId/feed-items/flat
```

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
`numFeedItems`	Integer	Specifies the number of feed items returned. Valid values are between 0 and 100. Default value is 25. Because feed items may 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`	Token	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 between 0 and 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

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	Returns information about the logged-in or specified user's profile.
/chatter/users/batch/***user_list***	Returns information about one or more of the specified user's profile.
/chatter/users/me/conversations	Returns all private conversations for the logged–in user. Also used to search across all conversations.
/chatter/users/me/conversations/***conversationId***	Returns information about the specified private conversation for the logged-in 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 logged-in or specified user.
/chatter/users/me/files/	Returns information about files that a user owns. This does not files shared with a user, files a user follows, or general organization files. Can also be used to post a file. Available in versions 24.0 and later.
/chatter/users/me/files/filter/groups	Returns information about files posted to groups that the specified user is a member of. Available in versions 24.0 and later.
/chatter/users/me/files/filter/sharedwithme	Returns information about files that have been shared with the specified user. Available in versions 24.0 and later.
/chatter/users/me/followers	Returns information about the followers for the specified user.
/chatter/users/me/following	Returns a list of people, groups, records, and files that the specified user is following. Also used to follow records.
/chatter/users/me/groups	Returns the groups the specified user is a member of.
/chatter/users/me/messages	Returns all the messages for all the private conversations for the logged-in user. Also used to search across all messages and post a message.
/chatter/users/me/messages/***messageId***	Returns the specified message.
/chatter/users/me/photo	Returns a URL to the photo for the logged-in user. Also used to post photos.

Resource	Description
/chatter/users/me/recommendations	Returns the user, group, file, and record recommendations for the logged-in user. Available in versions 24.0 and later.
/chatter/users/me/recommendations/***action***	Returns the recommendations of the specified action for the logged-in user. Available in versions 24.0 and later.
/chatter/users/me/recommendations/***action***/***objectCategory***	Returns the recommendations of the specified action and object category for the logged-in user. Also used to delete recommendations.
/chatter/users/me/recommendations/***action***/***objectID***	Returns the recommendation of the specified action and object ID for the logged-in user. Also used to delete a recommendation. Available in versions 24.0 and later.
/chatter/users/me/status	Returns the most recent user status feed item. Also used to update or delete user status.

User Information

Returns information about all users in an organization.

Resource

```
/chatter/users
```

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
page	Token	A generated token that indicates the view of feed items in the feed.
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.
q	String	Specify the string to search. The search string must contain at least two characters, not including any wildcards. For more information about wildcards, see Wildcards on page 43.

Note: The query parameter `q` only searches user names, not email addresses, titles, and so on.

Response body

User Page

User Profile Information

Returns information about the logged-in or specified user's profile.

Resource

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

or

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

HTTP methods

GET or HEAD

Response body

User Detail

Example response body

```
{
   "aboutMe": "I&#39;m excited to be part of the team focused on building out our apps
 business
and showing our customers their future.",
   "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": 4
   },
   "chatterInfluence": {
      "percentile": "0.5",
      "rank": 1
   },
   "companyName": "Seattle Apps",
   "email": "mraven@seattleapps.com",
   "firstName": "Marion",
   "followersCount": 1,
   "followingCounts": {
      "people": 5,
      "records": 2,
      "total": 7
   },
   "groupCount": 3,
   "id": "005D0000001Kl6xIAC",
   "isActive": true,
   "isChatterGuest": false,
   "lastName": "Raven",
   "managerId": "000000000000000AAA",
   "managerName": "",
   "mySubscription": null,
   "name": "Marion Raven",
   "phoneNumbers": [{
      "number": "(206) 999-2846",
      "type": "Work"
   }],
   "photo": {
      "largePhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/F",
      "photoVersionId": "729D00000000BDV",
      "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/T"
   },
   "title": null,
   "type": "User",
   "url": "/services/data/v25.0/chatter/users/005D0000001Kl6xIAC",
   "username": "mraven@seattleapps.com"
}
```

Batch User Information

Returns information about one or more of the specified user’s profile.

Resource

```
/chatter/users/batch/user_list
```

Where *user_list* is a comma separated list of user IDs.

HTTP methods

GET or HEAD

Response body

Batch Results

User Conversations, General

Returns all private conversations for the logged–in user. Also used to search across all conversations. See also User Conversations, Specific on page 135. Also see User Messages, General on page 144.

Resource

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

or

```
/chatter/users/userId/conversations
```

Note: `userId` must be the same as the logged-in user.

HTTP methods

GET or HEAD

Request parameters for GET or HEAD

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 between 1 and 100. If you don't specify a size, the default is 25.

Request parameter for GET

Parameter Name	Type	Description
`q` (optional)	String	Specify the string to search. The search string must contain at least two characters, not including any wildcards. For more information about wildcards, see Wildcards on page 43. Available in versions 24.0 and later. **Note:** The query parameter only searches message text bodies, not user names, titles, and so on.

Response body

Conversation Summary Collection

User Conversations, Specific

Returns information about the specified private conversation for the logged-in 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 137.

Resource

```
/chatter/users/me/conversations/conversationId
```

or

```
/chatter/users/userId/conversations/conversationId
```

Note: `userId` must be the same as the logged-in user.

HTTP methods

GET, PATCH, or HEAD

Request parameter for GET

Parameter Name	Type	Description
`q` (optional)	String	Specify the string to search. The search string must contain at least two characters, not including any wildcards. For more information about wildcards, see Wildcards on page 43. Available in versions 24.0 and later. **Note:** The query parameter only searches message text bodies, not user names, titles, and so on.

Request body for PATCH

Available in versions 24.0 and later. In XML form, the root XML tag is expected to be `<conversation>`.

Name	Type	Description
`read`	Boolean	Changes the conversation status to unread (`false`) or read (`true`).

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 on page 44. 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
```

HTTP methods

POST or PUT

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 logged-in or specified user.

Resource

```
/chatter/users/me/conversations/unread-count
```

OR

```
/chatter/users/userId/conversations/unread-count
```

HTTP methods

GET or HEAD

Request body

Conversation Unread Count

Users Files, General

Returns information about files that a user owns. This does not files shared with a user, files a user follows, or general organization files. Can also be used to post a file. Available in versions 24.0 and later. See also Using POST, PATCH, or PUT for Input on page 44.

Important: Be aware that this resource may return a large amount of data and may take a while to process.

Resource

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

or

```
/chatter/users/userId/files/
```

HTTP methods

GET, POST, or HEAD

Request body for POST

Available in versions 24.0 and later. Describes a new file input. Assumes that a file blob is also be provided in a binary form part named `fileData`. In XML form, the root XML tag is expected to be `<fileInput>`.

Name	Type	Description
`description`	String	Description of the file to be uploaded
`title`	String	Title of the file

Request parameters for GET or HEAD

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 between 1 and 100. If you don't specify a size, the default is 25.

Request parameters for POST

Name	Type	Description
description	String	Description of the file
fileData	String *`multipart/form-data for a file`*	Contents of the file to be uploaded
title	String	Title of the file

Response body for GET or HEAD

File Page

Example response body

```
{
   "currentPageUrl": "/services/data/v25.0/chatter/users/005D0000001Kl6xIAC/files",
   "files": [
      {
         "contentSize": 7193018,
         "contentUrl": null,
         "description": null,
         "fileType": "PDF",
         "flashRenditionStatus": "SUCCESS",
         "id": "069D00000000yO6IAI",
         "modifiedDate": "2011-11-18T22:14:23.000Z",
         "origin": "C",
         "owner": {
            "companyName": "Seattle Apps",
            "firstName": "Marion",
            "id": "005D0000001Kl6xIAC",
            "isChatterGuest": false,
            "lastName": "Raven",
            "mySubscription": null,
            "name": "Marion Raven",
            "photo": {
              "largePhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/F",

               "photoVersionId": "729D00000000BDV",
               "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/T"

            },
            "title": null,
            "type": "User",
            "url": "/services/data/v25.0/chatter/users/005D0000001Kl6xIAC"
         },
         "pdfRenditionStatus": "NA",
         "thumb120By90RenditionStatus": "SUCCESS",
         "thumb240By180RenditionStatus": "SUCCESS",
         "thumb720By480RenditionStatus": "SUCCESS",
         "title": "Web Services API",
         "type": "ContentDocument",
         "url": "/services/data/v25.0/chatter/files/069D00000000yO6IAI",
         "versionNumber": "1"
      },
      {
         "contentSize": 371033,
         "contentUrl": null,
         "description": "Sharing feed items policy.",
         "fileType": "PDF",
         "flashRenditionStatus": "SUCCESS",
         "id": "069D00000000xNXIAY",
         "modifiedDate": "2011-11-10T19:53:04.000Z",
         "origin": "H",
         "owner": {
            "companyName": "Seattle Apps",
            "firstName": "Marion",
            "id": "005D0000001Kl6xIAC",
            "isChatterGuest": false,
            "lastName": "Raven",
            "mySubscription": null,
            "name": "Marion Raven",
            "photo": {
              "largePhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/F",
               "photoVersionId": "729D00000000BDV",
               "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000BDV/T"

            },
            "title": null,
            "type": "User",
            "url": "/services/data/v25.0/chatter/users/005D0000001Kl6xIAC"
         },
         "pdfRenditionStatus": "NA",
         "thumb120By90RenditionStatus": "SUCCESS",
         "thumb240By180RenditionStatus": "SUCCESS",
         "thumb720By480RenditionStatus": "SUCCESS",
         "title": "ReviewForJohn",
         "type": "ContentDocument",
         "url": "/services/data/v25.0/chatter/files/069D00000000xNXIAY",
         "versionNumber": "2"
      }
   ],
   "nextPageUrl": null,
   "previousPageUrl": null
}
```

User Files, Filtered by Group

Returns information about files posted to groups that the specified user is a member of. Available in versions 24.0 and later.

Resource

```
/chatter/users/me/files/filter/groups
```

or

```
/chatter/users/userId/files/filter/groups
```

HTTP methods

GET or HEAD

Request parameters

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 between 1 and 100. If you don't specify a size, the default is 25.

Response body

File Page

User Files, Filtered by Sharing

Returns information about files that have been shared with the specified user. Available in versions 24.0 and later.

Resource

```
/chatter/users/me/files/filter/shared-with-me
```

OR

```
/chatter/users/userId/files/filter/shared-with-me
```

Note: The resource can also be accessed with final keyword as one word:

```
/chatter/users/me | userid/files/filter/sharedwithme
```

HTTP methods

GET or HEAD

Request parameters

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 between 1 and 100. If you don't specify a size, the default is 25.

Response body

File Page

User Followers

Returns information about the followers for the specified user.

Resource

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

or

```
/chatter/users/userId/followers
```

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description

`page`	Token	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/v25.0/`
		`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 between 1 and 1000. If you don't specify a size, the default is 25.

Response body

Follower Page

Following Users

Returns a list of people, groups, records, and files that the specified user is following. Also used to follow records. See Using POST, PATCH, or PUT for Input on page 44.

Note: You can't use the users resource to add a user to a group. Use the groups resource instead.

Resource

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

or

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

HTTP methods

GET, POST, or HEAD

Request body for POST

In XML form, the root XML tag is expected to be `<following>`.

Name	Type	Description
`subjectId`	String	The ID of what to follow, for example, a user ID, a record ID, a file ID, and so on. **Note:** Use Group Member Input for adding members to groups.

Request parameter for POST

Parameter Name	Type	Description
`subjectId`	String	This request enables the user to follow the specified record. This could be a user ID, a file ID, and so on.

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 feeds to see the available key prefixes.
`page`	Token	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`.
`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

User Groups

Returns the groups the specified user is a member of.

Resource

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

OR

```
/chatter/users/userId/groups
```

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
page	Token	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`.
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.

Response body

Group Page

User Messages, General

Returns all the messages for all the private conversations for the logged-in user. Also used to search across all messages and post a message. See Using POST, PATCH, or PUT for Input on page 44. See also User Conversations, General on page 134.

Resource

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

or

```
/chatter/users/userId/messages
```

Note: `userId` must be the same as the logged-in user.

HTTP methods

GET, POST, or HEAD

Request body for POST

In XML form, the root XML tag is expected to be `<message>`.

Name	Type	Description
`body`	String	Text of the message body
`inReplyTo`	String	ID of an existing message that identifies which conversation this message is part of
`recipients`	User Message Recipient List Input	List of users who are the intended message recipients, up to 9. Comma-separated list of user IDs.

Note: Either `recipients` or `inReplyTo` is required. You must specify one or the other, not both.

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
`inReplyTo`	ID *`messageID`*	Either `recipients` or `inReplyTo` is required	ID of an existing message that identifies which conversation this message is part of
`recipients`	ID *`userID`*	Either `recipients` or `inReplyTo` is required	List of users who are the intended message recipients, up to 9. Comma-separated list of user IDs.
`text`	String	Required	The text of the message, up to 10,000 characters

Request parameters for GET or HEAD

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 between 1 and 100. If you don't specify a size, the default is 25.

Request parameter for GET

Parameter Name	Type	Description
`q` (optional)	String	Specify the string to search. The search string must contain at least two characters, not including any wildcards. For more information about wildcards, see Wildcards on page 43. Available in versions 24.0 and later. **Note:** The query parameter only searches message text bodies, not user names, titles, and so on.

Response body for GET or HEAD

Message Collection

Response body for POST

Message

Example response body for GET or HEAD

```
{
   "messages": [
      {
         "id": "03JD000000000GJMAY",
            "sender": {
               "name": "Uriah Jones",
               "title": "Head SE South",
               "firstName": "Uriah",
               "lastName": "Jones",
               "companyName": "Seattle Apps",
               "mySubscription": null,
               "photo": {
                "largePhotoUrl": "https://instance_name/profilephoto/729D00000000AQ4/F",

                  "photoVersionId": "729D00000000AQ4",
                "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000AQ4/T",

               "isChatterGuest": "false",
               "id": "005D0000001GLowIAB",
               "url": "/services/data/v25.0/chatter/users/005D0000001GLowIAB",
               "type": "User",
            "url": "/services/data/v25.0/chatter/users/me/messages/03JD000000000GJMAY"

            "body": {
               "text": "Bob and Mary, I'll schedule the weekly meeting.",
               "messageSegments": [
               ],
            "recipients": [
               {
                  "name": "Bob Johnson",
                  "id": "005D0000001GOYpIAO",
                  "url": "/services/data/v25.0/chatter/users/005D0000001GOYpIAO"
                  "title": null,
                  "firstName": "Bob",
                  "lastName": "Johnson",
                  "companyName": "Seattle Apps",
                  "photo": {
                     "largePhotoUrl":
"https://instance_name/profilephoto/729D00000000XA5/F",
                     "smallPhotoUrl":
"https://instance_name/profilephoto/729D00000000XA5/T"
               }
               {
                  "name": "Mary Smith",
                  "title": null,
                  "firstName": "Mary",
                  "lastName": "Smith",
                  "companyName": "Seattle Apps",
                  "mySubscription": null,
                  "photo": {
                     "largePhotoUrl":
"https://instance_name/profilephoto/729D00000000KJ6/F",
                     "photoVersionId": "729D00000000KJ6",
                     "smallPhotoUrl":
"https://instance_name/profilephoto/729D00000000KJ6/T"
               }
                  "isChatterGuest": "false",
                  "id": "005D0000001GbdHIAS",
                  "url": "/services/data/v25.0/chatter/users/005D0000001GbdHIAS"
                  "type": "User",
            ],
            "conversationId": "03MD0000000008KMAQ",
            "conversationUrl":
"/services/data/v25.0/chatter/users/me/conversations/03MD0000000008KMAQ",
            "sentDate": "2011-03-29T20:54:48.000Z",
      }
      {
         "id": "03JD000000000FVMAY",
            "sender": {
               "name": "Mary Smith",
               "id": "005D0000001GbdHIAS",
               "url": "/services/data/v25.0/005D0000001GbdHIAS"
               "title": null,
               "firstName": "Mary",
               "lastName": "Smith",
               "companyName": "Seattle Apps",
               "photo": {
                "largePhotoUrl": "https://instance_name/profilephoto/729D00000000KJ6/F",

                "smallPhotoUrl": "https://instance_name/profilephoto/729D00000000KJ6/T"

            "url": "/services/data/v25.0/chatter/users/me/messages/03JD000000000FVMAY"

            "body": {
               "text": "Please review the quarterly numbers by Friday.",
               "messageSegments": [
               ],
            "recipients": [
               {
                  "name": "Uriah Jones",
                  "id": "005D0000001GLowIAB",
                  "url": "/services/data/v25.0/chatter/users/005D0000001GLowIAB"
                  "title": "Head SE South",
                  "firstName": "Uriah",
                  "lastName": "Jones",
                  "companyName": "Seattle Apps",
                  "photo": {
                     "largePhotoUrl":
"https://instance_name/profilephoto/729D000000008vV/F",
                     "smallPhotoUrl":
"https://instance_name/profilephoto/729D000000008vV/T"
               }
            ],
            "createdDate": "2011-03-29T20:16:31.000Z",
            "read": "true",
            "conversationId": "03MD00000000080MAA",
            "conversationUrl":
"/services/data/v25.0/chatter/users/me/conversations/03MD00000000080MAA",
      }
   ]
   "currentPageUrl: "/services/data/v25.0/chatter/users/me/messages",
   "nextPageUrl": null,
}
```

User Messages, Specific

Returns the specified message.

Resource

```
/chatter/users/me/messages/messageId
```

or

```
/chatter/users/userId/messages/messageId
```

Note: `userId` must be the same as the logged-in user.

HTTP methods

GET or HEAD

Response body

Message

User Photos

Returns a URL to the photo for the logged-in user. Also used to post photos.

If you're building an app that needs to display user profile photos in a feed, you should cache the user photos. Then use the `photoVersionId` to determine when you need to update a photo. This helps you to avoid running over limits and may improve mobile client connections.

Resource

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

OR

```
/chatter/users/userId/photo
```

HTTP methods

GET, POST or HEAD

Request body for POST

In XML form, the root XML tag is expected to be `<photo>`. Available in versions 25.0 and higher.

Name	Type	Description
`fileId`	String	18–character Id of an already existing file to be used as the photo
`version`	Integer	Version number of the already existing file

Request parameters for POST

You can specify either `fileUpload` or `fileId`. You'll receive an error if you specify both.

Parameter Name	Type	Description
`fileUpload`	String *`multipart/form-data for a file`*	Contents of file to be uploaded
`fileId`	String *`ID`*	ID of a file already uploaded. The file must have a content type of image, and be smaller than 2 MB.

Note: Photos are processed asynchronously and may not be visible right away.

Response body for GET or HEAD

Photo

Response body for POST

Photo

User Recommendations, General

Returns the user, group, file, and record recommendations for the logged-in user. Available in versions 24.0 and later.

Resource

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

OR

```
/chatter/users/userId/recommendations
```

Note: *`userId`* must be the same as the logged-in user.

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
`maxResults`	Integer	Maximum number of recommendation results; default is 10. Value must be greater than 0.

Response body

Zero or more Recommendation responses

Example response body for GET with no request parameters

```
{
   "object" : {
      "name" : "Uriah Jones",
      "title" : "Head SE South",
      "firstName" : "Uriah",
      "lastName" : "Jones",
      "companyName" : "Seattle Chatter",
      "mySubscription" : null,
      "photo" : {
         "largePhotoUrl" : "https://instance_name/profilephoto/729D000000008vV/F",
         "photoVersionId" : "729D000000008vV",
         "smallPhotoUrl" : "https://instance_name/profilephoto/729D000000008vV/T"
      },
      "isChatterGuest" : false,
      "id" : "005D0000001GLowIAB",
      "url" : "/services/data/v25.0/chatter/users/005D0000001GLowIAB"
      "type" : "User",
   },
   "action" : "follow",
   "explanation" : {
      "detailsUrl" :
"/services/data/v25.0/chatter/users/me/recommendations/follow/005D0000001GLowIAB/
explanation/UserViewingSameRecords",
      "type" : "UserViewingSameRecords",
      "summary" : "Shares interest in 2 records"
   },
   "actOnUrl" :
"/services/data/v25.0/chatter/users/005D0000001AzOAIA0/following?subjectId=
005D0000001GLowIAB"
}
{
   "object" : {
      "name" : "All Seattle Apps",
      "owner" : {
         "name" : "Marion Raven",
         "title" : null,
         "firstName" : "Marion",
         "lastName" : "Raven",
         "companyName" : "Seattle Apps",
         "mySubscription" : null,
         "photo" : {
            "largePhotoUrl" : "https://instance_name/profilephoto/729D00000000BDV/F",
            "photoVersionId" : "729D00000000BDV",
            "smallPhotoUrl" : "https://instance_name/profilephoto/729D00000000BDV/T"
         },
         "isChatterGuest" : false,
         "id" : "005D0000001Kl6xIAC",
         "url" : "/services/data/v25.0/chatter/users/005D0000001Kl6xIAC"
         "type" : "User",
      },
      "description" : "Get company announcements and other important updates",
      "visibility" : "PublicAccess",
      "memberCount" : 15,
      "fileCount" : 0,
      "canHaveChatterGuests" : false,
      "mySubscription" : null,
      "myRole" : "NotAMember",
      "photo" : {
         "largePhotoUrl" : "https://instance_name/profilephoto/729D00000000BDQ/F",
         "photoVersionId" : "729D00000000BDQ",
         "smallPhotoUrl" : "https://instance_name/profilephoto/729D00000000BDQ/T"
      },
      "lastFeedItemPostDate" : "2011-05-06T22:42:34.000Z",
      "id" : "0F9D00000000FG0KAM",
      "url" : "/services/data/v25.0/chatter/groups/0F9D00000000FG0KAM"
      "type" : "CollaborationGroup",
   },
   "action" : "join",
   "explanation" : {
      "detailsUrl" : null,
      "type" : "GroupNew",
      "summary" : "Created in the last month"
   },
   "actOnUrl" : "/services/data/v25.0/chatter/groups/0F9D00000000FG0KAM/members?userId=
005D0000001AzOAIA0"
}
```

User Recommendations for a Specific Action

Returns the recommendations of the specified action for the logged-in user. Available in versions 24.0 and later.

Resource

```
/chatter/users/me/recommendations/action
```

or

```
/chatter/users/userId/recommendations/action
```

Note: *`userId`* must be the same as the logged-in user.

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. Available in versions 25.0 and higher.

HTTP methods

GET or HEAD

Request parameters

Parameter Name	Type	Description
`maxResults`	Integer	Maximum number of recommendation results; default is 10. Value must be greater than 0.

Response body

Zero or more Recommendation responses

User Recommendations for a Specific Action and Object Category

Returns the recommendations of the specified action and object category for the logged-in user. Also used to delete recommendations.

Resource

```
/chatter/users/me/recommendations/action/objectCategory
```

or

```
/chatter/users/userId/recommendations/action/objectCategory
```

Note: *`userId`* must be the same as the logged-in user.

- Valid values for *`action`*:
  - `follow`
  - `join`
  - `view` (Available in versions 25.0 and higher.)
- Valid values for *`object category`*:
  - If action is `follow`, *`object category`* is `users`, `files`, or `records`.
  - If action is `join`, *`object category`* is `groups`.
  - If action is `view`, *`object category`* is `users`, `files`, `groups`, or `records`.

HTTP methods

GET, DELETE, or HEAD

DELETE is available as of version 24.0. DELETE is used to remove a recommendation and is only valid for the following resources:

- `/chatter/users/me | `***`userId`***`/recommendations/follow/`***`recommendedUserId`***
- `/chatter/users/me | `***`userId`***`/recommendations/follow/`***`recommendedFileId`***
- `/chatter/users/me | `***`userId`***`/recommendations/follow/`***`recommendedRecordId`***
- `/chatter/users/me | `***`userId`***`/recommendations/join/`***`recommendedGroupId`***

Note: *`userId`* must be the same as the logged-in user.

Request parameters for GET or HEAD

Parameter Name	Type	Description
`maxResults`	Integer	Maximum number of recommendation results; default is 10. Value must be greater than 0.

Parameter Name	Type	Description
`followed`	String	ID of a user. Provides new recommendations based on the context of this user ID. Available in versions 24.0 and later.
`viewed`	String	ID of a file. Provides new recommendations based on the context of this file ID. Available in versions 25.0 and higher.

Example request URL with `followed` request parameter

```
/chatter/users/me/recommendations/follow/users?followed=005D0000001GLowIAB
```

Response body for GET with `followed` request parameter

Zero or one user Recommendation response

Response body for GET or HEAD without request parameters

Zero or more Recommendation responses

Response body for DELETE

204: Successful Delete

User Recommendations for a Specific Action and Object ID

Returns the recommendation of the specified action and object ID for the logged-in user. Also used to delete a recommendation. Available in versions 24.0 and later.

Resource

```
/chatter/users/me/recommendations/action/objectID
```

or

```
/chatter/users/userId/recommendations/action/objectID
```

Note: *`userId`* must be the same as the logged-in user.

- Valid values for *`action`*:
  - `follow`
  - `join`
  - `view` (Available in versions 25.0 and higher.)
- Valid values for *`object ID`*:
  - If action is `follow`, *`object ID`* is a user ID, file ID, or record ID.

- ◊ If action is `join`, *`object ID`* is a group ID.
- ◊ If action is `view`, *`object ID`* is a user ID, file ID, group ID, or record ID.

HTTP methods

GET, DELETE, or HEAD

DELETE is used to remove a recommendation and is only valid for the following resources:

- `/chatter/users/me` | ***`userId`***`/recommendations/follow/`***`recommendedUserId`***
- `/chatter/users/me` | ***`userId`***`/recommendations/follow/`***`recommendedFileId`***
- `/chatter/users/me` | ***`userId`***`/recommendations/follow/`***`recommendedRecordId`***
- `/chatter/users/me` | ***`userId`***`/recommendations/join/`***`recommendedGroupId`***

Note: *`userId`* must be the same as the logged-in user.

Request parameters for GET or HEAD

Parameter Name	Type	Description
`maxResults`	Integer	Maximum number of recommendation results; default is 10. Value must be greater than 0.
`followed`	String	ID of a user. Provides new recommendations based on the context of this user ID. Available in versions 24.0 and later.

Response body for GET or HEAD without request parameters

Zero or one Recommendation response

Response body for DELETE

204: Successful Delete

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 on page 44.

Resource

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

or

```
/chatter/users/userId/status
```

HTTP methods

GET, DELETE, POST or HEAD

Request body for POST

In XML form, the root XML tag is expected to be `<userStatus>`.

Name	Type	Description
`body`	Message Body Input	Description of message body

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

Example response body

```
<?xml version="1.0" encoding="UTF-8"?>
<currentStatus>
  <body>
    <messageSegments>
      <segment>
        <text>Will send you latest updates tonight</text>
        <type>Text</type>
      </segment>
    </messageSegments>
    <text>Will send you latest updates tonight</text>
  </body>
  <parentId>005D0000001GLbn</parentId>
  <url>/services/data/v25.0/chatter/users/005D0000001GLbn/status</url>
</currentStatus>
```

Chapter 5

JSON and XML Request Bodies

Request bodies should be formatted in either XML or JSON, and included in POST, PATCH, or PUT requests.

The following are the top-level request bodies, and can contain one or more of the lower-level request bodies:

- Comment
- Dashboard Component Snapshot
- Feed Favorite
- Feed Item
- File
- Following
- Photo
- Group Member
- User Message
- User Status
- User Stream (Conversation)

Top-level request bodies can contain one or more lower-level request bodies.

Attachment Input: Existing Content

Name	Type	Description
`contentDocumentId`	String	ID of the existing content

Attachment Input: Link

Name	Type	Description
`url`	String	URL included in the attachment
`urlName`	String	Optional. Name of the URL. If not provided, a name is generated from the domain name of the URL

Attachment Input: New File Upload

The HTTP request should also carry a file upload part holding the file itself.

Note: If you're uploading a new file, you must include a binary file in the multipart request.

Name	Type	Description
`desc`	String	Description of the file
`fileName`	String	Name of the file
`title`	String	Title of the file

Comment Input

In XML form, the root XML tag is expected to be `<comment>`.

Name	Type	Description
`attachment`	Depends on type.	Optional Values are: • Attachment Input: Existing Content • Attachment Input: Link • Attachment Input: New File Upload **Note:** If you're uploading a new file, you must include a binary file in the multipart request.
`body`	Message Body Input	Description of message body

Dashboard Component Snapshot Input

Available in versions 25.0 and higher.

In XML form, the root XML tag is expected to be `<dashboardComponentSnapshot>`.

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.
`feedItemBody`	Message Body Input	Contains the text that accompanies the feed item associated with 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.

Feed Favorite Input

Available in versions 24.0 and later. In XML form, the root XML tag is expected to be `<favorite>`.

The following input is valid for the HTTP method POST:

Name	Type	Description
searchText	String	Text of the search that comprises the favorite. This parameter can only be used with saved searches and not list views.

The following input is valid for the HTTP method PATCH:

Name	Type	Description
updateLastViewDate	Boolean	Sets the last view date of the specified feed favorite to the current system time. Default value is false.

Feed Item Input

In XML form, the root XML tag is expected to be <feedItem>.

Name	Type	Description
attachment	Depends on type	Optional. You can specify only one of these values in the attachment parameter. Values are: • Attachment Input: Existing Content • Attachment Input: Link • Attachment Input: New File Upload **Note:** If you're uploading a new file, you must include a binary file in the multipart request.
body	Message Body Input	Description of message body
isBookmarkedByCurrentUser	Boolean	Specifies if the new feed item should be bookmarked for the user (true) or not (false. Default value is false. This parameter is only available for PATCH methods.
originalFeedItemId	String	If this feed item is shared, this is the original feed item 18-character Id **Note:** This parameter takes precedence over all other parameters. For example, if you share a post and also try to upload a file, the generated feed-item only contains the shared post. You should only share a post or upload a file, not both in the same post.

File Input

Available in versions 24.0 and later. Describes a new file input. Assumes that a file blob is also be provided in a binary form part named `fileData`. In XML form, the root XML tag is expected to be `<fileInput>`.

Name	Type	Description
`description`	String	Description of the file to be uploaded
`title`	String	Title of the file

Following Input

In XML form, the root XML tag is expected to be `<following>`.

Name	Type	Description
`subjectId`	String	The ID of what to follow, for example, a user ID, a record ID, a file ID, and so on. **Note:** Use Group Member Input for adding members to groups.

Group Member Input

In XML form, the root XML tag is expected to be `<member>`.

Name	Type	Description
`userId`	String	The user ID of the member to be added to the group

Message Body Input

Name	Type	Description
`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

Message Segment Input: Hash Tag

Represents an inline hash tag. If a hash tag is provided as input without whitespace text before it, a space is inserted. If a hash tag is provided as input without whitespace text after it, a space is inserted.

Name	Type	Description
`tag`	String	Text of the hash tag without the hash tag
`type`	String	Value is `Hashtag`

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.

Name	Type	Description
`type`	String	Value is `Link`
`url`	URL	URL to be used for the link

Message Segment Input: Mention

Represents an inline mention of a user. You must provide a user ID. No spaces are added.

Name	Type	Description
`id`	String	ID of the user to be mentioned
`type`	String	Value is `Mention`

Message Segment Input: Text

Represents plain inline text. Text prefixed with a # and surrounded by space is converted into a hash tag segment. Text that looks like a link and is surrounded by space is converted into a link segment.

Name	Type	Description
`text`	String	Plain text for this segment
`type`	String	Only value is `Text`

Photo

In XML form, the root XML tag is expected to be `<photo>`. Available in versions 25.0 and higher.

Name	Type	Description
`fileId`	String	18–character Id of an already existing file to be used as the photo
`version`	Integer	Version number of the already existing file

User Message

In XML form, the root XML tag is expected to be `<message>`.

Name	Type	Description
body	String	Text of the message body
inReplyTo	String	ID of an existing message that identifies which conversation this message is part of
recipients	User Message Recipient List Input	List of users who are the intended message recipients, up to 9. Comma-separated list of user IDs.

Note: Either recipients or inReplyTo is required

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"
}
```

User Message Recipient Input

In XML form, the root XML tag is expected to be <recipient>.

Name	Type	Description
recipient	String	18-character Id of the user to receive the message

User Message Recipient List Input

In XML form, the root XML tag is expected to be <message>.

Name	Type	Description
recipients	User Message Recipient Input[]	Collection of recipients to receive the message

User Status Input

Note: Setting a user status is no longer available as of version 25.0. Just post feed items to the feed instead, using Feed Item Input.

In XML form, the root XML tag is expected to be `<userStatus>`.

Name	Type	Description
`body`	Message Body Input	Description of message body

User Stream (Conversation) Input

Available in versions 24.0 and later. In XML form, the root XML tag is expected to be `<conversation>`.

Name	Type	Description
`read`	Boolean	Changes the conversation status to unread (`false`) or read (`true`).

Chapter 6

Chatter API Resource Returns

The Chatter API resources return an HTTP response code. They can also return a response body, in either JSON or XML.

Response Bodies

The following are the response bodies returned by successful execution of a resource.

Address

Name	Type	Description
`city`	String	Name of the city
`country`	String	Name of the country
`formattedAddress`	String	Address formatted by locale. Available in versions 25.0 and higher.
`state`	String	Name of the state, province, or so on
`street`	String	Street number
`zip`	String	Zip or postal code

Batch Results

Name	Type	Description
`results`	Batch Result Item[]	Collection of batch result items

Batch Result Item

Name	Type	Description
`result`	Various	Dependent on the type of batch. Value could be one of the following: • Error Message • User Summary for `/chatter/users/batch/` ***`userIds`***

Name	Type	Description
		- Group for `/chatter/groups/batch/` ***`userIds`***
`statusCode`	Integer	An HTTP status code

Chatter Activity

Name	Type	Description
`commentCount`	Integer	Total number of comments in the organization made by the user
`commentReceivedCount`	Integer	Total number of comments in the organization received by the user
`likeReceivedCount`	Integer	Total number of likes on posts and comments in the organization received by the user
`postCount`	Integer	Total number of posts in the organization made by the user

Chatter Directory

Available in versions 24.0 and later.

Name	Type	Description
`feeds`	String	Returns a URL to the general feeds resource for the organization
`groups`	String	Returns a URL to the general groups resource for the organization
`organization`	String	Returns a URL to the general connect organization resource
`users`	String	Returns a URL to the general users resource for the organization

Chatter Influence

Available in versions 24.0 and later.

Name	Type	Description
`percentile`	String	Percentile value for the user's influence rank within the organization

Name	Type	Description
`rank`	Integer	Number indicating the user's influence rank, relative to all other users within the organization

ClientInfo

Name	Type	Description
`applicationName`	String	Name of the remote access application used for authentication. For more information, see "Remote Access Application Overview" in the online help.
`applicationUrl`	String	Value from the `Info URL` field of the remote access application used for authentication

Comment

Name	Type	Description
`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. Available in versions 24.0 and later.
`body`	Feed Body	Message body comment
`clientInfo`	ClientInfo	Information about the remote access application used to authenticate the connection
`createdDate`	String	ISO8601 date string, for example, 2011-02-25T18:24:31.000Z
`feedItem`	Reference	Information about the feed item for the comment
`id`	String	18-character `Id` of the comment
`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
`isDeleteRestricted`	Boolean	Returns `true` if a comment cannot be deleted by the logged in user. If it returns `false`, possibly the logged in users can

Name	Type	Description
		delete the comment. Available in versions 25.0 and higher.
`myLike`	Reference	If the logged-in user has liked the comment, returns a reference to that like or returns `null`
`parent`	Parent	Information about the parent feed-item for this comment
`type`	String	Indicates the type of comment. Available in versions 24.0 and later. Valid values are: • `ContentComment`—the comment contains an attachment • `TextComment`—the comment contains only text
`url`	String	URL to this comment
`user`	User Summary	Information about the comment author

Comment Page

Name	Type	Description
`comments`	Comment[]	Collection of comments
`currentPageUrl`	String	URL to the first page of comments, that is, the most recent comments. The default is 25 comments per page.
`nextPageUrl`	String	URL to the next page of comments, that is, older comments. The default is 25 comments per page.
`total`	Integer	Total number of comments across all pages for the parent feed item

Conversation Detail

Name	Type	Description
`id`	String	18-character `Id` of the conversation
`members`	User Summary[]	Collection of users participating in this conversation
`messages`	Message Collection	Messages in this conversation
`read`	Boolean	Does the context user have unread messages in this conversation?
`url`	String	URL to the conversation detail page

Conversation Summary

Name	Type	Description
id	String	18-character Id of the conversation
latestMessage	Message	The most recent message in this conversation
members	User Summary[]	Collection of users participating in the conversation
read	Boolean	Does the context user have unread messages in this conversation?
url	String	URL to the conversation detail page

Conversation Summary Collection

Name	Type	Description
conversations	Conversation Summary[]	Collection of one or more conversations
currentPageUrl	String	URL to the first page of conversations, that is, the conversations with the most recent messages
nextPageUrl	String	URL to the next page of messages, that is, older conversations

Conversation Unread Count

Name	Type	Description
hasMore	Boolean	true if the actual number of unread conversations is greater than 50, false otherwise
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.

Error Message

Name	Type	Description
errorCode	String	HTTP error code, such as 501, 404, and so on
message	String	Description of error

Explanation Details

Available in versions 24.0 and later.

Name	Type	Description
data	Various	More information about the recommendation. What is returned depends on the type. • 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. • 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	Explanation of recommendation
type	String	Type of recommendation explanation. Values are: • UserFollowsSameUsers • GroupMembersFollowed • UserViewingSameRecords • RecordParentOfFollowed

Explanation Summary

Name	Type	Description
detailsUrl	String	URL for explanation details. If the recommendation doesn't have a detailed explanation, the string "null". Available in versions 24.0 and later.
summary	String	Explanation of recommendation
type	String	Type of recommendation explanation, such as UserFollowsSameUsers. Available in versions 24.0 and later.

Favorite

Available in versions 24.0 and later.

Name	Type	Description
createdBy	User Summary	Information about the original creator of the favorite
feedUrl	String	URL to the feed item for this favorite
id	String	18-character Id of the favorite
lastViewDate	String	ISO8601 date string, for example, 2011-02-25T18:24:31.000Z
name	String	Name of the favorite
searchText	String	If the favorite is from a search, contains the search text, if not, an empty string
type	String	An empty string or one of the following values: • listView • search
url	String	URL to this favorite
user	User Summary	Information about the user who saved this favorite

Favorites Page

Available in versions 24.0 and later.

Name	Type	Description
favorites	Favorite[]	Collection of one or more favorites
total	Integer	Total number of favorites

Features

Available in versions 24.0 and later.

Name	Type	Description
chatterActivity	Boolean	Specifies whether the user details include information about Chatter activity
chatterGlobalActivity	Boolean	Specifies whether the user details include global Chatter activity
chatterMessages	Boolean	Specifies whether Chatter messages are enabled for the organization
feedPolling	Boolean	Specifies whether the is-modified resource is enabled for the Chatter API
files	Boolean	Specifies whether files can act as resources for Chatter API

Name	Type	Description
`filesOnComments`	Boolean	Specifies whether files can be attached to comments
`trendingTopics`	Boolean	Specifies whether trending topics are enabled
`viralInvitesAllowed`	Boolean	Specifies whether existing Chatter users can invite people in their company to use Chatter

Feed

Name	Type	Description
`feedItemsUrl`	String	URL of feed items
`isFeedModifiedUrl`	String	URL of the isFeedModified resource with the polling token prepended. Returns `null` if feed is not a news feed.

Feed Body

Name	Type	Description
`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 • Message Segment: Field Change Value • Message Segment: Hashtag • Message Segment: Link • Message Segment: Mention • Message Segment: More Changes • Message Segment: Text
`text`	String	Unformatted text of the message. This can be an easy summary to use if you don’t want to process the message segments.

Feed Directory

Name	Type	Description
`favorites`	Favorite[]	Collection of favorites that the logged-in user has saved. Available in versions 24.0 and later.
`feeds`	Feed Directory Item[]	Collection of feed directory items that the logged-in user is allowed to view

Feed Directory Item

Name	Type	Description
`feedItemsUrl`	String	URL to the resource of the feed items for a specific feed. For example: `/services/data/v25.0.0/ chatter/feeds/files/me/feed-items`
`feedUrl`	String	URL to the resource of a specific feed. For example: `/services/data/v25.0.0/ chatter/feeds/files/me`
`label`	String	Name of the feed, such as, Groups, People, Company, and so on.

Feed Item

Name	Type	Description
`actor`	Depends on parent	The entity that created the feed item. Values are: • Record Summary—if a record • User Summary—if a user
`attachment`	Depends on type	Depends on the value of the attachment. Values include: • `ContentPost`—Feed Item Attachment: Content or `null` if the content is unavailable • `LinkPost`—Feed Item Attachment: Link • `TrackedChange`—Feed Item Attachment: Content if the change is on a document, or `null` otherwise All other types return `null`.

Name	Type	Description
body	Feed Body	Information about the feed item
clientInfo	ClientInfo	Information about the remote access application used to authenticate the connection
comments	Comment Page	Comments for this feed item
createdDate	String	ISO8601 date string, for example, 2011-02-25T18:24:31.000Z
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, nullotherwise.
event	Boolean	true if feed item created due to an event change, false otherwise
id	String	18-character Id of the feed item
isBookmarkedByCurrentUser	Boolean	true if the current user has bookmarked this feed item, false otherwise. Available in versions 24.0 and later.
isDeleteRestricted	Boolean	Returns true if a feed item cannot be deleted by the logged in user. If it returns false, possibly the logged in users can delete the feed item. Available in versions 25.0 and higher.
isLikedByCurrentUser	Boolean	true if the current user has liked this feed item, false otherwise
likes	Like Page	Likes for this feed item
myLike	Reference	If the logged-in user has liked the comment, this property is a reference to the specific like, null otherwise
modifiedDate	String	ISO8601 date string, for example, 2011-02-25T18:24:31.000Z
originalFeedItem	Reference	A reference to the original feed item if this feed item is a shared feed item, null otherwise. Available in versions 24.0 and later.
originalFeedItemActor	Depends on type	Value depends on type of original feed item. Available in versions 24.0 and later. Values are: • Record Summary–if original feed item is a record

Name	Type	Description
		- User Summary–if original feed item is a user - `null` if this feed item is not a shared feed item
`parent`	Depends on type	Value depends on type of parent. Values are: - File–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
`photoUrl`	String	URL of the photo associated with the feed item
`type`	String	Values are: - `ActivityType` - `ApprovalPost` - `AttachArticleEvent`—Available in versions 25.0 and higher. - `ContentPost` - `EmailMessageEvent` - `LinkPost` - `TextPost` - `TrackedChange` - `UserStatus`
`url`	String	URL to this feed item

Feed Item Attachment: Content

Name	Type	Description
`description`	String	Description of the attachment
`downloadUrl`	String	URL to the content
`fileSize`	String	Size of the file in bytes. If size cannot be determined, returns `unknown`.
`fileType`	String	Type of file
`hasImagePreview`	Boolean	`true` if the file has a preview image available, `false` otherwise
`hasPdfPreview`	Boolean	`true` if the file has a PDF preview available, `false` otherwise
`id`	String	18-character `Id` of the content
`mimeType`	String	MIME type content of the file.

Name	Type	Description
`title`	String	Title of the file
`versionId`	String	Version number of the file

Feed Item Attachment: Link

Name	Type	Description
`title`	String	Description of the link
`url`	String	URL to the attachment

Feed Item Page

Name	Type	Description
`currentPageUrl`	String	URL to the first page of feed items
`isFeedModifiedUrl`	String	URL of the isFeedModified resource with the polling token prepended. Returns `null` if feed is not a news feed.
`items`	Feed Item[]	Collection of one or more feed items
`nextPageUrl`	String	URL to the next page of feed items
`previousPageUrl`	String	URL to the previous page of feed items

Feed Modified Info

Name	Type	Description
`isModified`	Boolean	`true` if the feed is a news feed and has been modified since the last time it was polled, `false` if the news feed hasn't been modified. Returns `null` if feed is not a news feed.
`nextPollUrl`	String	URL to the next set of feed results

File

Available in versions 24.0 and later.

Name	Type	Description
`contentSize`	Integer	Size of the file in bytes
`contentUrl`	String	If the file is a link, returns the URL, otherwise the string "null"
`description`	String	Description of the file

Name	Type	Description
fileType	String	Type of file, such as PDF, PowerPoint, and so on
flashRenditionStatus	String	Specifies if a flash preview version of the file has been rendered
id	String	18-character Id of the file
modifiedDate	String	ISO8601 date string, for example, 2011-02-25T18:24:31.000Z
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
owner	User Summary	Owner of the file
name	String	Name of the file
pdfRenditionStatus	String	Specifies if a PDF preview version of the file has been rendered
thumb120By90RenditionStatus	String	Specifies if a 120x90–pixel–sized preview image of the file has been rendered
thumb240By180RenditionStatus	String	Specifies if a 240x180–pixel–sized preview image of the file has been rendered
thumb720By480RenditionStatus	String	Specifies if a 720x480–pixel–sized preview image of the file has been rendered
title	String	Title of the file
type	String	ContentDocument
url	String	URL for the file
versionNumber	String	Version number of the file

File Details

Available in versions 24.0 and later.Includes all of the File response body with the additional item:

Name	Type	Description
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.

File Page

Available in versions 24.0 and later.

Name	Type	Description
`currentPageUrl`	String	URL for the first page of files
`files`	File[]	Collection of one or more files
`nextPageUrl`	String	URL for the next page of files
`previousPageUrl`	String	URL for the previous page of files

File Shares Page

Available in versions 24.0 and later.

Name	Type	Description
`currentPageUrl`	String	URL for the first page of file shares
`shares`	Shares[]	Collection of one or more file shares
`nextPageUrl`	String	URL for the next page of file shares
`previousPageUrl`	String	URL for the previous page of file shares

Flat Feed

Available in versions 25.0 and higher.

Name	Type	Description
`currentPageUrl`	String	URL for the first page of items
`items`	Flat Feed Item[]	Collection of one or more flat feed items
`nextPageUrl`	String	URL for the next page of items

Flat Feed Item

Available in versions 25.0 and higher.

Name	Type	Description
`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
`type`	String	The type of item. Valid values are: • `Comment` • `FeedItem`

Follower Page

Name	Type	Description
`currentPageUrl`	String	URL to the current page
`followers`	Subscription[]	Collection of one or more followers or subscriptions
`mySubscription`	Reference	**Note:** This field is no longer available as of version 24.0. If the logged-in user is following this user, this contains information about the subscription, else returns `null`
`nextPageUrl`	String	URL to the next page
`previousPageUrl`	String	URL to the previous page
`total`	Integer	Total number of followers

Following Counts

Name	Type	Description
`people`	Integer	Number of people user is following
`records`	Integer	Number of records user is following
`total`	Integer	Total number of items user is following

Following Page

Name	Type	Description
`currentPageUrl`	String	URL to the current page
`following`	Subscription[]	Collection of one or more subscriptions
`nextPageUrl`	String	URL of the next page
`previousPageUrl`	String	URL of the previous page
`total`	Integer	Total number of records being followed

Group

Name	Type	Description
`canHaveChatterGuests`	Boolean	`true` if this group allows Chatter customers, `false` otherwise
`description`	String	Text-only description of the group

Name	Type	Description
fileCount	Integer	Number of files posted to this group. Available in versions 24.0 and later.
id	String	18-character Id of the group
lastFeedItemPostDate	Date	Date of the most recent feed item posted to the group. Available in versions 24.0 and later.
memberCount	Integer	Total number of group members
myRole	String	Values are: • GroupOwner • GroupManager • NotAMember • NotAMemberPrivateRequested • StandardMember
mySubscription	Reference	If the logged-in user is a member of this group, this contains information about that subscription, or returns null otherwise.
name	String	Name of the group
owner	String	Name of the owner of the group. Available in versions 24.0 and later.
photo	Photo	Information about the group photo
type	String	CollaborationGroup.
url	String	Relative URL string to the group
visibility	String	Values are: • PrivateAccess • PublicAccess

Group Member

Name	Type	Description
id	String	18-character Id of the user
role	String	Values are: • GroupOwner • GroupManager • NotAMember • NotAMemberPrivateRequested • StandardMember
url	String	URL to this membership

Name	Type	Description
user	User Summary	Information about the user who is subscribed to this group

Group Member Page

Name	Type	Description
currentPageUrl	String	URL to this page of members
members	Group Member[]	Collection of group members
myMembership	Reference	If the logged-in user is a member of this group, returns information about that membership, or returns null otherwise
nextPageUrl	String	URL to the next page of members
previousPageUrl	String	URL to the previous page of members
totalMemberCount	Integer	Total number of group members

Group Page

Name	Type	Description
currentPageUrl	String	URL to this page of groups
groups	Group[]	Collection of groups
nextPageUrl	String	URL to the next page of groups
previousPageUrl	String	URL to the previous page of groups
total	Integer	Total number of groups

Influence Thresholds

Available in versions 24.0 and later.

Name	Type	Description
name	String	Description of this influencer category. For example: Active Influencer
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.

Library

Available in versions 24.0 and later.

Name	Type	Description
id	String	18-character Id of the library
name	String	Name of the library
type	String	Type of library, for example, ContentWorkspace
url	String	URL to this library

Like

Name	Type	Description
id	String	18-character Id of the like
url	String	URL to this like
user	User Summary	Information about the user who liked this item

Like Page

Name	Type	Description
currentPageUrl	String	URL to the current page of likes
likes	Like[]	Collection of likes
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
nextPageUrl	String	URL to the next page of likes
previousPageUrl	String	URL to the previous page of likes
total	Integer	Total number of likes for the item

Message

Name	Type	Description
body	Feed Body	Text of this message
conversationId	String	18-character Id of the conversation this message is part of

Name	Type	Description
`conversationUrl`	String	URL to the conversation this message is part of
`id`	String	18-character `Id` of the message
`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`***
`sendDate`	String	ISO8601 date string, for example, 2011-02-25T18:24:31.000Z
`sender`	User Summary	Information about the user who sent this message
`url`	String	URL to this message

Message Collection

Name	Type	Description
`currentPageUrl`	String	URL to the first page of messages, that is, the most recent messages
`messages`	Message[]	Collection of one or more messages
`nextPageUrl`	String	URL to the next page of messages, that is, older messages

Message Segment: Entity Link

Represents an inline link to a URL. The URL is an object, such as a record or a group. Available in versions 25.0 and higher.

Name	Type	Description
`reference`	Reference	Information about the link
`text`	String	Plain text output for this segment

Name	Type	Description
`type`	String	`EntityLink`

Message Segment: Field Change

Represents an inline field change, for example, `"First Name from Bob to Joe"`. This segment can consist of multiple segments, for example:

```
[
   Text("Owner from "),
   FieldChangeValue("Bob Smith"),
   Text(" to "),
   FieldChangeValue("Dave Jones")
]
```

Name	Type	Description
`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
`text`	String	Formatted text of the change. For example: `"City from San Francisco to New Orleans"`
`type`	String	Value is `FieldChange`

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.

Name	Type	Description
`text`	String	Formatted text of the change. Possible values include: • the old name of the field

Name	Type	Description
		• the new name of the field
`type`	String	Value is `FieldChangeName`

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.

Name	Type	Description
`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
`type`	String	Value is `FieldChangeValue`
`url`	String	URL value if the field change is to a URL field (such as a web address)

Message Segment: Hashtag

Represents an inline hashtag.

Name	Type	Description
`tag`	String	Text of the hashtag without the hash symbol (#)
`text`	String	Text of the hashtag with the hash symbol (#)
`type`	String	Value is `Hashtag`
`url`	String	URL search for the tag in this format: `/feed-items?q=`***`tag`***

Message Segment: Link

Represents an inline link to a URL

Name	Type	Description
`text`	String	Formatted text of the feed item of this segment
`type`	String	Value is `Link`
`url`	String	URL value of the link

Message Segment: Mention

Represents an inline @mention.

Name	Type	Description
`accessible`	Boolean	Specifies whether the mentioned user would be able to see mention when it was posted: `true` if the mentioned user could see the post, `false` otherwise
`name`	String	Name of the user mentioned
`text`	String	Formatted text of the feed item of this segment
`type`	String	Value is `Mention`
`user`	User Summary	Information about the mentioned user, `null` if the logged-in user doesn’t have access.

Message Segment: More Changes

In feed items with a large number of tracked changes, the message is formatted as: "changed A, B, and made 3 more changes." The more changes segment contains the "3 more changes."

Name	Type	Description
`moreChangesCount`	Integer	Number of other changes
`text`	String	Formatted text of the number of changes. For example: `"2 other changes"` **Note:** This value is localized.
`type`	String	Value is `MoreChanges`

Message Segment: Text

Represents plain inline text.

Name	Type	Description
`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`
`type`	String	Value is `Text`

Organization

Available in versions 24.0 and later.

Name	Type	Description
`accessTimeout`	Integer	Amount of time after which the system prompts users who have been inactive to log out or continue working
`features`	Features	Information about features available in the organization
`orgId`	String	18-character `Id` for the organization
`userSettings`	User Settings	Information about the organization permissions for the user

Parent

Name	Type	Description
`id`	String	18-character `Id` of the parent
`url`	String	URL of the parent

Phone Number

Name	Type	Description
`number`	String	Phone number
`type`	String	Values are: • `Fax` • `Mobile` • `Work`

Photo

Name	Type	Description
`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.
`photoVersionId`	String	18-character `Id` to that version of the photo. Available in versions 24.0 and later.
`smallPhotoUrl`	String	URL to the small profile picture. The default size is 64x64 pixels.

Recommendation

Name	Type	Description
`action`	String	The recommended action. Values are: • `follow` • `join` • `view` Available in versions 25.0 and higher.
`actOnUrl`	String	URL that can be used with an HTTP method POST request to follow a recommended user, file, or record, or join a recommended group. For example: `/users/me/following/?subjectId=001D000000Iyu2p`
`explanation`	Explanation Summary	The recommendation explanation
`object`	Various	The object with which the receiver is recommended to take action. Possible objects are: • File Available in versions 24.0 and later. • Group • Record Summary • User Summary
`type`	String	No longer available as of version 24.0

Record Summary

Name	Type	Description
id	String	18-character Id of the record
mySubscription	Reference	If the logged in user is following this record, this contains information about the subscription, otherwise returns null.
name	String	Formatted display name
type	String	Type of record, such as CustomObject, ContentDocument, and so on
url	String	Relative URL to this record

Reference

Name	Value	Description
id	String	18-character Id of the object being referenced
url	String	URL to the resource endpoint

Shares

Available in versions 24.0 and later.

Name	Type	Description
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 • Library
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 logged-in user is the same as owner of the file. • N—no access • V—viewer

Subscription

Name	Type	Description
id	String	18-character Id of the subscription
subject	Depends	Information about the parent, that is, the thing or person being followed. Valid values are: • Group • Record Summary • User Summary
subscriber	Depends	Information about the subscriber, that is, the person following this item. Valid values are: • Group • Record Summary • User Summary
url	String	Relative URL to this specific subscription

Subscription Page

Name	Type	Description
currentPageUrl	String	Returns the current page of subscriptions
nextPageUrl	String	Returns the next page of subscriptions. If there is no next page, returns the string “null”
mySubscription	Reference	Returns information about the subscription, or null the logged-in user is not subscribed.
previousPageUrl	String	Returns the previous page of subscriptions. If there is no previous page, returns the string “null”
subscriptions	Subscription[]	Collection of subscriptions
total	Integer	Total number of subscriptions

Topics

Available in versions 24.0 and later.

Name	Type	Description
name	String	Name of the trending topic

User Detail

Name	Type	Description
`aboutMe`	String	Text from user's profile
`address`	Address	Address
`chatterActivity`	Chatter Activity	Chatter activity statistics
`chatterInfluence`	Chatter Influence	User's influence rank
`companyName`	String	Company name
`currentStatus`	User Status	**Note:** This property is no longer available as of version 25.0. User's current status
`email`	String	User's email address
`firstName`	String	User's first name
`followersCount`	Integer	Number of users following this user
`followingCounts`	Following Counts	Information about items the user is following
`groupCount`	Integer	Number of groups user is following
`id`	String	18-character `Id` of the user
`isActive`	Boolean	`true` if user is active, `false` otherwise
`isChatterGuest`	Boolean	`true` if user is a Chatter customer, `false` otherwise
`lastName`	String	User's last name
`managerId`	String	18-character `Id` of the user's manager
`managerName`	String	Locale-based concatenation of manager's first and last names
`mySubscription`	Reference	If the logged-in user is following this user, this contains information about the subscription, else returns `null`
`name`	String	Locale-based concatenation of user's first and last names
`phoneNumbers`	Phone Number[]	Collection of user's phone numbers
`photo`	Photo	Information about the user's photos
`title`	String	Title of the user
`type`	String	`User`
`url`	String	URL to the user's profile

Name	Type	Description
`username`	String	User name of the user, such as *`Admin@mycompany.com`*. Available in versions 24.0 and later.

User Page

Name	Type	Description
`currentPageUrl`	String	URL to the current page
`nextPageUrl`	String	URL to the next page, if there is no next page, returns the string "null"
`previousPageUrl`	String	URL to the previous page, if there is no previous page, returns the string "null"
`users`	User Detail[]	Collection of user detail information. Available in versions 24.0 and later.

User Settings

Available in versions 24.0 and later.

Name	Type	Description
`canFollow`	Boolean	Organization has Chatter customers
`canModifyAllData`	Boolean	User has "Modify all Data" permission
`canOwnGroups`	Boolean	User can be a group manager
`canViewAllData`	Boolean	User has "View all Data" permission
`canViewAllGroups`	Boolean	User has "View all Groups" permission
`canViewAllUsers`	Boolean	User has "View all Users" permission
`canViewFullUserProfile`	Boolean	User can see of other user's Chatter profiles
`canViewPublicFiles`	Boolean	User can see all files marked as public
`externalUser`	Boolean	User is a Chatter customer
`userId`	String	18-character `Id` of the user
`userLocale`	String	Locale of user

User Status

Note: This response is no longer available as of version 25.0.

Name	Type	Description
`body`	Feed Body[]	Body of the post
`parentId`	String	18-character `Id`
`url`	String	URL to this user status post

User Summary

Name	Type	Description
`companyName`	String	Name of the company
`firstName`	String	User's first name
`id`	String	18-character `Id`
`isActive`	Boolean	`true` if user is active, `false` otherwise. Available in versions 25.0 and higher.
`isChatterGuest`	Boolean	`true` if user is a Chatter customer, `false` otherwise
`lastName`	String	User's last name
`mySubscription`	Reference	If the logged-in user is following this user, this contains information about the subscription, else returns `null`
`name`	String	Locale-based concatenation of first and last names
`photo`	Photo	Information about the user's photos
`title`	String	Title of the user
`type`	String	`User`
`url`	String	URL to the user's profile

Response Body Encoding

Chatter API serves user-submitted content that is often not filtered at input and may come from many different sources including third-party mobile and web applications. Therefore, developers creating applications that consume Chatter API output must take care to properly process the output for the context in which they use the data. Failing to do so may expose their application and their users to Cross Site Scripting (XSS) exploits and other problems that may result in data loss, application failure and exposure of confidential company information.

Chatter API strings are minimally HTML entity encoded by default, which is suitable in most cases for display between HTML tags, but not necessarily in other HTML contexts.

Chatter API output may be used in many contexts. Developers should not assume that the default entity encoding is appropriate for all contexts. In particular, using Chatter 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.

Chatter API response returns are encoded on output. We escape the following characters in all strings:

Character	Escaped as
`<`	`&lt;`
`>`	`&gt;`
`"`	`&quot;`
`'`	`&#39;`
`\`	`&#92;`
`&`	`&amp;`

For non-HTML contexts, such as native mobile applications, Chatter API clients may request raw (unencoded) output by setting the `X-Chatter-Entity-Encoding` HTTP header in your request to `false`.

Chatter 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.

Status Codes and Error Responses

Either when an errors occur, or a response is successful, the response header contains an HTTP code, and the response body usually contains the HTTP response code, the message accompanying that error, and if appropriate, the field or object where the error occurred.

HTTP Response Code	Description
200	Success with GET or HEAD request
201	Success with POST request
204	Success with DELETE
400	The request could not be understood, usually because of an invalid ID, such as a `userId`, `feedItemId`, or `groupId` being incorrect.
401	The session ID or OAuth token used has expired or is invalid. 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.
500	An error has occurred within Force.com, so the request could not be completed. Contact salesforce.com Customer Support.
503	Too many requests in an hour.

Examples

The following are examples of what is returned for errors.

- Incorrect ID:

  Using a non-existent ID in a request using JSON or XML (*`request_body`*`.json` or *`request_body`*`.xml`)

  ```
  {
    "fields" : [ "Id" ],
    "message" : "User ID: id value of incorrect type: 001900K0001pPuOAAU",
    "errorCode" : "MALFORMED_ID"
  }
  ```

- Resource does not exist:

  Requesting a resource that does not exist, for example, you try to create a record using a misspelled object name

  ```
  {
    "message" : "The requested resource does not exist",
    "errorCode" : "NOT_FOUND"
  }
  ```

Index