



Salesforce.com: Summer '11

# Chatter REST API



Note: Any unreleased services or features referenced in this or other press releases or public statements are not currently available and may not be delivered on time or at all. Customers who purchase our services should make their purchase decisions based upon features that are currently available.

Last updated: April 28, 2011

© Copyright 2000-2011 salesforce.com, inc. All rights reserved. Salesforce.com is a registered trademark of salesforce.com, inc., as are other names and marks. Other marks appearing herein may be trademarks of their respective owners.



# Table of Contents

<b>Getting Started.....</b>	<b>3</b>
<b>Chapter 1: Introducing the Chatter REST API.....</b>	<b>3</b>
Using Chatter API.....	4
Understanding the Chatter API Resources.....	4
Send HTTP Requests with cURL.....	8
<b>Chapter 2: Quick Start.....</b>	<b>9</b>
Prerequisites.....	10
Step One: Obtain a Salesforce Developer Edition Organization.....	10
Step Two: Set Up Authorization.....	10
Step Three: Connect to Chatter API Using OAuth.....	11
<b>Reference.....</b>	<b>13</b>
<b>Chapter 3: Chatter API Resources Overview.....</b>	<b>13</b>
Chatter API Resources Relationships.....	14
Chatter API: How do I. . .	15
<b>Chapter 4: Chatter API Resources.....</b>	<b>19</b>
Comments Resource.....	19
Feed-Items Resource.....	20
Group-Memberships Resource.....	24
Groups Resource.....	25
Likes.....	27
News Feed Resource.....	28
Record Feed Resource.....	30
Subscriptions Resource.....	34
To Feed Resource.....	35
User-Profile Feed Resource.....	39
Users Resource.....	41
<b>Chapter 5: Chatter API Resource Returns.....</b>	<b>46</b>
Returns.....	47
Status Codes and Error Responses.....	57
<b>Index.....</b>	<b>58</b>



# GETTING STARTED

## Chapter 1

### Introducing the Chatter REST API

---

#### In this chapter ...

- [Using Chatter API](#)
- [Understanding the Chatter API Resources](#)
- [Send HTTP Requests with cURL](#)



**Note:** The Chatter API is currently available as a Developer Preview and is automatically available in all Developer Edition organizations. For information on enabling the Chatter API in other editions, contact [salesforce.com](https://salesforce.com).

The Chatter API is a REST API that provides 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. The 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, and it's optimized for Web 2.0 developers.

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

The following sections include:

- [The key characteristics and architecture](#) of the Chatter REST API. This will help you understand how your applications can best use the Chatter API resources.
- How to [set up your development environment](#) so you can start working with the Chatter API.
- [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 to Salesforce.
- 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.

### How is the Chatter REST API Different from Salesforce's other APIs?

Chatter API complements the Web services API by making it easy to interact with Chatter data, such as feeds, group information, and Chatter user-profile details. In addition, the Chatter REST API is optimized to work with Web 2.0 resources.

The following are some of the differences between Chatter API and the Web services 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.

## Understanding the Chatter API Resources

---

Each resource in the Chatter REST API is a URI used with an HTTP method (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 the Chatter API are:

Resource	Description
<a href="#">Comments Resource</a>	The specified comment.
<a href="#">Feed-Items Resource</a>	The specified feed item.
<a href="#">Group-Memberships Resource</a>	The specified group membership.
<a href="#">Groups Resource</a>	The specified group.
<a href="#">Likes</a>	The specified like.
<a href="#">News Feed Resource</a>	The Chatter news feed of either the current user or a specified user.
<a href="#">Record Feed Resource</a>	A feed for a record, such as an account, a user, or a group.
<a href="#">Subscriptions Resource</a>	The specified subscription.
<a href="#">To Feed Resource</a>	@mentions of either the current user or a specified user, as well as posts to that user's feed.

Resource	Description
<a href="#">User-Profile Feed Resource</a>	The user-profile feed of either the current user or a specified user.
<a href="#">Users Resource</a>	Either the current user or a specified user's Chatter profile details, such as the description they have in their “About Me” section. This is not a feed. To access the user profile feed, use <a href="#">User-Profile Feed Resource</a> .

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/feeds/users/me
```

You could also specify a particular user by user ID:

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

When you successfully access a resource, the return is in a structured format. Generally more than one return is included in the raw body. For example, the following is the structured return of the above resources (users). Note that it includes other formatted returns as well, such as address, current status, and phone numbers:

Name	Type	Description
aboutMe	String	Text from user's profile
address	<a href="#">Address</a>	Address
currentStatus	<a href="#">User Status</a>	User's current status
email	String	Users's email address
firstName	String	User's first name
id	String	18 character String ID
isActive	Boolean	<code>true</code> if user is active, <code>false</code> otherwise
lastName	String	User's last name
name	String	Locale-based concatenation of user's first and last names
managerId	String	18 character String ID
managerName	String	Locale-based concatenation of manager's first and last names
phoneNumbers	<a href="#">Phone Numbers[]</a>	Collection of user's phone numbers.
photo	<a href="#">Photo</a> on page 54	Information about the user's photos
title	String	Title of the user
url	String	URL to the user's Chatter profile

The output is as follows:

```
{
  "aboutMe": "Head SE South",
  "address": {
    "city": "New Orleans",
    "country": "US",
    "state": "LA",
    "street": "923 Bourbon Street",
    "zip": "70116"
  },
  "currentStatus": {
    "body": {
      "messageSegments": [
        {
          "name": "Joe Blogs",
          "text": "@Joe Blogs",
          "type": "Mention",
          "user": {
            "firstName": "Joe",
            "id": "005D0000001GLnuIAN",
            "lastName": "Blogs",
            "name": "Joe Blogs",
            "photo": {
              "largePhotoUrl": "https://na1.salesforce.com/profilephoto/005/F",
              "smallPhotoUrl": "https://na1.salesforce.com/profilephoto/005/T"
            },
            "title": null,
            "url": "/services/data/v22.0/chatter/users/005D0000001GLnuIB"
          }
        },
        {
          "text": " Please Review",
          "type": "Text"
        }
      ],
      "text": "@Joe Blogs Please Review"
    },
    "parentId": "005D0000001GLowIAB",
    "url": "/services/data/v22.0/chatter/users/005D0000001GLowIAB/status"
  },
  "email": "UJones@seattlechatter.com",
  "firstName": "Uriah",
  "id": "005D0000001GLowIAB",
  "isActive": true,
  "lastName": "Jones",
  "managerId": "005D0000001GLjkYBW",
  "managerName": "Dana Du Prie",
  "name": "Uriah Jones",
  "phoneNumbers": [
    {
      "number": "(504) 999-9904",
      "type": "Work"
    },
    {
      "number": "(504) 999-2245",
      "type": "Mobile"
    },
    {
      "number": "(504) 263-4145",
      "type": "Fax"
    }
  ],
  "photo": {
    "largePhotoUrl": "https://na1.salesforce.com/profilephoto/729D000000008vV/F",
    "smallPhotoUrl": "https://na1.salesforce.com/profilephoto/729D000000008vV/T"
  },
}
```



```
{  "title": "Head SE South",  "url": "/services/data/v22.0/chatter/users/005D0000001GLowIAB"}
```



**Note:** Salesforce runs on multiple server instances. The examples in this guide use the *na1* instance. The instance your organization uses may be different.

The following are important characteristics of the Chatter REST 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

All resources are named using a base URI that follows your Force.com URI, for example, `/services/data/v22.0/chatter/` is the base URI, with the named resource `feeds/news/me`. Put it together, the full URL is:

```
https://na1-salesforce.com/services/data/v22.0/chatter/feeds/news/me
```

### Authentication

The Chatter REST API supports [OAuth 2.0](#).



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

### 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 the SOAP-based Web services API. XML requests are supported in UTF-8 and UTF-16, and XML responses are provided in UTF-8.

### Localized names and times

Both the names and datetimes returned by the Chatter API are localized 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.

### Chatter API Connection Limits

The Chatter REST API uses the Web services API connection limits. Namely, the Concurrent API Request Limits and Total API Request Limits, both which are described fully in the [Web Services API Developer's Guide](#).



**Note:** The total API request limits are per user per organization. For example, an Enterprise Edition organization with 100 users would be allowed 100,000 Chatter API calls per 24-hour period.

## 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/](http://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 (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 Two: Set Up Authorization](#) on page 10.
- 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.
- 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.

# Chapter 2

## Quick Start

---

### In this chapter ...

- [Prerequisites](#)
- [Step One: Obtain a Salesforce Developer Edition Organization](#)
- [Step Two: Set Up Authorization](#)
- [Step Three: Connect to Chatter API Using OAuth](#)

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

## Prerequisites

---

Completing the prerequisites makes 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 familiar with the basic concepts and workflow.
- Read through all the steps before beginning this quick start. You may also wish to review the rest of this document to familiarize yourself with terms and concepts.

## Step One: Obtain a Salesforce Developer Edition Organization

---

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

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

Also ensure that you have the “REST Enabled” permission selected.

## Step Two: Set Up Authorization

---

Setting up OAuth 2.0 requires that you take some steps within the Salesforce app and in other locations. If any of the steps are unfamiliar, you can consult the Salesforce online help or 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. If your application is going to be used by many different organizations, you should create the remote access application in your Developer Edition organization. If your application is only going to be used in a single organization, you can create the remote access application in the same organization as your users, however, generally you would still use a Developer Edition organization.

2. In the appropriate Salesforce organization, navigate to **Your Name** ► **Setup** ► **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://nal.salesforce.com/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 Salesforce](#).

## Step Three: Connect to Chatter API Using OAuth

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



**Note:** The OAuth 2.0 specification uses “client” instead of “consumer.” Salesforce supports OAuth 2.0.

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	<code>client_id</code>
Consumer Secret	<code>client_secret</code>
Callback URL	<code>redirect_uri</code>

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

This example also uses the following values:

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

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@seattlechatter.com
```

```
--form password=1Lsfdc!  
https://na1.salesforce.com/services/oauth2/token
```

The following is the response that includes the access token:

```
{  
  "id": "https://login.na1.salesforce.com/id/00DD0000000FJ6TMAW/  
    005D00000001B5T4IAK",  
  "issued_at": "1302907727777",  
  "instance_url": "https://na1.salesforce.com",  
  "signature": "5jcevY5fUai0lWntuSxkwBzWcvRjd01RCOkIBZpyGv0=",  
  "access_token": "00DD0000000FJ6T!AQkAQpde_DMf2vGzddfZmBRS95GojDbtA  
    rKkgukAgZP0OVfYY5KkAqhLw9ejeKIlpJ3FgwGAWeRlBiWRt8mfXEuAZGbZNosk"  
}
```

## 2. Access the Chatter API using the access token.

```
curl -X GET https://na1.salesforce.com/services/data/v22.0/chatter/users/me  
-H 'Authorization: OAuth 00DD0000000FJ6T!AQkAQpde_DMf2vGzddfZmBRS95Goj  
  DbtArKkgukAgZP0OVfYY5KkAqhLw9ejeKIlpJ3FgwGAWeRlBiWRt8mfXEuAZGbZNosk'
```

# REFERENCE

## Chapter 3

### Chatter API Resources Overview

---

#### In this chapter ...

- [Chatter API Resources Relationships](#)
- [Chatter API: How do I . .](#)

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.

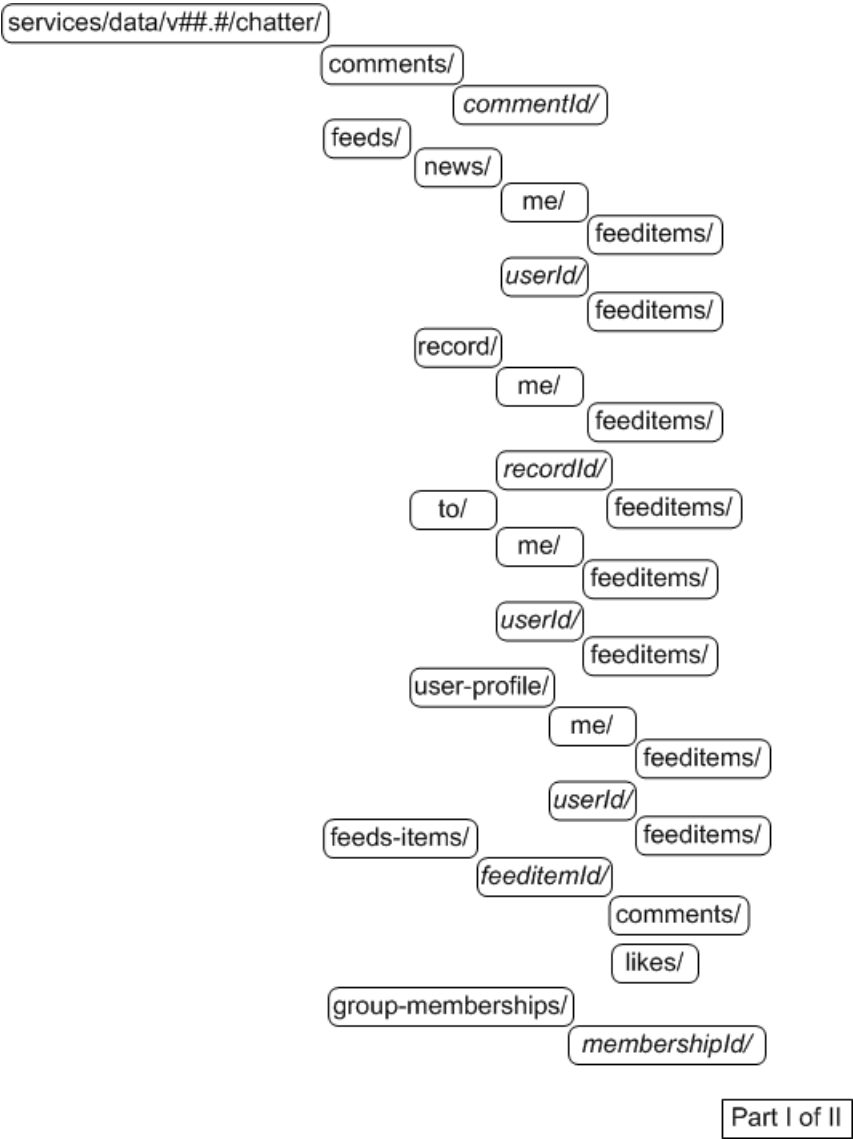


Figure 1: Chatter API Resources Relationships Part I



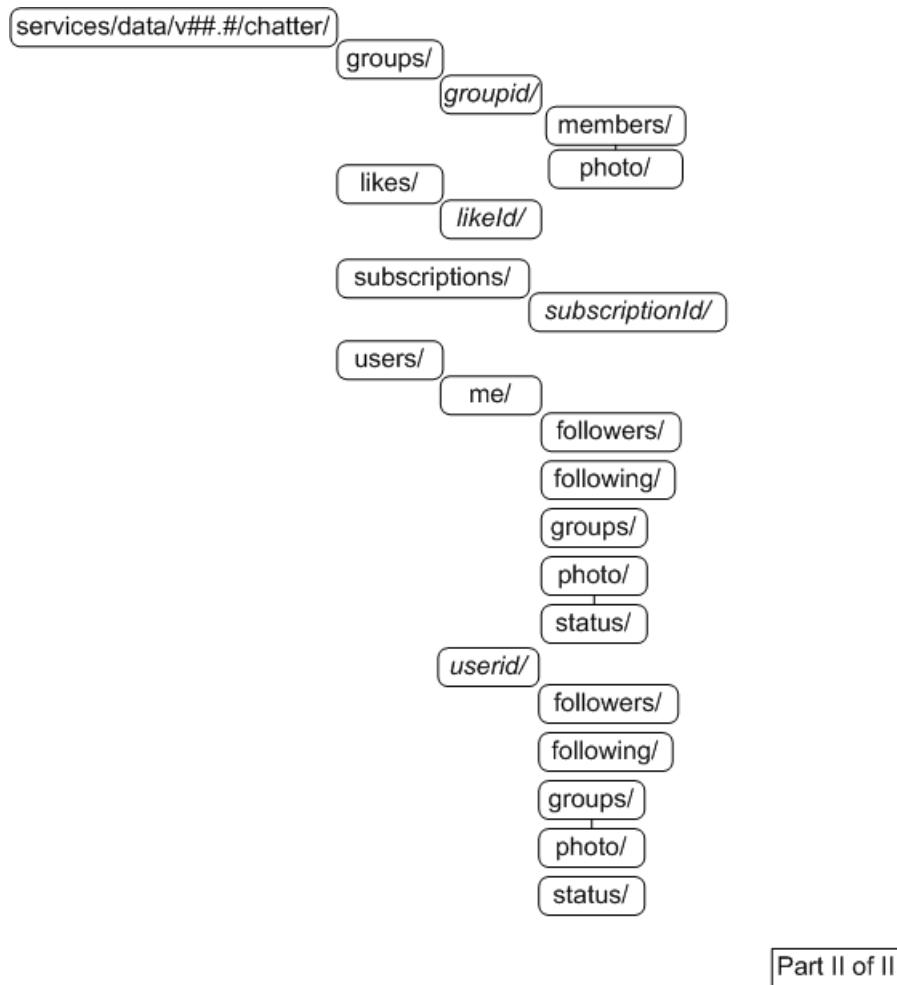


Figure 2: Chatter API Resources Relationships Part II

## Chatter API: How do I...

- [Update my status?](#)
- [Add a comment to a record or user?](#)
- [Like a feed item?](#)
- [Get my news feed?](#)
- [Get feed items posted to a record?](#)
- [Get the list of records a specific user follows?](#)
- [Join a group?](#)
- [Follow a record?](#)
- [Unfollow a record?](#)

### Update status

Use the following:

Resource:	/chatter/feeds/news/me/feed-items
-----------	-----------------------------------

HTTP method:	POST
Request parameter:	text
Example:	/services/data/v22.0/chatter/feeds/news /me/feed-items?text=New+post
Returns:	<a href="#">Feed Item</a>

### Add comments

Use the following:

Resource:	/chatter/feed-items/ <b>feedItemId</b> /comments
HTTP method:	POST
Request parameter:	text
Example:	/services/data/v22.0/chatter/feed-items/ 0D5D000000DaSbKAK?text=New+post
Returns:	<a href="#">Comment</a>

### Like feed items

Use the following:

Resource:	/chatter/feeds-items/ <b>feedItemId</b> /likes
HTTP method:	POST
Request parameter:	NA
Example:	/services/data/v22.0/chatter/feeds-items/0D5D000000DaZBKA0 /likes
Returns:	<a href="#">Like</a>

### Get news feeds

The following gets the news feed of the current user. Use the following:

Resource:	/chatter/feeds/news/me
HTTP method:	GET
Request parameter:	NA

Example:	<code>/services/data/v22.0/chatter/feeds/news/me</code>
Returns:	<a href="#">Feed</a>

### Get feed items for a record

Use the following:

Resource:	<code>/chatter/feeds/record/<b>recordId</b>/feed-items</code>
HTTP method:	GET
Request parameter:	page (optional)
Example:	<code>/services/data/v22.0/chatter/feeds/record/005x0000001oXx9AAE/feed-items?page=2011-03-31T20:57:24Z,0D5x0000001C2XPCA0</code>
Returns:	<a href="#">Feed Item Page</a>

### Get list of what user is following

Use the following:

Resource:	<code>/chatter/users/<b>userId</b>/following</code>
HTTP method:	GET
Request parameter:	page (optional)
Example:	<code>/services/data/v22.0/chatter/users/005D00000001GLowIAB/following?page=3</code>
Returns:	<a href="#">Subscription Page</a>

### Join a group

Use the following:

Resource:	<code>/chatter/groups/<b>groupId</b>/members</code>
HTTP method:	POST
Request parameter:	memberId
Example:	<code>/services/data/v22.0/chatter/groups/0F9D0000000006bOKAQ/members?memberId=005D00000001GLowIAB</code>
Returns:	<a href="#">Group Member</a>

### Follow a record

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

Resource:	<code>/chatter/users/me/following</code>
HTTP method:	POST
Request parameter:	<code>subjectId</code>
Example:	<code>/services/data/v22.0/chatter/users/me/following?subjectId=001D000000Iyu2p</code>
Returns:	<a href="#">Subscription</a>

### Unfollow records

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

Resource:	<code>/chatter/subscriptions/<i>subscriptionId</i></code>
HTTP method:	DELETE
Request parameter:	NA
Example:	<code>/services/data/v22.0/chatter/subscriptions/0E8D00000001JkFKAU</code>
Returns:	<a href="#">204: Successful Delete</a>

## Chapter 4

### Chatter API Resources

---

The following are the Chatter API resources:

Resource	Description
<a href="#">Comments Resource</a>	The specified comment.
<a href="#">Feed-Items Resource</a>	The specified feed item.
<a href="#">Group-Memberships Resource</a>	The specified group membership.
<a href="#">Groups Resource</a>	The specified group.
<a href="#">Likes</a>	The specified like.
<a href="#">News Feed Resource</a>	The Chatter news feed of either the current user or a specified user.
<a href="#">Record Feed Resource</a>	A feed for a record, such as an account, a user, or a group.
<a href="#">Subscriptions Resource</a>	The specified subscription.
<a href="#">To Feed Resource</a>	@mentions of either the current user or a specified user, as well as posts to that user's feed.
<a href="#">User-Profile Feed Resource</a>	The user-profile feed of either the current user or a specified user.
<a href="#">Users Resource</a>	Either the current user or a specified user's Chatter profile details, such as the description they have in their “About Me” section. This is not a feed. To access the user profile feed, use <a href="#">User-Profile Feed Resource</a> .

#### Comments Resource

---

The comments resource returns information about the specified comment.

##### Resource URL

The following is the resource URL for the comments resource, along with the allowed HTTP methods.

URL	GET	POST	DELETE	HEAD
<code>/chatter/comments/<i>commentId</i></code>	X		X	

## Returns

### Comment

## Example

You can find the ID for a specific comment from a feed, such as a news feed.

Use the following resource and the HTTP method GET to return information about a comment:

```
services/data/v22.0/chatter/comments/0D7D000000005NIKAY
```

This request returns:

```
{
  "body": {
    "messageSegments": [{
      "text": "Have you thought about adding colors?",
      "type": "Text"
    }],
    "text": "Have you thought about adding colors?"
  },
  "clientInfo": null,
  "createdDate": "2011-02-17T23:53:23.000+0000",
  "deletable": true,
  "id": "0D7D000000005NIKAY",
  "parentId": "0D5D00000000ChOCKA0",
  "url": "/services/data/v22.0/chatter/comments/0D7D000000005NIKAY",
  "user": {
    "firstName": "Uriah",
    "id": "005D00000001GLowIAB",
    "lastName": "Jones",
    "name": "Uriah Jones",
    "photo": {
      "largePhotoUrl": "https://na1.salesforce.com/profilephoto/729D000000008vV/F",
      "smallPhotoUrl": "https://na1.salesforce.com/profilephoto/729D000000008vV/T"
    },
    "title": "Head SE South",
    "url": "/services/data/v22.0/chatter/users/005D00000001GLowIAB"
  }
}
```

## Feed-Items Resource

---

The feed-items resource returns information about the specified feed items.

## Resource URLs

The following are the resource URLs for the feed-items resource, along with the allowed HTTP methods.

URL	GET	POST	DELETE	HEAD
/chatter/feed-items/ <i>feedItemId</i>	X		X	X
/chatter/feed-items/ <i>feedItemId</i> /comments	X	X		X
/chatter/feed-items/ <i>feedItemId</i> /likes	X	X		X

## Request Parameters

Additional request parameter for the following resources and the HTTP method GET or HEAD:

- /chatter/feed-items/*feedItemId*/comments
- /chatter/feed-items/*feedItemId*/likes

Parameter Name	Values	Description
page	Token	A generated token that indicates the view of feed items in the feed, 25 items at a time, ordered chronologically. The page token is automatically added to the URLs in feed items.

Additional request parameter for the resource /chatter/feed-items/*feedItemId*/comments and the HTTP method POST:

Parameter Name	Values	Description
text	String <i>Text to post</i>	Text of the item to be posted to the user's status

## Returns

- [Feed Item](#)
- [Comment Page](#)
- [Like Page](#)

## Example

The following example uses the resource /feed-items/0D5D0000000DaSbKAK and the HTTP method GET to return information about a specific feed item:

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

```

    },
    {
      "name": "Joe Blogs",
      "text": "@Joe Blogs",
      "type": "Mention",
      "user": {
        "firstName": "Joe",
        "id": "005D00000001GLnuIAN",
        "lastName": "Blogs",
        "name": "Joe Blogs",
        "photo": {
          "largePhotoUrl":
"https://na1.salesforce.com/profilephoto/729D00000000AQ4/F",
          "smallPhotoUrl": "https://na1.salesforce.com/profilephoto/729D00000000AQ4/T"
        }
      },
      "title": null,
      "url": "/services/data/v22.0/chatter/users/005D00000001GLnuIAG"
    }
  ],
  {
    "text": " did you see this?",
    "type": "Text"
  }
],
"text": "Hey @Joe Blogs did you see this?"
},
"clientInfo": null,
"comments": {
  "comments": [],
  "currentPageUrl": "/services/data/v22.0/chatter/feed-items/0D5D0000000DaSbKAK/comments",
  "nextPageUrl": null,
  "total": 0
},
"createdDate": "2011-04-04T22:29:28.000Z",
"event": false,
"id": "0D5D0000000DaSbKAK",
"likes": {
  "currentPageUrl":
"/services/data/v22.0/chatter/feed-items/0D5D0000000DaSbKAK/likes?page=0",
  "likes": [{
    "id": "0I0D0000000018AKAQ",
    "url": "/services/data/v22.0/chatter/likes/0I0D0000000018AKAQ",
    "user": {
      "firstName": "Uriah",
      "id": "005D00000001GLowIAB",
      "lastName": "Jones",
      "name": "Uriah Jones",
      "photo": {
        "largePhotoUrl": "https://na1.salesforce.com/profilephoto/729D000000008vV/F",
        "smallPhotoUrl": "https://na1.salesforce.com/profilephoto/729D000000008vV/T"
      }
    },
    "title": "Head SE South",
    "url": "/services/data/v22.0/chatter/users/005D00000001GLowIAB"
  }
  ],
  "nextPageUrl": null,
  "previousPageUrl": null,
  "total": 1
},
"modifiedDate": "2011-04-04T22:29:28.000Z",
"parentId": "005D00000001GLowIAB",
"parentName": "Uriah Jones",
"supplementalProperties": [],

```



```

"type": "UserStatus",
"url": "/services/data/v22.0/chatter/feed-items/0D5D0000000DaSbKAK",
"user": {
  "firstName": "Uriah",
  "id": "005D0000001GLowIAB",
  "lastName": "Jones",
  "name": "Uriah Jones",
  "photo": {
    "largePhotoUrl": "https://na1.salesforce.com/profilephoto/729D000000008vV/F",
    "smallPhotoUrl": "https://na1.salesforce.com/profilephoto/729D000000008vV/T"
  },
  "title": "Head SE South",
  "url": "/services/data/v22.0/chatter/users/005D0000001GLowIAB"
}
}

```

The following example uses this resource and the HTTP method POST to post a new comment to a feed item:

```
/feed-items/0D5D0000000DaSbKAK/comments?text=What+are+the+final+review+dates?
```

```

{
  "body": {
    "messageSegments": [{
      "text": "What are the final review dates?",
      "type": "Text"
    }],
    "text": "What are the final review dates?"
  },
  "clientInfo": {
    "applicationName": "SeattleChatter",
    "applicationUrl": "http://www.seattlechatter.com"
  },
  "createdDate": "2011-04-18T22:21:03.000+0000",
  "deletable": true,
  "id": "0D7D000000006xoKAA",
  "parentId": "0D5D0000000DaSbKAK",
  "url": "/services/data/v22.0/chatter/comments/0D7D000000006xoKAA",
  "user": {
    "firstName": "Uriah",
    "id": "005D0000001GLowIAB",
    "lastName": "Jones",
    "name": "Uriah Jones",
    "photo": {
      "largePhotoUrl": "https://na1.salesforce.com/profilephoto/729D000000008vV/F",
      "smallPhotoUrl": "https://na1.salesforce.com/profilephoto/729D000000008vV/T"
    },
    "title": "Head SE South",
    "url": "/services/data/v22.0/chatter/users/005D0000001GLowIAB"
  }
}

```

The following example uses this resource and the HTTP method GET to access a specific page of comments:

```
/feed-items/0D5D0000000DjKvKAK/comments?page=2011-04-18T22:54:15Z,0D7D000000006yZKAQ
```

```

{
  "comments": [
    {
      "body": {
        "messageSegments": [{
          "text": "Have you tried dividing the report by regions?",
          "type": "Text"
        }],
        "text": "Have you tried dividing the report by regions?"
      }
    ]
  }
}

```

```
    },
    "clientInfo": null,
    "createdDate": "2011-04-18T22:53:44.000+0000",
    "deletable": true,
    "id": "0D7D000000006y7KAA",
    "parentId": "005D0000001GLowIAG",
    "url": "/services/data/v22.0/chatter/comments/0D7D000000006y7KAA",
    "user": {
      "firstName": "Uriah",
      "id": "005D0000001GLowIAB",
      "lastName": "Jones",
      "name": "Uriah Jones",
      "photo": {
        "largePhotoUrl": "https://na1.salesforce.com/profilephoto/729D000000008vV/F",
        "smallPhotoUrl": "https://na1.salesforce.com/profilephoto/729D000000008vV/T"
      }
    },
    "title": "Head SE South",
    "url": "/services/data/v22.0/chatter/users/005D0000001GLowIAB"
  },
  {
    "body": {
      "messageSegments": [{
        . . .
      }],
      "currentPageUrl": "/services/data/v22.0/chatter/feed-items/0D5D00000000DjKvKAK/comments?page=2011-04-18T22:54:15Z,0D7D000000006yZKAQ",
      "nextPageUrl": "/services/data/v22.0/chatter/feed-items/0D5D00000000DjKvKAK/comments?page=2011-04-18T22:53:44Z,0D7D000000006y7KAA",
      "total": 40
    }
  }
}
```

## Group-Memberships Resource

The group-memberships resource is used to access members of a group.

### Resource URL

The following is the resource URL for the group-membership resource, along with the allowed HTTP methods.

URL	GET	POST	DELETE	HEAD
/chatter/group-memberships/ <i>membershipId</i>	X		X	X

### Returns

[Group Member](#)

### Example

You can find membership IDs (used in the following request) from the groups resource, for example,

```
/groups/0F9D000000006bOKA
```

Use the following resource with the HTTP method GET to return information about a particular subscriber to a group:

```
/chatter/group-memberships/0FBD00000000F110AE
```

To return the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<groupMember>
  <id>0FBD00000000F110AE</id>
  <role>GROUP_OWNER</role>
  <user>
    <firstName>Uriah</firstName>
    <id>005D00000001GLowIAB</id>
    <lastName>Jones</lastName>
    <name>Uriah Jones</name>
    <photoUrl>https://na1-salesforce.com/profilephoto/729D000000008vV/T</photoUrl>
    <title>Head SE South</title>
    <url>/services/data/v22.0/chatter/users/005D00000001GLowIAB</url>
  </user>
</groupMember>
```

## Groups Resource

The groups resource is used to get information about the specified group, such as members in the group.

### Resource URLs

The following are the resource URLs for the groups resource, along with the allowed HTTP methods.

URL	GET	POST	DELETE	HEAD
/chatter/groups/ <b>groupId</b>	X			X
/chatter/groups/ <b>groupId</b> /members	X	X		X
/chatter/groups/ <b>groupId</b> /photo	X	X	X	X

### Request Parameters

Additional request parameters for /chatter/groups/**groupId**/photo and HTTP methods POST:

Parameter Name	Values	Description
fileUpload	String <i>multipart/form-data</i> for a file	Contents of file to be uploaded



**Note:** Photos are processed asynchronously and are activated for the group only when processing has completed.

Additional request parameters for /chatter/groups/**groupId**/members and HTTP method POST:

Parameter Name	Values	Description
userId	ID <i>userId</i>	ID of the user to add to the group.

Additional request parameters for `/chatter/groups/groupId/members` and HTTP method GET:

Parameter Name	Values	Description
page	Token	A generated token that indicates the view of feed items in the feed, 25 items at a time, ordered chronologically. The page token is automatically added to the URLs in feed items.

## Returns

- [Group](#)
- [Group Member Page](#)
- [Photo](#)
- [Group Member](#)

## Examples

The following resource with the HTTP method GET returns information about the specified group:

```
/groups/0F9D000000006bOKAQ
```

Returns the following information:

```
{
  "description": "Get company announcements and other important updates",
  "id": "0F9D000000006bOKAQ",
  "memberCount": 29,
  "name": "All Seattle Chatter",
  "photo": {
    "largePhotoUrl": "https://na1.salesforce.com/profilephoto/729D000000002Sx/F",
    "smallPhotoUrl": "https://na1.salesforce.com/profilephoto/729D000000002Sx/T"
  },
  "url": "/services/data/v22.0/chatter/groups/0F9D000000006bOKAQ",
  "visibility": "PUBLIC_ACCESS"
}
```

Using the following resource with the HTTP method POST adds a member to the specified group:

```
/groups/0F9D000000006bOKAQ/members?userid=005D0000001GLnu
```

Returns the following:

```
{
  "id": "0FBD00000000UTMOA2",
  "role": "STANDARD_MEMBER",
  "user": {
    "firstName": "Joe",
    "id": "005D0000001GLnuIAN",
    "lastName": "Blogs",
    "name": "Joe Blogs",

```

```

    "photo": {
      "largePhotoUrl": "https://na1.salesforce.com/profilephoto/729D00000000AQ4/F",
      "smallPhotoUrl": "https://na1.salesforce.com/profilephoto/729D00000000AQ4/T"
    },
    "title": null,
    "url": "/services/data/v22.0/chatter/users/005D00000001GLnuIAN"
  }
}

```

## Likes

The likes resource returns information about the specified like.

### Resource URLs

The following is the resource URLs for the likes resource, along with the allowed HTTP methods.

URL	GET	POST	DELETE	HEAD
/chatter/likes/ <i>likeId</i>	X		X	

### Returns

- [Like Page](#)
- [Like](#)

### Examples

You can get the ID of a specific like from a general feed, such as a news feed.

The following resource and the HTTP method GET returns information about the specified like:

```
/chatter/likes/0I0D000000001KRKAY
```

Returns the following information:

```

{
  "id": "0I0D000000001KRKAY",
  "url": "/services/data/v22.0/chatter/likes/0I0D000000001KRKAY",
  "user": {
    "firstName": "Uriah",
    "id": "005D00000001GLowIAB",
    "lastName": "Jones",
    "name": "Uriah Jones",
    "photo": {
      "largePhotoUrl": "https://na1.salesforce.com/profilephoto/729D000000008vV/F",
      "smallPhotoUrl": "https://na1.salesforce.com/profilephoto/729D000000008vV/T"
    },
    "title": "Head SE South",
    "url": "/services/data/v22.0/chatter/users/005D00000001GLowIAB"
  }
}

```

## News Feed Resource

The news feed resource returns the news feed of either the current or specified user.

### Resource URLs

The following are the resource URLs for the news feed resource, along with the allowed HTTP methods.

URL	GET	POST	DELETE	HEAD
/chatter/feeds/news/me	X			X
/chatter/feeds/news/me/feed-items	X	X <sup>1</sup>		X
/chatter/feeds/news/ <i>userId</i>	X			X
/chatter/feeds/news/ <i>userId</i> /feed-items	X	X <sup>2</sup>		X

### Request Parameters

Additional request parameters for /chatter/feeds/news/me | *userId*/feed-items and HTTP methods GET or HEAD:

Parameter Name	Values	Description
page	Token	A generated token that indicates the view of feed items in the feed, 25 items at a time, ordered chronologically. The page token is automatically added to the URLs in feed items.

Additional request parameters for /chatter/feeds/news/me | *userId*/feed-items and HTTP method POST, used to create a new feed item. This is automatically a status update when using me, or when *userId* is the same as the current, logged-in user.

Parameter Name	Values	Description	Required or Optional	Type of Post
contentDocumentId	String	ID of an existing file to be posted.	Either contentDocumentId or feedItemFileUpload is required for file posts	File post
fileName	String	File name of a file to be uploaded and posted.	Required	File post

<sup>1</sup> This is also a status update

<sup>2</sup> This is also a status update when *userId* is the same as the current, signed in user

Parameter Name	Values	Description	Required or Optional	Type of Post
feedItemFileUpload	String	Contents of file to be uploaded. Must be an absolute path.	Either contentDocumentId or feedItemFileUpload is required for file posts	File post
desc	String	Description of file being posted. Only used with fileName.	Optional	File post
text	String	Text of the feed item.	Required for text posts, optional for link or file post	All posts
url	String	Href of the URL.	Required	Link post
urlName	String	Description name of the URL	Optional	Link post

## Returns

- [Feed](#)
- [Feed Item](#)

## Examples

The following is an example of updating a news feed with a file post. Note that this is also a status update.

```
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: OAuth 00Dx0000000M5Li!ARIAQCZ4DkPlHzGMaYKgVubVN.
K3ztqnibMTXDGDs8zEjbB7wVC6xA0pcfOMFWwJkjmNhNIH5RbVZ722hBeN0VE_c8jNFrAv'
https://na1-salesforce.com/services/data/v22.0/chatter/feeds/news/me/feed-items
```

Use the following resource with the HTTP method POST to create a link post to the feed of the current user:

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

Which returns the following information:

```
{
  "body": {
    "messageSegments": [{
      "text": "Did you see this?",
      "type": "Text"
    }],
    "text": "Did you see this?"
  },
  "clientInfo": {
    "applicationName": "SeattleChatter",
    "applicationUrl": "http://www.seattlechatter.com"
  },
  "comments": {
    "comments": [],
    "currentPageUrl": "/services/data/v22.0/chatter/feed-items/0D5D0000000Dpr5KAC/comments",
  }
}
```

```

    "nextPageUrl": null,
    "total": 0
  },
  "createdDate": "2011-04-19T23:20:15.000Z",
  "event": false,
  "id": "0D5D0000000Dpr5KAC",
  "likes": {
    "currentPageUrl":
"/services/data/v22.0/chatter/feed-items/0D5D0000000Dpr5KAC/likes?page=0",
    "likes": [],
    "nextPageUrl": null,
    "previousPageUrl": null,
    "total": 0
  },
  "modifiedDate": "2011-04-19T23:20:15.000Z",
  "parentId": "005D0000001GLowIAB",
  "parentName": "Uriah Jones",
  "supplementalProperties": [{
    "data": {
      "title": "www.chatter.com",
      "url": "http://www.chatter.com"
    },
    "type": "link"
  }],
  "type": "LinkPost",
  "url": "/services/data/v22.0/chatter/feed-items/0D5D0000000Dpr5KAC",
  "user": {
    "firstName": "Uriah",
    "id": "005D0000001GLowIAB",
    "lastName": "Jones",
    "name": "Uriah Jones",
    "photo": {
      "largePhotoUrl": "https://na1.salesforce.com/profilephoto/729D000000008vV/F",
      "smallPhotoUrl": "https://na1.salesforce.com/profilephoto/729D000000008vV/T"
    },
    "title": "Head SE South",
    "url": "/services/data/v22.0/chatter/users/005D0000001GLowIAB"
  }
}

```

## Record Feed Resource

---

The record feed resource returns feeds of the specified record.



## Resource URLs

The following are the resource URLs for the record feed resource, along with the allowed HTTP methods.

URL	GET	POST	DELETE	HEAD
/chatter/feeds/record/me/	X			X
/chatter/feeds/record/me/feed-items	X	X <sup>3</sup>		X
/chatter/feeds/record/ <b>recordId</b>	X			X
/chatter/feeds/record/ <b>recordId</b> /feed-items	X	X <sup>4</sup>		X

## Request Parameters

Additional request parameters for /chatter/feeds/record/me | **recordId**/feed-items and HTTP methods GET or HEAD:

Parameter Name	Values	Description
page	Token	A generated token that indicates the view of feed items in the feed, 25 items at a time, ordered chronologically. The page token is automatically added to the URLs in feed items.

Additional request parameters for /chatter/feeds/record/me | **recordId**/feed-items and HTTP method POST, used to create a new feed item. If *recordId* is a User ID, this is also a status update.

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

<sup>3</sup> This is also a status update

<sup>4</sup> If the specified *recordId* is a userId, and it is the same as the current, signed in user, this is also a status update

Parameter Name	Values	Description	Required or Optional	Type of Post
text	String	Text of the feed item.	Required for text posts, optional for link or file post	All posts
url	String	Href of the URL.	Required	Link post
urlName	String	Description name of the URL	Optional	Link post

## Returns

- [Feed](#)
- [Feed Item](#)

## Examples

Use the following resource with the HTTP method GET to return all the feed items for the specified record, in this example, an account:

```
/feeds/record/001D000000Iyu21
```

Which returns the following information:

```
{
  "feedItems": {
    "items": [
      {
        "body": {
          "messageSegments": [
            {
              "text": "Have they seen the latest presentation by ",
              "type": "Text"
            },
            {
              "name": "Joe Blogs",
              "text": "@Joe Blogs",
              "type": "Mention",
              "user": {
                "firstName": "Joe",
                "id": "005D0000001GLnuIAN",
                "lastName": "Blogs",
                "name": "Joe Blogs",
                "photo": {
                  "largePhotoUrl":
                    "https://na1.salesforce.com/profilephoto/729D00000000AQ4/F",
                  "smallPhotoUrl":
                    "https://na1.salesforce.com/profilephoto/729D00000000AQ4/T"
                },
                "title": null,
                "url": "/services/data/v22.0/chatter/users/005D0000001GLnuIAN"
              }
            },
            {
              "text": " ?",
              "type": "Text"
            }
          ],
          "text": "Have they seen the latest presentation by @Joe Blogs ?"
        },
        "clientInfo": null,
        "comments": {
          "comments": [],

```

```

        "currentPageUrl":
"/services/data/v22.0/chatter/feed-items/0D5D0000000DqRRKA0/comments",
        "nextPageUrl": null,
        "total": 0
    },
    "createdDate": "2011-04-20T16:37:15.000Z",
    "event": false,
    "id": "0D5D0000000DqRRKA0",
    "likes": {
        "currentPageUrl":
"/services/data/v22.0/chatter/feed-items/0D5D0000000DqRRKA0/likes?page=0",
        "likes": [],
        "nextPageUrl": null,
        "previousPageUrl": null,
        "total": 0
    },
    "modifiedDate": "2011-04-20T16:37:15.000Z",
    "parentId": "001D000000Iyu2lIAB",
    "parentName": "Edge Communications",
    "supplementalProperties": [],
    "type": "TextPost",
    "url": "/services/data/v22.0/chatter/feed-items/0D5D0000000DqRRKA0",
    "user": {
        "firstName": "Uriah",
        "id": "005D0000001GLowIAB",
        "lastName": "Jones",
        "name": "Uriah Jones",
        "photo": {
            "largePhotoUrl": "https://na1.salesforce.com/profilephoto/729D000000008vV/F",
            "smallPhotoUrl": "https://na1.salesforce.com/profilephoto/729D000000008vV/T"
        },
        "title": "Head SE South",
        "url": "/services/data/v22.0/chatter/users/005D0000001GLowIAB"
    }
    },
    "nextPageUrl": null
}
}

```

Use the following resource with the HTTP method POST to add a new feed item to the specified record, in this example, an account:

```

/feeds/record/001D000000Iyu2m/feed-items?
text=Have+they+seen+the+latest+proposal?

```

Which returns the following information:

```

{
  "body": {
    "messageSegments": [{
      "text": "Have they seen the latest proposal ?",
      "type": "Text"
    }],
    "text": "Have they seen the latest proposal ?"
  },
  "clientInfo": {
    "applicationName": "SeattleChatter",
    "applicationUrl": "http://www.seattlechatter.com"
  },
  "comments": {
    "comments": [],
    "currentPageUrl": "/services/data/v22.0/chatter/feed-items/0D5D0000000DqUrKAK/comments",
    "nextPageUrl": null,
    "total": 0
  },
}

```

```
    "createdDate": "2011-04-20T17:25:13.000Z",
    "event": false,
    "id": "0D5D0000000DqUrKAK",
    "likes": {
      "currentPageUrl":
"/services/data/v22.0/chatter/feed-items/0D5D0000000DqUrKAK/likes?page=0",
      "likes": [],
      "nextPageUrl": null,
      "previousPageUrl": null,
      "total": 0
    },
    "modifiedDate": "2011-04-20T17:25:13.000Z",
    "parentId": "001D000000Iyu2mIAB",
    "parentName": "Burlington Textiles Corp of America",
    "supplementalProperties": [],
    "type": "TextPost",
    "url": "/services/data/v22.0/chatter/feed-items/0D5D0000000DqUrKAK",
    "user": {
      "firstName": "Uriah",
      "id": "005D0000001GLowIAB",
      "lastName": "Jones",
      "name": "Uriah Jones",
      "photo": {
        "largePhotoUrl": "https://na1.salesforce.com/profilephoto/729D000000008vV/F",
        "smallPhotoUrl": "https://na1.salesforce.com/profilephoto/729D000000008vV/T"
      },
      "title": "Head SE South",
      "url": "/services/data/v22.0/chatter/users/005D0000001GLowIAB"
    }
  }
}
```

## Subscriptions Resource

The subscriptions resource returns information about the specified subscription. You can find subscription IDs using the following resource:

```
/users/me | userID/following
```

### Resource URLs

The following is the resource URL for the subscriptions resource and the allowed HTTP methods:

URL	GET	POST	DELETE	HEAD
/chatter/subscriptions/ <i><b>subscriptionId</b></i>	X		X	X

### Returns

[Subscription](#)

### Examples

With the following resource and the HTTP method GET returns information about the specified subscription:

```
/subscriptions/0E8D00000001YLVKA2
```

The following information is returned:

```
{
  "id": "0E8D00000001YLVKA2",
  "subject": {
    "firstName": null,
    "id": "001D000000Iyu21IAB",
    "lastName": null,
    "name": "Edge Communications",
    "type": "Account",
    "url": "/services/data/v22.0/001D000000Iyu21IAB"
  },
  "subscriber": {
    "firstName": "Uriah",
    "id": "005D0000001GLowIAB",
    "lastName": "Jones",
    "name": "Uriah Jones",
    "type": "User",
    "url": "/services/data/v22.0/chatter/users/005D0000001GLowIAB"
  },
  "url": "/services/data/v22.0/chatter/subscriptions/0E8D00000001YLVKA2"
}
```

## To Feed Resource

The to feeds resource returns @mentions of either the current or specified user, as well as posts to the user's feed.

### Resource URLs

The following are the resource URLs for the to feeds resource, along with the allowed HTTP methods.

URL	GET	POST	DELETE	HEAD
/chatter/feeds/to/me	X			X
/chatter/feeds/to/me/feed-items	X	X <sup>5</sup>		X
/chatter/feeds/to/ <b>userId</b>	X			X
/chatter/feeds/to/ <b>userId</b> /feed-items	X	X <sup>6</sup>		X

### Request Parameters

Additional request parameters for /chatter/feeds/to/me | **userId**/feed-items and HTTP methods GET or HEAD:

<sup>5</sup> This is also a status update

<sup>6</sup> This is also a status update when *userId* is the same as the current, signed in user

Parameter Name	Values	Description
page	Token	A generated token that indicates the view of feed items in the feed, 25 items at a time, ordered chronologically. The page token is automatically added to the URLs in feed items.

Additional request parameters for `/chatter/feeds/to/me` | `userId/feed-items` and HTTP method POST, used to create a new feed item. This is also a status update.

Parameter Name	Values	Description	Required or Optional	Type of Post
contentDocumentId	String	ID of an existing file to be posted.	Either contentDocumentId or feedItemFileUpload is required for file posts	File post
fileName	String	File name of a file to be uploaded and posted.	Required	File post
feedItemFileUpload	String	Contents of file to be uploaded. Must be an absolute path.	Either contentDocumentId or feedItemFileUpload is required for file posts	File post
desc	String	Description of file being posted. Only used with fileName.	Optional	File post
text	String	Text of the feed item.	Required for text posts, optional for link or file post	All posts
url	String	Href of the URL.	Required	Link post
urlName	String	Description name of the URL	Optional	Link post

## Returns

- [Feed](#)
- [Feed Item](#)

## Examples

Use the following resource and the HTTP method GET to return the @mentions for the specified user:

```
/feeds/to/005D0000001GLnuIAN
```

Which returns the following:

```
{ "feedItems": {
  "items": [
    {
      "body": {
```

```

    "messageSegments": [
      {
        "text": "Have they seen the latest presentation from ",
        "type": "Text"
      },
      {
        "name": "Joe Blogs",
        "text": "@Joe Blogs",
        "type": "Mention",
        "user": {
          "firstName": "Joe",
          "id": "005D00000001GLnuIAN",
          "lastName": "Blogs",
          "name": "Joe Blogs",
          "photo": {
            "largePhotoUrl":
"https://na1.salesforce.com/profilephoto/729D00000000AQ4/F",
            "smallPhotoUrl":
"https://na1.salesforce.com/profilephoto/729D00000000AQ4/T"
          },
          "title": null,
          "url": "/services/data/v22.0/chatter/users/005D00000001GLnuIAN"
        }
      },
      {
        "text": "?",
        "type": "Text"
      }
    ],
    "text": "Have they seen the latest presentation from @Joe Blogs?"
  },
  "clientInfo": null,
  "comments": {
    "comments": [],
    "currentPageUrl":
"/services/data/v22.0/chatter/feed-items/0D5D0000000DqQuKAK/comments",
    "nextPageUrl": null,
    "total": 0
  },
  "createdDate": "2011-04-20T16:42:45.000Z",
  "event": false,
  "id": "0D5D0000000DqQuKAK",
  "likes": {
    "currentPageUrl":
"/services/data/v22.0/chatter/feed-items/0D5D0000000DqQuKAK/likes?page=0",
    "likes": [],
    "nextPageUrl": null,
    "previousPageUrl": null,
    "total": 0
  },
  "modifiedDate": "2011-04-20T16:42:45.000Z",
  "parentId": "001D000000Iyu2oIAB",
  "parentName": "Dickenson plc",
  "supplementalProperties": [],
  "type": "TextPost",
  "url": "/services/data/v22.0/chatter/feed-items/0D5D0000000DqQuKAK",
  "user": {
    "firstName": "Uriah",
    "id": "005D00000001GLowIAB",
    "lastName": "Jones",
    "name": "Uriah Jones",
    "photo": {
      "largePhotoUrl": "https://na1.salesforce.com/profilephoto/729D000000008vV/F",
      "smallPhotoUrl": "https://na1.salesforce.com/profilephoto/729D000000008vV/T"
    }
  },

```

```

        "title": "Head SE South",
        "url": "/services/data/v22.0/chatter/users/005D0000001GLowIAB"
      },
    ],
    "nextPageUrl": null
  }

```

Use the following resource and the HTTP method POST to add a link post the current user's news feed.

```
/feeds/to/me/feed-items?text=Interesting+blog&url=http://theenforcer.net/
```

This is the same as if you used either the news or the record resource, for example:

```
/feeds/news/me/feed-items?text=Interesting+blog&url=http://theenforcer.net/
```

```

{
  "body": {
    "messageSegments": [{
      "text": "Interesting blog",
      "type": "Text"
    }],
    "text": "Interesting blog"
  },
  "clientInfo": {
    "applicationName": "SeattleChatter",
    "applicationUrl": "http://www.seattlechatter.com"
  },
  "comments": {
    "comments": [],
    "currentPageUrl": "/services/data/v22.0/chatter/feed-items/0D5D0000000DqVGKA0/comments",
    "nextPageUrl": null,
    "total": 0
  },
  "createdDate": "2011-04-20T18:00:27.000Z",
  "event": false,
  "id": "0D5D0000000DqVGKA0",
  "likes": {
    "currentPageUrl":
"/services/data/v22.0/chatter/feed-items/0D5D0000000DqVGKA0/likes?page=0",
    "likes": [],
    "nextPageUrl": null,
    "previousPageUrl": null,
    "total": 0
  },
  "modifiedDate": "2011-04-20T18:00:27.000Z",
  "parentId": "005D0000001GLnuIAN",
  "parentName": "Joe Blogs",
  "supplementalProperties": [{
    "data": {
      "title": "theenforcer.net",
      "url": "http://theenforcer.net/"
    },
    "type": "link"
  }],
  "type": "LinkPost",
  "url": "/services/data/v22.0/chatter/feed-items/0D5D0000000DqVGKA0",
  "user": {
    "firstName": "Joe",
    "id": "005D0000001GLnuIAN",
    "lastName": "Blogs",
    "name": "Joe Blogs",
    "photo": {
      "largePhotoUrl": "https://na1.salesforce.com/profilephoto/729D00000000AQ4/F",

```



```

    "smallPhotoUrl": "https://na1.salesforce.com/profilephoto/729D00000000AQ4/T"
  },
  "title": null,
  "url": "/services/data/v22.0/chatter/users/005D0000001GLnuIAN"
}

```

## User-Profile Feed Resource

The user-profile feed resource returns the user profile feed of the specified user. This is different than `/chatter/feed-items`, which returns everything in a user's feed, including posts, status updates, group updates, and record updates.

### Resource URLs

The following are the resource URLs for the user-profile feed resource, along with the allowed HTTP methods.

URL	GET	POST	DELETE	HEAD
<code>/chatter/feeds/user-profile/me</code>	X			X
<code>/chatter/feeds/user-profile/me/feed-items</code>	X	X <sup>7</sup>		X
<code>/chatter/feeds/user-profile/<i>userId</i></code>	X			X
<code>/chatter/feeds/user-profile/<i>userId</i>/feed-items</code>	X	X <sup>8</sup>		X

### Request Parameters

Additional request parameters for `/chatter/feeds/user-profile/me` | `userId/feed-items` and HTTP methods GET or HEAD:

Parameter Name	Values	Description
page	Token	A generated token that indicates the view of feed items in the feed, 25 items at a time, ordered chronologically. The page token is automatically added to the URLs in feed items.

Additional request parameters for `/chatter/feeds/user-profile/me` | `userId/feed-items` and HTTP method POST, used to create a new feed item. This is also a status update.

<sup>7</sup> This is also a status update

<sup>8</sup> This is also a status update when *userId* is the same as the current, signed in user

Parameter Name	Values	Description	Required or Optional	Type of Post
contentDocumentId	String	ID of an existing file to be posted.	Either contentDocumentId or feedItemFileUpload is required for file posts	File post
fileName	String	File name of a file to be uploaded and posted.	Required	File post
feedItemFileUpload	String	Contents of file to be uploaded. Must be an absolute path.	Either contentDocumentId or feedItemFileUpload is required for file posts	File post
desc	String	Description of file being posted. Only used with fileName.	Optional	File post
text	String	Text of the feed item.	Required for text posts, optional for link or file post	All posts
url	String	Href of the URL.	Required	Link post
urlName	String	Description name of the URL	Optional	Link post

## Returns

- [Feed](#)
- [Feed Item](#)

## Examples

Using the following resource and the HTTP method GET to return the user profile feed for a newly created user:

```
/feeds/user-profile/005D0000001GOZd
```

Which returns the following information. Note that most of it is blank, as this is a newly created user without any posts yet.

```
{
  "feedItems": {
    "items": [
      {
        "body": {
          "messageSegments": [
            {
              "text": "created this user.",
              "type": "Text"
            }
          ],
          "text": "created this user."
        },
        "clientInfo": null,
        "comments": {
          "comments": [],
          "currentPageUrl":
"/services/data/v22.0/chatter/feed-items/0D5D0000000CyCXKA0/comments",
          "nextPageUrl": null,
          "total": 0
        },
        "createdDate": "2011-03-02T18:21:18.000Z",
        "event": true,
      }
    ]
  }
}
```

```

    "id": "0D5D00000000CyCXKA0",
    "likes": {
      "currentPageUrl":
"/services/data/v22.0/chatter/feed-items/0D5D00000000CyCXKA0/likes?page=0",
      "likes": [],
      "nextPageUrl": null,
      "previousPageUrl": null,
      "total": 0
    },
    "modifiedDate": "2011-03-02T18:21:18.000Z",
    "parentId": "005D00000001GOZdIAO",
    "parentName": "G Jones",
    "supplementalProperties": [],
    "type": "TrackedChange",
    "url": "/services/data/v22.0/chatter/feed-items/0D5D00000000CyCXKA0",
    "user": {
      "firstName": "Uriah",
      "id": "005D00000001GLowIAG",
      "lastName": "Jones",
      "name": "Uriah Jones",
      "photo": {
        "largePhotoUrl": "https://na1.salesforce.com/profilephoto/729D0000000008vV/F",
        "smallPhotoUrl": "https://na1.salesforce.com/profilephoto/729D0000000008vV/T"
      },
      "title": "Head SE South",
      "url": "/services/data/v22.0/chatter/users/005D00000001GLowIAB"
    }
  },
  "nextPageUrl": null

```

## Users Resource

---

The users resource returns information about the user, such as who is following the user or the latest user status.

## Resource URLs

The following are the resource URLs for the User resource, along with the allowed HTTP methods.

URL	GET	POST	DELETE	HEAD
/chatter/users/me	X			X
/chatter/users/me/followers	X	X		X
/chatter/users/me/following	X	X		X
/chatter/users/me/groups	X			X
/chatter/users/me/photo	X	X	X	X
/chatter/users/me/status	X	X	X	X
/chatter/users/ <b>userId</b>	X			X
/chatter/users/ <b>userId</b> /followers	X	X		X
/chatter/users/ <b>userId</b> /following	X	X		X
/chatter/users/ <b>userId</b> /groups	X			X
/chatter/users/ <b>userId</b> /photo	X	X	X	X
/chatter/users/ <b>userId</b> /status	X	X	X	X

## Request Parameters

Additional request parameter for HTTP methods GET or HEAD for the following resources:

- /chatter/users/me | **userId**/followers
- /chatter/users/me | **userId**/following
- /chatter/users/me | **userId**/groups

Parameter Name	Values	Description
page	Integer <i>page number</i>	Page number of the list of followers or what is being followed. Items are indexed in sets of 25. A value of 0 returns the first 25 items, 1 returns items 26–50, and so on.

Additional request parameter for `/chatter/users/me | userId/following` and the HTTP method POST:

Parameter Name	Values	Description
subjectId	ID <i>recordID</i>	This request enables the user to follow the specified record. This could be a group ID, a user ID, an account ID, and so on.

Additional request parameters for `/chatter/users/me | userId/photo` and HTTP method POST:

Parameter Name	Values	Description
fileUpload	String <i>multipart/form-data</i> for a file	Contents of file to be uploaded



**Note:** Photos are processed asynchronously and are activated for the user only when processing has completed.

Additional request parameters for `/chatter/users/me | userId/status` and HTTP method POST:

Parameter Name	Values	Description
text	String <i>Text to post</i>	Text of the item to be posted to the user's status

## Returns

- [User Summary](#)
- [Subscription Page](#)
- [Subscription](#)
- [Group Page](#)
- [Photo](#)
- [User Status](#)

## Examples

The following resource used with the HTTP method GET returns general information about the current user:

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

Returns the following information:

```
{
  "aboutMe": "Head SE South",
  "address": {
    "city": "New Orleans",
    "country": "US",
    "state": "LA",
    "street": "923 Bourbon Street",
    "zip": "70116"
  },
  "currentStatus": {
```

```

    "body": {
      "messageSegments": [
        {
          "name": "Joe Blogs",
          "text": "@Joe Blogs",
          "type": "Mention",
          "user": {
            "firstName": "Joe",
            "id": "005D00000001GLnuIAN",
            "lastName": "Blogs",
            "name": "Joe Blogs",
            "photo": {
              "largePhotoUrl":
"https://na1.salesforce.com/profilephoto/729D00000000AQ4/F",
              "smallPhotoUrl":
"https://na1.salesforce.com/profilephoto/729D00000000AQ4/T"
            },
            "title": null,
            "url": "/services/data/v22.0/chatter/users/005D00000001GLnuIAN"
          }
        },
        {
          "text": " Please Review",
          "type": "Text"
        }
      ],
      "text": "@Joe Blogs Please Review"
    },
    "parentId": "005D00000001GLowIAB",
    "url": "/services/data/v22.0/chatter/users/005D00000001GLowIAB/status"
  },
  "email": "ujones@salesforce.com",
  "firstName": "Uriah",
  "id": "005D00000001GLowIAB",
  "isActive": true,
  "lastName": "Jones",
  "managerId": "005D00000001GLKMegh",
  "managerName": "Diana Du Prie",
  "name": "Uriah Jones",
  "phoneNumbers": [
    {
      "number": "(504) 999-9904",
      "type": "Work"
    },
    {
      "number": "(504) 999-2245",
      "type": "Mobile"
    },
    {
      "number": "(504) 263-4145",
      "type": "Fax"
    }
  ],
  "photo": {
    "largePhotoUrl": "https://na1.salesforce.com/profilephoto/729D000000008vV/F",
    "smallPhotoUrl": "https://na1.salesforce.com/profilephoto/729D000000008vV/T"
  },
  "title": "Head SE South",
  "url": "/services/data/v22.0/chatter/users/005D00000001GLowIAB"
}

```

The following example shows how to follow an account. The following resource with the HTTP method POST:

```
/chatter/users/me/following?subjectId=0E8D00000001YLVKA2
```

Returns the following information:

```
{
  "id": "0E8D00000001IUAKA2",
  "subject": {
    "firstName": null,
    "id": "001D000000Iyu2mIAB",
    "lastName": null,
    "name": "Burlington Textiles Corp of America",
    "type": "Account",
    "url": "/services/data/v22.0/001D000000Iyu2mIAB"
  },
  "subscriber": {
    "firstName": "Uriah",
    "id": "005D0000001GLowIAB",
    "lastName": "Jones",
    "name": "Uriah Jones",
    "type": "User",
    "url": "/services/data/v22.0/chatter/users/005D0000001GLowIAB"
  },
  "url": "/services/data/v22.0/chatter/subscriptions/0E8D00000001IUAKA2"
}
```

This example updates a photo. With the following cURL example and the HTTP method POST:

```
curl --form fileUpload=@/home/ujones/Downloads/george_washington.jpg
--header 'Authorization: OAuth 00Dx0000000M5Li'
http://na1.salesforce.com/services/data/v22.0/chatter/users/005x0000001V8RsBBN/photo
```

Returns the following information:

```
<?xml version="1.0" encoding="UTF-8"?>
<photo>
  <largePhotoUrl>https://na1.salesforce.com/profilephoto/729x0000000CrMp/F</largePhotoUrl>
  <smallPhotoUrl>https://na1.salesforce.com/profilephoto/729x0000000CrMp/T</smallPhotoUrl>
</photo>
```

The following example updates a user's status with a text post. (To update a user's status with a file or link post, use one of the feeds instead, such as a [news feed](#) or [record feed](#).) With the following resource and the HTTP method POST:

```
/users/005D0000001GLbn/status?text=Will+send+you+latest+updates+tonight
```

Returns the following information:

```
<?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/v22.0/chatter/users/005D0000001GLbn/status</url>
</currentStatus>
```

# Chapter 5

## Chatter API Resource Returns

---

### In this chapter ...

- [Returns](#)
- [Status Codes and Error Responses](#)

The Chatter API resources return [structured information](#), either in JSON or XML.

In addition, if there's an error, the resources return an [error code and description](#).



## Returns

The following are the various structured returns from successful execution of resources.

### Address

Name	Type	Description
city	String	Name of the city
country	String	Name of the country
state	String	Name of the state
street	String	Street number
zip	String	Zip code

### 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 Salesforce online help.
applicationUrl	String	Value from the <code>InfoURL</code> field of the remote access application use for authentication

### Comment

Name	Type	Description
body	<a href="#">Feed Body</a>	Text of the comment
clientInfo	<a href="#">ClientInfo</a>	Information about the remote access application used to authenticate the connection
createdDate	String	ISO8601 date string, for example, 2011-02-25T18:24:31.000Z
id	String	18 character String ID
isDeletable	Boolean	Can the context user delete the comment
parentId	String	This comment's feed item's subject ID
url	String	URL to this comment
user	<a href="#">User Summary</a>	User summary information

## Comment Page

Name	Type	Description
comments	<a href="#">Comment</a> []	Collection of comments
currentPageUrl	String	URL to the first page of comments, that is, the most recent comments, ten comments to a page
nextPageUrl	String	URL to the next page of comments, that is, older comments, 10 comments to a page
total	Integer	Total number of comments for the parent feed item

## Data: Content

Name	Type	Description
downloadUrl	String	Absolute URL that can be used to download the file
fileSize	String	Size of the file in bytes
fileType	String	Type of the file. Possible values include: <ul style="list-style-type: none"> <li>• Plain Text</li> <li>• Powerpoint 2007</li> <li>• Image – jpg</li> </ul>
id	String	18 character String ID
title	String	Name of the file, either specified when file was uploaded, or if not specified, the name of the file
versionId	String	Version ID of the file

## Data: Link

Name	Type	Description
title	String	Description given to link by user
url	String	URL of link

## Error Message

Name	Type	Description
errorCode	String	<a href="#">HTTP error code</a> , such as 200, 404, and so on
message	String	Description of error

**Feed**

Name	Type	Description
feedItems	<a href="#">Feed Item Page[]</a>	Collection of feed items

**Feed Item**

Name	Type	Description
body	<a href="#">Feed Body</a>	Information about the feed item
clientInfo	<a href="#">ClientInfo</a>	Information about the remote access application used to authenticate the connection
comments	<a href="#">Comment Page</a>	Comments for this feed item
createdDate	String	ISO8601 date string, for example, 2011-02-25T18:24:31.000Z
event	Boolean	true if feed item created due to an event change, false otherwise
id	String	18 character String ID
likes	<a href="#">Like Page</a>	Likes for this feed item
modifiedDate	String	ISO8601 date string, for example, 2011-02-25T18:24:31.000Z
parentId	String	18 character String ID of the parent of this feed item
parentName	String	Localized rendering of the parent name, such as the name of the user, group, or record
supplementalProperties	<a href="#">Supplemental Properties[]</a>	Collection of additional information about posts
type	String	Values are: <ul style="list-style-type: none"> <li>CONTENT_POST</li> <li>LINK_POST</li> <li>TEXT_POST</li> <li>TRACKED_CHANGE</li> <li>USER_STATUS</li> </ul>
url	String	URL to this feed item
user	<a href="#">User Summary</a>	Author of this feed item

## Feed Body

Name	Type	Description
messageSegments	Segment[]	Collection of message segments, based on type. Segments can be one or more of the following: <ul style="list-style-type: none"> <li>• <a href="#">Message Segment: Field Change</a></li> <li>• <a href="#">Message Segment: Field Change Name</a></li> <li>• <a href="#">Message Segment: Field Change Value</a></li> <li>• <a href="#">Message Segment: Hashtag</a></li> <li>• <a href="#">Message Segment: Link</a></li> <li>• <a href="#">Message Segment: Mention</a></li> <li>• <a href="#">Message Segment: More Changes</a></li> <li>• <a href="#">Message Segment: Text</a></li> </ul>
text	String	Unformatted body of the text

## Feed Item Page

Name	Type	Description
currentPageUrl	String	URL to the first page of feed items, that is, the most recent items, 20 items on a page
items	<a href="#">Feed Item</a> []	Collection of one or more feed items
nextPageUrl	String	URL to the next page of feed items, that is, older items, 20 items on a page

## Like

Name	Type	Description
id	String	18 character String ID
url	String	URL to this like
user	<a href="#">User Summary</a>	Information about the user who liked this item

## Like Page

Name	Type	Description
currentPageUrl	String	URL to the current page of likes, 10 likes to a page
likes	<a href="#">Like</a> []	Collection of likes

Name	Type	Description
nextPageUrl	String	URL to the next page of likes, 10 likes to a page
previousPageUrl	String	URL to the previous page of likes, 10 likes to a page
total	Integer	Total number of likes for the item

## Group

Name	Type	Description
description	String	Text-only description of the group
id	String	18 character String ID
memberCount	Integer	Total number of group members
name	String	Name of the group
photo	<a href="#">Photo</a>	Information about the group photo
url	String	Relative URL string to the group
visibility	String	Values are: <ul style="list-style-type: none"> <li>PRIVATE_ACCESS</li> <li>PUBLIC_ACCESS</li> </ul>

## Group Member

Name	Type	Description
id	String	18 character String ID
role	String	Values are: <ul style="list-style-type: none"> <li>GROUP_OWNER</li> <li>GROUP_MANAGER</li> <li>STANDARD_MEMBER</li> </ul>
member	<a href="#">User Summary</a>	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	<a href="#">Group Member[]</a>	Collection of group members
nextPageUrl	String	URL to the next page of members
previousPageUrl	String	URL to the previous page of members

Name	Type	Description
total	Integer	Total number of group members

## Group Page

Name	Type	Description
currentPageUrl	String	URL to this page of groups
groups	<a href="#">Group</a> []	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

## Message Segment: Field Change

Name	Type	Description
segments	Segments[]	Collection of message segments, based on type. Segments can be one or more of the following: <ul style="list-style-type: none"> <li><a href="#">Message Segment: Field Change</a></li> <li><a href="#">Message Segment: Field Change Name</a></li> <li><a href="#">Message Segment: Field Change Value</a></li> <li><a href="#">Message Segment: Hashtag</a></li> <li><a href="#">Message Segment: Link</a></li> <li><a href="#">Message Segment: Mention</a></li> <li><a href="#">Message Segment: More Changes</a></li> <li><a href="#">Message Segment: Text</a></li> </ul>
text	String	Formatted text of the change. For example: <pre>"City from San Francisco to New Orleans"</pre>
type	String	Value is FieldChange

## Message Segment: Field Change Name

Name	Type	Description
text	String	Formatted text of the change. Possible values include: <ul style="list-style-type: none"> <li>the old name of the field</li> <li>the new name of the field</li> </ul>

Name	Type	Description
type	String	Value is <code>FieldChangeName</code>

### Message Segment: Field Change Value

Name	Type	Description
text	String	Formatted text of the change. Possible values include: <ul style="list-style-type: none"> <li>the words a blank value</li> <li>the old value of the field</li> <li>the new value of the field</li> </ul>
type	String	Value is <code>FieldChangeValue</code>
url	String	Relative URL to the feed-item or null

### Message Segment: Hashtag

Name	Type	Description
tag	String	Text of the hashtag without the hashtag
text	String	Text of the hashtag with the hashtag
type	String	Value is <code>Hashtag</code>
url	String	Relative URL to the feed-item or null

### Message Segment: Link

Name	Type	Description
text	String	Formatted text of the feed item of this segment
type	String	Value is <code>Link</code>
url	String	Relative URL to feed item

### Message Segment: Mention

Name	Type	Description
access	Boolean	Specifies if current user has access to the mentioned user; <code>true</code> if the user has access, <code>false</code> otherwise
name	String	Name of the user mentioned
text	String	Formatted text of the feed item of this segment

Name	Type	Description
type	String	Value is <code>Mention</code>
user	<a href="#">User Summary</a>	Information about the mentioned user

### Message Segment: 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"
type	String	Value is <code>MoreChanges</code>

### Message Segment: Text

Name	Type	Description
text	String	Formatted text associated with the item. Possible values include: <ul style="list-style-type: none"> <li>formatted text of a comment</li> <li>the name of the field that has been changed with the word <code>from</code></li> </ul>
type	String	Value is <code>Text</code>

### Phone Numbers

Name	Type	Description
number	String	Phone number
type	String	Values are: <ul style="list-style-type: none"> <li>FAX</li> <li>MOBILE</li> <li>WORK</li> </ul>

### Photo

Name	Type	Description
smallPhotoUrl	String	Absolute URL to the small profile picture
largePhotoUrl	String	Absolute URL to the large profile picture



## Record Summary

Name	Type	Description
firstName	String	Returns <code>null</code> for record types other than user
lastName	String	Returns <code>null</code> for record types other than user
id	String	18 character String ID
name	String	Formatted display name
type	String	Type of record, such as <code>User</code> , <code>Account</code> , <code>ContentDocument</code> , and so on
url	String	Relative URL to this record

## Subscription

Name	Type	Description
id	String	18 character String ID
subject	<a href="#">Record Summary</a>	Information about the parent, that is, the thing or person being followed.
subscriber	<a href="#">Record Summary</a>	Information about the subscriber, that is, the thing or person following this item
url	String	Relative URL to this specific subscription

## Subscription Page

Name	Type	Description
subscriptions	<a href="#">Subscription</a> []	Collection of subscriptions
total	Integer	Total number of subscriptions

## Supplemental Properties

Name	Type	Description
data	Data collection []	Collection of data. Format is one of the following: <ul style="list-style-type: none"> <li><a href="#">Data: Content</a></li> <li><a href="#">Data: Link</a></li> </ul>
type	String	Values are one of the following: <ul style="list-style-type: none"> <li><code>content</code></li> <li><code>link</code></li> </ul>

## User Detail

Name	Type	Description
aboutMe	String	Text from user's profile
address	<a href="#">Address</a>	Address
currentStatus	<a href="#">User Status</a>	User's current status
email	String	Users's email address
firstName	String	User's first name
id	String	18 character String ID
isActive	Boolean	true if user is active, false otherwise
lastName	String	User's last name
name	String	Locale-based concatenation of user's first and last names
managerId	String	18 character String ID
managerName	String	Locale-based concatenation of manager's first and last names
phoneNumbers	<a href="#">Phone Numbers</a> []	Collection of user's phone numbers.
photo	<a href="#">Photo</a> on page 54	Information about the user's photos
title	String	Title of the user
url	String	URL to the user's Chatter profile

## User Status

Name	Type	Description
body	<a href="#">Feed Body</a> []	Body of the post
parentId	String	18 character String ID
url	String	URL to this user status post

## User Summary

Name	Type	Description
firstName	String	User's first name
id	String	18 character String ID
lastName	String	User's last name
name	String	Locale-based concatenation of first and last names
photo	<a href="#">Photo</a>	Information about the user's photos
title	String	Title of the user

Name	Type	Description
Url	String	URL to the user's Chatter profile

## 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 <code>userId</code> , <code>feedItemId</code> , or <code>groupId</code> being incorrect.
401	The session ID or OAuth token used has expired or is invalid. The response body contains the message and <code>errorCode</code> .
403	The request has been refused. Verify that the context user has the appropriate permissions to access the requested data.
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.

## 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" : "Account 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

## C

Chatter API 3  
     comments resource 19  
     cURL 8  
     feed-items 20  
     follow a record 43  
     how to 15  
     likes resources 27  
     quick reference 15  
     subscriptions resource 34  
     using 4  
 Chatter API resources 4  
 Comments resource 19  
 cURL 9, 10

## E

Error responses 57

## F

Feed-items resource 20  
 Feeds  
     group-memberships resource 24  
     groups resource 25  
     news resource 28  
     record resource 30  
     to resource 35  
     user-profile resource 39  
     users resource 41  
 Feeds resources  
     news 28  
     record 30  
     to 35  
     user-profile 39  
 Following records 43

## G

Group-memberships resources 24  
 Groups resources 25

## J

JSON 9, 10

## L

Likes resources 27

## N

News feeds resource 28

## O

OAuth  
     access token 9  
 OAuth authentication 11  
 Organization 10

## Q

Quickstart  
     access token 10  
     Developer Edition 10  
     OAuth 10  
     prerequisites 10

## R

Record resource 30  
 Resources  
     comments 19  
     error responses 57  
     likes 27  
     relationships 14  
     returns 47  
     subscriptions 34  
 REST resources  
     Describe Global Resource 9  
     Discovery Resource 9  
     PATCH 9  
     Query 9  
     SObject Basic Information 9  
     SObject Describe 9  
     SObject Row 9  
     Versions 9

## S

Subscriptions resource 34

## T

To Me resource 35  
 To resource 35

## U

URI  
     base URI 9, 10  
 User-profile resource 39  
 Users resources 41

