Salesforce Connect API Basics

Salesforce Connect API provides a powerful way to access various external data sources from within Salesforce. It's a part of the broader set of Salesforce APIs that allow developers to create rich, integrated experiences.

Table of Contents

Introduction

The Connect API (formerly known as Chatter REST API) allows developers to access Chatter feeds and social data such as users, groups, and topics, as well as integrate with Salesforce records. It enables the creation of custom user experiences that leverage the social and collaborative aspects of the Salesforce platform.

Connect API Features

Feature	Description
Feed Items	Access and interact with feed items in a user's Chatter feed.
Groups	Create, read, update, and delete Chatter groups.
Users	Retrieve and manage user information.
Topics	Work with topics, including associating records with topics.
Files	Share and manage files within Chatter.

Using Connect API

To use the Connect API, you must authenticate with Salesforce using OAuth 2.0. After obtaining an access token, you can make HTTP requests to the API endpoints to interact with Salesforce data.

Authentication

```
POST /services/oauth2/token HTTP/1.1
Host: login.salesforce.com
Content-Type: application/x-www-form-urlencoded

grant_type=password&client_id=YOUR_CONSUMER_KEY&client_secret=YOUR_CONSUMER_SECRET&userr
```

Headers

Include the access token in the Authorization header for all API requests.

```
Authorization: Bearer YOUR_ACCESS_TOKEN
```

Code Example

The following example demonstrates how to post a message to a user's feed using the Connect API.

```
HttpRequest req = new HttpRequest();
req.setEndpoint('https://yourInstance.salesforce.com/services/data/vXX.0/chatter/feeds/r
items');
req.setMethod('POST');
req.setHeader('Content-Type', 'application/json');
req.setHeader('Authorization', 'Bearer ' + accessToken);

String body = '{"body" : {"messageSegments" : [{"type": "Text", "text" : "Hello from
the Connect API!"}]}}';
req.setBody(body);

Http http = new Http();
HTTPResponse res = http.send(req);
System.debug(res.getBody());
```

Best Practices

- **Handle Pagination**: When querying large sets of data, use pagination to manage the volume of records.
- **Rate Limits**: Be mindful of API rate limits and design your application to stay within these limits.
- **Secure Data**: Ensure that your application follows security best practices when handling sensitive data.
- **Error Handling**: Implement robust error handling to manage API exceptions and provide meaningful feedback to users.

Additional Resources

- Connect API Developer Guide
- Salesforce Developers Documentation
- Trailhead: Connect REST API

This markdown file provides a basic introduction to the Salesforce Connect API, highlighting its key features and providing a simple code example to get started.