Salesforce Bulk API(Postman)

In our previous article, we have seen the step-by-step procedure on "How to integrate Salesforce using REST API". In this article, we are going to see how to use Salesforce Bulk API" when there is a need for loading or deleting large sets of data.

Bulk API is based on REST principles and is developed for loading or deleting large sets of data. It is used to INSERT, UPDATE, UPSERT, DELETE and QUERY records from Salesforce asynchronously. Salesforce provides "Connected App" to connect with the platform with any other application.

Salesforce Connected App

"Connected App" is an application which connects Salesforce org with an external application. Connected Apps use standard SAML and OAuth protocols to authenticate, provide single sign-on, and provide tokens to use with Salesforce APIs.

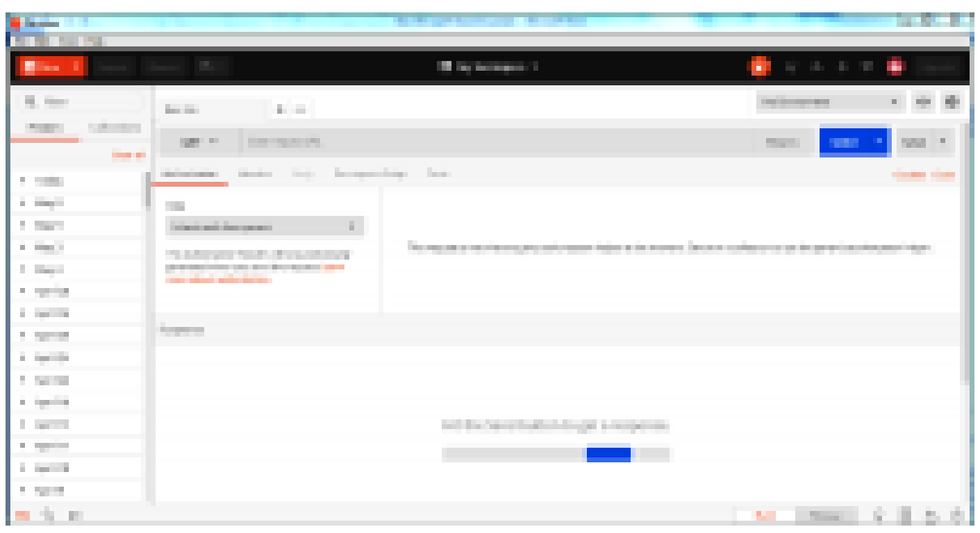
Create a **Salesforce connected App** to get "Consumer Key " and "Consumer secret", as you need these details to authenticate the external application. If you are new to Connected App, go through our previous article which has steps to create Connected App.

Postman

Postman is an application for interacting with HTTP APIs. It has powerful testing features and user friendly **GUI** for making requests and reading responses.

Getting started with Postman

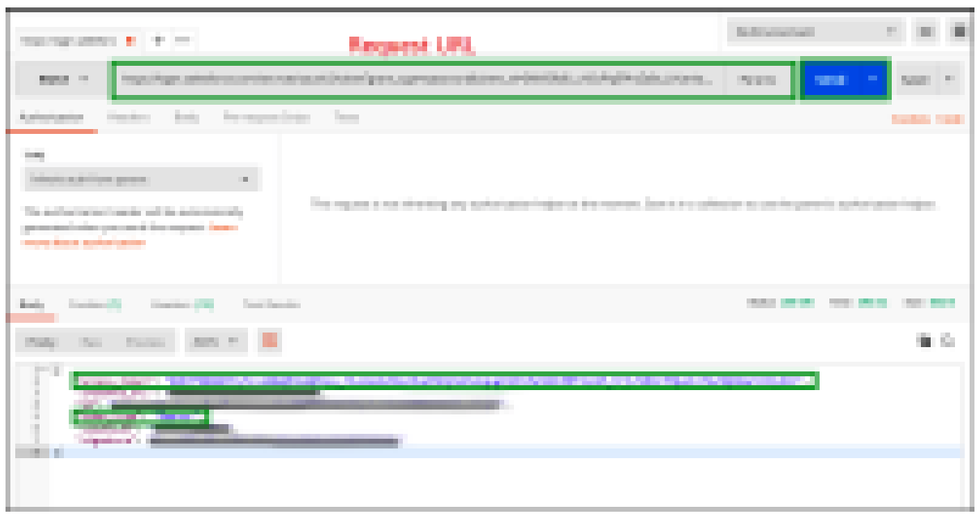
- Install Postman or it is also available as Add-On for Google Chrome, go to Google Chrome webstore->search for **Postman** ->Add it to chrome.
- Launch Postman, below screen will be shown



- Setting up the HTTP login request URL
- We are going to set request URL by (Base URL + Parameters), below is the structure to construct request URL for Salesforce org
- https://login.salesforce.com/services/oauth2/token?grant_type=password&client_**id=YourConsumerKey**&client_secret=**YourConsumerSecret**&username=SalesforceUserName&password=**SalesforcePassword**

- In the above URL replace **highlighted** text with
  - **YourConsumerKey:** Consumer Key obtained from Connected App.
  - **YourConsumerSecret:** Consumer Secret obtained from Connected App.
  - **SalesforceUserName:** Salesforce User ID.
  - **SalesforcePassword:** password

- Create a **POST** method and Copy the request URL as shown below and click on send.



- On successful login, you will get Instance-URL, Access token, Token-type. This Access token is further used as a parameter value in Header to send any HTTP requests and get the response from Salesforce. So, make a note of Access token.

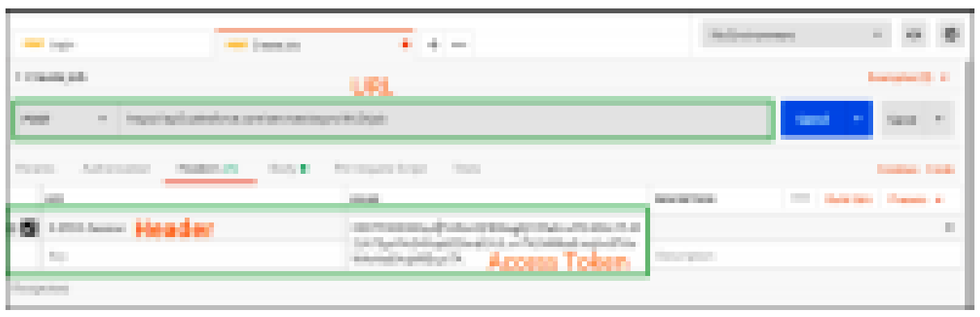
Getting Started with Bulk API

In bulk API a set of records are processed by creating a **Job** and that contains **Batches(One or More)**. Job specifies the type of operation and Object used to process records. A batch is a set of records. Following are the basic steps to process the data

- Create a new Job.
- Add one or more batches to the created Job.
- Close the Job.
- Get the result from the batch.

Steps to Create Job

- Now we will see how to create Job using **POST** Below is the Sample URL.
  https:// **instanceurl** /services/async/**API_Version**/job
- In the above URL replace **highlighted** text with
  - **Instanceurl:** Instanceurl obtained in **#4.**
  - **API_version:** Salesforce API version.
- Create a POST method, copy the URL and copy the Access Token in header section as shown below.



- In the body section provide the details of Job either in XML or JSON format. Here I'm providing job details in XML format. Following is the structure of XML to create Job.

```
<?xml version="1.0" encoding="UTF-8"?>
<jobInfo xmlns="http://www.force.com/2009/06/asyncapi/dataload">
        <operation>insert</operation>
        <object>Account</object>
        <contentType>XML</contentType>
</jobInfo>
```

- In the above structure **operation**, **object** and **contectType** are the three mandatory nodes to create a job.
  - **Operation**: operation can have only insert, query , queryall , upsert, update, delete and hardDelete as values . All the values should be in lower case. Here I've selected **insert**
  - **Object**: Any standard or custom object name. Here I've selected **Account**
  - **Content Type**: XML, JSON or CSV. Here I've selected **XML**
- Click on **send** In the response, you will get the Job Id if the job is created successfully else you will get the error message. Following is the response from salesforce.

```
<?xml version="1.0" encoding="UTF-8"?>
<jobInfo
   xmlns="http://www.force.com/2009/06/asyncapi/dataload">
    <id>7507F000008Q3hRQAS</id>
    <operation>insert</operation>
    <object>Account</object>
    <createdById>0057F000000IWSkQAO</createdById>
    <createdDate>2018-11-13T07:42:49.000Z</createdDate>
    <systemModstamp>2018-11-13T07:42:49.000Z</systemModstamp>
    <state>Open</state>
    <concurrencyMode>Parallel</concurrencyMode>
    <contentType>XML</contentType>
    <numberBatchesQueued>0</numberBatchesQueued>
    <numberBatchesInProgress>0</numberBatchesInProgress>
    <numberBatchesCompleted>0</numberBatchesCompleted>
    <numberBatchesFailed>0</numberBatchesFailed>
    <numberBatchesTotal>0</numberBatchesTotal>
    <numberRecordsProcessed>0</numberRecordsProcessed>
    <numberRetries>0</numberRetries>
    <apiVersion>44.0</apiVersion>
    <numberRecordsFailed>0</numberRecordsFailed>
    <totalProcessingTime>0</totalProcessingTime>
    <apiActiveProcessingTime>0</apiActiveProcessingTime>
    <apexProcessingTime>0</apexProcessingTime>
</jobInfo>
```

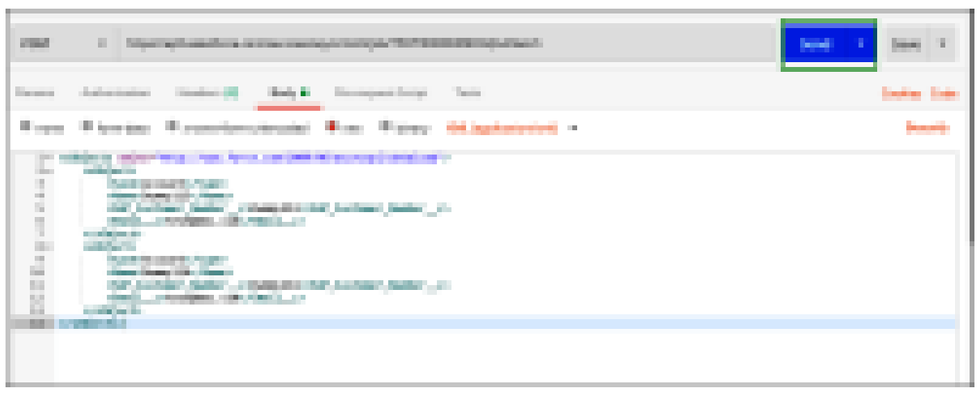
- In the above response, **id** node contains the JobId and **state** node denotes the status of the Job.
- Make a note of this Job id as you need this id while adding **Batches** to the job**.**

Steps to Add Batches for a Job

- Now we will see how to add **Batch** for above created **Job** using **POST** Below is the Sample URL.
  - https:// **instanceurl** /services/async/**API_Version**/job /**Job_Id**/batch
- In the above URL replace **highlighted** text with
  - **Instanceurl:** Instanceurl obtained in **#4.**
  - **API_version:** Salesforce API version.
  - **Job_Id:** Job Id obtained in **#11.**
- Create a **POST** method, copy the URL and copy the Access Token in header section. In the body, provide the details of records either in XML or JSON format based on the content type selected while creating the Job. Here I'm providing batch details in XML format. Following is the structure of XML to create batch.

```
<sObjects xmlns="http://www.force.com/2009/06/asyncapi/dataload">
        <sObject>
```

```
                <type>Account</type>
                <Name>Dummy113</Name>
                <SAP_Customer_Number__c>Dummy113</SAP_Customer_Number__c>
                <Email__c>test@abc.com</Email__c>
        </sObject>
        <sObject>
                <type>Account</type>
                <Name>Dummy114</Name>
                <SAP_Customer_Number__c>Dummy114</SAP_Customer_Number__c>
                <Email__c>test@abc.com</Email__c>
        </sObject>
</sObjects>
```



- Click on **Send** In the response, you will get the batch Id along with Job Id if the batch is created successfully else you will get the error message. Following is the batch response from salesforce.

```
<?xml version="1.0" encoding="UTF-8"?>
<batchInfo   xmlns="http://www.force.com/2009/06/asyncapi/dataload">
    <id>7517F00000BUPNOQA5</id>
    <jobId>7507F000008Q3hRQAS</jobId>
    <state>Queued</state>
    <createdDate>2018-11-13T12:51:10.000Z</createdDate>
    <systemModstamp>2018-11-13T12:51:10.000Z</systemModstamp>
    <numberRecordsProcessed>0</numberRecordsProcessed>
    <numberRecordsFailed>0</numberRecordsFailed>
    <totalProcessingTime>0</totalProcessingTime>
    <apiActiveProcessingTime>0</apiActiveProcessingTime>
    <apexProcessingTime>0</apexProcessingTime>
</batchInfo>
```

- In the above response, **id** node contains the batchId and **state** node denotes the status of the batch.
- Make a note of this batch id as you need this id while fetching the result from salesforce

Steps to close the Job:

- Now we will see how to close the above created **Job** using **POST** method. Below is the Sample URL.
  - https:// **instanceurl** /services/async/**API_Version**/job /**Job_Id**
- In the above URL replace **highlighted** text with
  - **Instanceurl:** Instanceurl obtained in **#4.**
  - **API_version:** Salesforce API version.
  - **Job_Id:** Job Id obtained in **#11.**

- Create a POST method, copy the URL and copy the Access Token in header section. In the body section, add the following structure to close the job.

```
<?xml version="1.0" encoding="UTF-8"?>
<jobInfo   xmlns="http://www.force.com/2009/06/asyncapi/dataload">
 <state>Closed</state>
</jobInfo>
```

- Click on send button. The following response will be shown.

```
<?xml version="1.0" encoding="UTF-8"?>
<jobInfo
   xmlns="http://www.force.com/2009/06/asyncapi/dataload">
    <id>7507F000008Q3hRQAS</id>
    <operation>insert</operation>
    <object>Account</object>
    <createdById>0057F000000IWSkQAO</createdById>
    <createdDate>2018-11-13T07:42:49.000Z</createdDate>
    <systemModstamp>2018-11-13T07:42:49.000Z</systemModstamp>
    <state>Closed</state>
    <concurrencyMode>Parallel</concurrencyMode>
    <contentType>XML</contentType>
    <numberBatchesQueued>0</numberBatchesQueued>
    <numberBatchesInProgress>0</numberBatchesInProgress>
    <numberBatchesCompleted>1</numberBatchesCompleted>
    <numberBatchesFailed>0</numberBatchesFailed>
    <numberBatchesTotal>1</numberBatchesTotal>
    <numberRecordsProcessed>2</numberRecordsProcessed>
    <numberRetries>0</numberRetries>
    <apiVersion>44.0</apiVersion>
    <numberRecordsFailed>0</numberRecordsFailed>
    <totalProcessingTime>581</totalProcessingTime>
    <apiActiveProcessingTime>474</apiActiveProcessingTime>
    <apexProcessingTime>182</apexProcessingTime>
</jobInfo>
```

- In the above response state node represents that the Job is closed.

Steps to get the Batch Result

- Now we will see how to get the batch result from Salesforce using **GET** Below is the Sample URL.
  - https:// **instanceurl** /services/async/**API_Version**/job /**Job_Id**/batch/**Batch_Id**/result
- In the above URL replace **highlighted** text with
  - **Instanceurl:** Instanceurl obtained in **#4.**
  - **API_version:** Salesforce API version.
  - **Job_Id:** Job Id obtained in **#11.**
  - **Batch_Id:** batch id for which you want to fetch the result.

- Create a **GET** method, copy the URL and copy the Access Token in header section. The following is the batch result received from salesforce.

```
<?xml version="1.0" encoding="UTF-8"?>
<results xmlns="http://www.force.com/2009/06/asyncapi/dataload">
    <result>
        <id>0017F00001LeT2OQAV</id>
        <success>true</success>
        <created>true</created>
    </result>
    <result>
```

```
        <id>0017F00001LeT2PQAV</id>
        <success>true</success>
        <created>true</created>
    </result>
</results>
```

Limitations of Bulk API

Following are the limitations of Salesforce Bulk APIs.

- Batches for data loads can consist of a single CSV, XML, or JSON file that is no larger than 10 MB.
- A batch can contain a maximum of 10,000 records.
- A batch can contain a maximum of 10,000,000 characters for all the data in a batch.
- A field can contain a maximum of 32,000 characters.
- A record can contain a maximum of 5,000 fields.
- A record can contain a maximum of 400,000 characters for all its fields.
- A batch must contain some content or an error occurs.