

SFDC REST API - AE WS 3.0

SFDC -

User needs Security Token (Sent by email)

Create 'Connected App' in Salesforce

To authenticate using OAuth, you must create a connected app that defines your application's OAuth settings for the Salesforce organization.

When you develop an external application that needs to authenticate with Salesforce, you need to define it as a new connected app within the Salesforce organization that informs Salesforce of this new authentication entry point.

- Use the following steps to create a new connected app.
- From Setup, click Create | Apps and click New to start defining a connected app.
- Enter the name of your application.
- Enter the contact email information, as well as any other information appropriate for your application.
- Select Enable OAuth Settings.
- Enter a Callback URL. Depending on which OAuth flow you use, this is typically the URL that a user's browser is redirected to after successful authentication. As this URL is used for some OAuth flows to pass an access token, the URL must use secure HTTP (HTTPS) or a custom URI scheme.
- Add all supported OAuth scopes to Selected OAuth Scopes. These scopes refer to permissions given by the user running the connected app.
- Enter a URL for Info URL. This is where the user can go for more information about your application.
- Click Save. The Consumer Key is created and displayed, and the Consumer Secret is created (click the link to reveal it).

AE

First you need to get an Authorization Token

Create REST CONN object - CONN.WS.SFDC_TOKEN

- General Tab
 - URL Endpoint - <https://login.salesforce.com/services/oauth2/token>
 - Authentication = None

Create REST JOBS object - JOBS.WS.REST.SFDC_TOKEN

- General Tab
 - Connection = CONN.WS.SFDC_TOKEN
 - Check Write Response Report
- Definition Tab

- Resource = Leave Empty
- Method = POST
- Media Type = application/x-www-form-urlencoded
- Headers & Query Parameters
 - Query Parameters

Parameter	Value
grant_type	password
client_id	<Consumer Key from Connected App>
client_secret	<Consumer Secret from Connected App>
username	<users login (usually an email address)>
password	<users password + users security token>

- Save and Run the job.
- In the Open the 'restResp_<runID>' on the Directory tab of the Job Report.
 - The JSON response should look like:

```
{
  "id": "https://login.salesforce.com/id/00Di0000000jkEsEAI/005i0000002oXHaaAM",
  "issued_at": "1421186788690",
  "token_type": "Bearer",
  "instance_url": "https://na15.salesforce.com",
  "signature": "WYc2Z8Og5VpgWCtjUcsoVMlpMB62b+AMrD8HbmLymTo=",
  "access_token": "00Di0000000jkEs!
  AQQAQPcXHoVopfjhjBG425A5hosy2ehyUGEy3v2pLzfZXeuBWJaGp4luEqWqRN9QaHSFixzdUgXLMyl
  tAhkMdhAKxxe7.pdqB"
}
```

The **instance_url** will be the base of all your URL endpoints.
 The **access_token** will be used in the HTTP Header for authorization.

***** Access Token expires so needs to be refreshed every day**

List of standard Resources areas

Name	URI
/subjects	/subjects
/query	/query
/search	/search
/chatter	/chatter

SFDC documentation of REST API References:

http://www.salesforce.com/us/developer/docs/api_rest/Content/resources_list.htm

Sample Scenario

All Contacts should have a Title populated for them. We need to:

- List the Fields and values for a Contact
- Update 'Title' field if empty

Create Connection object with the Resource URI related to Contacts -
CONN.WS.SFDC_CONTACTS

- General Tab
 - URL Endpoint = `https://<instance_url>/services/data/v20.0`
 - Authentication = None
- Resources Tab

Name	URI
Contact	/subjects/Contact
Contact/describe	/subjects/Contact/describe
Contact/{ID}	/subjects/Contact/&id#

- Headers & Query Parameters Tab
 - HTTP Headers

Header	Value
Authorization	OAuth <access_token>

List all the Fields and Values for a specific Contact

Create REST JOBS object - JOBS.WS.REST.SFDC.GET_CONTACT_INFO

- General Tab
 - Connection = CONN.WS.SFDC_CONTACTS
 - Check Write Response Report
- Definition Tab
 - Resource = `select "/subjects/Contact/&id#"`
 - Method = GET

The &id# can be populated in multiple ways, script variable, object variable, or promptset variable.

For this example we'll define the variable and value on the 'Variable and Prompts' tab of the job.

Key	Value
&id#	<contact id>

The Contact Id is found as part of the URL when you view the Contact in Salesforce. For example, if the URL is:
<https://eu2.salesforce.com/003b000000clNpx>

The Contact Id would be **003b000000clNpx**

Run the Job and look at the Response text file in the Job Report.