

Salesforce Sales Cloud - How to Test Revoking a Salesforce OAuth Token and Re-Authenticating Your Instances Through API

Last Modified on 06/12/2019 9:16 pm EDT

In this article, we will discuss how, if you are setting up a framework to re-authenticate instances of Salesforce Sales Cloud (SFDC), you can know how you can test that your process was successful. We will look into how to revoke an SFDC OAuth token and we will also go over a scenario that you can run through to test and ensure that your re-authentication process through API call is successful.

1. The first step is to access your SFDC account and create an OAuth/Connected App that has scopes of both 'full' and 'refresh_token, offline_access'. It is also important to pay attention to the 'Refresh Token Policy' that you set on your Connect App because that can impact your ability to re-authenticate with this described process as well.

Lightning Experience Migration Assistant

Get Started

Connected App Name: **Brody_App**

« Back to List: Custom Apps

Buttons: Edit, Delete, Manage

Warning: Allow from 2-10 minutes for your changes to take effect on the server before using the connected app.

Version	1.0
API Name	Brody_App
Created Date	2/6/2018 10:31 AM
By	Brody Taylor
Contact Email	[Redacted]
Contact Phone	[Redacted]
Last Modified Date	2/7/2018 10:06 PM
By	Brody Taylor
Description	
Info URL	

API (Enable OAuth Settings)

Consumer Key	[Redacted]	Consumer Secret	Click to reveal
Selected OAuth Scopes	Full access (full) Perform requests on your behalf at any time (refresh_token, offline_access)	Callback URL	https://auth.cloudelements.io/oauth
Enable for Device Flow	<input type="checkbox"/>	Require Secret for Web Server Flow	<input checked="" type="checkbox"/>
Token Valid for	0 Hour(s)	Include Custom Attributes	<input type="checkbox"/>
Include	<input type="checkbox"/>		

2. Next authenticate an instance through API. The required calls and payloads are already discussed in the developer docs .

3. Access your SAP Cloud Platform Open Connectors account, locate your new SFDC instance, and copy the auth header (user/org/element tokens) from the instance. For instance you can copy the full Authorization header from any endpoint after selecting your instance.

reports

GET /reports Retrieve a list of reports

Parameters Cancel

Name	Description
Authorization * required string (header)	The authorization tokens. The format for the header value is 'Element <token>, User <user secret>' User 3b5Unr3Jm7HChvz29Y3 [Redacted]

Execute

4. Now revoke the user's OAuth token either through the SFDC interface or through an API call to SFDC directly.

5. a) Revoke through the SFDC website by accessing: Setup -> Manage Apps -> Connected Apps OAuth Usage -> User Count.

- <https://{SFDC-Instance}.salesforce.com/services/oauth2/revoke?token=#####>

- https://help.salesforce.com/articleView?id=remoteaccess_revoke_token.htm&type=5

SFDC_RevokeToken Examples (0)

POST <https://na54.salesforce.com/services/oauth2/revoke?token={yourOauthToken}> Params Send Save

Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> token	{yourOauthToken}			
New key	Value	Description		

Authorization Headers (2) Body Pre-request Script Tests Cookies Code

Key	Value	Description	...	Bulk Edit	Presets
<input checked="" type="checkbox"/> Authorization	Bearer {YourBearerToken}				
<input checked="" type="checkbox"/> Content-Type	application/json				
New key	Value	Description			

7. Confirm that the SFDC instance is no longer authenticated.

Code Details

401 Error: Unauthorized

Response body

```
{  "requestId": "5a7beb7de4b06fa3c9f33aa5",  "message": "Session expired or invalid"}
```

Response headers

```
content-type: application/json; charset=UTF-8
```

8. Make an API call to GET /oauth/url just as you did when you initially provisioned your instance of SFDC.

GET <https://staging.cloud-elements.com/elements/api-v2/elements/23/oauth/url?callbackUrl=https://auth.cloudeleme..> Params Send Save

Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> callbackUrl	https://auth.cloudelements.io/oauth			
<input checked="" type="checkbox"/> apiKey	[REDACTED]			
<input checked="" type="checkbox"/> apiSecret	[REDACTED]			
New key	Value	Description		

Authorization Headers (2) Body Pre-request Script Tests Cookies Code

Key	Value	Description	...	Bulk Edit	Presets
<input checked="" type="checkbox"/> Authorization	User 3b5Unr3jm7HChvh...				
<input checked="" type="checkbox"/> Content-Type	application/json				
New key	Value	Description			

9. Access the OAuth URL generated, login to SFDC, and retrieve the provider code.

10. Make an API call to PATCH /instances with the full SAP Cloud Platform Open Connectors auth header retrieved previously, blank oauth.user.refresh_token and oauth.user.token values, and the providerData.code value in the body.

The screenshot displays a REST client interface for a PATCH request to the URL `https://staging.cloud-elements.com/elements/api-v2/instances`. The interface includes a "Send" button and a "Save" button. The "Headers (2)" tab is selected, showing the following headers:

Key	Value	Description
<input checked="" type="checkbox"/> Authorization	User 3b5Unr3jm7HChvhz...	
<input checked="" type="checkbox"/> Content-Type	application/json	
<input type="text" value="New key"/>	<input type="text" value="Value"/>	<input type="text" value="Description"/>

The "Body" tab is also active, showing the request body in JSON format:

```
1 {
2   "configuration": {
3     "oauth.user.refresh_token": "",
4     "oauth.user.token": ""
5   },
6   "providerData": {
7     "code": "..."
8   }
9 }
```

11. Confirm a successful response, and verify that the SFDC instance was successfully re-authenticated.

Note that there can also be scenarios where re-authentication through API must be made via a PUT /instances containing the full instance configuration as opposed to a PATCH /instances. This can be necessary if the token has expired or if the OAuth token is revoked through alternate means than what was discussed in this article.