

# Salesforce Sales Cloud Authenticate a Connector Instance

Last Modified on 02/17/2022 7:22 am EST

You can authenticate with Salesforce to create your own instance of the Salesforce Sales Cloud connector through the UI or through APIs. Once authenticated, you can use the connector instance to access the different functionality offered by the Salesforce platform.

Note the following when authenticating instances:

- The UI uses default application permissions when asking for access to your Salesforce applications. If you would like custom application permissions, you will need to create your own connected app and configure the application access in your custom UI. Follow the instructions in [Salesforce Sales Cloud API Provider Setup](#).
- If you create a custom application, you will need to create instances via the APIs to use this new app. You only need to create this application once and you can use it for all of your customers.
- Once an instance is created we automatically refresh the token behind the scenes so that you will never have to connect your application again.

**Note:** If you are authenticating with **OAuth2Password** and Salesforce has provided a security token, then the security token must be appended to the user's password. For example, if a user's password is mypassword, and their security token is XXXXXXXXXXXX, then the value provided for this parameter must be mypasswordXXXXXXXXXX. If a user is using MFA, then the MFA code can be appended to the password with respective settings enabled in Salesforce.

## Authenticate Through the UI

Use the UI to authenticate with Salesforce and create a connector instance. Because you authenticate with Salesforce via OAuth 2.0, all you need to do is add a name for the instance. After you create the instance, you'll log in to Salesforce to authorize SAP Open Connectors access to your account. For more information about authenticating a connector instance, see [Authenticate a Connector Instance \(UI\)](#).

After successfully authenticating, we give you several options for next steps. [Make requests using the API docs](#) associated with the instance, [map the instance to a common resource](#), or [use it in a formula template](#).

## Authenticate Through API

Authenticating through API is a multi-step process that involves:

1  
Redirect URL



2

Authenticate Users

>

3

Authenticate Instance

- [Getting a redirect URL](#). This URL sends users to the vendor to log in to their account.
- [Authenticating users and receiving the authorization grant code](#). After the user logs in, the vendor makes a callback to the specified url with an authorization grant code.
- [Authenticating the connector instance](#). Using the authorization code from the vendor, authenticate with the vendor to create a connector instance at SAP Open Connectors.

## Getting a Redirect URL

1

Redirect URL

>

2

Authenticate Users

>

3

Authenticate Instance

Use the following API call to request a redirect URL where the user can authenticate with the vendor. Replace `{keyOrId}` with the connector key, `sfdc`.

```
GET /elements/{keyOrId}/oauth/url?apiKey=&apiSecret=&callbackUrl=&siteAddress=
```

### Query Parameters

Query Parameter	Description
apiKey	The key obtained from registering your app with the provider. This is the Consumer Key that you noted at the end of <a href="#">Salesforce Sales Cloud API Provider Setup</a> .
apiSecret	The secret obtained from registering your app with the provider. This is the Consumer Secret that you noted at the end of <a href="#">Salesforce Sales Cloud API Provider Setup</a> .
callbackUrl	The URL that will receive the code from the vendor to be used to create a connector instance. This is the Callback URL that you noted at the end of <a href="#">Salesforce Sales Cloud API Provider Setup</a> .
siteAddress	<i>Optional</i> If using a sandbox account, specify a site address of <a href="http://test.salesforce.com">http://test.salesforce.com</a> . Otherwise, the default <a href="http://login.salesforce.com">http://login.salesforce.com</a> is used.

## Example cURL

```
curl -X GET
-H 'Content-Type: application/json'
'https://api.openconnectors.us2.ext.hana.ondemand.com/elements/api-v2/elements/sfdc/oauth?url?apiKey=fake_salesforce_api_key&apiSecret=fake_salesforce_api_secret&callbackUrl=https://www.mycoolapp.com/auth&state=sfdc'
```

## Example Response

Use the `oauthUrl` in the response to allow users to authenticate with the vendor.

```
{
  "element": "sfdc",
  "oauthUrl": "https://login.salesforce.com/services/oauth2/authorize?response_type=code&client_id=fake_salesforce_api_key&client_secret=xyz789&scope=full%20refresh_token&redirect_uri=https://www.mycoolapp.com/auth&state=sfdc"
}
```

## Authenticating Users and Receiving the Authorization Grant Code

1  
Redirect URL

>

2  
Authenticate Users

>

3

Authenticate Instance

Provide the response from the previous step to the users. After they authenticate, the endpoint provides an authorization grant code and state as query parameters, as shown in the example below.

```
https://?code=xoz8AFqScK2ngM04kSSM&state=sfdc
```

## Query Parameters

Query Parameter	Description
code	The code required by SAP Open Connectors to retrieve the OAuth access and refresh tokens from the endpoint.
state	The connector key ( <code>sfdc</code> ).

**Note:** If the user denies authentication and/or authorization, there will be a query string parameter called `error` instead of the `code` parameter. In this case, your application can handle the error gracefully.

## Authenticating the Connector Instance

1

Redirect URL

>

2

Authenticate Users

>

3

Authenticate Instance

Use the `/instances` endpoint to authenticate with Salesforce and create a connector instance. If you are configuring events, see the [Events section](#).

**Note:** The endpoint returns a connector token upon successful completion. Retain the token for all subsequent requests involving this connector instance.

To create a connector instance:

1. Construct a JSON body as shown below (see [Parameters](#)):

```
{
  "element": {
    "key": "sfdc"
  },
  "providerData": {
    "code": ""
  },
  "configuration": {
    "oauth.callback.url": "",
    "oauth.api.key": "",
    "oauth.api.secret": "",
    "filter.response.nulls": true
  },
  "tags": [
    ""
  ],
  "name": ""
}
```

2. Call the following, including the JSON body you constructed in the previous step:

```
POST /instances
```

**Note:** Make sure that you include the User and Organization keys in the header. For more information, see [Authorization Headers](#), [Organization Secret](#), and [User Secret](#).

3. Locate the `token` in the response and save it for all future requests using the connector instance.

## Example cURL

```
curl -X POST \
  https://api.openconnectors.us2.ext.hana.ondemand.com/elements/api-v2/instances \
  -H 'authorization: User , Organization ' \
  -H 'content-type: application/json' \
  -d '{
  "element": {
    "key": "sfdc"
  },
  "providerData": {
    "code": "xoz8AFqSck2ngM04kSSM"
  },
  "configuration": {
    "oauth.callback.url": "",
    "oauth.api.key": "",
    "oauth.api.secret": ""
  },
  "tags": [
    "For Docs"
  ],
  "name": "SFDCviaAPI1"
}'
```

## Parameters

API parameters not shown in SAP Open Connectors are in `code formatting` .

**Note:** Event related parameters are described in [Events](#).

Parameter	Description	Data Type
'key'	The connector key. sfdc	string
Name <code>name</code>	The name for the connector instance created during authentication.	Body
<code>oauth.callback.url</code>	The Callback URL from Salesforce. This is the Callback URL that you noted at the end of <a href="#">Salesforce Sales Cloud API Provider Setup</a> .	
<code>oauth.api.key</code>	The Consumer Key from Salesforce. This is the Consumer Key that you noted at the end of <a href="#">Salesforce Sales Cloud API Provider Setup</a> .	string
<code>oauth.api.secret</code>	The Consumer Secret from Salesforce. This is the Consumer Secret that you noted at the end of <a href="#">Salesforce Sales Cloud API Provider Setup</a> .	string
Filter null values from the response <code>filter.response.nulls</code>	<i>Optional.</i> Determines if null values in the response JSON should be filtered from the response. Yes or <code>true</code> indicates that SAP Open Connectors will filter null values. Default: <code>true</code> .	boolean
tags	<i>Optional.</i> User-defined tags to further identify the instance.	string

## Example Response

```
{
  "id": 123,
  "name": "test",
  "token": "3sU/S/kZD36BaABPS7EAuSGHF+1wsthT+mvoukiE",
  "element": {
    "id": 39,
    "name": "Salesforce.com",
    "key": "sfdc",
    "description": "The Salesforce.com allows you to deliver revolutionary CRM automation functionality, such as account and contact creation, from anywhere, anytime, on any device.",
    "active": true,
    "deleted": false,
    "typeOauth": true,
    "trialAccount": false,
    "configDescription": "If you do not have a Salesforce.com account, you can create one at Salesforce.com Signup",
    "signupURL": "http://www.salesforce.com"
  },
  "provisionInteractions": [],
  "valid": true,
  "eventsEnabled": true,
  "disabled": false,
  "maxCacheSize": 0,
  "cacheTimeToLive": 0,
  "cachingEnabled": false
}
```