

# Infusionsoft REST Authenticate a Connector Instance

Last Modified on 03/16/2020 9:27 pm EDT

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

## Authenticate Through the UI

Use the UI to authenticate with Infusionsoft and create a connector instance. Because you authenticate with Infusionsoft 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 Infusionsoft to authorize SAP Cloud Platform 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 similar to authenticating via the UI. Instead of clicking and typing through a series of buttons, text boxes, and menus, you will instead send a request to our `/instances` endpoint. The end result is the same, though: an authenticated connector instance with a **token** and **id**.

Authenticating through API follows a multi-step OAuth 2.0 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 call back 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 Cloud Platform 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 service provider. Replace `{keyOrId}` with the connector key, `infusionsoftrest`.

```
curl -X GET /elements/{keyOrId}/oauth/url?apiKey=&apiSecret= &callbackUrl=
```

### Query Parameters

| Query Parameter | Description                                                                                                                                                                 |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| apiKey          | The API key or client ID obtained from registering your app with the provider. This is the <b>Key Name</b> that you recorded in <a href="#">API Provider Setup</a> .        |
| apiSecret       | The client secret obtained from registering your app with the API provider. This is the <b>Secret Name</b> that you recorded in <a href="#">API Provider Setup</a> .        |
| callbackUrl     | The URL that the API provider returns a user to after they authorize access. This is the <b>Callback URL Name</b> that you recorded in <a href="#">API Provider Setup</a> . |

### Example Request

```
curl -X GET \  
'https://api.openconnectors.us2.ext.hana.ondemand.com/elements/api-v2/elements/infusionsoftrest/oauth/url?apiKey=Rand0MAP1-key&apiSecret=fak3AP1-s3Cr3t&callbackUrl=https:%3A%2F%2Fwww.mycoolapp.com%2auth' \  
' \
```

### Example Response

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

```
{
  "oauthUrl": "https://signin.infusionsoft.com/app/oauth/authorize?scope=files.readwrite+offline_access&response_type=code&redirect_uri=https%3A%2F%2Fhttpbin.org%2Fget&state=infusionsoftrest&client_id=Rand0MAP1-key",
  "element": "infusionsoftrest"
}
```

## Authenticating Users and Receiving the Authorization Grant Code

1  
Redirect URL

>

2  
Authenticate Users

>

3  
Authenticate Instance

Provide the `oauthUrl` in the response from the previous step to the users. After users authenticate, Infusionsoft provides the following information in the response:

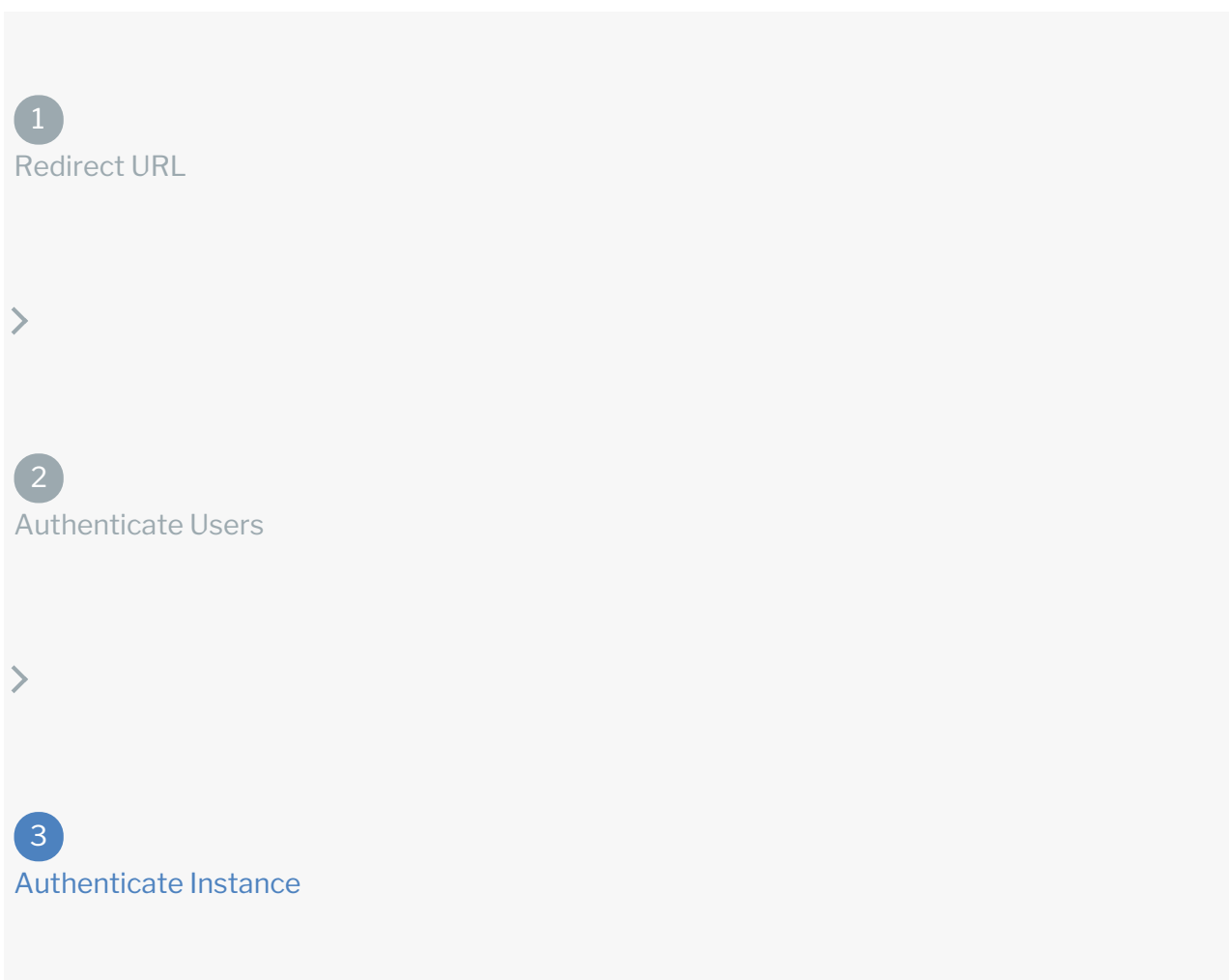
- code
- state

| Response Parameter | Description |
|--------------------|-------------|
|                    |             |

|                         |                                                                                                                                                                                                                                              |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Response code Parameter | <b>Description:</b> The authorization grant code returned from the API provider in an OAuth 2.0 authentication workflow. SAP Cloud Platform Open Connectors uses the code to retrieve the OAuth access and refresh tokens from the endpoint. |
| state                   | A customizable identifier, typically the connector key ( <code>infusionsoftrest</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



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

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

To authenticate a connector instance:

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

```
{
  "element": {
    "key": "infusionsoftrest"
  },
  "providerData": {
    "code": ""
  },
  "configuration": {
    "oauth.api.key": "",
    "oauth.api.secret": "",
    "oauth.callback.url": ""
  },
  "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` and `id` in the response and save them for all future requests using the connector instance.

## Example Request

```

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": "infusionsoftrest"
  },
  "providerData": {
    "code": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  },
  "configuration": {
    "oauth.api.key": "Rand0MAP1-key",
    "oauth.api.secret": "fak3AP1-s3Cr3t",
    "oauth.callback.url": "https://mycoolapp.com"
  },
  "tags": [
    "Docs"
  ],
  "name": "API Instance"
}'

```

## Authentication Parameters

API parameters in the UI are **bold**, while parameters available in the instances API are in `code formatting`.

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

| Parameter                  | Description                                                                                                                                                                                                              | Data Type |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>key</b>                 | The connector key.<br>infusionsoftrest                                                                                                                                                                                   | string    |
| <b>code</b>                | The authorization grant code returned from the API provider in an OAuth 2.0 authentication workflow. SAP Cloud Platform Open Connectors uses the code to retrieve the OAuth access and refresh tokens from the endpoint. | string    |
| <b>Name</b><br><b>name</b> | The name of the connector instance created during authentication.                                                                                                                                                        | string    |
|                            | The API key or client ID obtained from registering your app                                                                                                                                                              |           |

| Parameter                       | Description                                                                                                                                                              | Data Type |
|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <code>oauth.api.key</code>      | with the provider. This is the <b>Key Name</b> that you noted in <a href="#">API Provider Setup</a> .                                                                    | string    |
| <code>oauth.api.secret</code>   | The client secret obtained from registering your app with the API provider. This is the <b>Secret Name</b> that you noted in <a href="#">API Provider Setup</a> .        | string    |
| <code>oauth.callback.url</code> | The URL that the API provider returns a user to after they authorize access. This is the <b>Callback URL Name</b> that you noted in <a href="#">API Provider Setup</a> . | string    |
| Tags<br><code>tags</code>       | <i>Optional.</i> User-defined tags to further identify the instance.                                                                                                     | string    |

## Example Response for an Authenticated Connector Instance

In this example, the instance ID is `12345` and the instance token starts with "ABC/D...". The actual values returned to you will be unique: make sure you save them for future requests to this new instance.

```
{
  "id": 12345,
  "name": "API Instance",
  "createdDate": "2017-08-07T18:46:38Z",
  "token": "ABC/Dxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "element": {
    "id": 11748,
    "name": "Infusionsoft REST",
    "key": "infusionsoftrest",
    "description": "Add an Infusionsoft Instance to connect your existing Infusionsoft account to the CRM Hub, allowing you to manage contacts, leads, accounts, opportunities etc. across multiple CRM Connectors. You will need your Infusionsoft account information to add an instance.",
    "image": "elements/provider_infusionsoftrest.png",
    "active": true,
    "deleted": false,
    "typeOauth": false,
    "trialAccount": false,
    "bulkDownloadEnabled": false,
    "resources": [ ],
    "transformationsEnabled": true,
    "bulkDownloadEnabled": true,
    "bulkUploadEnabled": true,
    "cloneable": true,
    "extendable": true,
    "beta": false,
    "authentication": {
```



```
    "type": "oauth2"
  },
  "extended": false,
  "hub": "crm",
  "protocolType": "http",
  "parameters": [ ],
  "private": false
},
"elementId": 11748,
"tags": [
  "Docs"
],
"provisionInteractions": [],
"valid": true,
"disabled": false,
"maxCacheSize": 0,
"cacheTimeToLive": 0,
"providerData": {
  "code": "evcfr8vfne4smad3nbnw652e"
},
"configuration": { },
"eventsEnabled": false,
"traceLoggingEnabled": false,
"cachingEnabled": false,
"externalAuthentication": "none",
"user": {
  "id": 12345
}
}
```