# Sage Business Cloud Accounting Authenticate a Connector

Last Modified on 11/02/2021 3:44 am EDT

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

## Authenticate Through the UI

Use the UI to authenticate with Sage and create a connector instance. Because you authenticate with Sage 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 Sage 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:

- 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

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

```
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 **Client ID** that you recorded in API Provider Setup. |
| apiSecret | The client secret obtained from registering your app with the provider. This is the **Client Secret** that you recorded in API Provider Setup. |

| callbackUrl Query Parameter | Description 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 recorded in API Provider Setup. |
|---|---|

## Example cURL

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

## Example Response

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

```
{
"element": "sageone",
"oauthUrl": "https://www.sageone.com/oauth2/auth?scope=full_access&response_type=code&redirect_uri=https%3A%2F%2Fhttpbin.org%2Fget&state=sageone&client_id=b8cb21f24d3f1a2d8c5d"
}
```

# Authenticating Users and Receiving the Authorization Grant Code

Provide the response from the previous step to the users. After they authenticate, Sage Business Cloud Accounting provides the following information in the response:

- code
- country
- state

| Response Parameter | Description |
|---|---|
| code | The Authorization Grant Code required by SAP Open Connectors to retrieve the OAuth access and refresh tokens from the endpoint. |
| country | The country code associated with your account at Sage Business Cloud Accounting |
| state | A customizable identifier, typically the connector key ( `sageone` ). |

> ℹ **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 `/instances` endpoint to authenticate with Sage Business Cloud Accounting and create a connector instance. If you are configuring events, see the Events section.

> ℹ **Note:** The endpoint returns a Connector id and token upon successful completion. Retain the token and id 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": "sageone"
  },
  "providerData": {
    "code": ""
  },
  "configuration": {
    "oauth.api.key": "",
    "oauth.api.secret": "",
    "signature.secret": "",
    "apim.subscription.key": "",
    "country": "",
    "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. Note the **Token** and **ID** and save them 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": "sageone"
  },
  "providerData": {
    "code": "8aa74ff8ae16ba3ca19d12cbdea83aff16bddcd7"
  },
  "configuration": {
    "oauth.api.key": "xxxxxxxxxxxxxxxxxx",
    "oauth.api.secret": "xxxxxxxxxxxxxxxxxxxxxx",
    "signature.secret": "xxxxxxxxxxxxxxxxxxxxxxxxx",
    "apim.subscription.key": "xxxxxxxxxxxxxxxxxxxxxxxxx",
    "country": "US",
    "oauth.callback.url": "https://mycoolapp.com"
  },
  "tags": [
    "Test"
  ],
  "name": "SageOneInstance"
}'
```

## 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.<br>sageone | string |
| Name<br>`name` | The name for the connector instance created during authentication. | Body |
| `oauth.callback.url` | The Callback URL for the connected app you created for Sage Business Cloud Accounting. This is the Callback URL that you recorded in the API Provider Setup section. | |
| `oauth.api.key` | The key obtained from registering your app with the provider. This is the **Client ID** that you recorded in the API Provider Setup section. | string |
| `oauth.api.secret` | The client secret obtained from registering your app with the provider. This is the **Client Secret** that you recorded in the API Provider Setup section. | string |
| `country` | The two digit country code associated with the account of the authenticating user. | string |
| tags | *Optional.* User-defined tags to further identify the instance. | string |

## Example Response

```
{
    "id": 123,
    "name": "SageOneInstance",
    "token": "xxxxxxxxxxxxxxxxxx",
    "element": {
        "id": 3458,
        "name": "Sage Business Cloud Accounting",
        "key": "sageone",
        "description": "Add a Sage Business Cloud Accounting Instance to connect your existing Sage Business Cloud Accounting account to the Finance Hub, allowing you manage your customers, employees, invoices, etc. across multiple Finance connectors. You will need your Sage Business Cloud Accounting account information to add an instance.",
        "image": "http://www.merchantmaverick.com/wp-content/uploads/2015/09/sage-one-logo1.jpg",
        "active": true,
        "deleted": false,
        "typeOauth": false,
        "trialAccount": false,
        "resources": [ ],
        "transformationsEnabled": true,
        "bulkDownloadEnabled": false,
        "bulkUploadEnabled": false,
        "cloneable": true,
        "extendable": false,
        "beta": false,
        "authentication": {
            "type": "oauth2"
        },
        "extended": false,
        "hub": "finance",
        "protocolType": "http",
        "parameters": [],
        "private": false
    },
```

```
    "elementId": 3458,
    "provisionInteractions": [],
    "valid": true,
    "disabled": false,
    "maxCacheSize": 0,
    "cacheTimeToLive": 0,
    "configuration": {
      "base.url": "https://api.accounting.sage.com/v3.1/contact",
      "country": "US",
      "event.vendor.type": "polling",
      "bulk.query.date_mask": "yyyy-MM-dd'T'HH:mm:ss'Z'",
      "pagination.type": "page",
      "oauth.callback.url": "https://httpbin.org/get",
      "oauth.user.refresh_token": "7cc8cd2b4d619e5a09eb5a006bbe5deb7a84d695",
      "oauth.user.refresh_interval": "3600",
      "oauth.token.revoke_url": "https://oauth.na.sageone.com/revoke",
      "oauth.api.key": "b8cb21f24d3f1a2d8c5d",
      "signature.secret": "c956dd3431bc13c5bf9ee1d72d253d2df3fa871a",
      "event.notification.enabled": "false",
      "oauth.api.secret": "d29b5ea1b9b026b6f89e57666897be5a406563df",
      "resource_owner_id": "6ca18a4d6d3443bea14cfe2b48374b5c",
      "oauth.token.url": "https://oauth.na.sageone.com/token",
      "pagination.max": "200",
      "oauth.scope": "full_access",
      "oauth.token.refresh_url": "https://oauth.na.sageone.com/token",
      "oauth.user.token": "b6edb717088721befc7699eda91928d04430ab77",
      "apim.subscription.key": "9317ebcbf83c49ecbafec63becdb1500",
      "oauth.authorization.url": "https://www.sageone.com/oauth2/auth",
      "bulk.query.download_format": "JSON",
      "event.poller.refresh_interval": "15",
      "event.notification.callback.url": null,
      "authentication.time": "",
      "oauth.user.refresh_time": "1496258839283",
      "event.poller.configuration": "{\"customers\":{\"url\":\"/hubs/finance/customers?where=lastModifiedDate>='${date:
yyyy-MM-dd'T'HH:mm:ss'Z'}' and attributes='created_at,updated_at'\",\"idField\":\"id\",\"datesConfiguration\":{\"update
dDateField\":\"updated_at\",\"updatedDateFormat\":\"yyyy-MM-dd'T'HH:mm:ss'Z'\",\"updatedDateTimezone\":\"GMT\",\"c
reatedDateField\":\"created_at\",\"createdDateFormat\":\"yyyy-MM-dd'T'HH:mm:ss'Z'\",\"createdDateTimezone\":\"GMT\"
}}}",
      "oauth.basic.header": "true",
      "pagination.page.startindex": "1"
    },
    "eventsEnabled": false,
    "traceLoggingEnabled": false,
    "cachingEnabled": false,
    "externalAuthentication": "none",
    "user": {
      "id": 123456
    }
}
```