

QuickBooks Online Authenticate a Connector

Last Modified on 11/15/2021 1:10 am EST

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

You can authenticate with QuickBooks using [OAuth 2.0 authentication](#).

As of December 17, 2019, all existing OAuth 1.0 access tokens have been revoked and no new OAuth 1.0 connections are allowed. [Click here](#) to access and learn more about scripts to help you migrate from OAuth1 to OAuth 2.

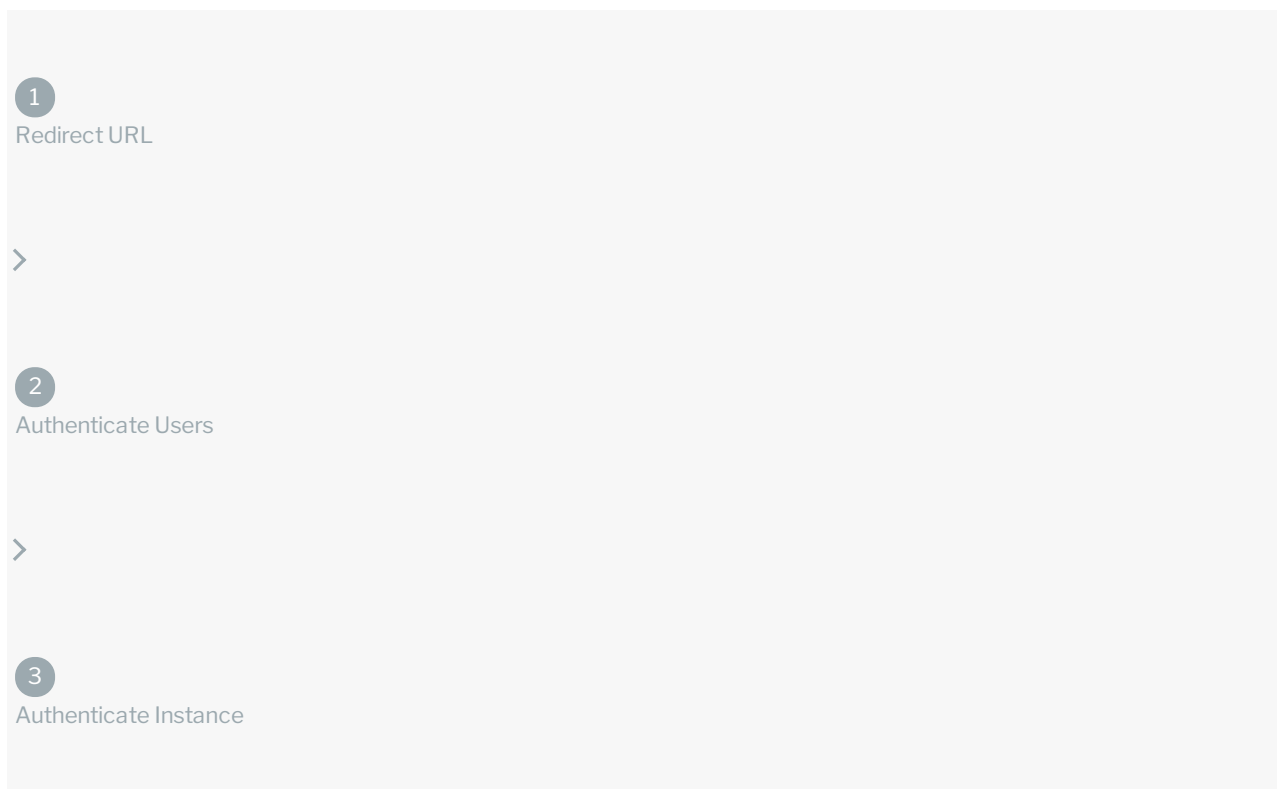
Authenticating through the UI

Use the UI to authenticate with QuickBooks and create a connector instance. When you authenticate with QuickBooks 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 QuickBooks Online 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](#).

Authenticating through API

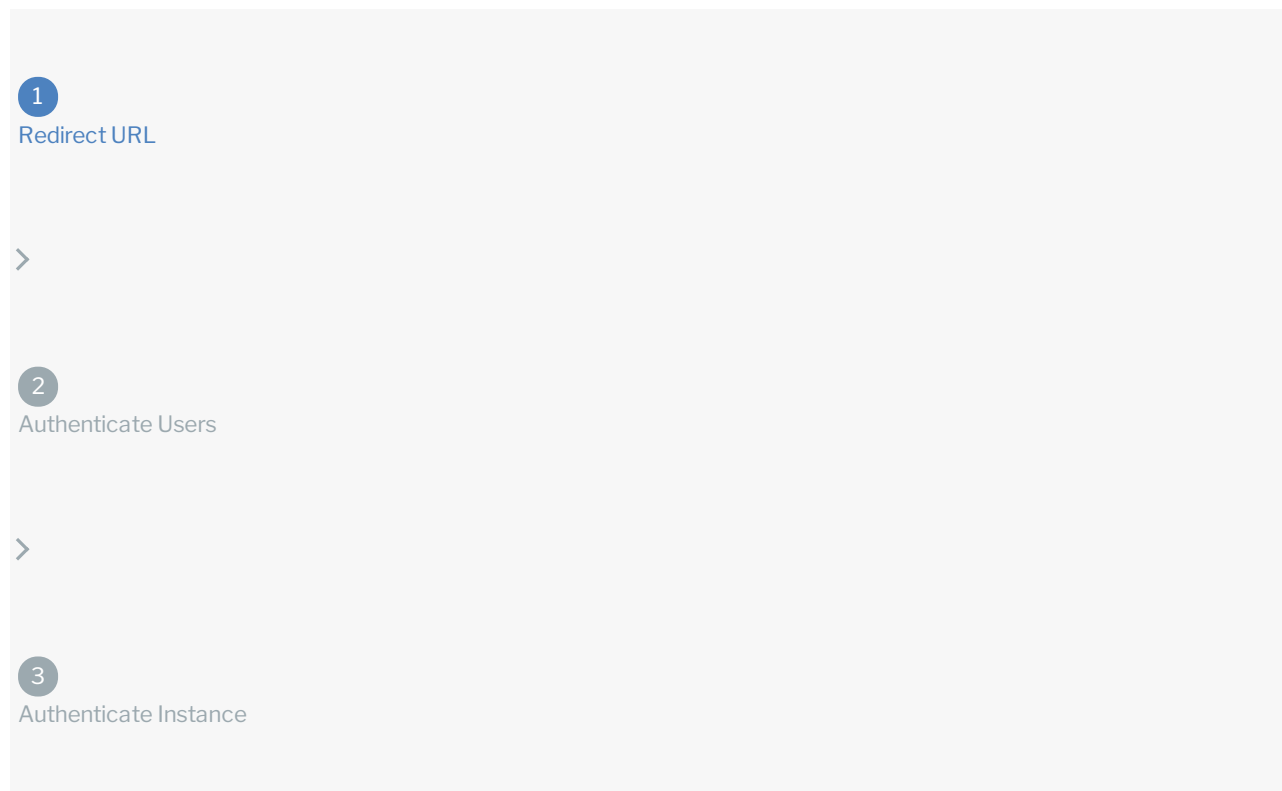
Authenticating through API follows a multi-step OAuth 2.0 process that involves:



- [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.

Note: You can connect to sandbox accounts by specifying `connect.to.sandbox` as `true` when you authenticate a connector instance via API.

Getting a Redirect URL



Use the following API call to request a redirect URL where the user can authenticate with the service provider. Replace `{keyOrd}` with the connector key, `quickbooks`. Note the `scope` and `authentication.type` parameters that are unique to QuickBooks Online.

```
curl -X GET /elements/{keyOrd}/oauth/url?apiKey=&apiSecret=&callbackUrl=&siteAddress=&scope=com.intuit.quickbooks.accounting&authentication.type=oauth2
```

Query Parameters

Query Parameter	Description
apiKey	The API key or client ID obtained from registering your app with the provider. This is the Client ID that you recorded in QuickBooks Online API Provider Setup .
apiSecret	The client secret obtained from registering your app with the API provider. This is the Client Secret that you recorded in QuickBooks Online API Provider Setup .

Query Parameter	Description
callbackUrl	The URL that the API provider returns a user to after they authorize access. This is the Redirect URI that you recorded in QuickBooks Online API Provider Setup .
scope	The scope provided is required to access data in QuickBooks online.
authentication.type	Identifies that you are authenticating with OAuth 2.0.

Example cURL

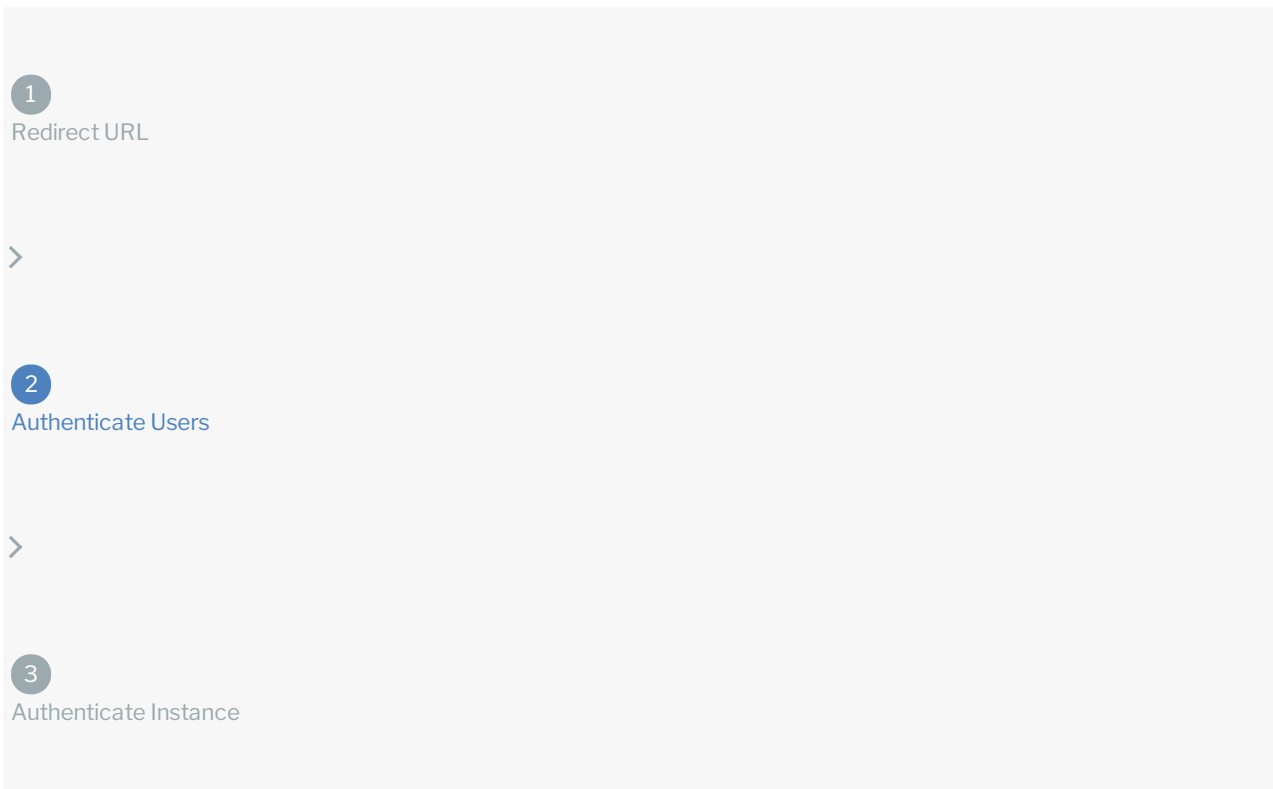
```
curl -X GET \
  'https://api.openconnectors.us2.ext.hana.ondemand.com/elements/api-v2/elements/{keyOrID}oauth/url?apiKey=****
*v&apiSecret=*****&callbackUrl=https://auth.cloudelements.io/oauth&state=quickbooks&authentication.type=oauth
2' \
  -H 'content-type: application/json' \
```

Example Response

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

```
{
  "oauthUrl": "https://appcenter.intuit.com/connect/oauth2?scope=com.intuit.quickbooks.accounting&response_type=co
de&redirect_uri=https%3A%2F%2Fhttpbin.org%2Fget&state=quickbooks&client_id=fiu8Ef123jFHIPEuui19asfji8689EY8
9Fjhfh11I52fs6GEkl",
  "element": "quickbooks"
}
```

Authenticating Users and Receiving the Authorization Grant Code



Provide the response from the previous step to the users. After they authenticate, QuickBooks Online provides the following

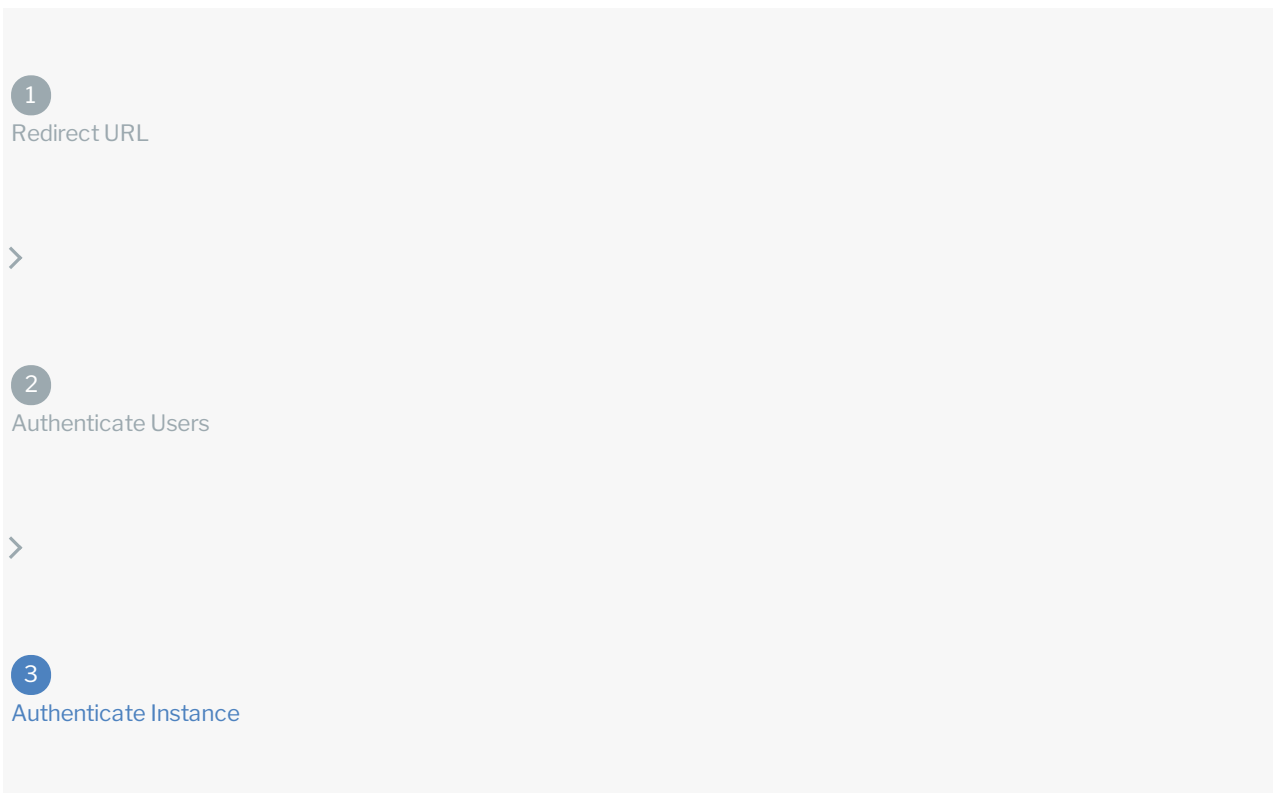
information in the response:

```
{
  "code": "Q011xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "realmId": "1231xxxxxxxxxxxx",
  "state": "quickbooks"
}
```

Response Parameter	Description
code	The authorization grant code returned from the API provider in an OAuth 2.0 authentication workflow. SAP Open Connectors uses the code to retrieve the OAuth access and refresh tokens from the endpoint.
realmID	An identifier unique to QuickBooks that identifies the company that the authenticated instance is connected to.
state	A customizable identifier, typically the connector key (<code>quickbooks</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 `/instances` endpoint to authenticate with QuickBooks Online 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": "quickbooks"
  },
  "providerData": {
    "code": "",
    "realmId": "",

    "dataSource": "QBO"
  },
  "configuration": {
    "oauth.callback.url": "",
    "oauth.api.key": "",
    "oauth.api.secret": "",
    "authentication.type": "oauth2"
  },
  "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 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": "quickbooks"
    },
    "providerData": {
      "code": "xxxxxxxxxxxxxxxxxxxxxxxx",
      "realmId": "xxxxxxxxxxxxxxxxxxxx"
    },
    "configuration": {
      "oauth.callback.url": "https://auth.cloudelements.io/oauth",
      "oauth.api.key": "xxxxxxxxxxxxxxxxxxxx",
      "oauth.api.secret": "xxxxxxxxxxxxxxxxxxxxxxxx",
      "authentication.type": "oauth2"
    },
    "tags": [
      "Docs"
    ],
    "name": "QBO_Instance-="
  }'

```

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
<code>key</code>	The connector key. quickbooks	string
<code>code</code>	The authorization grant code returned from the API provider in an OAuth 2.0 authentication workflow. SAP Open Connectors uses the code to retrieve the OAuth access and refresh tokens from the endpoint.	string
<code>realmId</code>	The realmId returned upon user authorization. An identifier unique to QuickBooks that identifies the company that the authenticated instance is connected to.	string
<code>name</code>	The name of the connector instance created during authentication.	string
<code>oauth.api.key</code>	The API key or client ID obtained from registering your app with the provider. This is the Client ID that you noted in QuickBooks Online API Provider Setup .	string
<code>oauth.api.secret</code>	The client secret obtained from registering your app with the API provider. This is the Client Secret that you noted in QuickBooks Online API Provider Setup .	string
<code>oauth.callback.url</code>	The API key or client ID obtained from registering your app with the provider. This is the Redirect URI that you noted in QuickBooks Online API Provider Setup .	string
<code>authentication.type</code>	Identifies the authentication type to use with the request.	string
<code>quickbooks.minorversion</code>	<i>Optional.</i> A specific version of the API other than the generally available version. For more information, see Minor Versions in the QuickBooks docs .	STRING
<code>scope</code>	Identifies the QuickBooks API access that your application is requesting.	string

tags	<i>Optional.</i> User-defined tags to further identify the instance.	string
Parameter	Description	Type

Example Response for an OAuth 2.0 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.

you are still required to have all of the OAuth information ahead of time.

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": "quickbooks"
    },
    "configuration": {
      "oauth.callback.url": "",
      "quickbooks.datasource": "",
      "oauth.user.refresh_interval": "",
      "quickbooks.realm.id": "",
      "oauth.user.token": "",
      "oauth.user.token.secret": "",
      "oauth.api.key": "",
      "oauth.api.secret": ""
    },
    "tags": [
      "QBO Token"
    ],
    "name": "QBO Token",
    "externalAuthentication": "initial"
  }'
```

Parameters for Token Based Authentication

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. quickbooks	string
Name name	The name for the connector instance created during authentication.	Body
"oauth.user.refresh_interval"	In seconds, the amount of time that should pass before a refresh needs to take place. The default for QuickBooks Online is 151 days or 13046400.	
oauth.user.token	The token retrieve in the Getting an OAuth Token step .	
oauth.user.token.secret	A secret to establish the ownership of the token.	
quickbooks.realm.id	The unique Identifier for the authorized QuickBooks company.	
state	This should always be quickbooks	
quickbooks.dataSource	This value determines what data source should be used for the connection.	
oauth.callback.url	The Callback URL from QuickBooks. This is the Callback URL that you noted in QuickBooks Online API Provider Setup .	
	The Consumer Key from QuickBooks. This is the Consumer Key that you	string

<code>oauth.api.key</code>	noted in QuickBooks Online API Provider Setup .	Data
Parameter	Description	Type
<code>oauth.api.secret</code>	The Consumer Secret from QuickBooks. This is the Consumer Secret that you noted in QuickBooks Online API Provider Setup .	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
Merge Version and Object ids	<p>QuickBooks Online resources return the <code>ID</code> in api response, which is combination of <code>systemId (raw record ID)</code> and <code>syncToken</code> . When a user performs an api call (like <code>GET /invoices</code>) the response object will have <code>ID</code> : <code>1234 3</code> , where <code>1234</code> is the <code>record id</code> and <code>3</code> is <code>sync-token</code></p> <p>This sync token helps manage concurrent requests, whenever user tries to patch any record, only the request with the most recent sync tokens will succeed.” We can use this field to enable / disable this <code>synctokens</code> by default on an instance</p> <ul style="list-style-type: none"> • If <code>Merge Version and Object ids</code> is set to <code>true</code> , the <code>ID</code> will be combination of <code> </code> , example : <code>1234 3</code> • If <code>Merge Version and Object ids</code> is set to <code>false</code> , the <code>ID</code> will be just <code>_</code> , example : <code>1234</code> 	

Example Response

```
{
  "id": 123,
  "name": "test",
  "token": "2jidFlehYe6B2uilUFI82fsdf+1thuVT+mvoFj3",
  "element": {
    "id": 39,
    "name": "QuickBooks Online",
    "key": "quickbooks",
    "description": "Add a QuickBooks Online Instance to connect your existing QuickBooks Online account to the Finance Hub, allowing you to manage your customers, employees, invoices, purchase orders etc. across multiple Finance connectors. You will need your QuickBooks Online account information to add an instance.",
    "active": true,
    "deleted": false,
    "typeOAuth": true,
    "trialAccount": false,
    "configDescription": "If you do not have an QuickBooks account, you can create one at \"http://quickbooks.intuit.com/signup/\" target=\"_blank\">QuickBooks Signup",
    "signupURL": "http://quickbooks.intuit.com/signup/"
  },
  "provisionInteractions": [],
  "valid": true,
  "eventsEnabled": true,
  "disabled": false,
  "maxCacheSize": 0,
  "cacheTimeToLive": 0,
  "cachingEnabled": false
}
```