QuickBooks Online Events

Last Modified on 01/19/2021 12:42 pm EST

On this page

SAP Cloud Platform Open Connectors supports events via polling or webhooks depending on the API provider. For more information about our Events framework, see Events Overview.

Supported Events and Resources

SAP Cloud Platform Open Connectors supports polling events and webhooks for QuickBooks Online. After receiving an event, SAP Cloud Platform Open Connectors standardizes the payload and sends an event to the configured callback URL of your authenticated connector instance.

Polling

You can set up polling for the `events` resource. You can also copy the `events` configuration to poll other resources. See Configure Polling Through API for more information.

> **Note:** Unless configured for a specific time zone, polling occurs in UTC.

You can set up events for the following resources:

- bill-payments
- bills
- classes
- credit-memos
- credit-terms
- currencies
- customers
- employees
- invoices
- journal-entries
- ledger-accounts
- payment-methods
- payments
- products
- purchase-orders
- refund-receipts
- sales-receipts
- tax-codes
- tax-rates
- time-activities
- vendor-credits
- vendors

Configure Polling Through the UI

To configure polling through the UI, follow the same steps to authenticate a connector instance, and then turn on events. Select the resources to poll, and then click **Create Instance**. For more information, see Authenticate an Connector Instance with Events (UI) or the connector-specific authentication topic.

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 object, or use it in a formula template.

Configure Polling Through API

Use the `/instances` endpoint to authenticate with QuickBooks Online and create a connector instance with polling enabled.

> **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 with polling:

1. Complete the authentication steps for the authentication up to constructing the final authentication JSON body.
2. Construct a JSON body as shown below (see Parameters):

```
   {
     "element":{
       "key":"quickbooks"
     },
     "providerData":{
       "code": "",
       "realmId": ""
     },
     "configuration":{
       "oauth.callback.url": "",
       "oauth.api.key": "",
        "oauth.api.secret": "",
       "authentication.type" : "oauth2",
       "scope" : "com.intuit.quickbooks.accounting openid profile email phone address",
       "event.notification.enabled": true,
       "event.vendor.type": "polling",
       "event.notification.callback.url": "https://api.openconnectors.us2.ext.hana.ondemand.
com/elements/api-v2/events/quickbooks/",
       "event.notification.signature.key": "",
        "event.poller.refresh_interval": "15",
       "event.poller.urls": "bill-paymentsbillsclassescredit-memoscredit-termscurrenciescust
omersemployeesinvoicesjournal-entriesledger-accountspayment-methodspaymentsproductspurchase-
ordersrefund-receiptssales-receiptstax-codestax-ratestime-activitiesvendor-creditsvendors"
     },
     "tags":[
       ""
     ],
     "name":""
   }

}
```

3. 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.

4. Locate the `token` and `id` in the response and save them for all future requests using the connector instance.

Example cURL with Polling

```
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": "xxxxxxxxxxxxxxxxxxxxx",
    "realmId": "xxxxxxxxxxxxxxx"
  },
  "configuration": {
    "oauth.callback.url": "https;//mycoolapp.com",
    "oauth.api.key": "xxxxxxxxxxxxxxxxx",
    "oauth.api.secret": "xxxxxxxxxxxxxxxxxxxxxx"
    "authentication.type" : "oauth2",
    "scope" : "com.intuit.quickbooks.accounting openid profile email phone address",
    "event.notification.enabled": true,
    "event.vendor.type": "polling",
    "event.notification.callback.url": "https://api.openconnectors.us2.ext.hana.ondemand.com/elem
ents/api-v2/events/quickbooks/",
    "event.notification.signature.key": "xxxxxxxxxxxxxxxxxxxx",
    "event.poller.refresh_interval": "15",
    "event.poller.urls": "bill-paymentsbillsclassescredit-memoscredit-termscurrenciescustomersemp
loyeesinvoicesjournal-entriesledger-accountspayment-methodspaymentsproductspurchase-ordersrefund-
receiptssales-receiptstax-codestax-ratestime-activitiesvendor-creditsvendors"
    },
  "tags": [
    "Docs"
  ],
  "name": "API Instance"
}'
```

Polling Parameters

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

Parameter	Description	Data Type
Events Enabled `event.notification.enabled`	*Optional.* Identifies that events are enabled for the connector instance. Default: `false`.	boolean
Vendor Event Type `event.vendor.type`	*Optional.* Identifies the type of events enabled for the instance, either `webhook` or `polling`.	string
Event Notification Callback URL `event.notification.callback.url`	The URL where you want SAP Cloud Platform Open Connectors to send the events.	string
Callback Notification Signature Key `event.notification.signature.key`	*Optional.* A user-defined key for added security to show that events have not been tampered with.	string
Event poller refresh interval (mins)	A number in minutes to identify how often the poller should	number

Parameter	Description	Data Type
`event.poller.refresh_interval`	check for changes.	
`event.poller.urls`	The objects that should be polled.	
tags	*Optional.* User-defined tags to further identify the instance.	string

Webhooks

You can configure webhooks through the UI or in the JSON body of the `/instances` API request .

Configure Webhooks Through the UI

To configure webhooks through the UI, follow the same steps to authenticate a connector instance, and then turn on events. For more information, see Authenticate an Connector Instance with Events (UI) or the connector-specific authentication topic.

Configure Webhooks Through API

Use the `/instances` endpoint to authenticate with and create a connector instance with webhooks enabled.

> **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 with webhooks:

1. Get an authorization grant code by completing the steps in Getting a redirect URL and Authenticating users and receiving the authorization grant code.

2. Construct a JSON body as shown below (see Parameters):

```
{
  "element":{
    "key":"quickbooks"
  },
  "providerData":{
    "code": "",
    "realmId": ""
  },
  "configuration":{
    "oauth.callback.url": "",
    "oauth.api.key": "",
     "oauth.api.secret": "",
    "authentication.type" : "oauth2",
    "scope" : "com.intuit.quickbooks.accounting openid profile email phone address",
    "event.notification.enabled": true,
    "event.vendor.type": "webhooks",
    "event.notification.callback.url": "https://api.openconnectors.us2.ext.hana.ondemand.
com/elements/api-v2/events/quickbooks/",
    "event.notification.signature.key": ""
   },
  "tags":[
    ""
  ],
  "name":""
}
```

3. 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. See the Overview for details.

4. Locate the `token` and `id` in the response and save them for all future requests using the connector instance.

Example cURL with Polling

```
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": "xxxxxxxxxxxxxxxxxxxxx",
    "realmId": "xxxxxxxxxxxxxxx"
  },
  "configuration": {
    "oauth.callback.url": "https;//mycoolapp.com",
    "oauth.api.key": "xxxxxxxxxxxxxxxxx",
    "oauth.api.secret": "xxxxxxxxxxxxxxxxxxxxxx"
    "authentication.type" : "oauth2",
    "scope" : "com.intuit.quickbooks.accounting openid profile email phone address",
    "event.notification.enabled": true,
    "event.vendor.type": "webhook",
    "event.notification.callback.url": "https://api.openconnectors.us2.ext.hana.ondemand.com/elem
ents/api-v2/events/quickbooks/",
    "event.notification.signature.key": "xxxxxxxxxxxxxxxxxxxx"
    },
  "tags": [
    "Docs"
  ],
  "name": "API Instance"
}'
```

Webhook Parameters

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

Parameter	Description	Data Type
Events Enabled `event.notification.enabled`	*Optional.* Identifies that events are enabled for the connector instance. Default: `false`.	boolean
Vendor Event Type `event.vendor.type`	*Optional.* Identifies the type of events enabled for the instance, either `webhook` or `polling`.	string
Event Notification Callback URL `event.notification.callback.url`	The URL where you want SAP Cloud Platform Open Connectors to send the events.	string

Parameter	Description	Data Type
Callback Notification Signature Key `event.notification.signature.key`	*Optional.* A user-defined key for added security to show that events have not been tampered with.	string
tags	*Optional.* User-defined tags to further identify the instance.	string