Authenticate a Connector Instance with Events (API)

Last Modified on 08/26/2021 7:37 am EDT

On this page

This section provides a summary of how to authenticate a connector instance with events. Each connector is different, so take a look at the Events section of the Connector Guide. You can authenticate via SAP Open Connectors or APIs.

Authenticate a Connector Instance with Polling

Authenticating a connector instance with events works the same as authenticating an instance, you just need to turn on events and set a few more parameters.

To authenticate a connector instance with polling events add the polling configuration to the JSON body of your `POST /instances` request. See the Authenticate and Events sections of the Connector Guide for the connector that you want to monitor for connector-specific steps.

For more information about each field described here, see Polling Parameters.

Polling Configuration

In the `configuration` JSON object, add the following event and polling configuration parameters when authenticating a connector instance with polling events:

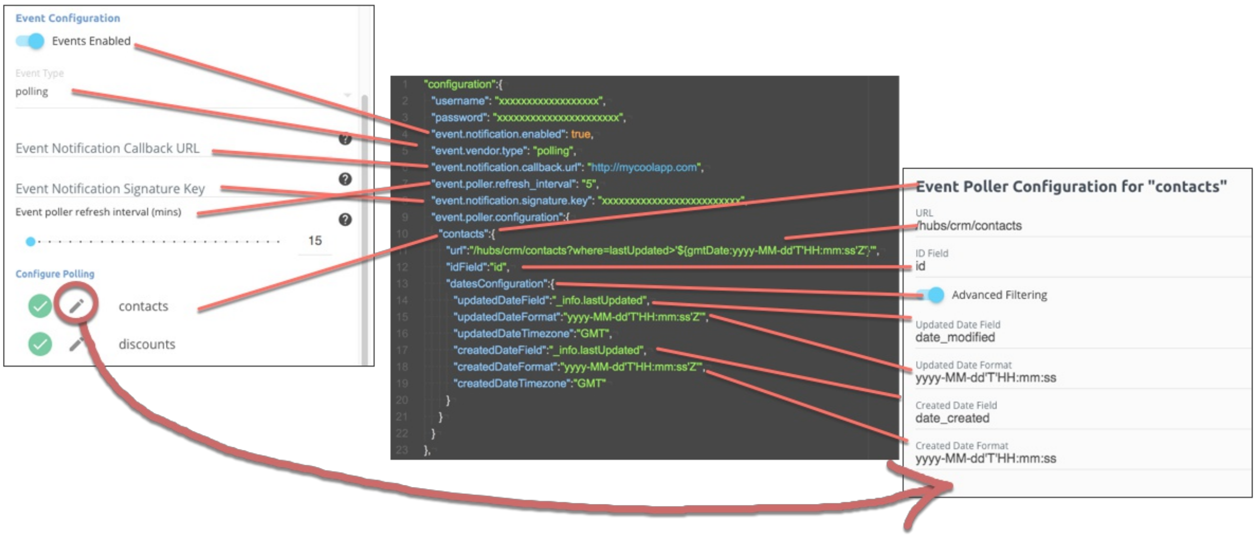
- `event.notification.enabled: true`
- `event.vendor.type: polling`
- `event.notification.callback.url:`
- `event.notification.signature.key:`
- `event.poller.refresh_interval:`
- `event.poller.configuration:`

Here is an example of a connector that uses Basic authentication with the required polling configuration values. The usual body to authenticate a connector instance only includes the `username` and `password` parameters, the rest are event and polling-specific parameters.

```
{
  "element":{
    "key":"connectorKey"
  },
  "configuration":{
    "username": "xxxxxxxxxxxxxxxxxx",
    "password": "xxxxxxxxxxxxxxxxxxxxxx",
    "event.notification.enabled": true,
    "event.vendor.type": "polling",
    "event.notification.callback.url": "http://mycoolapp.com",
    "event.poller.refresh_interval": "",
    "event.notification.signature.key": "xxxxxxxxxxxxxxxxxxxxxxxxx",
    "event.poller.configuration":{
      "contacts":{
        "url":"/hubs/crm/contacts?where=lastUpdated>'${gmtDate:yyyy-MM-dd'T'HH:mm:ss'Z'}'",
        "idField":"id",
        "datesConfiguration":{
          "updatedDateField":"_info.lastUpdated",
          "updatedDateFormat":"yyyy-MM-dd'T'HH:mm:ss'Z'",
          "updatedDateTimezone":"GMT",
          "createdDateField":"_info.lastUpdated",
          "createdDateFormat":"yyyy-MM-dd'T'HH:mm:ss'Z'",
          "createdDateTimezone":"GMT"
        }
      }
    }
  },
  "tags":[
    ""
  ],
  "name":""
}
```

Polling Parameters

Labels and buttons on the UI correspond to parameters in the JSON. The table below shows UI labels and buttons in **bold** and the equivalent parameters in the configuration JSON object in `code formatting`.



Event Configuration

Events Enabled

Event Type
polling

Event Notification Callback URL

Event Notification Signature Key

Event poller refresh interval (mins)
15

Configure Polling
contacts
discounts

```
"configuration":{
  "username": "xxxxxxxxxxxxxxxxxx",
  "password": "xxxxxxxxxxxxxxxxxxxxxx",
  "event.notification.enabled": true,
  "event.vendor.type": "polling",
  "event.notification.callback.url": "http://mycoolapp.com",
  "event.poller.refresh_interval": "5",
  "event.notification.signature.key": "xxxxxxxxxxxxxxxxxxxxxxxxxx",
  "event.poller.configuration":{
    "contacts":{
      "url":"/hubs/crm/contacts?where=lastUpdated>'${gmtDate:yyyy-MM-dd'T'HH:mm:ss'Z'}'",
      "idField":"id",
      "datesConfiguration":{
        "updatedDateField":"_info.lastUpdated",
        "updatedDateFormat":"yyyy-MM-dd'T'HH:mm:ss'Z'",
        "updatedDateTimezone":"GMT",
        "createdDateField":"_info.lastUpdated",
        "createdDateFormat":"yyyy-MM-dd'T'HH:mm:ss'Z'",
        "createdDateTimezone":"GMT"
      }
    }
  }
},
```

Event Poller Configuration for "contacts"

URL
/hubs/crm/contacts

ID Field
id

Advanced Filtering

Updated Date Field
date_modified

Updated Date Format
yyyy-MM-dd'T'HH:mm:ss

Created Date Field
date_created

Created Date Format
yyyy-MM-dd'T'HH:mm:ss

Parameter	Description	Data Type
Events Enabled `event.notification.enabled`	Identifies that events are enabled for the connector instance. Default: `false`.	boolean

Parameter	Description	Data Type
Event Type `event.vendor.type`	The type of event, either `polling` or `webhook` .	string
Event Notification Callback URL `event.notification.callback.url`	The URL where you want SAP Open Connectors to send the events.	string
Event poller refresh interval (mins) `event.poller.refresh_interval`	A number in minutes to identify how often the poller should check for changes.	number
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
Configure Polling `event.poller.configuration`	Configuration parameters for polling. ☑ **Tip:** The default polling configuration represents the optimal configuration. Although you can change anything in the poller configuration, we recommend that you do so rarely and in conjunction with SAP Open Connectors support.	JSON object
Resource to Poll `resourceName` (e.g., `contacts`)	The polling event configuration of the resource that you will monitor.	JSON object
URL `url`	The url to query for updates to the resource.	String
ID Field `idField`	The field in the resource that is used to uniquely identify it.	String
Advanced Filtering `datesConfiguration`	Configuration parameters for dates in polling.	JSON Object
Updated Date Field `updatedDateField`	The field that identifies an updated object.	String
Updated Date Format `updatedDateFormat`	The date format of the field that identifies an updated object.	String
Created Date Field `createdDateField`	The field that identifies a created object.	String
Created Date Format `createdDateFormat`	The date format of the field that identifies a created object.	String

Authenticate a Connector Instance with Webhooks

Authenticating a connector instance with events works the same as authenticating an instance, you just need to turn on events and set a few more parameters.

To authenticate a connector instance with webhook events add the webhook configuration to the JSON body of your `POST /instances` request. See the Authentication and Events sections of the Connector Guide for the connector that you want to monitor for connector-specific steps.

> ℹ **Note:** When you authenticate a connector instance with webhook events, make sure that you check the Connector Guide for any additional setup you need at the API provider.

For more information about each field described here, see Webhooks Parameters.

Webhook Configuration

In the `configuration` JSON object, add the following event and webhook configuration parameters when authenticating a connector instance with webhook events:

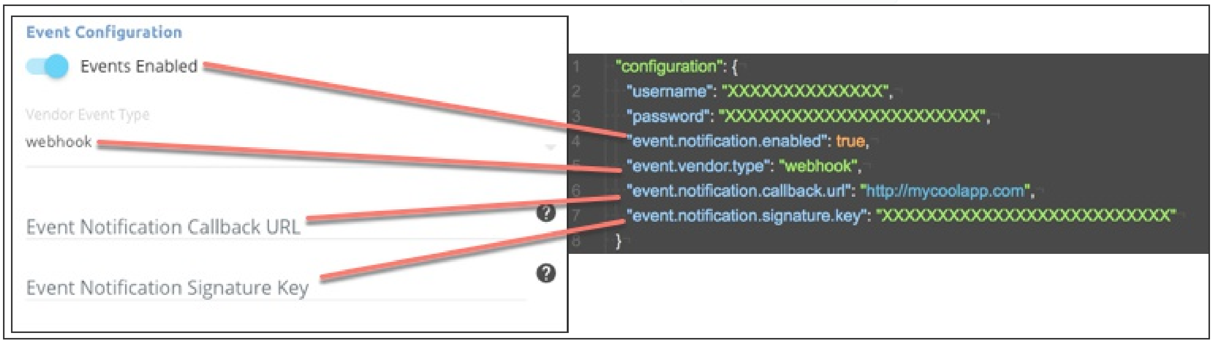
- `event.notification.enabled: true`
- `event.vendor.type: webhooks`
- `event.notification.callback.url:`
- `event.notification.signature.key:`

Here is an example of a connector that uses Basic authentication with the required polling configuration values. The usual body to authenticate a connector instance only includes the `username` and `password` parameters, the rest are event and polling-specific parameters.

```
{
  "element":{
    "key":"connectorKey"
  },
  "configuration":{
    "username": "xxxxxxxxxxxxxxxxxx",
    "password": "xxxxxxxxxxxxxxxxxxxxxx",
    "event.notification.enabled": true,
    "event.vendor.type": "webhooks",
    "event.notification.callback.url": "http://mycoolapp.com",
    "event.notification.signature.key": "xxxxxxxxxxxxxxxxxxxxxxxxx"
  },
  "tags":[
    ""
  ],
  "name":""
}
```

Webhook Parameters

Labels and buttons on the UI correspond to parameters in the JSON. The table below shows UI labels and buttons in **bold** and the equivalent parameters in the configuration JSON object in `code formatting`.



Event Configuration

Events Enabled

Vendor Event Type: webhook

Event Notification Callback URL

Event Notification Signature Key

```
"configuration": {
  "username": "XXXXXXXXXXXXXX",
  "password": "XXXXXXXXXXXXXXXXXXXXXXX",
  "event.notification.enabled": true,
  "event.vendor.type": "webhook",
  "event.notification.callback.url": "http://mycoolapp.com",
  "event.notification.signature.key": "XXXXXXXXXXXXXXXXXXXXXXXXXXX"
}
```

Parameter	Description	Data Type
Events Enabled `event.notification.enabled`	Identifies that events are enabled for the connector instance. Default: `false` .	boolean
Event Type `event.vendor.type`	The type of event, either `polling` or `webhook` .	string

Parameter	Description	Data Type
Event Notification Callback URL `event.notification.callback.url`	The URL where you want SAP 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