GitHub Events

Last Modified on 01/07/2019 11:56 am EST

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 webhook events for GitHub. After receiving an event, SAP Cloud Platform Open Connectors standardizes the payload and sends an event to the configured callback URL of your authenticated instance. For more information about webhooks at GitHub including the currently available webhooks, see their webhooks documentation .

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 GitHub 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": "github"
  },
  "providerData": {
    "code": ""
  },
  "configuration": {
    "oauth.api.key": "",
    "oauth.api.secret": "",
    "oauth.callback.url": "",
    "github.organization": "",
    "github.repository": "",
    "event.notification.enabled": true,
    "event.vendor.type": "webhooks",
    "event.notification.callback.url": "",
    "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. 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

```
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": "github"
  },
  "providerData": {
    "code": "xoz8AFqScK2ngM04kSSM"
  },
  "configuration": {
    "oauth.api.key": "Rand0MAP1-key",
    "oauth.api.secret": "fak3AP1-s3Cr3t",
    "oauth.callback.url": "https://mycoolapp.com",
    "github.organization": "myOrg",
    "github.repository": "myRepo"
    "event.notification.enabled": true,
    "event.vendor.type": "webhooks",
    "event.notification.callback.url": "https://mycoolapp.com/events",
    "event.notification.signature.key": "xxxxxxxxxxxxxxxxxxxxxxxx"
  },
  "tags": [
    "Docs"
  ],
  "name": "API Instance"
}'
```

Parameters

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

Parameter	Description	Data Type
`key`	The connector key. github	string
`code`	The authorization grant code returned from the API provider in an OAuth 2.0 authentication workflow. SAP Cloud Platform Open Connectors uses the	string

Parameter	Description	Data Type
	code to retrieve the OAuth access and refresh tokens from the endpoint.	
Name `name`	The name of the connector instance created during authentication.	Body
`oauth.api.key`	The API key or client ID 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 API provider. This is the **Client Secret** that you recorded in the API Provider Setup section.	string
`oauth.callback.url`	The URL that the API provider returns a user to after they authorize access. This is the **Authorization callback URL** that you recorded in the API Provider Setup section.	string
GitHub Organization `github.organization`	The organization associated with the repository you're connecting to.	string
GitHub Repository `github.repository`	The name of the repository you're connecting to.	string
Events Enabled `event.notification.enabled`	*Optional.* Identifies that events are enabled for the connector instance. Default: `false`.	boolean
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
tags	*Optional.* User-defined tags to further identify the instance.	string

Configure Webhooks at GitHub

After you authenticate a connector instance with webhooks, you must set up webhooks at

GitHub. Take a look at their documentation for complete details.

1. Retrieve the Webhook URL from the connector instance.
    - Navigate to the connector instance, click **Edit** on the connector instance card, and then copy the **Webhook URL**.
    - Make a `GET/instances/{id}/events` request and look for `encodedid` in the response. Append that to the `event.notification.callback.url` you used when authenticating the connector instance. This is the **Webhook URL**.
2. Go to your repository in Github, and then click the **Settings** tab.
3. On the left, click **Webhooks**.
4. Click **Add webhook**.
5. In **Payload URL**, enter the **Webhook URL** you identified in step 1.
6. Change the **Content Type** to **application/json**.
7. Complete the form as needed for your use case, and then click **Add Webhook**.