

Acuity Scheduling Events

Last Modified on 12/20/2019 2:48 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 webhook events for Acuity Scheduling. 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. For more information about webhooks at Acuity Scheduling including the currently available webhooks, see [Acuity Scheduling's webhooks documentation](#).

After you set up webhook events in SAP Cloud Platform Open Connectors, you also need to configure webhooks at Acuity Scheduling with the Webhook URL provided after authentication. See [Webhook Configuration](#).

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.

After successfully authenticating, open the authenticated instance that you just created, and then copy the Webhook URL. You'll use the url to [configure webhooks at Acuity Scheduling](#).

Configure Webhooks Through API

Use the `/instances` endpoint to authenticate with Acuity Scheduling 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": "acuityscheduling"
  },
  "providerData": {
    "code": ""
  },
  "configuration": {
    "oauth.api.key": "",
    "oauth.api.secret": "",
    "oauth.callback.url": "",
    "event.notification.enabled": true,
    "event.notification.callback.url": ""
  },
  "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.
5. Set up [webhooks at the API provider][#webhook-configuration].

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": "acuityscheduling"
    },
    "providerData": {
      "code": "xoz8AFqScK2ngM04kSSM"
    },
    "configuration": {
      "oauth.api.key": "Rand0MAP1-key",
      "oauth.api.secret": "fak3AP1-s3Cr3t",
      "oauth.callback.url": "https://mycoolapp.com",
      "event.notification.enabled": true,
      "event.notification.callback.url": "https://mycoolapp.com/events"
    },
    "tags": [
      "Docs"
    ],
    "name": "API Instance"
  }'
```

Parameters

API parameters not shown in the SAP Cloud Platform Open Connectors are in

`code formatting`.

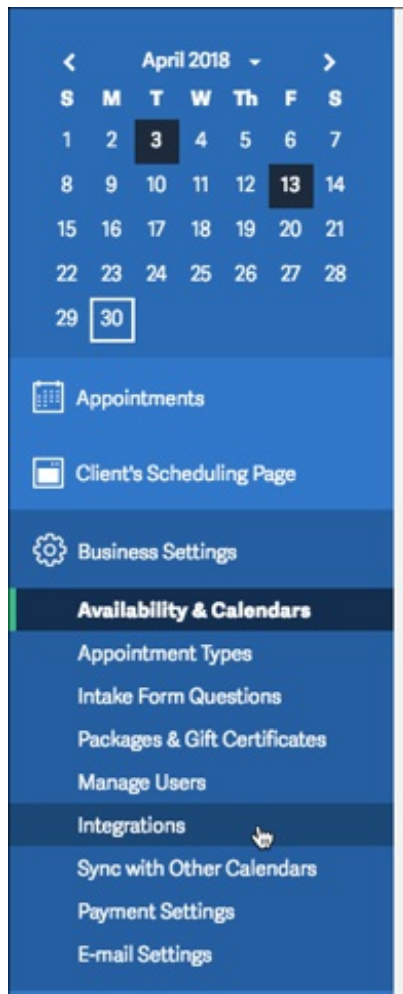
Parameter	Description	Data Type
<code>key</code>	The connector key. acuityscheduling	string
<code>code</code>	The authorization grant code returned from the API provider in an OAuth 2.0 authentication workflow. SAP Cloud Platform Open Connectors uses the code to retrieve the OAuth access and refresh tokens from the endpoint.	string

Name Parameter	Description	Data Type
<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 recorded in 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 recorded in API Provider Setup .	string
<code>oauth.callback.url</code>	The URL that the API provider returns a user to after they authorize access. This is the Redirect URI that you recorded in API Provider Setup .	string
Events Enabled <code>event.notification.enabled</code>	<i>Optional.</i> Identifies that events are enabled for the connector instance. Default: <code>false</code> .	boolean
Event Notification Callback URL <code>event.notification.callback.url</code>	The URL where you want SAP Cloud Platform Open Connectors to send the events.	string
tags	<i>Optional.</i> User-defined tags to further identify the instance.	string

Webhook Configuration

Use the Webhook URL we generated after you authenticated the connector Instance to set up webhooks at Acuity Scheduling. Before you begin, [review the Acuity Scheduling documentation on webhooks](#).

1. Log in to your account at [Acuity Scheduling](#).
2. On the left, click **Business Settings**, and then click **Integrations**.



3. Scroll to the API section, find **Webhooks**, and then click **Set Up**.
4. In the new window, enter the Webhook URL from the authenticated connector instance in any of the webhook types that you want to receive events for.
5. Click **Save Settings**.