

Freshservice Events

Last Modified on 02/12/2019 4:12 pm EST

SAP Cloud Platform Open Connectors currently supports events via **polling** or **webhooks** depending on the endpoint.

- **Polling** is a mechanism where SAP Cloud Platform Open Connectors executes the configured query every `n` minutes and captures the changed information.
- **Webhooks** are when the provider lets SAP Cloud Platform Open Connectors know what information has changed. Note that additional endpoint setup may be required prior to creating your connector instance.

Note: SAP Cloud Platform Open Connectors normalizes only the `objectId`, `objectType`, and `eventType` event data. Event data also contains raw data, which the provider returns. The raw data varies based on the endpoint.

If you would like to see more information on our Events framework, please see [Events Overview](#).

Freshservice requires an extra specification to be entered once a connector instance has been created. The Freshservice Instance ID must be 64 base encoded, then included in the webhook callback URL. This document will walk you through the entire workflow:

- create an instance
- retrieve the instance ID
- 64base Encode Instance ID
- webhook setup

Step 1. Create an Instance

To provision your Freshservice connector, use the `/instances` API.

Below is an example of the provisioning API call.

- **HTTP Headers:** Authorization- User , Organization
- **HTTP Verb:** POST
- **Request URL:** `/instances`
- **Request Body:** Required – see below
- **Query Parameters:** none

Description: token is returned upon successful execution of this API. This token needs to be

retained by the application for all subsequent requests involving this connector instance.

A sample request illustrating the /instances API is shown below.

HTTP Headers:

```
Authorization: User , Organization
```

This instance.json file must be included with your instance request. Please fill your information to provision. The “key” into SAP Cloud Platform Open Connectors Freshservice is “freshdesk”. This will need to be entered in the “key” field below depending on which connector you wish to instantiate.

```
{
  "element": {
    "key": "freshservice"
  },
  "configuration": {
    "username": "",
    "password": "",
    "subdomain": "",
    "event.notification.enabled": "true",
    "event.notification.callback.url": ""
  },
  "tags": [
    ""
  ],
  "name": "",
  "eventsEnabled": "true"
}
```

Here is an example cURL command to create an instance using /instances API.

Example Request:

```
curl -X POST
-H 'Authorization: User , Organization '
-H 'Content-Type: application/json'
-d @instance.json
'https://api.openconnectors.us2.ext.hana.ondemand.com/elements/api-v2/instances'
```

If the user does not specify a required config entry, an error will result notifying her of which

entries she is missing.

Below is a successful JSON response:

```
"id": 1234,
"name": "Freshservice Test",
"token": "/sO6vVsB2eXhOlgvNR/p+G7wC/+rhY5M=",
"element": {
  "id": 130,
  "name": "Freshservice Beta",
  "key": "freshservice",
  "description": "Add a Freshservice Instance to connect your existing Freshservice account to the Help Desk Hub, allowing you to manage your incidents, priorities, statuses, users, etc. across multiple Help Desk connectors. You will need your Freshservice account information to add an instance.",
  "image": "https://pbs.twimg.com/profile_images/3159951933/1511f0f59e3f239a8ef707b1db3a42e3.png",
  "active": true,
  "deleted": false,
  "typeOauth": false,
  "trialAccount": false,
  "hub": "helpdesk",
  "parameters": [
    {
      "id": 38,
      "createdDate": "2015-04-08T19:53:48Z",
      "name": "subdomain",
      "vendorName": "subdomain",
      "type": "configuration",
      "vendorType": "path",
      "source": "request",
      "elementId": 130,
      "required": false
    }
  ]
},
"provisionInteractions": [],
"valid": true,
"disabled": false,
"maxCacheSize": 0,
"cacheTimeToLive": 0,
"configuration": {
  "base.url": "https://{subdomain}.freshservice.com",
  "pagination.offset": "false",
  "password": "password",
  "pagination.max": "30",
  "subdomain": "https://sample.freshservice.com",
  "username": "sample@sample.com",
  "event.notification.callback.url": "https://www.mycoolapp.com/events"
```

```
},  
"eventsEnabled": true,  
"cachingEnabled": false  
}
```

Note: Make sure you have straight quotes in your JSON files and cURL commands. Please use plain text formatting in your code. Make sure you do not have spaces after the in the cURL command.

Retrieve Instance ID from the create instance response:

```
{  
  "id": 1234,  
  "name": "Freshservice Test",  
  "token": "/sO6vVsB2eXhOlgvNR/p+G7wC/+rhY5M=",  
}
```

Copy the Instance ID and go to a 64Base Encoding website.

Here is a sample site that will 64 base encode your instance ID: <https://www.base64encode.org/>.

Copy the ID, encode it, then copy the encoded ID.

Place the ID in the following URL:

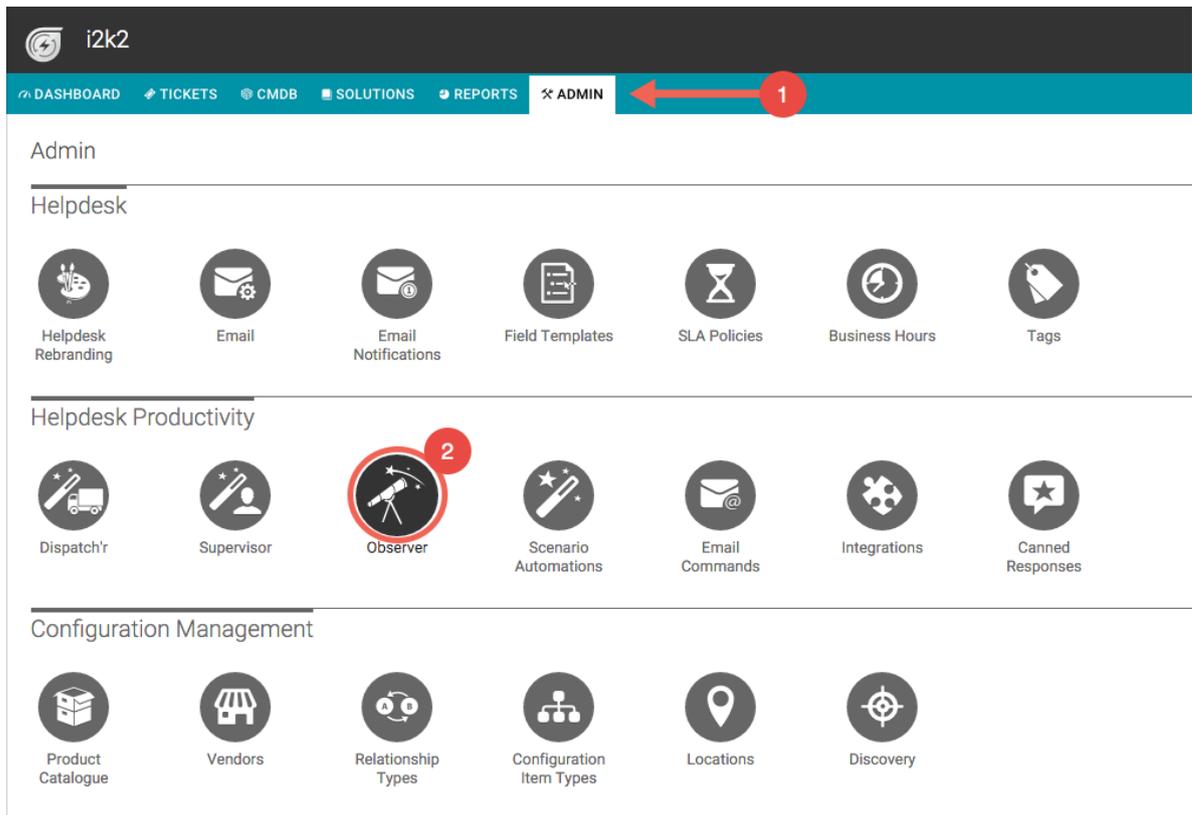
```
https://api.openconnectors.us2.ext.hana.ondemand.com/elements/api-  
v2/events/freshservice/{INSERT_64BASE_ENCODED_INSTANCE_ID}
```

An example of the URL once the Instance ID has been encoded:

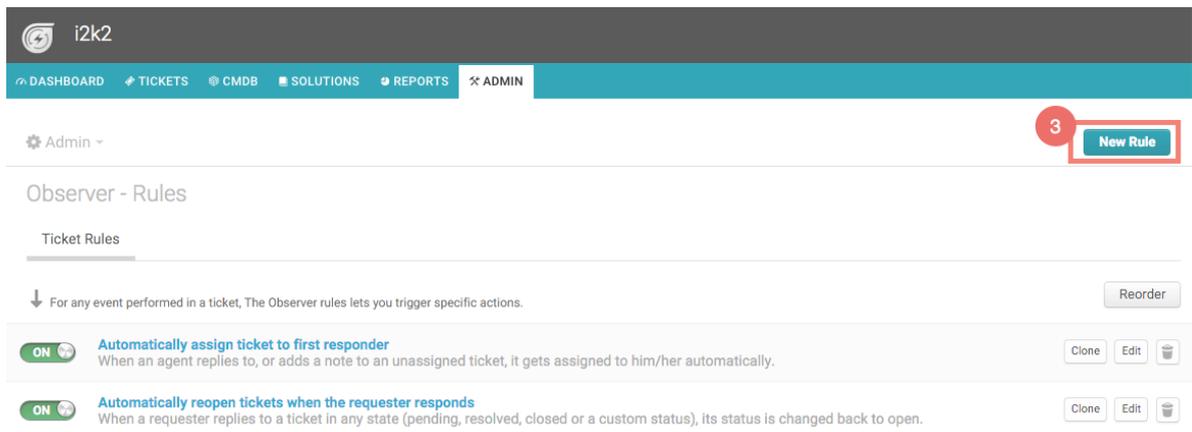
```
https://api.openconnectors.us2.ext.hana.ondemand.com/elements/api-  
v2/events/freshservice/MjA5MzE=
```

Log in to your Freshservice domain.

1. Click "Admin"
2. Under Helpdesk Productivity, select "Observer"



3. Select “New Rule”



4. Name the Rule

5. Add an Event

6. Select person to perform events

7. Select condition(s)

8. Under “Select Actions”

9. Choose “Trigger Webhook”

10. Select POST

11. Insert the following URL with the base64 encoded Instance ID:

```
https://api.openconnectors.us2.ext.hana.ondemand.com/elements/api-v2/events/freshservice/{INSERT_BASE64_ENCODED_INSTANCE_ID}
```

Content should be set to JSON

12. Select Content

13. Click “Save”

The screenshot shows a configuration page for a webhook. At the top right, there are 'Cancel' and 'Save' buttons, with a red circle containing the number '13' pointing to the 'Save' button. Below this, there is a section 'on ticket with these properties' with radio buttons for 'Match ANY of the below' and 'Match ALL of the below'. A dropdown menu shows 'Select Condition'. Below that is a section 'perform these actions' with a dropdown menu showing 'Trigger Webhook'. Under 'Request Type', a dropdown menu shows 'POST', with a red circle '10' and an arrow pointing to it. The 'Callback URL' field contains the URL `https://api.cloud-elements.com/elements/api-v2/events/freshservice/{INSERT_64BASE_ENCODED_INSTANCE_ID}`, with a red circle '11' and an arrow pointing to it. Below the URL are options for 'Requires Authentication', 'Encoding' (JSON, XML, X-FORM-URLENCODED), and 'Content' (Simple, Advanced). The 'Content' section is highlighted in yellow and contains a list of ticket properties: 'Ticket ID {{ticket.id}}', 'Subject {{ticket.subject}}', 'Description {{ticket.description}}', and 'Ticket URL {{ticket.url}}'. A red circle '12' and an arrow point to this list. At the bottom, there is an 'Add new action' button.

Events are now implemented for Freshservice.