

BrightTALK Authenticate a Connector Instance

Last Modified on 03/16/2020 5:52 am EDT

On this page

You can authenticate with BrightTALK to create your own instance of the BrightTALK connector through the UI or through APIs. Once authenticated, you can use the connector instance to access the different functionality offered by the BrightTALK platform.

Authenticate Through the UI

Use the UI to authenticate with BrightTALK and create a connector instance. You will need your BrightTALK username (entered in the **API Key** field) and password (entered in the **API Secret** field).

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 resource](#), or [use it in a formula template](#).

Authenticate Through API

You can provision instances through SAP Cloud Platform Open Connectors or through APIs. This section describes both methods and includes parameter details, a request sample, and a response sample.

To provision your BrightTALK Connector, use the `/instances` API.

Note: token is returned upon successful execution of this API. Retain the token for all subsequent requests involving this connector instance.

The provisioning API call includes:

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

Request Body

You must include an instance.json file with your instance request. See [Parameters](#) for information about each parameter. The Boolean parameters show default values.

Note: If you don't specify a required parameter, your response results in an error.

```
{
  "element": {
    "key": "brighttalk"
  },
  "configuration": {
    "brighttalk.api.key": "",
    "brighttalk.api.secret": "",
    "brighttalk.isstaging": false,
    "brighttalk.bulkignore.emailnotpresent": true,
    "filter.response.nulls": true
  },
  "tags": [
    ""
  ],
  "name": ""
}
```

Note: Make sure that you use straight quotes in your JSON files and cURL commands. Use plain text formatting in your code and ensure that you do not include spaces in the cURL command.

Parameters

The following table shows the parameters used to provision an instance. We noted optional parameters in the table, but if you do not specify values, the connector will be provisioned with default values for the optional parameters.

Parameter	Description	Data Type
Instance Name <code>name</code>	The name for the instance created during provisioning.	String
API Key <code>brighttalk.api.key</code>	The code that identifies the calling connector instance to BrightTALK. This is your BrightTALK username.	String

Parameter	Description	Data Type
API Secret <code>brighttalk.api.secret</code>	The secret associated with the API Key that identifies the calling connector instance to BrightTALK. This is your BrightTALK password.	String
Staging <code>brighttalk.isstaging</code>	<i>Optional.</i> Indicates whether the instance should connect to a production or staging environment. No or <code>false</code> indicates production. Default: <code>false</code>	Boolean
Bulk ignore if email not present <code>brighttalk.bulkignore.emailnotpresent</code>	<i>Optional.</i> Specifies what to do with a record during bulk download if it is missing an email . Set to Yes or <code>true</code> to ignore the record during download. Default: <code>true</code>	Boolean
Filter null values from the response <code>filter.response.nulls</code>	<i>Optional.</i> Determines if null values in the response JSON should be filtered from the response. Yes or <code>true</code> indicates that SAP Cloud Platform Open Connectors will filter null values. Default: <code>true</code>	Boolean
Tags	<i>Optional.</i> User-defined tags to further identify the instance.	object

Sample Request

Below is an example cURL request:

```
curl -X POST -H "Content-Type: application/json" -H "Authorization: User rapG956KSQJ/1Zo20kdg2uurkG+wYU836miXluQQT4k=, Organization 976c3406ee321baf50acfe6cf5eac1ac" -H "Cache-Control: no-cache" -H "Postman-Token: 5b56f6d5-e58e-d53f-a757-671f7cab58fa" -d '{
  "element": {
    "key": "brighttalk"
  },
  "configuration": {
    "brighttalk.api.key": "xxxxxx",
    "brighttalk.api.secret": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "brighttalk.isstaging": false,
    "brighttalk.bulkignore.emailnotpresent": true,
    "filter.response.nulls": true
  },
  "tags": [
    "Docs"
  ],
  "name": "BrightTalkforDocs"
}' "https://api.openconnectors.us2.ext.hana.ondemand.com/elements/api-v2/instances"
```

Sample Response

Below is an abridged successful JSON response:

```

{
  "id": 411183,
  "name": "BrightTalkforDocs",
  "createdDate": "2017-03-14T19:20:16Z",
  "token": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxx=",
  "element": {
    "id": 184,
    "name": "BrightTALK",
    "hookName": "BrightTALK",
    "key": "brighttalk",
    "description": "Add an BrightTALK Instance to connect your existing Bri
ghtTALK account to the Marketing Hub, allowing you to manage your channels,
subscribers, activities, etc. You will need your BrightTALK account informa
tion to add an instance.",
    "image": "https://www.brighttalk.com/resources/images/logo_b.png?v=973
73",
    "active": true,
    "deleted": false,
    "typeOauth": false,
    "trialAccount": false,
    "signupURL": "http://www.brighttalk.com",
    "defaultTransformations": []
  },
  "configuration": {
    "brighttalk.api.secret": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "filter.response.nulls": "true",
    "brighttalk.bulkignore.emailnotpresent": "true",
    "brighttalk.isstaging": "false",
    "brighttalk.api.key": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  }
}

```

Instance Configuration

The content in the `configuration` section or nested object in the body posted to the `POST /instances` or `PUT /instances/{id}` APIs varies depending on which connector is being instantiated. However, some configuration properties are common to all connectors and available to be configured for all connectors. These properties are -

- `event.notification.enabled` : This property is a `boolean` property, and determines if event reception (via `webhook` or `polling`) is enabled for the connector instance. This property defaults to *false*.
- `event.vendor.type` : When `event.notification.enabled` property is set to *true*, this property determines the mechanism to use to receive or fetch changed events from the service endpoint. The supported values are `webhook` and `polling` . Most

connectors support one mechanism or the other, but some like Salesforce.com support both mechanisms. This property is *optional*.

- `event.notification.type` : This property can be used to determine how an event notification should be sent to the consumer of the connector instance, in most cases your application. Currently, `webhook` is the only supported value for this property. This means that when an event is received by the connector instance, it will get forwarded to the provided `event.notification.callback.url` via a `webhook` to you. This property is *optional*.
 - `event.notification.callback.url` : As mentioned above, the value of this property is an `http` or `https` URL to which we will post the event for consumption by your application. This property is *optional*.
 - `filter.response.nulls` : This property defaults to *true*, i.e., it's `boolean` property, and determines if `null` values in the response `JSON` should or should not be filtered from the response returned to the consuming application. By default, all `null` values are filtered from the response before sending the response to the consuming application.
-