SmartHub.ai

SmartHub INFER IoT Center

v3.0.0

API Guide

You can find the most up-to-date technical documentation at: https://www.smarthub.ai/

SmartHub Inc.
4332 Holt Street, Union City, CA, 94587, USA
www.smarthub.ai

Table of contents

1 Introduction to the API

The SmartHub INFER IoT Center API offers a way for third-party systems to interact with SmartHub INFER IoT Center.

SmartHub INFER IoT Center provides programmable REST APIs to integrate with your existing enterprise solution. With the SmartHub INFER IoT Center APIs, you can create, view, edit, and delete various SmartHub INFER IoT Center entities such as Devices, Campaigns, Alerts, Notifications, Groups, and Users, programmatically. All the core features of the SmartHub INFER IoT Center provide REST APIs.

1.1 Terminology

This document uses the following terms.

API: API is an acronym for Application Programming Interface. It is a name used to refer to a special framework some web applications or services provide which allows a user to connect to the system and perform some number of discrete actions such as running functions, requesting data, or updating information.

Call: A call is another name for a request or a communication sent by a user to the API, in the form of a URL string, which invokes a specific action on one particular endpoint, and can also include additional parameters or values.

REST: REST is an acronym for Representational State Transfer. It is a form of software architecture that is primarily used for designing a web service.

HTTP: HTTP is an acronym for Hyper-Text Transport Protocol. It is one of the key architectural components behind how web-based content on the Internet is accessed through web browsers.

JSON: JSON is an acronym for JavaScript Object Notation. It is a format for information, based on the JavaScript language, that is intended for consumption by a programmed function.

Method: HTTP provides support for four methods which each describe a type of result a user might want to achieve through a given communication with a web server or API. The four methods are:

GET	GET is used for retrieving or querying records from a system.
PUT	PUT is used for inserting or creating records into a system.
POST	POST is used for updating existing records in a system.
DELETE	DELETE is used for removing records from a system.

Note: The required permissions that are listed for each API are for invoking the API operation. Some APIs require additional permissions to complete the operation.

1.2 APIs provided by SmartHub INFER

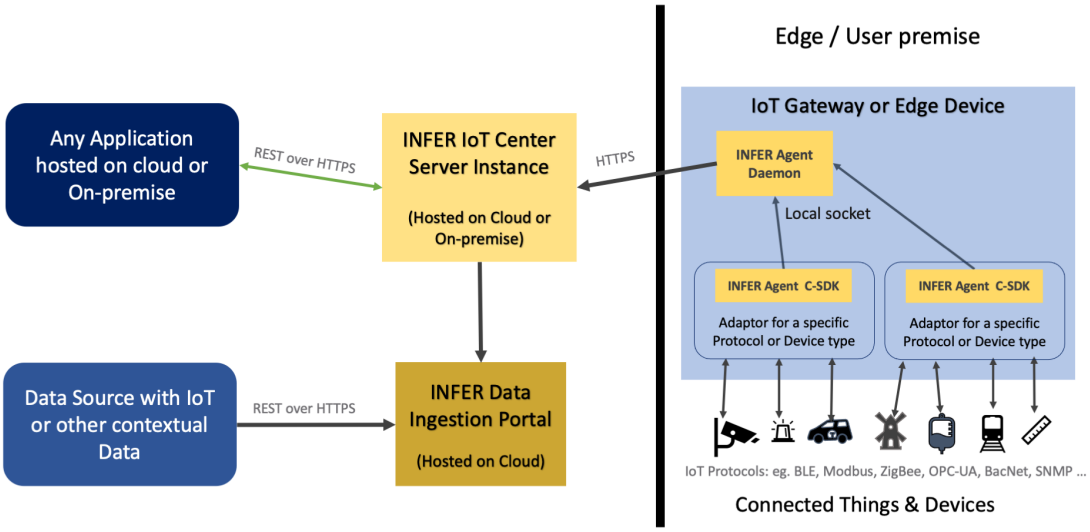

Figure 1-1. SmartHub INFER API Overview Diagram

SmartHub INFER provides 3 APIs for 3 different types of Integration as depicted in the above diagram.

1.2.1 Server APIs

This is a set of REST APIs that are provided by SmartHub INFER Server to enable applications to read data stored in INFER, write data into INFER and control actions performed by INFER on IoT & Edge devices. All of the functionality provided by SmartHub INFER Console (UI) is implemented using this same set of REST APIs. Note that the consumer of these APIs needs to have inbound HTTPS access to the INFER instance - regardless of whether it deployed on-premise or in the cloud.

1.2.2 Data Ingestion APIs

This is a set of REST APIs provided by **INFER Data Ingestion Portal**, which is hosted in the cloud with an unchanging URL. It that enables third party applications to send device-related data, Alert data or other contextual data into a central place which then gets forwarded to its intended INFER Instance. The INFER instance can be running in the Cloud or on-premise.

1.2.3 Agent APIs

This is a set of C Functions that is provided as an SDK library to enable Edge Adapters or other applications running at the Edge to inject IoT data into INFER via INFER Agent Daemon. The SDK library can be consumed by C/C++, Python, Go or other language programs that implement a particular protocol to interface with devices.

1.3 User Authentication

Use the following APIs to create and issue an authentication token for a user.

1.3.1 Acquire Token

Header

If the user name is available across multiple organizations, enter the following header:

```
x-org-domain-name:<domain>
```

API `/api/tokens`

Method `GET`

Required Parameters `None`

Response

```
{
    "accessToken": “string”,
    "expiresInSecs": "1543317540",
    "accessTokenExpiresAt": "1543317540",
    "refreshToken": “string”,
    "refreshTokenExpiresAt": "1544519940"
}
```

1.3.2 Issue Access Token Using Refresh Token

API `/api/tokens/refresh`

Method `GET`

Required Parameters `None`

Response

```
{
  "accessToken": "string",
  "accessTokenExpiresAt": "1543317540"
  "expiresInSecs": 0,
  "refreshToken": null
}
```

1.4 Device Authentication

Use the following APIs to issue device token and device credentials.

1.4.1 Create Device Credential

Required Permissions

You must have the **Create Device Credential** permission to perform this operation.

API `api/device-credentials/{id}`

Method `POST`

Input Examples

JWT_NATIVE	`{}`
PROPERTY_NATIVE	`{"requestParams":"{\"DeviceKey\":\"1234\"}"}`
TPM_NATIVE	`{"requestParams":"{\"tpm_ek\":\"123456\"}"}`

Response

```
{
  "credentials": "string"
}
```

1.4.2 Get Device Token

Required Permissions

You must have the **Get Device Token** permission to perform this operation.

API `/api/device-tokens`

Header `x-device-auth`

Enter the device credential that you created in the Create Device Credential API.

Method `GET`

Required Parameters

Name	Type	Description
`id`	string	Device ID

Response

```
{
  "deviceId": "string",
  "accessToken": "string"
}
```

1.5 HTTP Response Codes

The following table lists the response codes used by SmartHub INFER IoT Center. The response codes adhere to the standard HTTP and REST conventions.

HTTP Status Codes

Status Code	Error Message
200	Success
401	Unauthorized
403	Permission denied
404	Not found

1.6 API Headers

APIs must include the following headers:

1.6.1 API Version

```
Content-Type: application/json
Accept: application/json;api-version=<api-version>
```

To get the current API version, use the following API:

API `/api/versions`

Method `GET`

Sample Response

```
{
  "currentApiVersion": "0.2",
  "supportedApiVersions": [
    "0.1",
    "0.2"
  ]
}
```

1.6.2 User Access Token

You require a user access token to perform API operations.

```
Authorization : {UserAccessToken}
```

For information about generating user access tokens, see the User Authentication section.

1.6.3 Set the Current Organization ID

Use this header to set the current organization ID for which you want to run the APIs.

```
x-current-org-id:<orgId>
```

1.7 Restricted Characters

The following characters are restricted when creating a template name, device name, custom property, and metric name.

1.7.1 Template Name

```
< > % $ ( ) { }
```

1.7.2 Device Name

```
< > % $ ( ) { } [ ]
```

1.7.3 Custom Property

```
< > . % $ ( ) { }
```

1.7.4 Metric Name

```
: { } & "
```

2 Server APIs

This is a set of REST APIs that are provided by SmartHub INFER Server to enable other applications to read data stored in INFER, write data into INFER and control actions performed by INFER on IoT & Edge devices. Note that the consumer of these APIs needs to have inbound HTTPS access to the INFER instance - regardless of whether it deployed on-premise or in the cloud.

2.1 Swagger UI

SmartHub INFER's Server APIs are *OpenAPI* compliant and provides a *Swagger UI* as part the INFER server instance. A Swagger UI provides detailed documentation for all of the RESTful APIs offered by the Server. It also provides the ability to invoke the APIs interactively with a live server connection. More details about Swagger UI can be found at https://swagger.io

Users of SmartHub INFER can access the swagger UI by pointing their browser to `https://<INFER-SERVER-FQDN>/openapi/index.html` where `<INFER-SERVER-FQDN>` is the Fully Qualified Domain Name or IP address of the INFER Server Instance.

2.2 Using the Server API

To invoke APIs on the Server, one needs to be authenticated first. This is done by invoking the following APIs in sequence.

1. Ensure that the **Servers** drop-down list at the top-left of the page is showing the correct FQDN for the INFER instance you would like to connect to.
2. Click the **Authorize** button in the top right corner to perform `Basic Auth` to the INFER server. Click **Close** once you are logged in.
3. Next, get the API versions supported by the server by invoking API Version call. For all subsequent calls, ensure that you provide a supported version in the **Accept header**.
4. Acquire an **Access Bearer token**, that you can use for all subsequent calls in the session. This can be done by invoking the Acquire Token call. The response will provide two tokens - `accessToken` and `refreshToken` .
5. Authorize all other REST calls using the value of `accessToken` as *BearerAuth* token. You can do this by clicking the gray-colored lock at the right, for any of the REST APIs to open the popup and enter the value of `accessToken` returned by Acquire Token call.

2.3 Types of Server APIs

The rest of this section provides a high-level overview of the Server APIs along with some sample use cases.

- **Device Management** - APIs for performing operations on devices, device templates, device authentication, device commands, and files
- **Campaign Management** - Create, Modify, Delete, Start, Stop and other operations on Campaigns. Create, Delete and management of Packages
- **Alerting** - APIs to get, create, update, and delete alerts and alert definitions
- **Identity and Access Management (IAM)** - APIs to perform tenant management, sub-org management, role management, user management, and group management operations
    - **Organization Management** - Create, view, update, and delete an organization

    - **User Management** - APIs to create, fetch, update, and delete users
    - **Role Management** - APIs to create, fetch, update, and delete roles
    - **Group Management** - APIs to create, fetch, update, and delete user groups
    - **Permission Management** - API to fetch permissions
    - **Organization Settings** - APIs to view and update your organization settings
    - **Password Management** - APIs to generate a password recovery link and reset the password
    - **Token Management** - APIs to generate access and refresh tokens
- **Notification** - APIs for Notification Destinations, Notification Definitions, and Notification Instances
    - **Notification Definition** - APIs to create, update, get, and delete notification definitions
    - **Notification Instances** - APIs to retrieve notification instances
    - **System Notifications** - APIs to view system notifications. System notifications are generated when there is a system downtime. The notifications are sent to the users through email or displayed on the UI
- **Certificate Management** - APIs to create, update and delete certificates
- **Advanced Search** - APIs to create, get, update, and delete a filtered device list
- **Metric APIs** - APIs to query metrics
- **Audit APIs** - APIs to get audit logs, get audit types, and get entity types

3 Data Ingestion APIs

This is a set of REST APIs provided by **SmartHub INFER Data Ingestion Portal**, that enables third party applications to send device, Alert or other contextual data into any SmartHub INFER. The Data Ingestion portal is hosted in the cloud with an unchanging URL and can scale to accomodate data ingestion for any number of INFER instances.

As the Data Ingestion Portal is a single public entry point for data to be sent into any INFER instance, users of this portal need to authenticate themselves using an **API Key** and tag the data using a combination of **Client Instance** and **Datastream Instance**. More details are provided below.

3.1 API Specification

URL Endpoint

`https://vmi.us02.smarthubai.net`

API

`/api/valert`

Header

This API uses an **API key** as its authentication mechanism and it needs to be specified in the request header for every invocation. Please contact SmartHub.ai to get your API key.

```
x-api-key:  <YOUR KEY>
```

Request Method

`POST`

Request JSON structure

```
{
  "viid":"< YOUR ORG IDENTIFIER >",
  "diid":"<YOUR value>",
  "timeStamp":"",
  “payLoads”: [
    {
      "payLoadType":"Alert",
      "alertType": "tailgating",
      “alertID”: “214495-bafc-24-4fcdaa“,
      "location":"",
      "cameraName":"",
      "cameraIP":"",
      "modelNumber":"",
      "serialNumber":"",
      "firmware":"",
      "alertStatus":"",
      "severity":"",
      "securityProfile":"",
      "alertStatus":"",
      "severity":"",
      .....
   },
   {
      "payLoadType":"Properties",
      "deviceType":"Camera",
      "location":"",
      "cameraName":"",
```

```
      "cameraIP":"",
      "modelNumber":"",
      "serialNumber":"",
      "firmware":"",
      "alertStatus":"",
      "severity":"",
      "securityProfile":"",
      "alertStatus":"",
      "severity":"",
      .....
    }
  ]
}
```

payLoads is an array of one or more payloads

payLoadType can have one of the following values:

Properties	When properties of the device needs to sent at a specific frequency
Metrics	When metrics collected from the device needs to sent in a time-series mode
Event	Specific events related to the application/device, only if applicable
Alert	Alerts in the source application to be sent as alerts into SmartHub INFER platform

These are the **Mandatory keys** that need to be present in the JSON for every call:

viid	Identifies your client instance
diid	Identifies your datastream instance to the SmartHub INFER platform
timeStamp	Timestamp of the occurrence in the source system
payLoads	Array of payload objects
payLoadType	Identifies the payload type (One for each of the payLoad object in the array)

3.2 API Invocation example using curl

```
curl --request POST \
  --url https://vmi.us02.smarthubai.net/api/valert \
  --header 'content-type: application/json;api-version=0.17' \
  --header 'x-api-key:  <YOUR KEY> ' \
  --data '<JSON INPUT>'
```