

B1 IoT – How to setup SAP Cloud Platform IoT Services

STEP 1: INTERNET OF THINGS SERVICE SETUP

This is a very quick and simple walkthrough to setup your Internet of Things Service within your [SAP Cloud Platform account](#), if you haven't already done so.

If you have, you may skip to [Step 2: Adding your SensorTag Device in the Internet of Things Cockpit](#).

The IoT services in SCP is comprised of two main areas:

Remote Device Management Service (RDMS) – this is used to onboard new devices, like our iPhone, SensorTag, etc.

Message Management Service (MMS) – this is used to manage messages between device, cloud platform and business applications. This is also where we will see the sensor data stored in SAP Cloud Platform.

Enabling the IoT Service for your SAP Cloud Platform Account

This will give you access to the Internet of Things Cockpit which leads you to the Remote Device Management Service.

First go to your SAP Cloud Platform Account and select **Services** from the side menu.

Note: Make sure you are at the account level at the top. Then on you may see the Services in the left menu.

The screenshot shows the SAP Cloud Platform Cockpit interface. On the left, a sidebar menu is open under the 'Services' category. The 'Internet of Things' service is highlighted and expanded, showing its details. The service is labeled 'Enabled' and has a brief description: 'Develop, customize, and operate IoT business applications in the cloud.' To the right, other services are listed: 'Remote Data Sync' (Enabled) and 'Development & Operations, std' (Enabled). The 'Development & Operations, std' service has a description: 'Build and run mobile apps for B2E and B2B use cases.'

Scroll down until you see, **Internet of Things Services** and then click “Enable”.

SAP Cloud Platform Cockpit

Overview Home [Europe (Rot) - Trial] / Europe (Rot) - Trial / p1558868588 / Internet of Things

Service: Internet of Things - Overview

Enabled

Disable

Service Description

The SAP Cloud Platform Internet of Things service for the Cloud Foundry environment connects devices to SAP Cloud Platform to provide scalable ingestion of IoT data and device management. The respective services provide a secure connection to remote devices using a broad variety of IoT protocols and manage the device lifecycle from onboarding to decommissioning.

Take Action

Configure Service

Go to Service

You now have enabled the IoT Services for your SCP account. The “IoT-User” role has now been assigned to you. Next, we will install the Message Management Service (MMS).

Deploying the Message Management Service (MMS)

As mentioned above, this is used to manage messages between your iPhone, the platform, and business applications. This is also where we will see the sensor data stored in SAP Cloud Platform. From your SAP Cloud Platform Cockpit, let's go back to the **Internet of Things Service**. We will enter the IoT Service Cockpit by clicking “**Go to Service**“.

You will now see “**Deploy Message Management Service**” on the bottom left tile. Click on that.

Internet of Things Services Cockpit

Device Management

3 Device Types

5 Message Types

2 Devices

All Device Types All Message Types All Registered Devices

Message Management

Deploy Message Management Service

Send and view messages, or perform other actions

Enter your account ID, username, and password.

Note: Your account ID is likely your (i/d/p) number or email address with the word “trial” on the end. Your User Name is likely to be your (i/d/p) number or email address itself.

The screenshot shows the deployment configuration for the Message Management Service. It includes fields for Host (https://hanatrial.ondemand.com), Account ID (p1558868588trial), User Name (P1558868588), and Password (*****). A green 'Deploy' button is at the bottom right.

Deploy the Message Management Service

The Message Management Service receives and processes messages sent from devices. In addition, it provides interfaces to push messages to devices. The component will be deployed on the SAP Cloud Platform user account specified.

Account Settings

Host: https://hanatrial.ondemand.com

Account ID: p1558868588trial

User Settings

User Name: P1558868588

Password: *****

Deploy

We now have deployed the IoT Services application and the Message Management Service (MMS). Lastly, we need to add the IoT-MMS-User Role to our UserID.

Adding the IoT-MMS-User Role to your User ID

From the SAP Cloud Platform Cockpit, select **Java Applications** and click on **iotmms**. This will take us to the Java Application Overview screen.

The screenshot shows the Java Application Overview screen in the SAP Cloud Platform Cockpit. The sidebar lists various categories like Overview, Applications, Java Applications, etc. The main area shows a table of applications with columns for State and Name. One application, 'iotmms', is currently starting.

SAP Cloud Platform Cockpit

Overview Applications Java Applications HTML5 Applications HANA XS Applications Subscriptions Services Solutions SAP HANA / SAP ASE Connectivity Security Repositories Integration Tokens

Home [Europe (Rot) - Trial] / Europe (Rot) - Trial / p1558868588

Subaccount: p1558868588 - Java Applications

All: 4

Deploy Application

State	Name
Stopped	helpdesk
Starting	iotmms
Stopped	s3
Stopped	simplesensortag

On the left-hand side, select **Security > Roles**.

We now see the predefined **IoT-MMS-User** Role at the top.

The screenshot shows the SAP Cloud Platform Cockpit interface. The left sidebar has 'Overview', 'Monitoring', 'Configuration', and 'Security' sections, with 'Roles' selected. The main content area shows a breadcrumb path: Home [Europe (Rot) - Trial] / Europe (Rot) - Trial / p1558868588 / iotmms. The title is 'Java Application: iotmms - Roles'. Below it, a table lists 'Roles (All: 1)'. A single row is shown: Name 'IoT-MMS-User' and Type 'Predefined'. Below the table, a note says 'IoT-MMS-User Predefined: Provisioned by the application'. At the bottom, there's a table for 'Individual Users' with one entry 'User ID: p1558868588' and an 'Actions' column containing a 'Unassign' button.

Verify that the IoT-MMS-Role is selected. Then below, click on “**Assign**”

Add in your User ID and select “**Assign**“.

Note: Your User ID is not your account name. The user ID doesn't not include the word “trial” on the end.

The screenshot shows the SAP Cloud Platform Cockpit interface with the 'Roles' page. A modal dialog box titled 'Assign role "IoT-MMS-User" to user' is open. It contains a 'User ID' field with the value 'p1558868588' and a note 'Note: Changes will affect new sessions only'. At the bottom right of the dialog are 'Assign' and 'Cancel' buttons.

Now we have added the Internet of Things Service, installed the MMS, and assigned our user to the IoT-MMS-User Role. Next, we will add our SensorTag Device!

STEP 2: ADDING YOUR SENSORTAG DEVICE IN THE INTERNET OF THINGS COCKPIT

From your SAP Cloud Platform Cockpit, let's go back to the **Internet of Things Service**. We will enter the IoT Service Cockpit by clicking "**Go to Service**". We are launching the Internet of Things Cockpit, within SAP Cloud Platform. The Internet of Things Services cockpit is the main interface for users to interact with the Remote Device Management Service (RDMS). It can be used to register new Devices (like our SensorTag device), to define the schema of messages (Device Types and Message Types) they can send and/or receive, as well as to establish the necessary trust relationship needed by devices to interact with Message Management Service (MMS).

As mentioned above, in order for our SensorTag device to successfully communicate with SAP Cloud Platform, there are a few prerequisites. The internet of Things Service requires a specific syntax used to register new devices in the device registry. We will need to create 3 entities:

Device Type: A device type specifies a group of devices that share the same specification. In the IoT services, this specification is interpreted as the capability of sending, receiving and processing certain types of messages from our SensorTag Device.

Message Type: A message type defines the type of data contained in a message that is sent or received by a device. Message types are specified for device types. For example, we will need to create a Message Type with the following attributes: sensor, value, and timestamp.

Device: A device is a physical object that can be registered with IoT services. After successful registration it is able to send or receive messages. This is of course our SensorTag device itself.

So, let's begin!

From your SAP Cloud Platform Cockpit, let's go back to the **Internet of Things Service**. We will enter the IoT Service Cockpit by clicking "**Go to Service**".

The screenshot shows the SAP Internet of Things Service Cockpit interface. At the top, there is a header bar with the SAP logo, the title "Internet of Things Service Cockpit", and a user profile for "Jacob Tan (P1558868588)". Below the header, the interface is divided into two main sections: "Device Management" and "Message Management".

Device Management: This section contains three cards:

- "All Message Types": Shows 4 items, represented by a speech bubble icon.
- "All Device Types": Shows 3 items, represented by a gear icon.
- "All Registered Devices": Shows 3 items, represented by a smartphone icon.

Message Management: This section contains two cards:

- "Redeploy the Message Management": Shows an "Up to Date" status with a download icon.
- "Send and view messages, or perform other": Shows an MMS Cockpit icon.

At the bottom left, there is a blue button with a warning icon and the number "1". At the very bottom of the screen, there is a footer bar with links: Legal Disclosure, Privacy, Terms of Use, Copyright, Trademark, and SAP.com.

2.1: Creating our Message Type

We will create a Message Type for the SensorTag device. Click on the “Message Types” tile.

Click on the plus (+) button at the bottom.

Name the new Message Type (B1IoT SensorTag Message Type) for the new device.

For the Fields, click the plus (+) button to add more messages types (to have 9 in total) and data types. (of course, you may add more in the future)

timestamp: date

dev: string

accx: string

accy: string

accz: string

alt: string

long: string

lat: string

lux: string

Then press “Create” (bottom right corner).

The screenshot shows the 'Message Types' section of a software interface. On the left, there's a sidebar with a list of existing message types: 'B1IoT SensorTag Message Type' (selected), 'climateData', 'iOS Message Type', 'MessageFromDevice', and 'MessageToDevice'. The main area is titled 'B1IoT SensorTag Message Type' and contains two tabs: 'Information' (selected) and 'Fields'. The 'Fields' tab displays a table with 9 rows, each representing a field with its position, name, type, and optional settings. The fields listed are: 1. timestamp (date), 2. dev (string), 3. accx (string), 4. accy (string), 5. accz (string), 6. alt (string), 7. long (string), 8. lat (string), and 9. lux (string). At the bottom of the screen, there are three buttons: a blue '+' button, a red warning icon with '1', and a 'Delete' button.

Position	Name	Type	Optional Settings
1	timestamp	date	
2	dev	string	
3	accx	string	
4	accy	string	
5	accz	string	
6	alt	string	
7	long	string	
8	lat	string	
9	lux	string	

Please note your Message Type ID. I recommend you copy/paste somewhere for easy access later on when we need it.

The screenshot shows the SAP IoT Service Cockpit interface. On the left, there's a sidebar with a back arrow, the title 'Message Types', and search/filter icons. Below this are three entries:

- B1IoT SensorTag Message Type**
ID: 2de0e4992cd7b86ad115
- climateData**
ID: a1d6dbb3b5bf0c68025a
- iOS Message Type**
ID: e137ac01638076ce523c

On the right, a detailed view for the 'B1IoT SensorTag Message Type' is displayed. It shows the message type name and ID. Below the name are two circular icons: one with an 'i' labeled 'Information' and another with a document icon labeled 'Fields'. At the bottom right of this view is the ID: 2de0e4992cd7b86ad115.

2.2: Creating our Device Type

Press the back arrow (top left) to return to the IoT Services Cockpit. Then press the “**Devices Type**” tiles.

The screenshot shows the SAP IoT Service Cockpit interface. The top navigation bar includes the SAP logo, the title 'Internet of Things Service Cockpit', a refresh icon, and a user dropdown for 'Jacob Tan (P1558868588)'.

The main area is divided into sections:

- Device Management**: Contains three cards with counts:
 - All Message Types: 5 (Message Types)
 - All Device Types: 4 (Device Types)
 - All Registered Devices: 3 (Devices)
- Message Management**: Contains two cards with links:
 - Redeploy the Message Management: Up to Date (with a download icon)
 - Send and view messages, or perform other: MMS Cockpit (with a cluster icon)

Then click on the Plus (+) symbol at the bottom to add a new device type.

Name your device type (B1 IoT SensorTag Device Type), assign it (by clicking Add Message Type) with the Message Type we just created, and chose **From Device** for the direction. press "Create".

The screenshot shows the 'Create Device Type' page. On the left, there's a sidebar with a search bar and three listed device types: 'Device Simulator' (ID: 2f68f7d4673d5dfd5e1e), 'iOS Device Type' (ID: e2eff2627c29adbbdfba), and 'TesselClimate' (ID: 21a6c19c5efd62b1c454). The main area has tabs for 'Information' and 'Message Types'. In the 'Information' tab, the 'Name' field is filled with 'B1 IoT SensorTag Device Type'. In the 'Message Types' tab, there's a table with columns 'Assignment Name', 'Message Type', and 'Direction'. A new row is being added, with 'Assignment Name' empty ('Enter a name (optional)'), 'Message Type' set to 'B1IoT SensorTag Message Type', and 'Direction' set to 'From Device'. At the bottom, there are 'Create' and 'Cancel' buttons, and a status indicator showing '1 warning'.

Assignment Name	Message Type	Direction
<input type="text" value="Enter a name (optional)"/>	B1IoT SensorTag Message Type	From Device

Create **Cancel**

⚠ 1

2.3: Creating our Device

Press the back arrow (top left) to return to the IoT Services Cockpit. Then press the “**Devices**” tiles.

The screenshot shows the SAP Internet of Things Service Cockpit interface. At the top, there is a header bar with the SAP logo, the title "Internet of Things Service Cockpit", a refresh icon, and a user profile for "Jacob Tan (P1558868588)". Below the header, the main content area is divided into two main sections: "Device Management" and "Message Management".

Device Management:

- All Message Types: 5 Message Types (represented by a speech bubble icon)
- All Device Types: 4 Device Types (represented by a gear icon)
- All Registered Devices: 3 Devices (represented by a smartphone icon)

Message Management:

- Redeploy the Message Management: Up to Date (represented by a download icon)
- Send and view messages, or perform other: MMS Cockpit (represented by a cluster of circles icon)

Once again, press the (+) button on the bottom to create a new device. Give it a name (B1 IoT SensorTag Device), select the Device Type we created earlier (SensorTag Device Type) and the press “Create” (bottom right).

The screenshot shows the 'Create Device' interface. On the left, a sidebar lists existing devices: DevelopmentTessel, DeviceSimulator, and iOS Device. The main area is titled 'Information' and contains fields for 'Name' (B1 IoT SensorTag Device), 'Device Type' (B1 IoT SensorTag Device Type), and 'Further Details' (Enter a URL to call up further details). Below this is a 'Custom Attributes' section with a 'Add Custom Attribute' button. At the bottom, a status bar shows a warning icon with '1' and buttons for 'Create' and 'Cancel'.

*Important: After you create the device, a pop up will appear with a token value. Copy/Paste this as well because we need this to input into our B1 IoT app. If you lose it, click into the **Authentication** tab and you may generate a new token.*

The screenshot shows the 'Devices' list with the newly created 'B1 IoT SensorTag Device' selected. A modal window provides detailed information about the device, including its ID, Device Type (B1 IoT SensorTag Device Type), and Authentication Type (OAuth). A sub-modal window shows the 'OAuth Access Token' has been created, with the token value displayed as 'ada6a44d4840efe9ba5ba8e1fdfcae5a'. Buttons for 'Close' and 'OK' are visible at the bottom of the sub-modal.

Also, please note your Device ID after you click “Close”. You will see the “ID” listed under your “Device Type” on your “Device Page”.

I recommend you copy/paste somewhere for easy access later on when we need it. This will be needed to build your URL endpoint in the next step.

The screenshot shows a web-based interface for managing devices. On the left, there is a sidebar titled "Devices" with a search bar and a refresh icon. Below the search bar, a list of devices is displayed:

- B1 IoT SensorTag Device
ID: 7d6a8683-e692-4c3f-aba2-31ddbf41bd0
- DevelopmentTessel
ID: c6974a51-2a29-453e-b56e-1b87e3ef2510
- DeviceSimulator
ID: 2f5c5ccc-d939-4749-9af6-e705bcd337e1
- iOS Device
ID: 6784af44-d8bc-441d-8d2c-eb77fc1e0e4b

On the right, a detailed view of the "B1 IoT SensorTag Device" is shown. The title "B1 IoT SensorTag Device" is at the top. Below it are two circular icons: one with an "i" for "Information" and one with a shield for "Authentication". To the right of these icons, the device's ID and type are listed:

ID: 7d6a8683-e692-4c3f-aba2-31ddbf41bd0
Device Type: **B1 IoT SensorTag Device Type**
Authentication Type: OAuth

At the bottom of the interface, there are several buttons: a gear icon, a plus sign, a warning icon with a "1", and links for "Open Chart", "Edit", and "Delete".

Getting your HTTP API Datapoint

Go back to your IoT Service Cockpit > Click on **MMS Cockpit** (Bottom right)

The screenshot shows the SAP Internet of Things Service Cockpit interface. At the top, there's a header bar with the SAP logo, the title "Internet of Things Service Cockpit", a refresh icon, and a user dropdown for "Jacob Tan (P1558868588)". Below the header, the main content area is divided into two main sections: "Device Management" and "Message Management".

Device Management: This section contains three cards:

- All Message Types: Shows 5 message types.
- All Device Types: Shows 4 device types.
- All Registered Devices: Shows 4 registered devices.

Message Management: This section contains two cards:

- Update the Message Management: Includes a download icon and text "New Version: 2.56.0".
- Send and view messages, or perform other: Includes a gear icon and text "MMS Cockpit".

Click on **HTTP API** (2nd row first Tile)

The screenshot shows the SAP Message Management Service Cockpit interface. At the top, there's a header bar with the SAP logo, the title "Message Management Service Cockpit", and a user dropdown for "Jacob Tan (P1558868588)". Below the header, the main content area is divided into two main sections: "Core Services" and "Data Services".

Core Services: This section contains five tiles:

- Display Stored Messages (Application Data)
- Push Messages to Devices (Push Service)
- Configure the Message Management Service (Configuration)
- Configure Processing Service Mappings (Configuration)
- View Registered Devices And Device Types (IoT Service Cockpit)

Data Services: This section contains four tiles:

- Messaging through HTTP (HTTP API)
- Messaging through MQTT over TCP (MQTT TCP API)
- Messaging through WebSockets (WebSocket API)
- Messaging through MQTT over WS (MQTT WebSocket API)

Copy & Paste the Data Endpoint to somewhere which at the end we will edit the parameters.

The screenshot shows the SAP Cloud Platform HTTP API client interface. At the top, there's a back arrow icon and the title "HTTP API". Below that is a "Description" section containing a note about the client's purpose: "This client can be used to test the HTTP API. It can send messages to the Message Management Service by submitting HTTP POST requests to a device-specific HTTP data endpoint. In addition, the client can receive acknowledgments by sending HTTP POST requests to a device-specific HTTP acknowledgement endpoint." Underneath this is a "Send Message" section. It contains two input fields: "Data Endpoint" with the value "<https://iotmmsp1558868588trial.hanatrial.ondemand.com/com.sap.iotservices.mms/v1/api/http/data/d000-e000-v000-i000-c000-e001>" and "Message" with the JSON content: {"mode": "sync", "messageType": "m0t0y0p0e1", "messages": [{"sensor": "sensor1", "value": "20", "timestamp": "1413191650"}]}. To the right of the message field is a "Send" button with a circular arrow icon.

We now have everything setup to use our B1 IoT app with SAP Cloud Platform!

1. **MessageType ID:** e.g. 2de0e4992cd7b86ad112
2. **MessageType fields:** You'd need to know what naming convention you use for each data field (e.g. timestamp, accx, accy, etc.)
3. **Device ID:** e.g. 7d6a8683-e692-4c23-aba2-31ddbfe41bd0
4. **OAuth Access Token:** e.g. ada6a44d4840efe12a5ba8e1fdfcae5a
5. **HTTP API Data Endpoint:** e.g.
<https://iotmmsp1558868588trial.hanatrial.ondemand.com/com.sap.iotservices.mms/v1/api/http/data/replace-this-part-with-your-device-id>
e.g.
<https://iotmmsp1558868588trial.hanatrial.ondemand.com/com.sap.iotservices.mms/v1/api/http/data/7d6a8683-e692-4c23-aba2-31ddbfe41bd0>

2.4 (Optional) Testing with Postman

In this step, we will formulate a simple example on how you can send data through postman to SCP. Basically, in actual scenario, the sensor itself will be streaming data in this format to SCP.

POST

<https://iotmmsp1558868588trial.hanatrial.ondemand.com/com.sap.iotservices.mms/v1/api/http/data/7d6a8683-e692-4c23-aba2-31ddbfe41bd0>

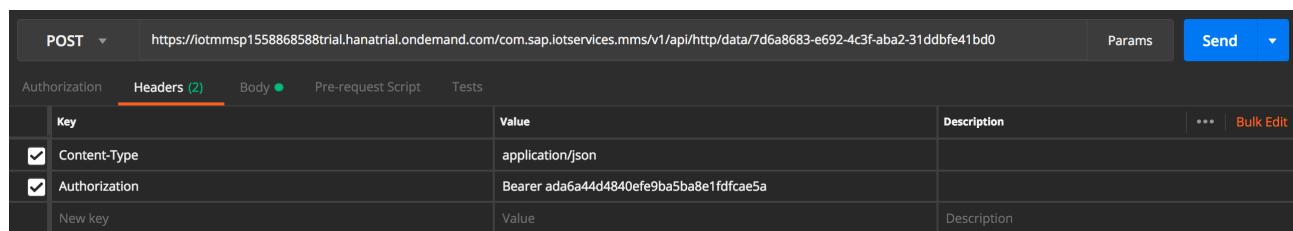
Headers

Content-Type: application/json

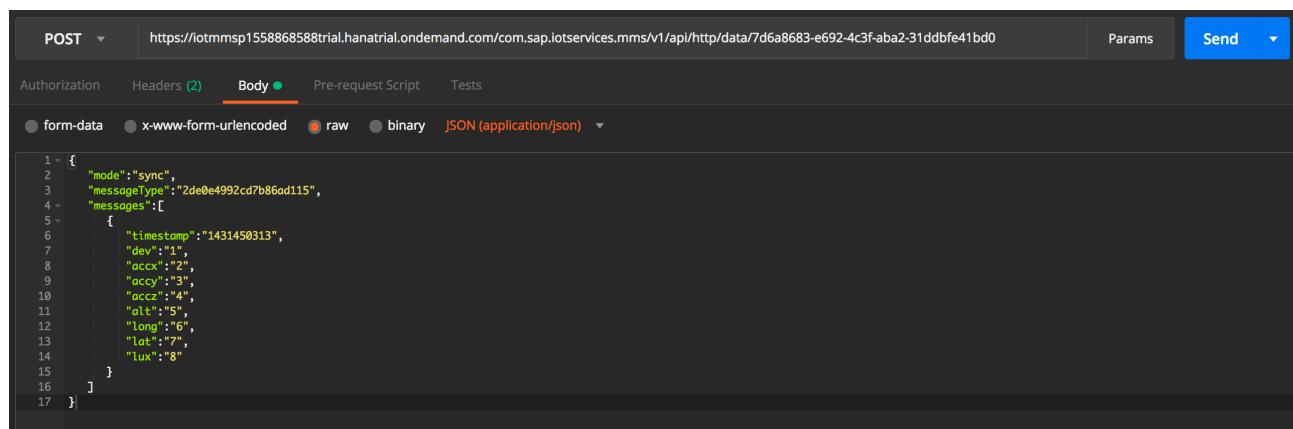
Authorization: Bearer ada6a44d4840efe9ba5ba8e1fdfcae5a

Body:

```
{  
  "mode": "sync",  
  "messageType": "2de0e4992cd7b86ad112",  
  "messages": [  
    {  
      "timestamp": "1431450313",  
      "dev": "1",  
      "accx": "2",  
      "accy": "3",  
      "accz": "4",  
      "alt": "5",  
      "long": "6",  
      "lat": "7",  
      "lux": "8"  
    }  
  ]  
}
```



Screenshot of the Postman Headers tab. The URL is https://iotmmsp1558868588trial.hanatrial.ondemand.com/com.sap.iotservices.mms/v1/api/http/data/7d6a8683-e692-4c23-aba2-31ddbfe41bd0. The Headers section shows two entries: Content-Type (application/json) and Authorization (Bearer ada6a44d4840efe9ba5ba8e1fdfcae5a). There is also a New key entry with Value.



Screenshot of the Postman Body tab. The URL is https://iotmmsp1558868588trial.hanatrial.ondemand.com/com.sap.iotservices.mms/v1/api/http/data/7d6a8683-e692-4c23-aba2-31ddbfe41bd0. The Body section is set to form-data. The raw JSON content is displayed in a code editor:

```
1 ~ {  
2   "mode": "sync",  
3   "messageType": "2de0e4992cd7b86ad115",  
4   "messages": [  
5     {  
6       "timestamp": "1431450313",  
7       "dev": "1",  
8       "accx": "2",  
9       "accy": "3",  
10      "accz": "4",  
11      "alt": "5",  
12      "long": "6",  
13      "lat": "7",  
14      "lux": "8"  
15     }  
16   ]  
17 }
```

STEP 3: SETUP YOUR B1 IOT APP TO SEND SENSOR DATA TO SAP CLOUD PLATFORM

Now that all of the heavy lifting is out of the way, let's have fun with the B1 IoT app. B1 IoT is an app that allows you to visualize your SensorTag sensor data in real-time. B1 IoT allows you to stream this sensor data to the Internet of Things service within SAP Cloud Platform (via HTTP POST).

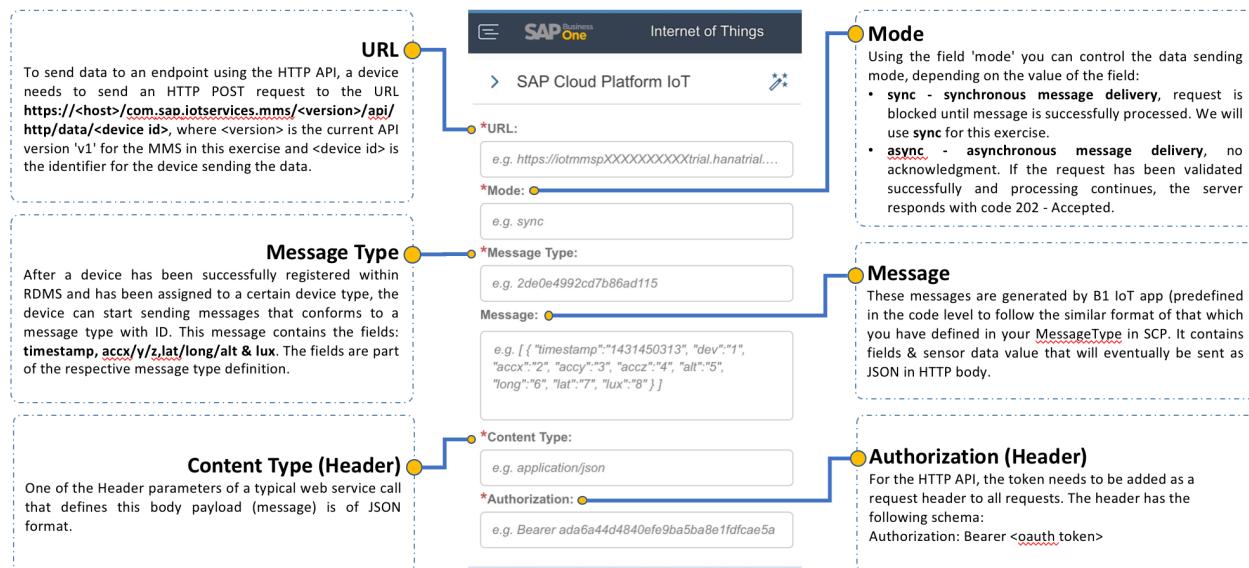
The left screenshot shows the SAP Business One Internet of Things app. It has a header with icons for info, connectivity, location, and alerts. Below it, a section for 'Sensor Data' shows a 'Connected' status with a 'Stream to SCP' button and a switch set to 'OFF'. A 'Movement' section displays X, Y, and Z-axis values from an Accelerometer Sensor. Another section shows 'Thing Position' and 'Light Intensity' with a value of 138 Lux.

The right screenshot shows the 'SAP Cloud Platform IoT' configuration screen. It includes fields for 'URL' (set to [https://iotmmsp1558868588trial.hanatrial.ondemand...](https://iotmmsp1558868588trial.hanatrial.ondemand.com)), 'Mode' (set to 'sync'), 'Message Type' (set to '2de0e4992cd7b86ad115'), and a 'Message' field containing a JSON array: [{ "timestamp": "1431450313", "dev": "1", "accx": "2", "accy": "3", "accz": "4", "alt": "5", "long": "6", "lat": "7", "lux": "8" }]. It also includes fields for 'Content-Type' (set to 'application/json') and 'Authorization' (set to 'Bearer ada6a44d4840efe9ba5ba8e1fdfcae5a').

The second screen is the **Configuration** module of “**SAP Cloud Platform IoT Service**”. This tab is the most important because it allows you to enter your SAP Cloud Platform IoT Service credentials into the provided form to seamlessly stream your sensor data in real-time. The form is comprised of *URL*, *mode*, *messageType*, and *messages*. You will also notice header values with two fields: *Content-Type* and *Authorization*. This is needed to send data to the IoT service from an external source like our SensorTag device, outside of the authorized network. Once entered your details, you may test it by pressing the wizard button.

In the first screen, there is a sensor configuration where it allows you to constantly send your updated sensor data in intervals of *Every 2 Seconds*, *Every 5 Seconds*, and/or *Every 10 Seconds*. Let's dig deeper into the form below.

What to enter in the “IoT Service” Form



Your form's inputs should look something similar to mine below. At the moment you have to enter this manually, so I suggest you put your information into an email and send it to your Phone. This will allow you to easily copy/paste the information to the appropriate fields.

URL: <https://iotmmsi840337trial.hanatrial.ondemand.com/com.sap.iotservices.mms/v1/api/http/data/74b90c1d-3492-499c-9757-505ce4756bff>

mode: sync

messageType: 4c0781b3ff1e7506f77b

messages: *automatically generated*

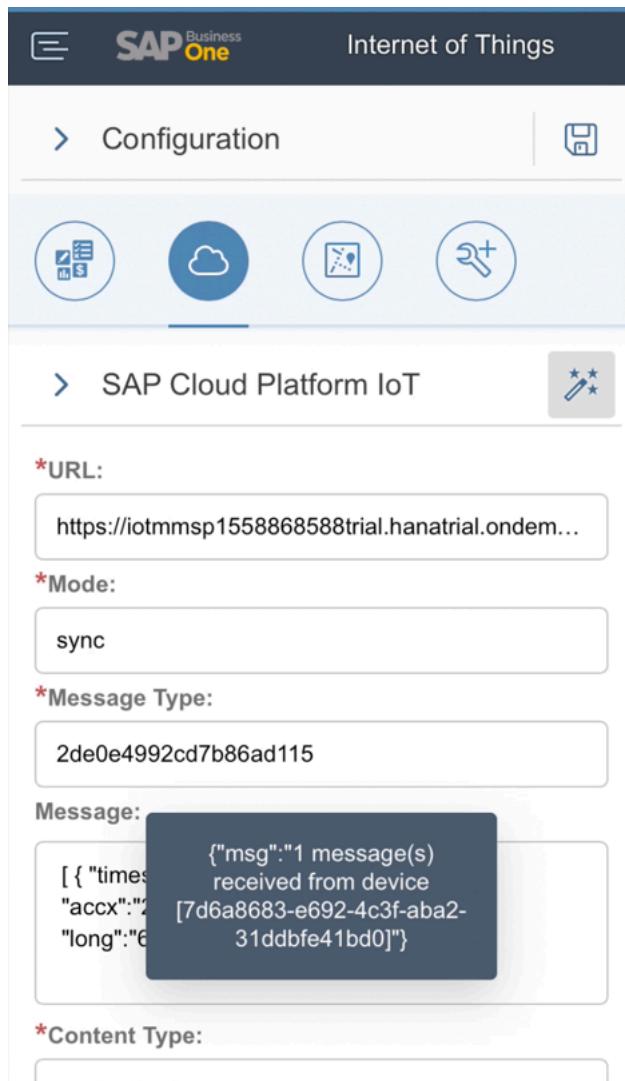
Header

Content-Type: application/JSON; charset=utf-8

Authorization: Bearer *insert authorization token here*

Note: It is extremely important to add the word “Bearer” and a space before your device’s authorization token.

After you have successfully setup your B1 IoT app, tap  to test and you will see a response from SAP Cloud Platform that looks something like this:



If you'd like to see the raw data in a table, you can easily do so from the IoT Service Cockpit. From your SAP Cloud Platform Cockpit, let's go back to Services, and select the Internet of Things Service. We will enter the IoT Service Cockpit by clicking "Go to Service". You will now see the IoT Services Cockpit.

The screenshot shows the SAP Cloud Platform Cockpit with the "Internet of Things Service Cockpit" selected. The interface is divided into two main sections: "Device Management" and "Message Management".

Device Management:

- All Message Types: 5 Message Types
- All Device Types: 4 Device Types
- All Registered Devices: 3 Devices

Message Management:

- Redeploy the Message Management: Up to Date
- Send and view messages, or perform other actions: MMS Cockpit

Select the "**MMS Cockpit - Select and view messages, or perform other actions**" tile (in the bottom right-hand side).

The screenshot shows the "Message Management Service Cockpit". It is organized into two main sections: "Core Services" and "Data Services".

Core Services:

- Display Stored Messages (Application Data)
- Push Messages to Devices (Push Service)
- Configure the Message Management Service (Configuration)
- Configure Processing Service Mappings (Configuration)
- View Registered Devices And Device Types (IoT Service Cockpit)

Data Services:

- Messaging through HTTP (HTTP API)
- Messaging through MQTT over TCP (MQTT TCP API)
- Messaging through WebSockets (WebSocket API)
- Messaging through MQTT over WS (MQTT WebSocket API)

Select the "**Display Stored Messages**" tile (in the top left-hand corner).

Application Data

Table Chart

REFRESH Last updated on 6/20/2018, 2:50:55 AM

7 Tables

T_IOT_2DE0E4992CD7B86AD115
NEO_5XBN2L7P450NSUN1JN9XX7ALV 1 >

T_IOT_3F676714E256C6F959C2
NEO_5XBN2L7P450NSUN1JN9XX7ALV 1708 >

T_IOT_A1D6DBB3B5BF0C68025A
NEO_5XBN2L7P450NSUN1JN9XX7ALV 8 >

T_IOT_ACKSTORE
NEO_5XBN2L7P450NSUN1JN9XX7ALV 2 >

T_IOT_CONFIG
NEO_5XBN2L7P450NSUN1JN9XX7ALV 67 >

T_IOT_HTTP_PUSH
NEO_5XBN2L7P450NSUN1JN9XX7ALV >

T_IOT_MONITOR_LOG
NEO_5XBN2L7P450NSUN1JN9XX7ALV 519411 >

OData API

Locate the newly created table by looking for your Message Type ID (Example: The table would look something like **T_IOT_2DE0E4992CD7B86AD115**. The long part after the second underscore would be your message type). Then select your table and you will see the messages being sent! Refresh the page to see the new data being fed in.

Application Data

Table Chart

REFRESH Last updated on 6/20/2018, 2:51:15 AM

Table NEO_5XBN2L7P450NSUN1JN9XX7ALV.T_IOT_2DE0E4992CD7B86AD115
(1 row(s) out of 1 loaded. Newest on top.)

OData API

G_DEVICE	G_CREATED	C_TIMESTAMP	C_DEV	C_ACCX	C_ACCY	C_ACCZ	C_ALT	C_LONG	C_LAT	C_LUX
7d6a8683-e692-4c3f-aba2-31ddbf41bd0	Wed Jun 20 2018 02:45:55	Wed May 13 2015 01:05:13	1	2	3	4	5	6	7	8
			GMT+0800 (+08)							



Application Data



Table



Chart

▼ Configuration

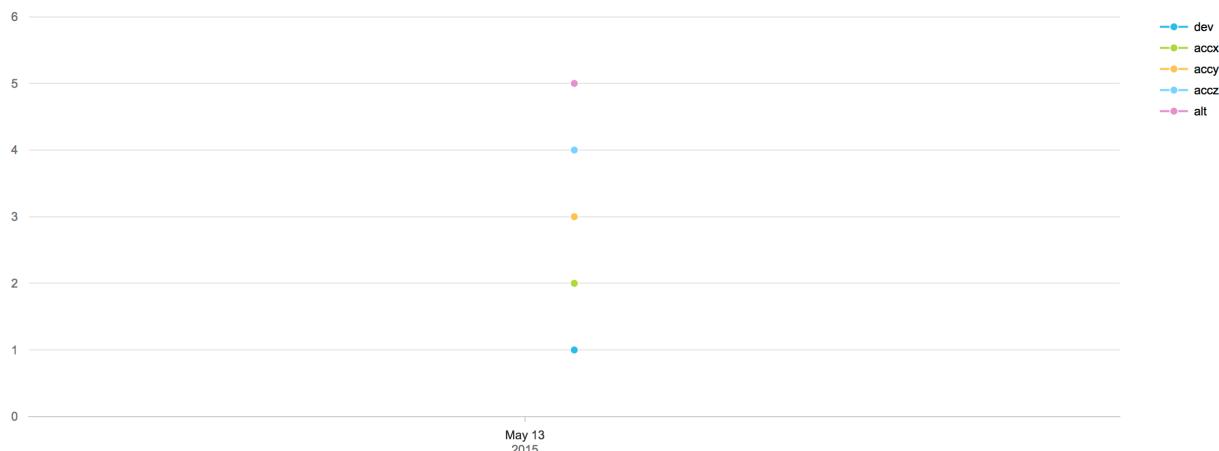
*Devices: B1 IoT SensorTag Device



*Message Type: B1IoT SensorTag Message Type



*Time Axis: timestamp

*Measures: dev accx accy accz alt Show Last Values: 10 50 100 250 500Chart Auto-Refresh: OFF

www.sap.com/contactsap

© 2017 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. See <http://www.sap.com/corporate-en/legal/copyright/index.epx> for additional trademark information and notices.

