Setup & Connect SensorTag on SMB IoT App with SAP Cloud Platform IoT 4.0 Service Cockpit

Part I – Setup Device's Digital Twin on SAP Cloud Platform IoT 4.0 Service Cockpit.

Part II – Setup a Cloud Foundry (CF) NodeJS Application, consuming SAP Cloud Platform IoT 4.0 certificate (taken from Part I); CF App serving as a Proxy between SMB IoT Demo App to SAP Cloud Platform IoT 4.0.

What will you need:

1. SAP Cloud Platform IoT 4.0 Service (Productive / Demo Tenant)
   (contact your instructor if you do not have one)

2. Texas Instrument SensorTag CC2650STK

As of 2 January, 2019

SAP TECHED

TABLE OF CONTENTS

1 PART I – SETUP DEVICE’S DIGITAL TWIN ON SAP CLOUD PLATFORM IOT 4.0 SERVICE COCKPIT

Setup Device’s Digital Twin on SAP Cloud Platform IoT 4.0 Service Cockpit.

This hand-on session covers the data ingestion and consumption the *SAP Cloud Platform Internet of Things for the Cloud Foundry Environment* (Internet of Things Service).

1.1 SAP Cloud Platform Internet of Things

The SAP Cloud Platform Internet of Things Service enables customers and partners to develop, customize, and operate Internet of Things business applications in the cloud.

The Internet of Things Service connects devices to the 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.

The Internet of Things Service collects and process sensor data at scale already at the edge or in the cloud and store it on the SAP Cloud Platform for use by other applications. Moreover, the Internet of Things Service provides a multi-tenant architecture allowing role-based access to device data through easy-to-use APIs.

1.2 Links

- **Documentation**: https://help.sap.com/iot
- **Internet of Things Service Cockpit**: [IOT_INSTANCE_URL]/iot/cockpit
- **Internet of Things API Service**: [IOT_INSTANCE_URL]/iot/core/api/v1/doc
- **Internet of Things Starter Kit**: https://github.com/SAP/iot-starterkit/tree/master/cf

2 DEVICE MODEL

The device model for the IoT service consists of the entities depicted in Figure 1. Further details can be found in the documentation.

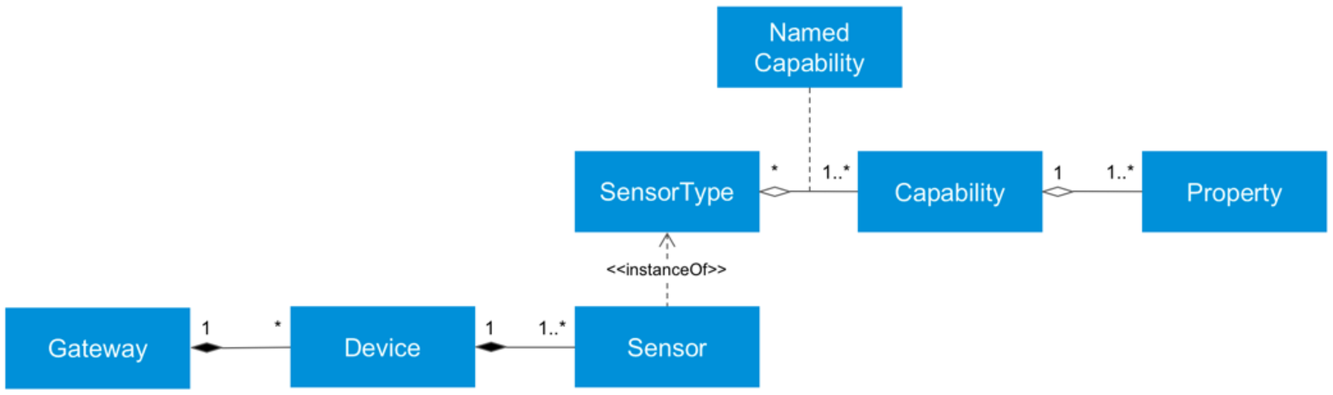
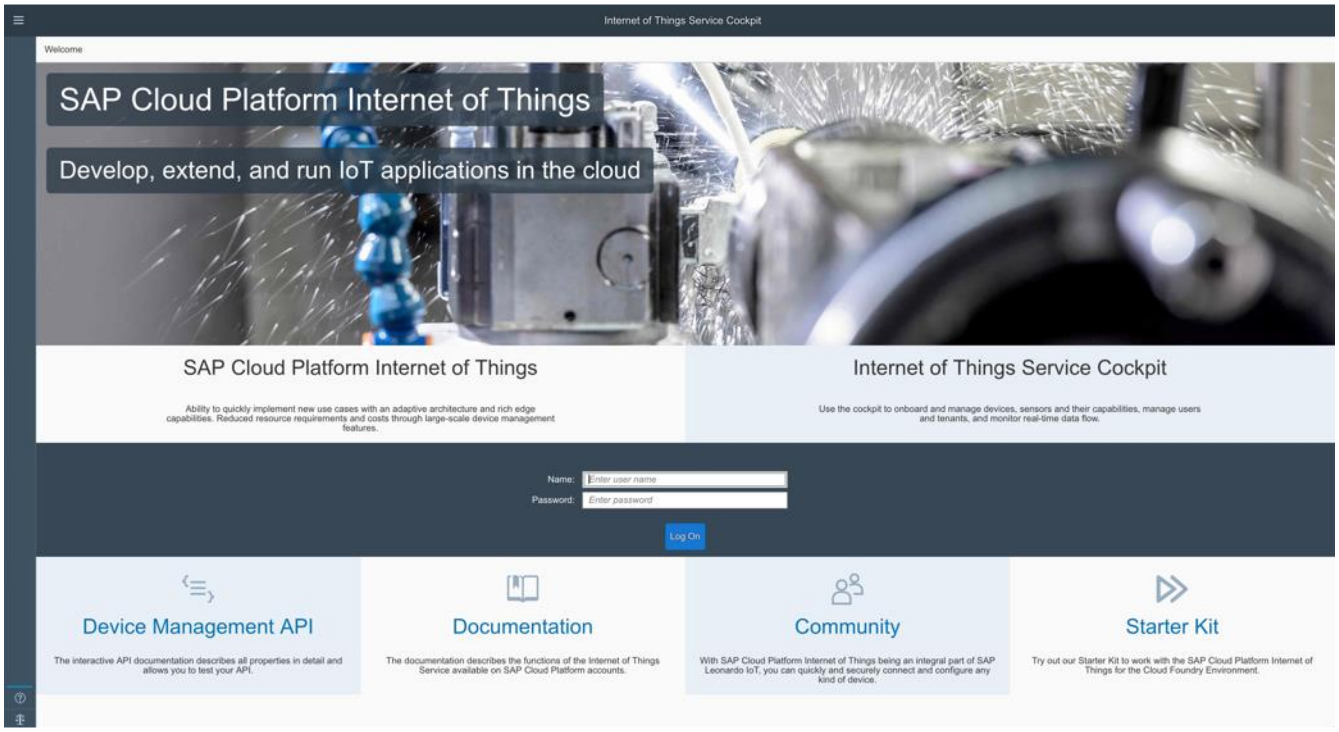


Figure 1: Device Model

You will be given a demo tenant of the SAP Cloud Platform IoT Service Cockpit.
Please use Google Chrome browser and access your demo account. You should see the following.



SAP Cloud Platform Internet of Things

Develop, extend, and run IoT applications in the cloud

SAP Cloud Platform Internet of Things

Ability to quickly implement new use cases with an adaptive architecture and rich edge capabilities. Reduced resource requirements and costs through large-scale device management features.

Internet of Things Service Cockpit

Use the cockpit to onboard and manage devices, sensors and their capabilities, manage users and tenants, and monitor real-time data flow.

Device Management API

The interactive API documentation describes all properties in detail and allows you to test your API.

Documentation

The documentation describes the functions of the Internet of Things Service available on SAP Cloud Platform accounts.

Community

With SAP Cloud Platform Internet of Things being an integral part of SAP Leonardo IoT, you can quickly and securely connect and configure any kind of device.

Starter Kit

Try out our Starter Kit to work with the SAP Cloud Platform Internet of Things for the Cloud Foundry Environment.

And the end of the exercise, you should be getting the following values. Keep the following template handy.
You will need these details for Part II.

Device ID	
Device Alternate ID	
Sensor ID	
Sensor Type Alternate ID	
Capability Alternate ID	
P12 Certificate Secret Key	
Certificate Name	

3 DEVICE ONBOARDING

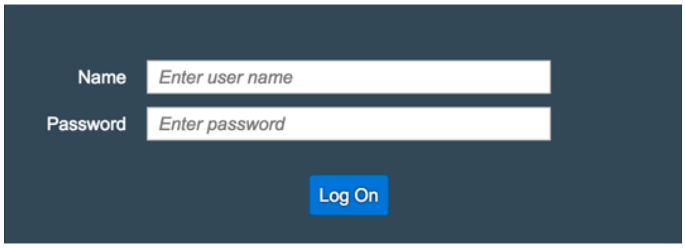
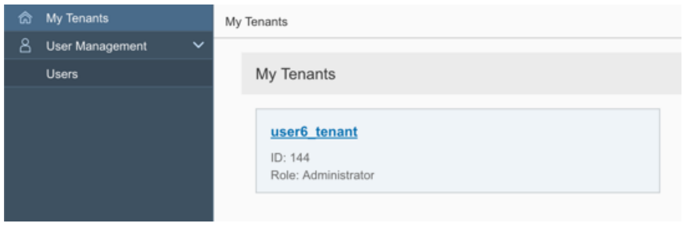
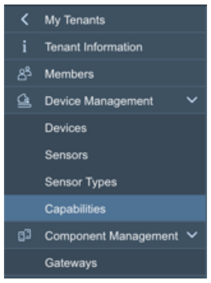
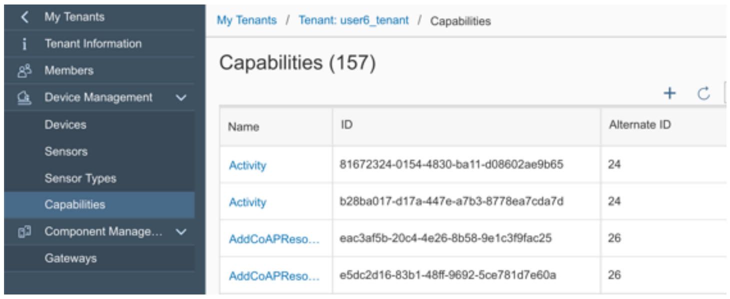
In order to send data to the IoT service a device is required. The according device entity must have at least one sensor assigned. In case, no sensor is created in beforehand a Sensor will be automatically created during data ingestion (default behavior).

3.1 Creating a Device Entity Using the IoT Service Cockpit

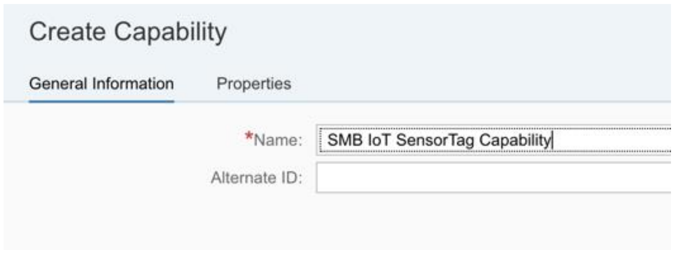
In the following a device is created using the IoT Service Cockpit. The device will have one sensor, which is of a custom sensor type. Therefore, a capability and a sensor type are created initially.

3.1.1 Create a Capability using the IoT Service Cockpit

In the following a capability is created. The capability will contain 8 properties required for SMB IoT SensorTag Demo. The capability entity will be assigned to a sensor type in the following step.

Explanation	Screenshot
1. Log on to the **Internet of Things Service Cockpit**. [IOT_INSTANCE_URL] Enter your **user name**, your **password**, and choose Log On. USER: [USER] Password: [PASSWORD] Your user is already assigned to a tenant and has the role **administrator**. **Click** into your Tenant.	 Name: Enter user name Password: Enter password Log On  My Tenants My Tenants user6_tenant ID: 144 Role: Administrator
2. Use the main menu to navigate to the **Capabilities** section.	 My Tenants Tenant Information Members Device Management Devices Sensors Sensor Types Capabilities Component Management Gateways
3. Choose **+** (Create a capability) to start the capability creation process.	 My Tenants Tenant Information Members Device Management Devices Sensors Sensor Types Capabilities Component Manage… Gateways My Tenants / Tenant: user6_tenant / Capabilities Capabilities (157) Name | ID | Alternate ID Activity | 81672324-0154-4830-ba11-d08602ae9b65 | 24 Activity | b28ba017-d17a-447e-a7b3-8778ea7cda7d | 24 AddCoAPReso… | eac3af5b-20c4-4e26-8b58-9e1c3f9fac25 | 26 AddCoAPReso… | e5dc2d16-83b1-48ff-9692-5ce781d7e60a | 26

4. In the **General Information** section, enter “SMB IoT SensorTag Capability” as **Name.**



Create Capability

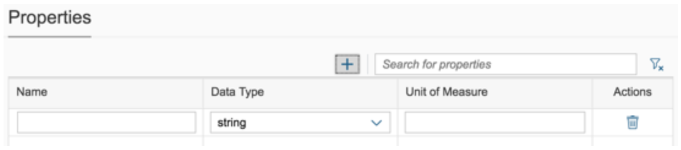
General Information | Properties

*Name: SMB IoT SensorTag Capability

Alternate ID:

5. In the **Properties** section, specify one property by choosing **+** (add 8 properties).

8 properties will be captured for SMB IoT demo.

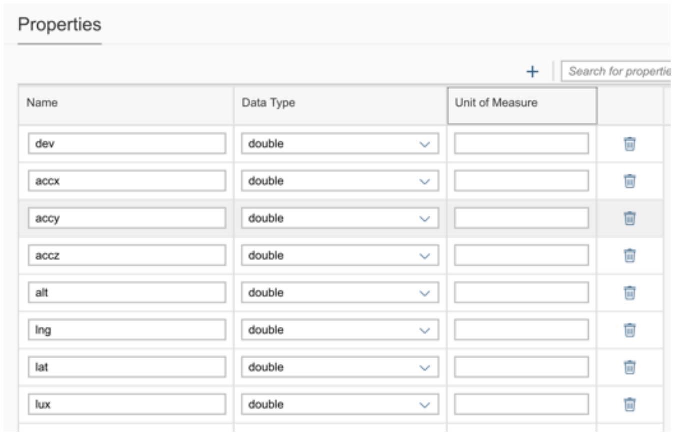


Properties

Name	Data Type	Unit of Measure	Actions
	string		

6. Make sure the following **Properties are as exact value & order**, if not your demo app will not work as the mobile app is taking these values mapping.

Property Name
dev
accx
accy
accz
alt
lng
lat
lux



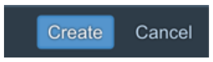
Properties

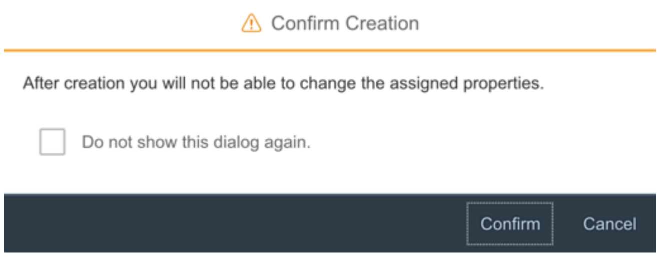
Name	Data Type	Unit of Measure
dev	double	
accx	double	
accy	double	
accz	double	
alt	double	
lng	double	
lat	double	
lux	double	

7. Choose **Create**.

Double check if you need to.
Make sure exact Value & Order.

Click **Confirm**.





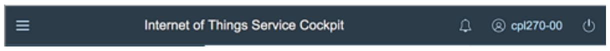
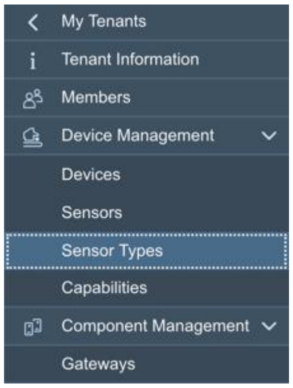
Confirm Creation

After creation you will not be able to change the assigned properties.

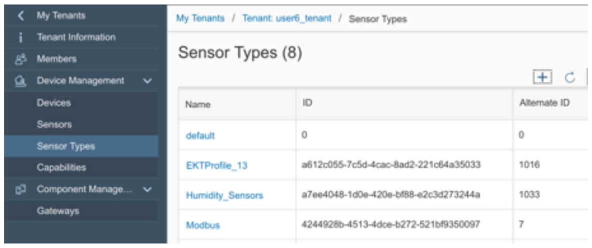
☐ Do not show this dialog again.

3.1.2 *Create a Sensor Type using the IoT Service Cockpit*

In the following a sensor type is created. The previously capability will be assigned as measure to the sensor type.

Explanation	Screenshot
1. Open the **Internet of Things Service Cockpit**.	
2. Use the main menu to navigate to the **Sensor Type** section.	 My Tenants Tenant Information Members Device Management Devices Sensors Sensor Types Capabilities Component Management Gateways

3. Choose **+** (**Create a sensor type**) to start the sensor type creation process.

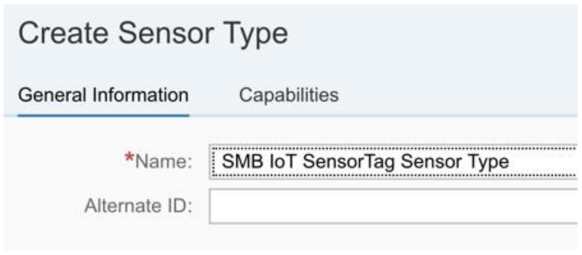


My Tenants / Tenant: user6_tenant / Sensor Types

Sensor Types (8)

Name	ID	Alternate ID
default	0	0
EKTProfile_13	a612c055-7c5d-4cac-8ad2-221c64a35033	1016
Humidity_Sensors	a7ee4048-1d0e-420e-bf88-e2c3d273244a	1033
Modbus	4244928b-4513-4dce-b272-521bf9350097	7

4. In the **General Information** section, enter "SMB IoT SensorTag Sensor Type" as **Name.**



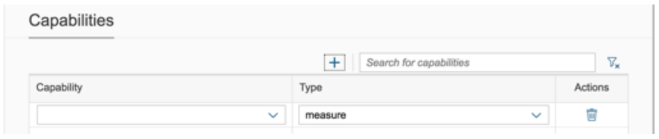
Create Sensor Type

General Information   Capabilities

*Name: SMB IoT SensorTag Sensor Type

Alternate ID:

5. In the **Capabilities** section, add a capability by choosing **+** (Add a capability).

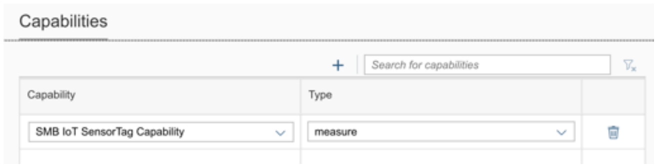


Capabilities

Capability	Type	Actions
	measure	

6. Select the previously created capability "SMB IoT SensorTag Capability" as **Capability** from the drop-down box.

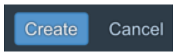
   Select "*measure*" for **Type**.

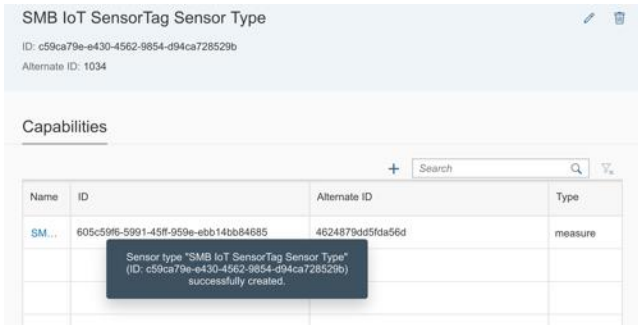


Capabilities

Capability	Type
SMB IoT SensorTag Capability	measure

7. Choose **Create**.





SMB IoT SensorTag Sensor Type

ID: c59ca79e-e430-4562-9854-d94ca728529b

Alternate ID: 1034

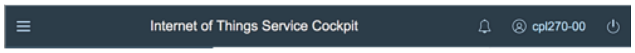
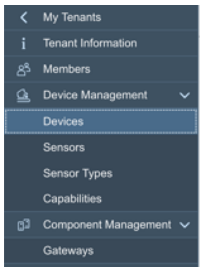
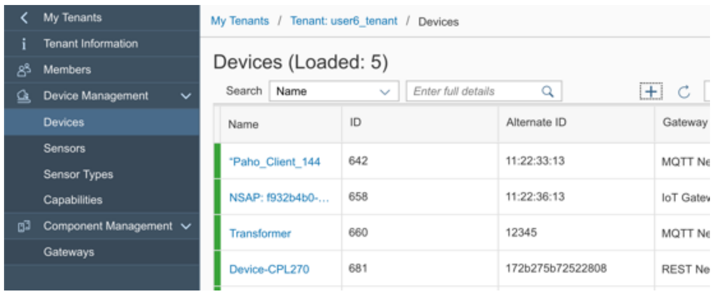
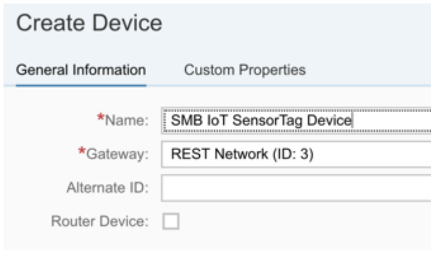
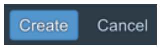
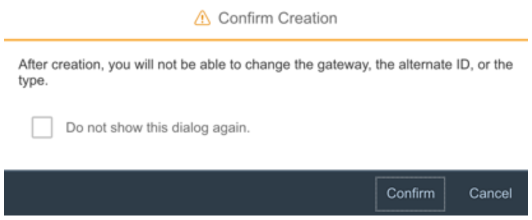
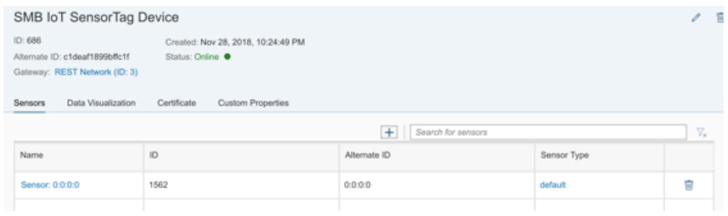
Capabilities

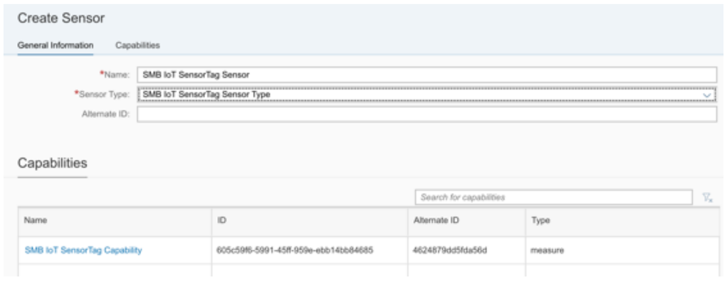
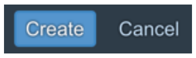
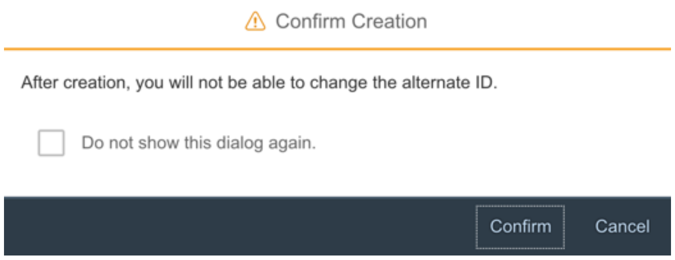
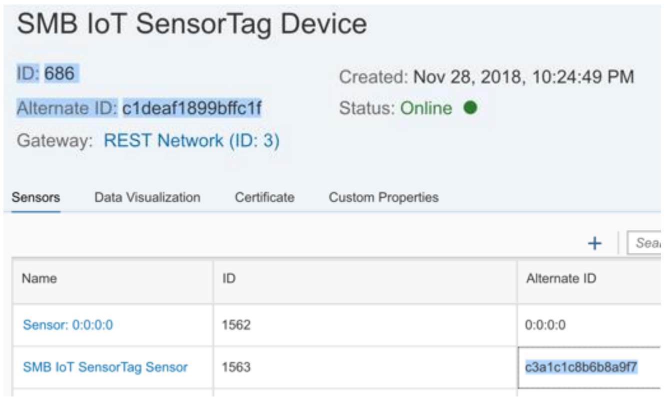
Name	ID	Alternate ID	Type
SM...	605c59f6-5991-45ff-959e-ebb14bb84685	4624879dd5fda56d	measure

Sensor type "SMB IoT SensorTag Sensor Type" (ID: c59ca79e-e430-4562-9854-d94ca728529b) successfully created.

3.1.3 *Create a Device with a Sensor using the IoT Service Cockpit*

In the following a device is created. The device entity does not have any sensors attached, yet. The device will be assigned to one gateway (REST).

Explanation	Screenshot
1. Open the **Internet of Things Service Cockpit** with your user credentials.	
2. Use the main menu to navigate to the **Devices** section.	 My Tenants Tenant Information Members Device Management Devices Sensors Sensor Types Capabilities Component Management Gateways
3. Choose **+** (Create a device) to start the sensor type creation process.	 My Tenants Tenant Information Members Device Management Devices Sensors Sensor Types Capabilities Component Management Gateways My Tenants / Tenant: user6_tenant / Devices Devices (Loaded: 5) Search Name Enter full details Name | ID | Alternate ID | Gateway *Paho_Client_144 | 642 | 11:22:33:13 | MQTT Ne NSAP: f932b4b0-... | 658 | 11:22:36:13 | IoT Gatev Transformer | 660 | 12345 | MQTT Ne Device-CPL270 | 681 | 172b275b72522808 | REST Ne
4. In the **General Information** section, enter "*SMB IoT SensorTag Device*" as **Name.** Select "*REST Network*" as **Gateway.**	 *Name: SMB IoT SensorTag Device *Gateway: REST Network (ID: 3) Alternate ID: Router Device: ☐
5. Choose **Create**. Double check if you need to. Click **Confirm**.	  After creation, you will not be able to change the gateway, the alternate ID, or the type. ☐ Do not show this dialog again.
6. In the ***SMB IoT SensorTag Device*** Page > **Sensor** section, choose **+** (add a sensor).	 ID: 686 Alternate ID: c1deaf1899bffc1f Gateway: REST Network (ID: 3) Created: Nov 28, 2018, 10:24:49 PM Status: Online ● Sensors Data Visualization Certificate Custom Properties Search for sensors Name | ID | Alternate ID | Sensor Type Sensor: 0:0:0:0 | 1562 | 0:0:0:0 | default

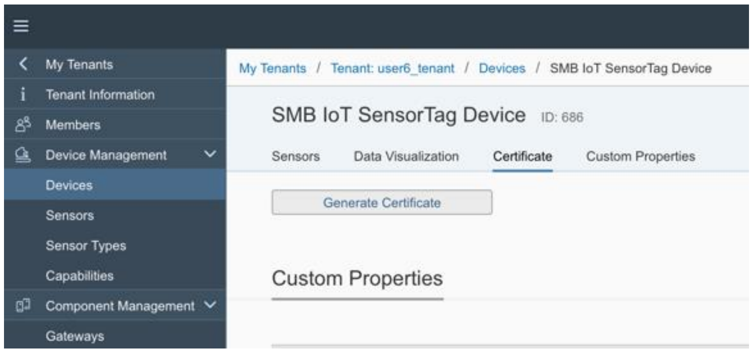
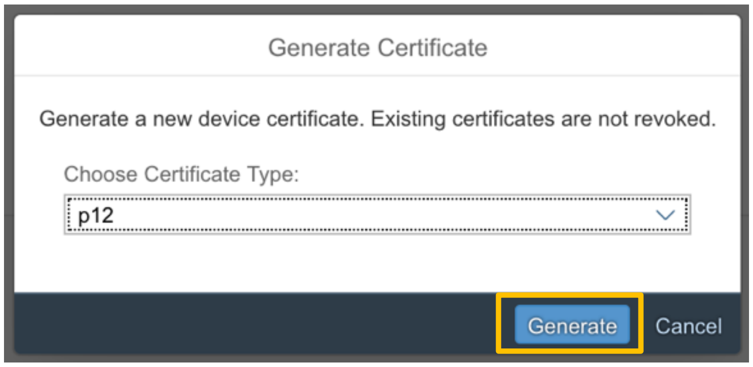
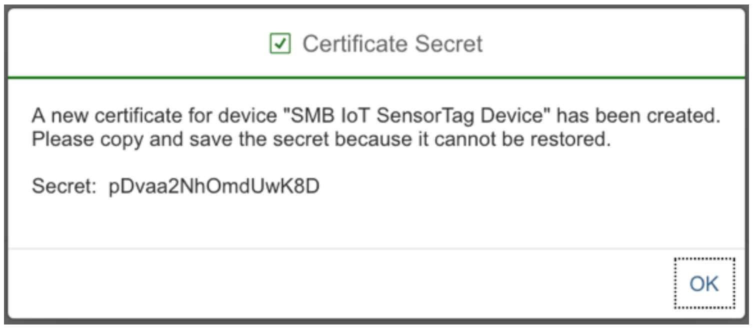
Explanation	Screenshot
7. In the **General Information** section, enter "SMB IoT SensorTag Sensor" as **Name.** Select the previously created sensor type **"**SMB IoT SensorTag Sensor Type**"** as **Sensor Type**.	Create Sensor General Information · Capabilities *Name: SMB IoT SensorTag Sensor *Sensor Type: SMB IoT SensorTag Sensor Type Alternate ID: Capabilities Search for capabilities Name: SMB IoT SensorTag Capability · ID: 605c59f6-5991-45ff-959e-ebb14bb84685 · Alternate ID: 4624879dd5fda56d · Type: measure
8. Choose **Create**. Double check if you need to. Click **Confirm**.	 Confirm Creation After creation, you will not be able to change the alternate ID. Do not show this dialog again.
9. Note down the **ID** & **Alternate ID** of the created **Device**. Note down the **Alternate ID** of the created **Sensor**. **Device ID**: [DEVICE_ID] **Device Alternate ID**: [DEVICE_ALT_ID] **Sensor Alternate ID**: [SENSOR_ALT_ID] e.g. **Device ID**: 686 **Device Alternate ID**: c1deaf1899bffc1f **Sensor Alternate ID**: c3a1c1c8b6b8a9f7	SMB IoT SensorTag Device ID: 686 Alternate ID: c1deaf1899bffc1f Gateway: REST Network (ID: 3) Created: Nov 28, 2018, 10:24:49 PM Status: Online ● Sensors · Data Visualization · Certificate · Custom Properties Name · ID · Alternate ID Sensor: 0:0:0:0 · 1562 · 0:0:0:0 SMB IoT SensorTag Sensor · 1563 · c3a1c1c8b6b8a9f7

4 SEND MESSAGES VIA POSTMAN REST CLIENT (CHROME APP)

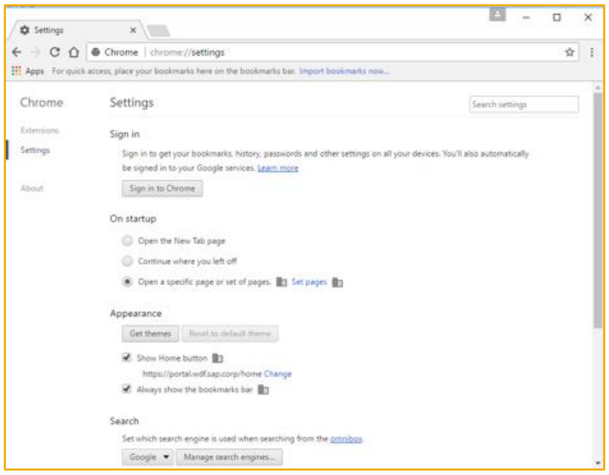
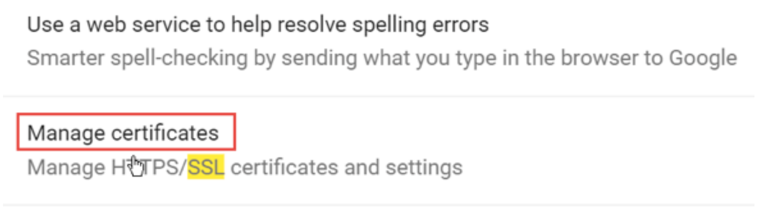
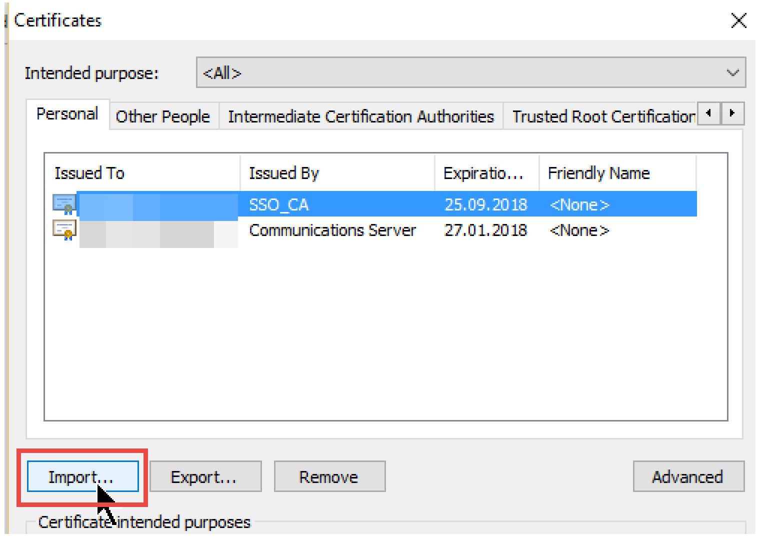
In this step, we will send the data from Chrome Postman Rest Client. We have already on-boarded this simulator device during previous steps. Once we send the data, it will be received by SAP Cloud Platform IoT Gateway Cloud and will be visible in the IoT Cockpit and via APIs.

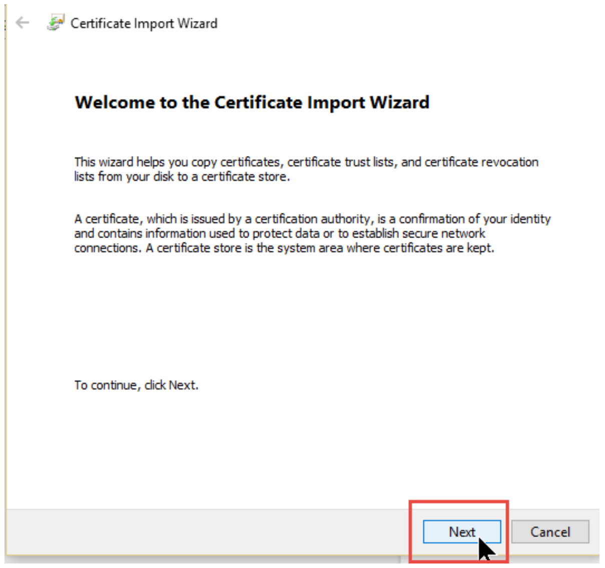
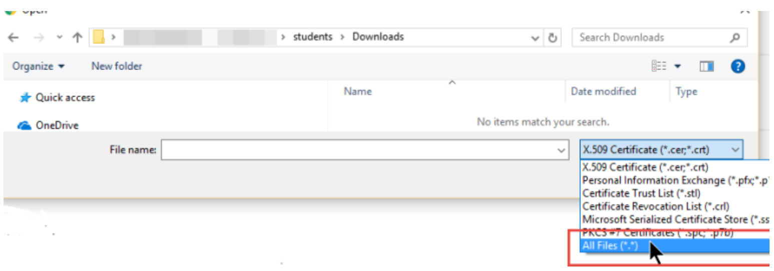
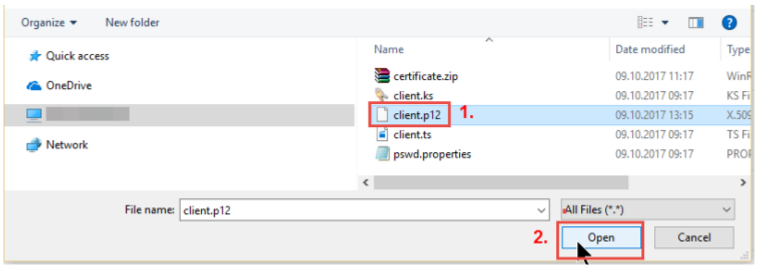
Make sure you're using the **Chrome App version of Postman Client** (not the Desktop App version).

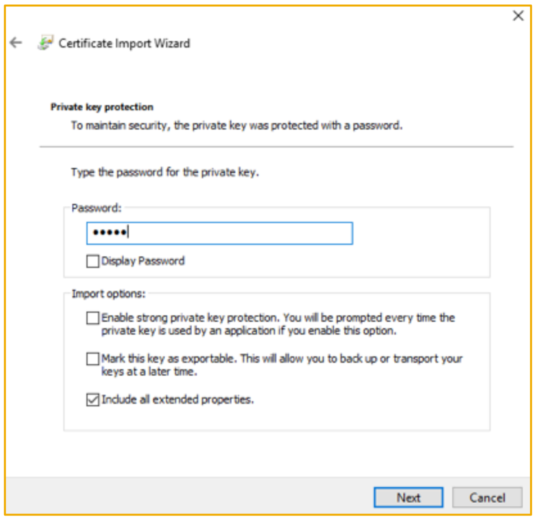
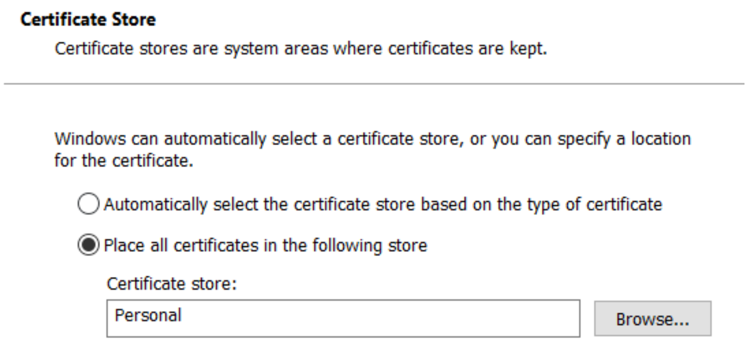
4.1 Download the certificates of Device

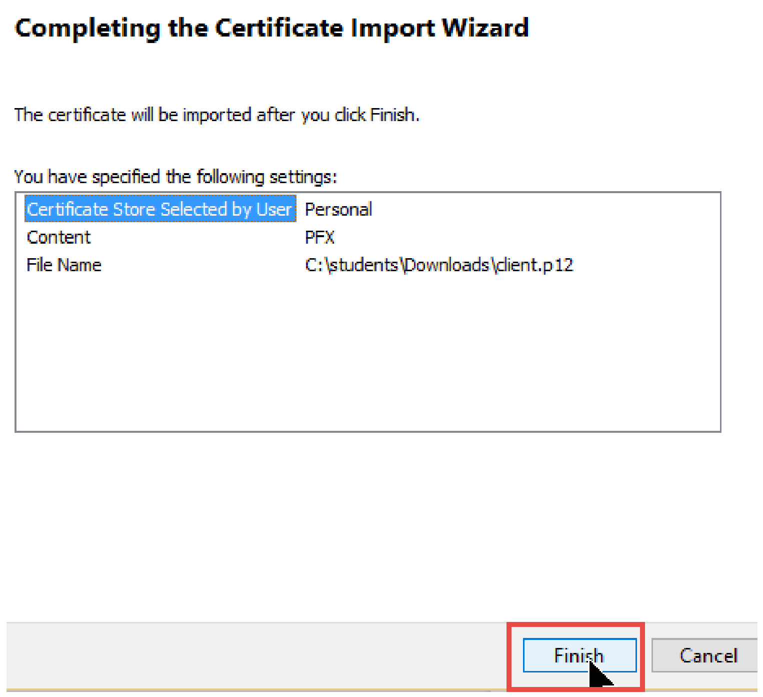
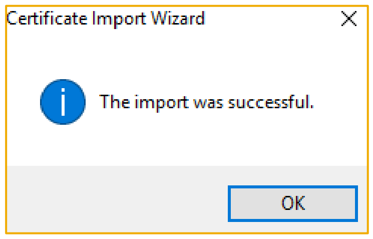
1. Navigate to IoT Cockpit **Main menu -> Device Management -> Devices** and select the device you have onboarded earlier. 2. Select **Certificate** tab and click on **Generate Device Certificate** Button.	 My Tenants Tenant Information Members Device Management Devices Sensors Sensor Types Capabilities Component Management Gateways My Tenants / Tenant: user6_tenant / Devices / SMB IoT SensorTag Device SMB IoT SensorTag Device ID: 686 Sensors Data Visualization Certificate Custom Properties Generate Certificate Custom Properties
3. Chose the Certificate Type as **P12** and Click on **Generate.**	 Generate a new device certificate. Existing certificates are not revoked. Choose Certificate Type: p12
4. This will trigger popup providing **Secret Key** which you must copy and **Save** in notepad. 5. Select **OK** to close the window.	 A new certificate for device "SMB IoT SensorTag Device" has been created. Please copy and save the secret because it cannot be restored. Secret: pDvaa2NhOmdUwK8D

4.2 Configure POSTMAN client with Device certificates for secure communication

Explanation	Screenshot
1. In Google chrome type **chrome://settings** in the address bar. It will open chrome settings.	 Settings Chrome | chrome://settings Apps For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now... Chrome Extensions Settings About Settings Search settings Sign in Sign in to get your bookmarks, history, passwords and other settings on all your devices. You'll also automatically be signed in to your Google services. Learn more Sign in to Chrome On startup Open the New Tab page Continue where you left off Open a specific page or set of pages. Set pages Appearance Get themes Reset to default theme Show Home button https://portal.wdf.sap.corp/home Change Always show the bookmarks bar Search Set which search engine is used when searching from the omnibox. Google Manage search engines...
2. Search for **SSL** in search text field. You get **HTTPS/SSL** as shown in the image Click on **Manage Certificates** button.	 Use a web service to help resolve spelling errors Smarter spell-checking by sending what you type in the browser to Google Manage certificates Manage HTTPS/SSL certificates and settings
3. Once the certificate manager launched click on **Import …** on **Personal** tab.	 Certificates Intended purpose: <All> Personal | Other People | Intermediate Certification Authorities | Trusted Root Certification Issued To | Issued By | Expiratio... | Friendly Name [redacted] | SSO_CA | 25.09.2018 | <None> [redacted] | Communications Server | 27.01.2018 | <None> Import... Export... Remove Advanced Certificate intended purposes

Explanation	Screenshot
4. **Certificate Import Wizard** will be opened. Click on Next.	 Certificate Import Wizard **Welcome to the Certificate Import Wizard** This wizard helps you copy certificates, certificate trust lists, and certificate revocation lists from your disk to a certificate store. A certificate, which is issued by a certification authority, is a confirmation of your identity and contains information used to protect data or to establish secure network connections. A certificate store is the system area where certificates are kept. To continue, click Next. Next Cancel
5. Browse to D:\students\Downloads and choose “All Files (*.*) when searching for the newly created p12 certificate file.	 [redacted] › students › Downloads Search Downloads Organize New folder Quick access OneDrive Name Date modified Type No items match your search. File name: X.509 Certificate (*.cer;*.crt) X.509 Certificate (*.cer;*.crt) Personal Information Exchange (*.pfx;*.p Certificate Trust List (*.stl) Certificate Revocation List (*.crl) Microsoft Serialized Certificate Store (*.ss PKCS #7 Certificates (*.spc;*.p7b) All Files (*.*)
6. Select **SMB IoT SensorTag Device-device_certificate.p12** file and choose Open.	 Organize New folder Quick access OneDrive [redacted] Network Name / Date modified / Type certificate.zip 09.10.2017 11:17 WinF client.ks 09.10.2017 09:17 KS Fi client.p12 **1.** 09.10.2017 13:15 X.509 client.ts 09.10.2017 09:17 TS Fi pswd.properties 09.10.2017 09:17 PROI File name: client.p12 All Files (*.*) **2.** Open Cancel

Explanation	Screenshot
7. Provide the **Secret** which you have obtained while downloading device certificate Hint: secret which is stored in notepad Click on **Next.**	 Certificate Import Wizard **Private key protection** To maintain security, the private key was protected with a password. Type the password for the private key. Password: ••••• ☐ Display Password Import options: ☐ Enable strong private key protection. You will be prompted every time the private key is used by an application if you enable this option. ☐ Mark this key as exportable. This will allow you to back up or transport your keys at a later time. ☑ Include all extended properties.
8. Click on **Next**	 **Certificate Store** Certificate stores are system areas where certificates are kept. Windows can automatically select a certificate store, or you can specify a location for the certificate. ○ Automatically select the certificate store based on the type of certificate ● Place all certificates in the following store Certificate store: Personal

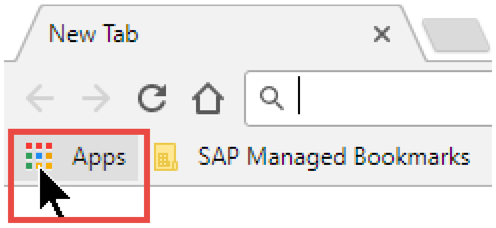
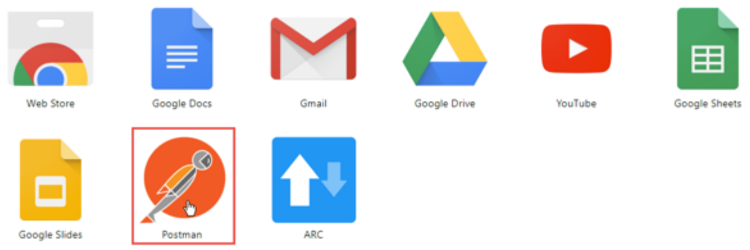
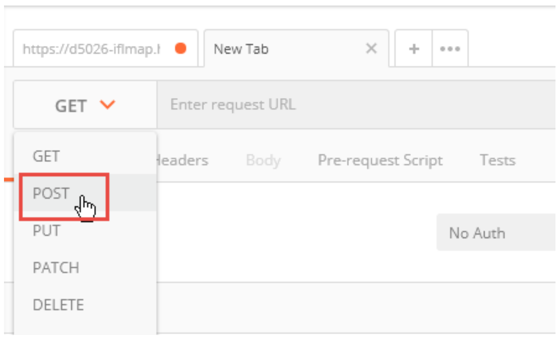
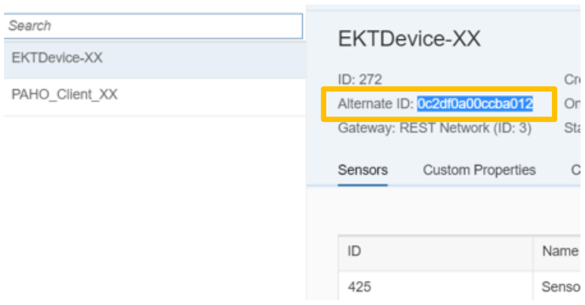
Explanation	Screenshot
9. Finally, click on **Finish** button	 Completing the Certificate Import Wizard The certificate will be imported after you click Finish. You have specified the following settings: Certificate Store Selected by User: Personal Content: PFX File Name: C:\students\Downloads\client.p12
10. You should receive the Import successful message.	 The import was successful.

For Mac User:

1. Open **Keychain Access**
2. Under Keychain, Select '**login'**
3. Click the ' **+** ' Button
4. **Import** the Certificate & Enter **Certificate Secret Key**

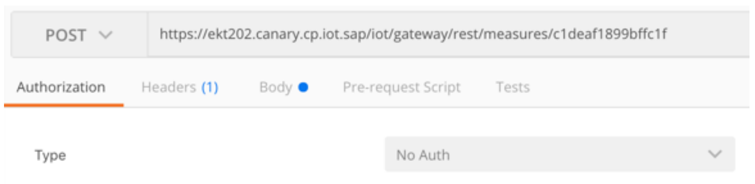
3.1 SENDING DATA FROM POSTMAN

3.1.1 *Sending Data from Postman using Device Certificates (for secure communication)*

Explanation	Screenshot
1. At the left corner at the top of your Google Chrome browser window choose **Apps**.	
2. Select Postman from the list of apps.	
3. Once Postman has been launched operation type to **Post**. Copy and paste the URL below into the request URL field. https://ektXXX.canary.cp.iot.sap/iot/gateway/rest/measures/[DEVICE_ALT_ID] **You can get device alternate ID from the cockpit or earlier you've copied.**	  EKTDevice-XX ID: 272 Alternate ID: 0c2df0a00ccba012 Gateway: REST Network (ID: 3) ID: 425

Explanation	Screenshot

4. Change to Authorization tab. Make sure that **No Auth** is selected.

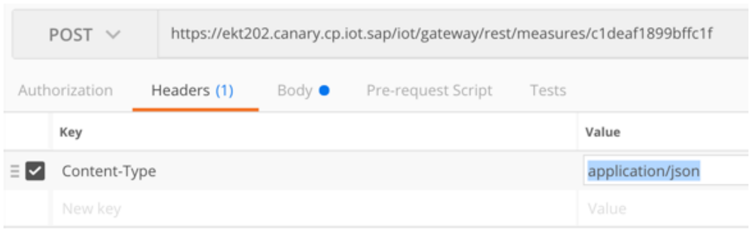


POST https://ekt202.canary.cp.iot.sap/iot/gateway/rest/measures/c1deaf1899bffc1f

Authorization | Headers (1) | Body | Pre-request Script | Tests

Type: No Auth

5. Change to Headers tab. Enter the following:
   - Key: **Content-Type**
   - Value: **application/json**



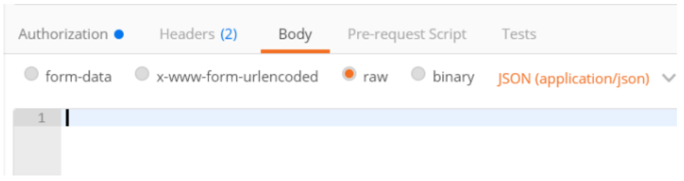
POST https://ekt202.canary.cp.iot.sap/iot/gateway/rest/measures/c1deaf1899bffc1f

Authorization | Headers (1) | Body | Pre-request Script | Tests

Key	Value
Content-Type	application/json
New key	Value

6. Change to Body tab and do the following:

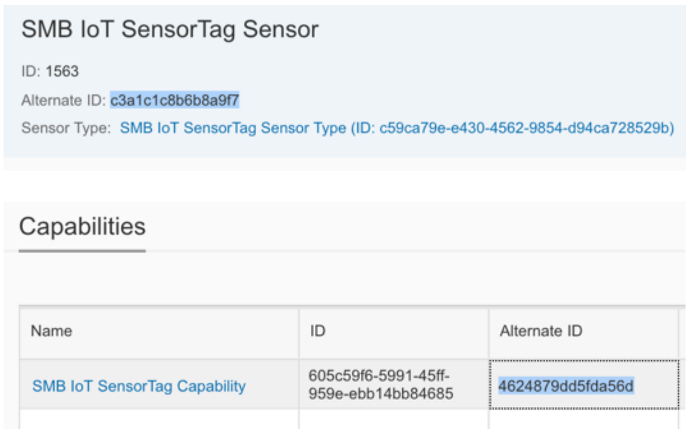
   - Select **raw**



Authorization | Headers (2) | Body | Pre-request Script | Tests

form-data | x-www-form-urlencoded | raw | binary | JSON (application/json)

7. **Sensor Alternate ID** & **Capability Alternate ID** for SMB IoT SensorTag can be obtained from IoT cockpit UI



SMB IoT SensorTag Sensor

ID: 1563

Alternate ID: c3a1c1c8b6b8a9f7

Sensor Type: SMB IoT SensorTag Sensor Type (ID: c59ca79e-e430-4562-9854-d94ca728529b)

Capabilities

Name	ID	Alternate ID
SMB IoT SensorTag Capability	605c59f6-5991-45ff-959e-ebb14bb84685	4624879dd5fda56d

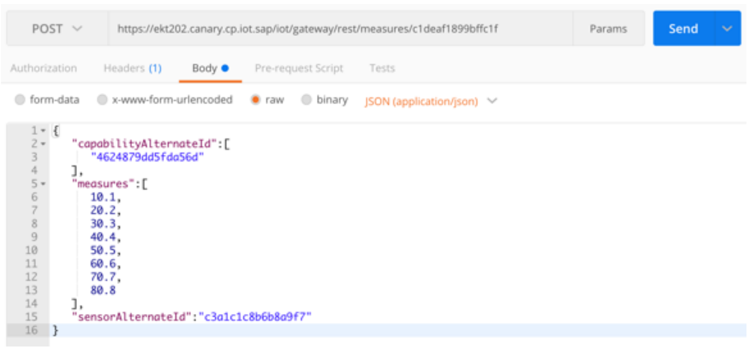
Payload should look like:

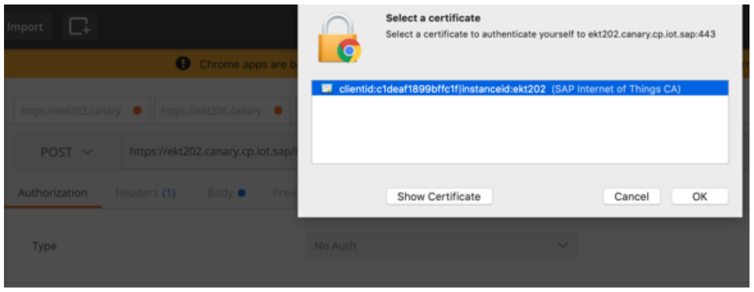
```
{
  "capabilityAlternateId":[
    "[CAPABILITY_ALT_ID]"
  ],
  "measures":[
    10.1,
    20.2,
    30.3,
    40.4,
    50.5,
    60.6,
    70.7,
    80.8
  ],
"sensorAlternateId":"[SENSOR_ALT_ID]"
}
```

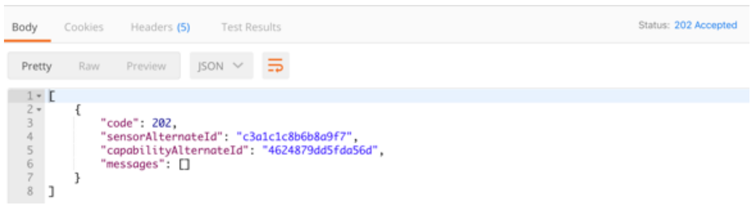
For example:

{"**capabilityAlternateId**":["4624879dd5fda56d"],"measures":[10.1,20.2,30.3,40.4,50.5,60.6,70.7,80.8],"**sensorAlternateId**":"c3a1c1c8b6b8a9f7"}

Explanation	Screenshot

8. Paste the payload in the Body section and finally click on **Send** button.	 POST https://ekt202.canary.cp.iot.sap/iot/gateway/rest/measures/c1deaf1899bffc1f Params Send
	Authorization Headers (1) Body Pre-request Script Tests
	form-data x-www-form-urlencoded raw binary JSON (application/json)
	`{`
	`"capabilityAlternateId":[`
	`"4624879dd5fda56d"`
	`],`
	`"measures":[`
	`10.1,`
	`20.2,`
	`30.3,`
	`40.4,`
	`50.5,`
	`60.6,`
	`70.7,`
	`80.8`
	`],`
	`"sensorAlternateId":"c3a1c1c8b6b8a9f7"`
	`}`

9. **You must get certificate pop up while sending the request.** Select the certificate and Click OK. **Hint**: If the certificate selection pop up <u>does not</u> appear and you get unauthorized: Close all **chrome web browser sessions** and exit from **Postman**. Open chrome and POSTMAN again. Once it restarts, it should reflect the certificate changes.	 **Select a certificate** Select a certificate to authenticate yourself to ekt202.canary.cp.iot.sap:443 clientid:c1deaf1899bffc1f|instanceid:ekt202 (SAP Internet of Things CA) Show Certificate Cancel OK

10. Make sure that that HTTP response code **202** is shown at the right lower corner which means the message has been sent successfully.	 Body Cookies Headers (5) Test Results Status: 202 Accepted
	Pretty Raw Preview JSON
	`[`
	`{`
	`"code": 202,`
	`"sensorAlternateId": "c3a1c1c8b6b8a9f7",`
	`"capabilityAlternateId": "4624879dd5fda56d",`
	`"messages": []`
	`}`
	`]`

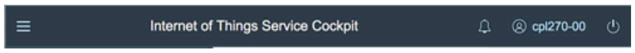
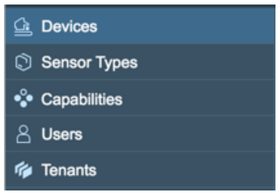
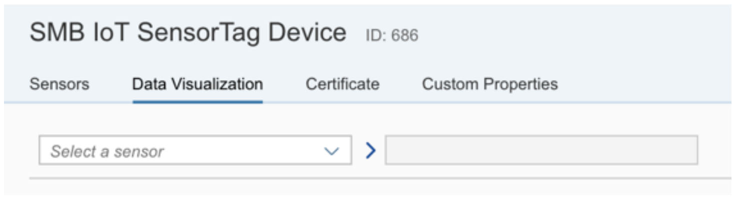
11. Send some further temperature data with Postman to SAP Cloud Platform Internet of things as follows: Modify the number of the **values** element and click on Send button.	

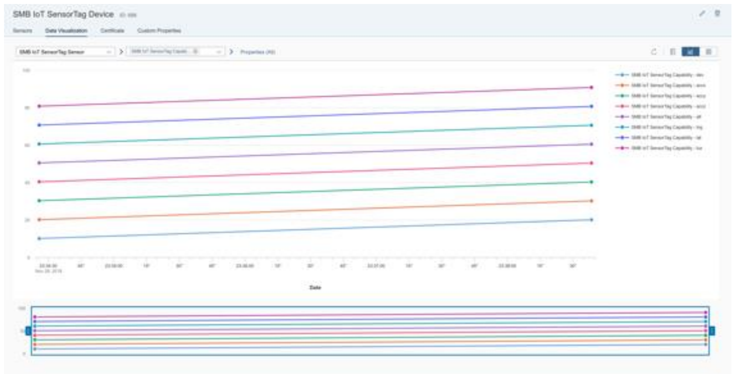
5 DATA CONSUMPTION

The incoming measures can be retrieved via API and the Internet of Things Service Cockpit in case persistency is enabled.

5.1 View Real-time Measures in IoT Service Cockpit

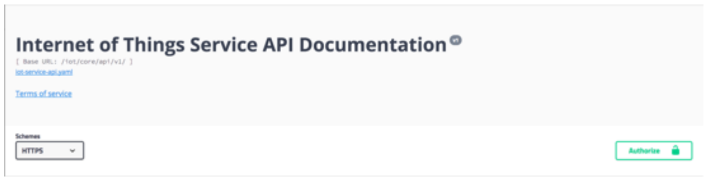
In the following the incoming data of a sensor is consumed in a chart.

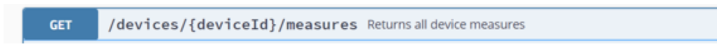
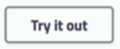
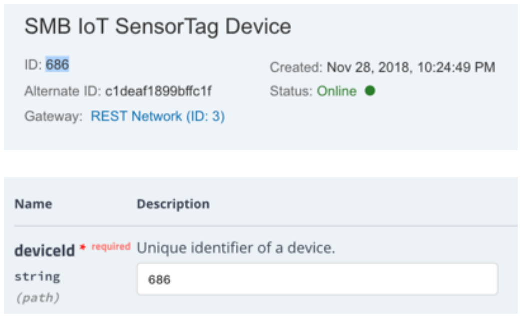
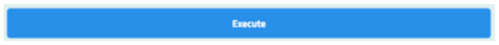
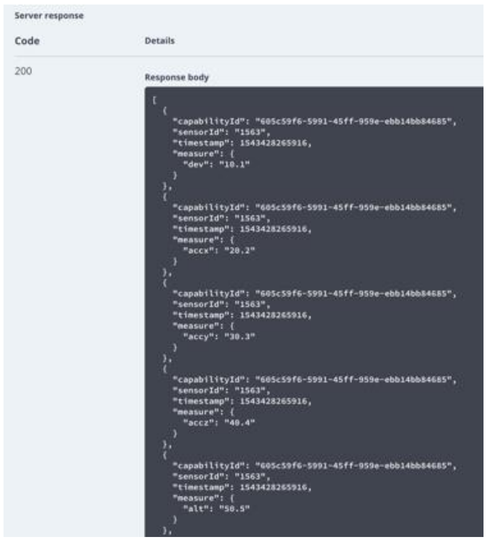
Explanation	Screenshot
1. Open the **Internet of Things Service Cockpit** with your user credentials.	
2. Use the main menu to navigate to the **Devices** section.	
3. Choose the previously created device (SMB IoT SensorTag Device).	
4. Choose the **Data Visualization** section**.**	 SMB IoT SensorTag Device ID: 686 Sensors Data Visualization Certificate Custom Properties Select a sensor

5. Select the previously created sensor "SMB IoT SensorTag Sensor" as **sensor** from the dropdown box. Select the previously created capability "SMB IoT SensorTag Capability" as **capability** from the dropdown box. Select the property "Humidity" as **property**.	 SMB IoT SensorTag Device Sensors Data Visualization Certificate Custom Properties SMB IoT SensorTag Sensor Properties (All) Date
6. The values will be displayed in a line chart. Optional: **Run** again the java sample and view the incoming data in the chart.	

5.2 Retrieve Measures using the API

In the following measures of a device are retrieved via API.

Explanation	Screenshot
1. Open the **Internet of Things API Documentation** and choose **Authorize.**	 Internet of Things Service API Documentation [Base URL: /iot/core/api/v1/] iot-service-api.yaml Terms of service

2. Choose the GET request entry **/devices/{deviceid}/measures** (Reads device measures).	 GET `/devices/{deviceId}/measures` Returns all device measures
3. Choose **Try it out**.	 Try it out
4. Enter the **Device id** of the previously created device for the field **id** (**not** the Device Alternate ID)	 SMB IoT SensorTag Device ID: 686 Created: Nov 28, 2018, 10:24:49 PM Alternate ID: c1deaf1899bffc1f Status: Online ● Gateway: REST Network (ID: 3) Name: deviceId * required string (path) Description: Unique identifier of a device. 686
5. Choose **Execute**.	 Execute
6. Scroll down to the response body to thee the measures.	 Server response Code: 200 Details: Response body

```
[
  {
    "capabilityId": "605c59f6-5991-45ff-959e-ebb14bb84685",
    "sensorId": "1563",
    "timestamp": 1543428265916,
    "measure": {
      "dev": "10.1"
    }
  },
  {
    "capabilityId": "605c59f6-5991-45ff-959e-ebb14bb84685",
    "sensorId": "1563",
    "timestamp": 1543428265916,
    "measure": {
      "accx": "20.2"
    }
  },
  {
    "capabilityId": "605c59f6-5991-45ff-959e-ebb14bb84685",
    "sensorId": "1563",
    "timestamp": 1543428265916,
    "measure": {
      "accy": "30.3"
    }
  },
  {
    "capabilityId": "605c59f6-5991-45ff-959e-ebb14bb84685",
    "sensorId": "1563",
    "timestamp": 1543428265916,
    "measure": {
      "accz": "40.4"
    }
  },
  {
    "capabilityId": "605c59f6-5991-45ff-959e-ebb14bb84685",
    "sensorId": "1563",
    "timestamp": 1543428265916,
    "measure": {
      "alt": "50.5"
    }
  },
```

6 *Part II – Setup a Cloud Foundry (CF) NodeJS Application, consuming SAP Cloud Platform IoT 4.0 certificate (taken from Part I); CF App serving as a Proxy between SMB IoT Demo App to SAP Cloud Platform IoT 4.0.*

Setup a Cloud Foundry (CF) NodeJS Application, consuming SAP Cloud Platform IoT 4.0 certificate (taken from Part I); CF App serving as a Proxy between SMB IoT Demo App to SAP Cloud Platform IoT 4.0.

Note: In an IoT scenario, usually a server-side component (e.g. Raspberry PI) will be the gateway between the sensors (SensorTag) and the platform (SCP IoT 4.0). With that, server-side program could directly consume the certificate and stream data. Given our scenario having our SMB IoT App (client-side) as a hybrid application (JavaScript base), we make use of a Cloud Foundry (CF) application as a proxy to process the certificate and stream the data from the SMB IoT App to the CF App. Thus, the logic could be further enhanced & developed in the CF App. For now, it is simply just a Sensor Data "middleman".

From Part I, you should have the following details:

URL: https://ektXXX.canary.cp.iot.sap/iot/gateway/rest/measures/[SENSOR_ID]

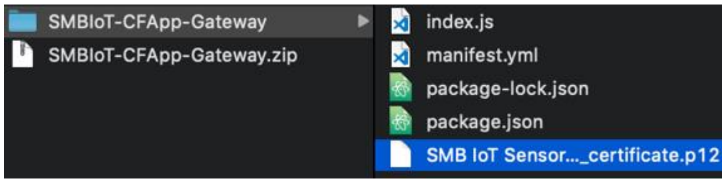
Sensor Alternate ID: xxxx

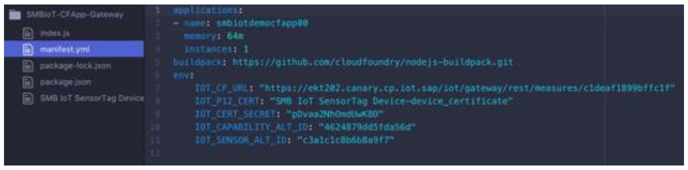
Capability Alternate ID: xxxx

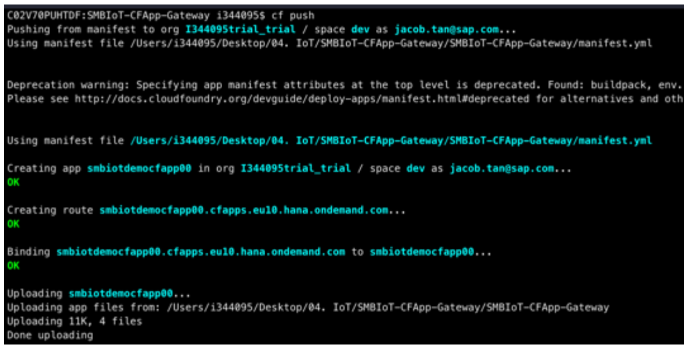
Name of Certificate (p12): xxxx.p12

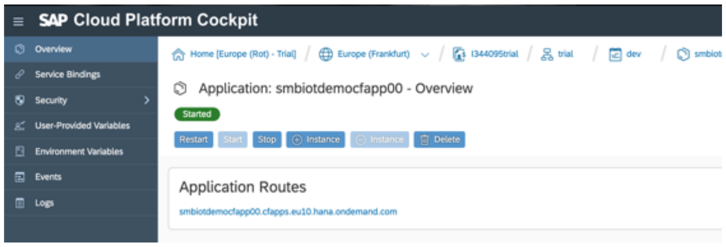
Secret Key to Certificate: xxxx

6.1 Deploy Cloud Foundry Application with Digital Device Certificate (from Part I)

Explanation	Screenshot
1. **Download** SMBIoT-CFApp-Gateway.zip. **Extract** & **Upload** the Certificate into the folder.	 SMBIoT-CFApp-Gateway SMBIoT-CFApp-Gateway.zip index.js manifest.yml package-lock.json package.json SMB IoT Sensor..._certificate.p12

2. **Open** manifest.yml.
 	
Modify Environment Variables to your details.
 	
For Example:
 	
IOT_CF_URL: "https://ekt202.canary.cp.iot.sap/iot/gateway/rest/measures/c1deaf1899bffc1f"
IOT_P12_CERT: "SMB IoT SensorTag Device-device_certificate.p12"
IOT_CERT_SECRET: "pDvaa2NhOmdUwK8D"
IOT_CAPABILITY_ALT_ID: "4624879dd5fda56d"
IOT_SENSOR_ALT_ID: "c3a1c1c8b6b8a9f7"
	 SMBIoT-CFApp-Gateway
	index.js
	manifest.yml
	package-lock.json
	package.json
	SMB IoT SensorTag Device
	`applications:`
	`- name: smbiotdemocfapp00`
	`  memory: 64m`
	`  instances: 1`
	`buildpack: https://github.com/cloudfoundry/nodejs-buildpack.git`
	`env:`
	`    IOT_CF_URL: "https://ekt202.canary.cp.iot.sap/iot/gateway/rest/measures/c1deaf1899bffc1f"`
	`    IOT_P12_CERT: "SMB IoT SensorTag Device-device_certificate"`
	`    IOT_CERT_SECRET: "pDvaa2NhOmdUwK8D"`
	`    IOT_CAPABILITY_ALT_ID: "4624879dd5fda56d"`
	`    IOT_SENSOR_ALT_ID: "c3a1c1c8b6b8a9f7"`
	
	You may give a **unique name** for your application.
	Maybe replace the number at the back to your initials.
	e.g. smbiotdemocfappjt

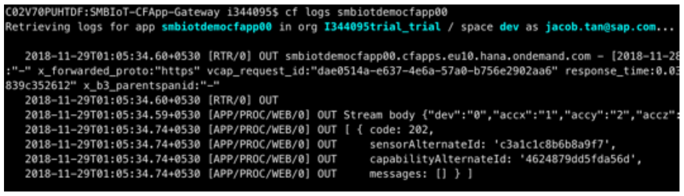
3. **Deploy your CF App into SAP Cloud Platform Cloud Foundry.**
 	
For more info on how to install Cloud Foundry CLI & Login via your Command Prompt / Terminal, please refer to this tutorial.
 	
Once logged into CF successfully and **terminal / command prompt currently in your active folder**, run the following commands.
 	
$ cf push
or
$ cf push --random-route
 	
random-route will avoid name collisions with others that deploy this same app on SCP. You can also choose your own app name by changing the manifest.yml file.
	
	`C02V70PUHTDF:SMBIoT-CFApp-Gateway i344095$ cf push`
	`Pushing from manifest to org I344095trial_trial / space dev as jacob.tan@sap.com...`
	`Using manifest file /Users/i344095/Desktop/04. IoT/SMBIoT-CFApp-Gateway/SMBIoT-CFApp-Gateway/manifest.yml`
	
	`Deprecation warning: Specifying app manifest attributes at the top level is deprecated. Found: buildpack, env.`
	`Please see http://docs.cloudfoundry.org/devguide/deploy-apps/manifest.html#deprecated for alternatives and oth`
	
	`Using manifest file /Users/i344095/Desktop/04. IoT/SMBIoT-CFApp-Gateway/SMBIoT-CFApp-Gateway/manifest.yml`
	
	`Creating app smbiotdemocfapp00 in org I344095trial_trial / space dev as jacob.tan@sap.com...`
	`OK`
	
	`Creating route smbiotdemocfapp00.cfapps.eu10.hana.ondemand.com...`
	`OK`
	
	`Binding smbiotdemocfapp00.cfapps.eu10.hana.ondemand.com to smbiotdemocfapp00...`
	`OK`
	
	`Uploading smbiotdemocfapp00...`
	`Uploading app files from: /Users/i344095/Desktop/04. IoT/SMBIoT-CFApp-Gateway/SMBIoT-CFApp-Gateway`
	`Uploading 11K, 4 files`
	`Done uploading`

4. Once successfully deployed, you should be able to see your CF App in your SCP Cockpit. The Application Routes will be displayed there. Typically, it will be <app_name>.cfapps.eu10.hana.ondemand.com Now we will test it.	 SAP Cloud Platform Cockpit Overview Service Bindings Security User-Provided Variables Environment Variables Events Logs Home [Europe (Rot) - Trial] Europe (Frankfurt) I344095trial trial dev smbiot Started Restart Start Stop Instance Instance Delete Application: smbiotdemocfapp00 - Overview Application Routes smbiotdemocfapp00.cfapps.eu10.hana.ondemand.com

6.2 *Test your Cloud Found Application with POSTMAN Client*

1. [OPTIONAL] Test & Debug

   $ cf logs <app_name>



```
C02V70PUHTDF:SMBIoT-CFApp-Gateway i344095$ cf logs smbiotdemocfapp00
Retrieving logs for app smbiotdemocfapp00 in org I344095trial_trial / space dev as jacob.tan@sap.com...

   2018-11-29T01:05:34.60+0530 [RTR/0] OUT smbiotdemocfapp00.cfapps.eu10.hana.ondemand.com - [2018-11-28
:"-" x_forwarded_proto:"https" vcap_request_id:"dae0514a-e637-4e6a-57a0-b756e2902aa6" response_time:0.03
839c352612" x_b3_parentspanid:"-"
   2018-11-29T01:05:34.60+0530 [RTR/0] OUT
   2018-11-29T01:05:34.59+0530 [APP/PROC/WEB/0] OUT Stream body {"dev":"0","accx":"1","accy":"2","accz":
   2018-11-29T01:05:34.74+0530 [APP/PROC/WEB/0] OUT [ { code: 202,
   2018-11-29T01:05:34.74+0530 [APP/PROC/WEB/0] OUT     sensorAlternateId: 'c3a1c1c8b6b8a9f7',
   2018-11-29T01:05:34.74+0530 [APP/PROC/WEB/0] OUT     capabilityAlternateId: '4624879dd5fda56d',
   2018-11-29T01:05:34.74+0530 [APP/PROC/WEB/0] OUT     messages: [] } ]
```

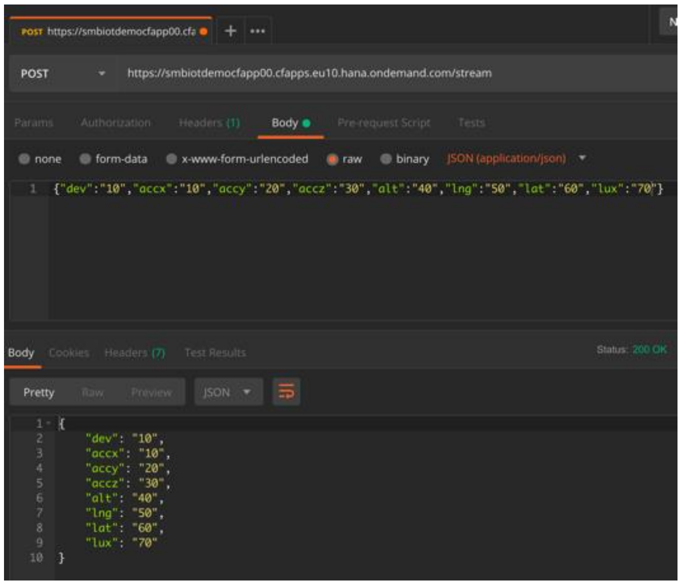
2. Test your Cloud Foundry App in **POSTMAN Rest Client** with a **POST request**.

   Request Type:
   POST

   URL:
   Application Route as captured above + **/stream**
   e.g.
   https://smbiotdemocfapp00.cfapps.eu10.hana.ondemand.com/stream

   Body:
   {"dev":"10","accx":"10","accy":"20","accz":"30","alt":"40","lng":"50","lat":"60","lux":"70"}

   Expected Response:
   Status:
   200 OK



POST https://smbiotdemocfapp00.cfapps.eu10.hana.ondemand.com/stream

```
{"dev":"10","accx":"10","accy":"20","accz":"30","alt":"40","lng":"50","lat":"60","lux":"70"}
```

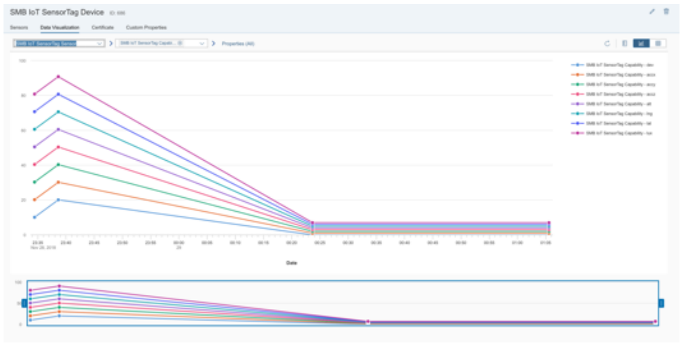
Status: 200 OK

```
{
    "dev": "10",
    "accx": "10",
    "accy": "20",
    "accz": "30",
    "alt": "40",
    "lng": "50",
    "lat": "60",
    "lux": "70"
}
```

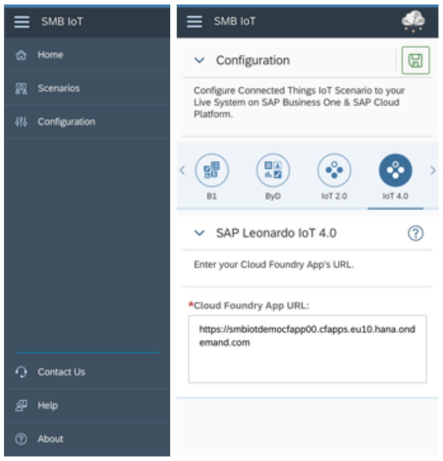
3. Data Steaming successfully.

   Open Internet of Things Service Cockpit.

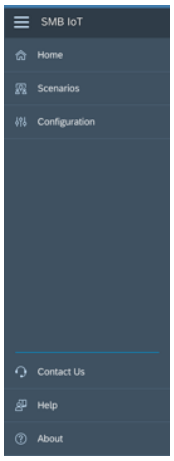
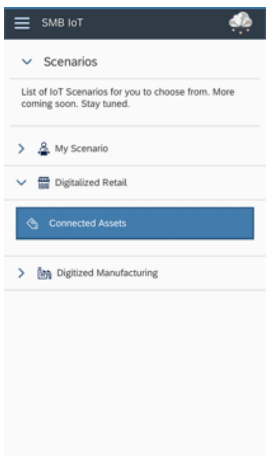
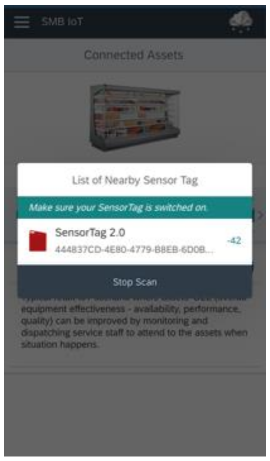
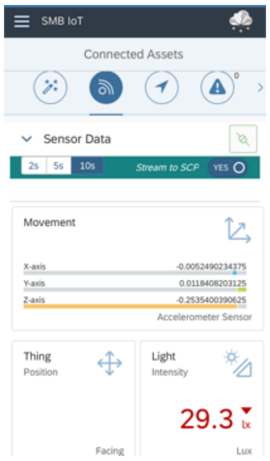
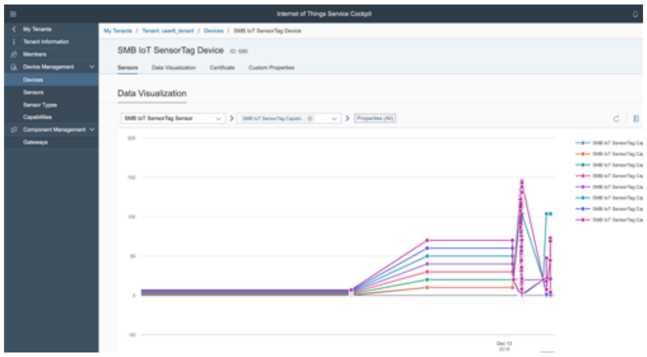
   Device Management > Devices > Data Visualization > Select Device & Capability



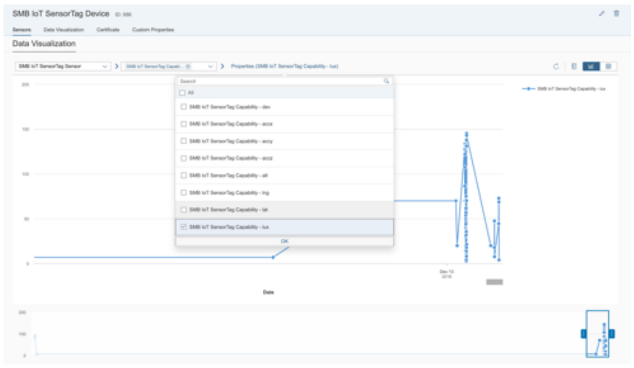
6.3 Setup SMB IoT App on your Phone and Start Streaming Data from SensorTag

Explanation	Screenshot
1. **Configure SMB.io App with CF URL** **Open** SMB.io App from your Phone. Go to **Menu** > **Configuration** > **IoT 4.0**. **Enter** the Cloud Foundry URL. The URL is referring to the Application Route from previous step. *e.g.* https://smbiotdemocfapp00.cfapps.eu10.hana.ondemand.com *Note: Please enter without the stream at the back of the URL.*	 SMB IoT Home Scenarios Configuration Contact Us Help About SMB IoT Configuration B1 ByD IoT 2.0 IoT 4.0 SAP Leonardo IoT 4.0 Configure Connected Things IoT Scenario to your Live System on SAP Business One & SAP Cloud Platform. Enter your Cloud Foundry App's URL. *Cloud Foundry App URL: https://smbiotdemocfapp00.cfapps.eu10.hana.ondemand.com App available on App Store (iOS)

6.3 Setup SMB IoT App on your Phone and Start Streaming Data from SensorTag

Explanation	Screenshot
2. <u>**Connect SensorTag to App & Start Streaming to SCP.**</u> **Open** SMB.io App from your Phone. Go to **Menu** > **Scenarios** > <u>**Connected Assets**</u>. **Switch** the Sensor to **ON**. **On** your SensorTag. **Select** your SensorTag in the list. Once connected, you should see a **Green Dot** indicating the app is connected to your SensorTag. **Navigate** to Tab and you should be able to see the Sensor Data sending to your SMB.io app. **Switch** the “Stream to SCP” to **ON**. *Note: By default, it will send sensor data to <u>SCP every 10 seconds</u>. You may change the frequency beside the switch.* Next, we will check in our SCP IoT 4.0 Service Cockpit if the data is coming.	 SMB IoT Home Scenarios Configuration Contact Us Help About  List of IoT Scenarios for you to choose from. More coming soon. Stay tuned. My Scenario Digitalized Retail Connected Assets Digitized Manufacturing  SMB IoT Connected Assets List of Nearby Sensor Tag Make sure your SensorTag is switched on. SensorTag 2.0 444837CD-4E80-4779-B8EB-6D0B... -42 Stop Scan equipment effectiveness - availability, performance, quality) can be improved by monitoring and dispatching service staff to attend to the assets when situation happens.  SMB IoT Connected Assets Sensor Data 2s 5s 10s Stream to SCP YES Movement X-axis -0.0052490234375 Y-axis 0.0118408203125 Z-axis -0.2535400390625 Accelerometer Sensor Thing Position Facing Light Intensity 29.3 lx Lux
3. <u>**Check in your SCP IoT Service Cockpit if the Data is received.**</u> **Devices** > **Data Visualization**	 Internet of Things Service Cockpit My Tenants Tenant Information Members Device Management Devices Sensors Sensor Types Capabilities Component Management Gateways SMB IoT SensorTag Device Sensors Data Visualization Certificate Custom Properties Data Visualization

6.3 Setup SMB IoT App on your Phone and Start Streaming Data from SensorTag

Explanation	Screenshot
4. You may choose specific properties to see the data patterns. **Congratulations, you've successfully setup the IoT connectivity with SensorTag and SCP IoT 4.0 services.**	

SMB IoT SensorTag Device

Sensors | Data Visualization | Certificate | Custom Properties

Data Visualization

SMB IoT SensorTag Sensor > SMB IoT SensorTag Capab... > Properties (SMB IoT SensorTag Capability - lux)

- [ ] All
- [ ] SMB IoT SensorTag Capability - dev
- [ ] SMB IoT SensorTag Capability - accx
- [ ] SMB IoT SensorTag Capability - accy
- [ ] SMB IoT SensorTag Capability - accz
- [ ] SMB IoT SensorTag Capability - alt
- [ ] SMB IoT SensorTag Capability - lng
- [ ] SMB IoT SensorTag Capability - lat
- [✓] SMB IoT SensorTag Capability - lux

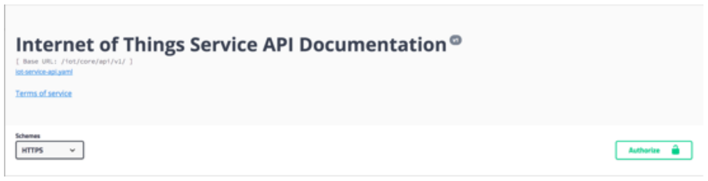
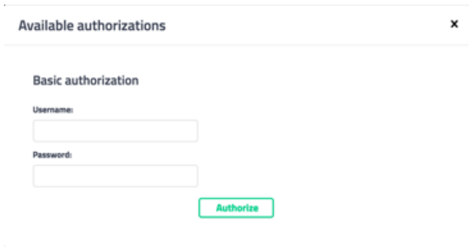
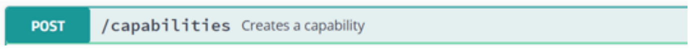
The following exercises are OPTIONAL. It is to demonstrate all that you've done earlier, which can also be done via API during run time.

3.1 [OPTIONAL] Creating a Device Entity with a Custom Sensor Type Using the API

In the following steps, a device is created using the API.
The device will have one sensor, which is of a custom sensor type.
Therefore, **capability and a sensor type are created initially**.

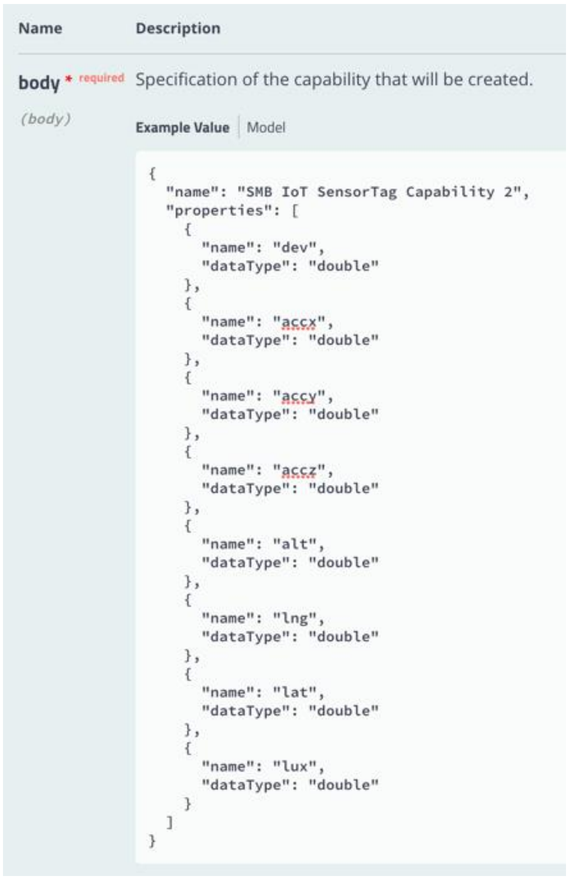
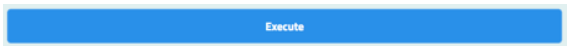
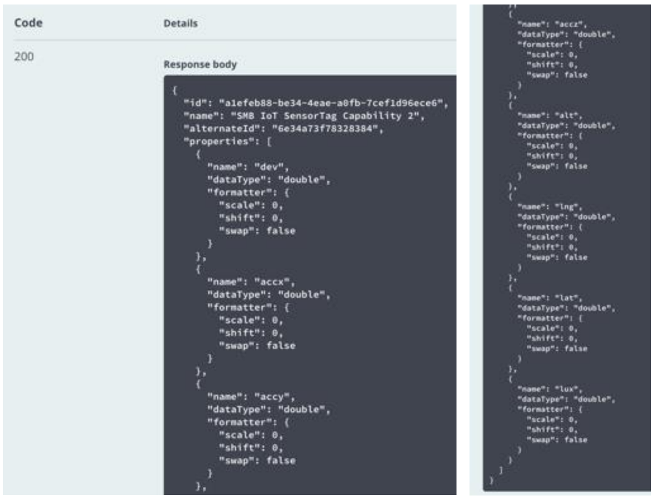
3.1.1 [OPTIONAL] Create a Capability Using the API

In the following a capability is created. A capability can be reused since it can be assigned to multiple sensor types. Each capability can have one or many properties.

Explanation	Screenshot
1. Open the **Internet of Things API Documentation** and choose **Authorize.** [IOT_INSTANCE_URL]/iot/core/api/v1/doc/	 Internet of Things Service API Documentation [Base URL: /iot/core/api/v1/] iot-service-api.yaml Terms of service
2. Enter your **user name** and **password** and choose **Authorize** to logon.	 Available authorizations × Basic authorization Username: Password: Authorize
3. Choose the POST request entry **/capabilities** (Creates a new capability).	 POST /capabilities Creates a capability
4. Choose **Try it out**.	

Explanation	Screenshot

5. Copy and replace the entire sample **JSON string** from the right which defines a capability with the **name** **“**SMB IoT SensorTag Capability 2*”* and **8 properties**.	{
	  "name": "SMB IoT SensorTag Capability 2",
	  "properties": [
	    {
	      "name": "dev",
	      "dataType": "double"
	    },
	    {
	      "name": "accx",
	      "dataType": "double"
	    },
	    {
	      "name": "accy",
	      "dataType": "double"
	    },
	    {
	      "name": "accz",
	      "dataType": "double"
	    },
	    {
	      "name": "alt",
	      "dataType": "double"
	    },
	    {
	      "name": "lng",
	      "dataType": "double"
	    },
	    {
	      "name": "lat",
	      "dataType": "double"
	    },
	    {
	      "name": "lux",
	      "dataType": "double"
	    }
	  ]
	}

Explanation	Screenshot
6. Paste the **JSON string** in the body field for POST request.	
7. Choose **Execute**.	
8. Scroll to the **response body.** In case of success the response code is 200 and the body contains all information of the created capability. <u>Note down the ==ID== of the capability.</u>	

Screenshot for step 6:

Name | Description

body * required (body) | Specification of the capability that will be created.

Example Value | Model

```
{
  "name": "SMB IoT SensorTag Capability 2",
  "properties": [
    {
      "name": "dev",
      "dataType": "double"
    },
    {
      "name": "accx",
      "dataType": "double"
    },
    {
      "name": "accy",
      "dataType": "double"
    },
    {
      "name": "accz",
      "dataType": "double"
    },
    {
      "name": "alt",
      "dataType": "double"
    },
    {
      "name": "lng",
      "dataType": "double"
    },
    {
      "name": "lat",
      "dataType": "double"
    },
    {
      "name": "lux",
      "dataType": "double"
    }
  ]
}
```

Screenshot for step 8:

Code | Details

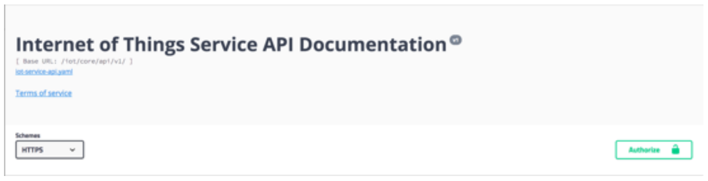
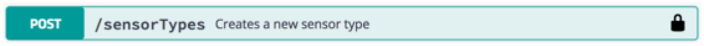
200 | Response body

```
{
  "id": "a1efeb88-be34-4eae-a8fb-7cef1d96ece6",
  "name": "SMB IoT SensorTag Capability 2",
  "alternateId": "6e34a73f78328384",
  "properties": [
    {
      "name": "dev",
      "dataType": "double",
      "formatter": {
        "scale": 0,
        "shift": 0,
        "swap": false
      }
    },
    {
      "name": "accx",
      "dataType": "double",
      "formatter": {
        "scale": 0,
        "shift": 0,
        "swap": false
      }
    },
    {
      "name": "accy",
      "dataType": "double",
      "formatter": {
        "scale": 0,
        "shift": 0,
        "swap": false
      }
    },
```

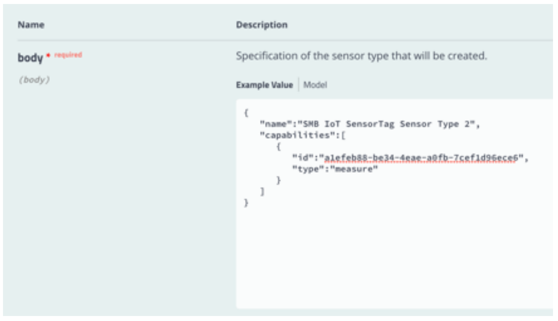
```
    {
      "name": "accz",
      "dataType": "double",
      "formatter": {
        "scale": 0,
        "shift": 0,
        "swap": false
      }
    },
    {
      "name": "alt",
      "dataType": "double",
      "formatter": {
        "scale": 0,
        "shift": 0,
        "swap": false
      }
    },
    {
      "name": "lng",
      "dataType": "double",
      "formatter": {
        "scale": 0,
        "shift": 0,
        "swap": false
      }
    },
    {
      "name": "lat",
      "dataType": "double",
      "formatter": {
        "scale": 0,
        "shift": 0,
        "swap": false
      }
    },
    {
      "name": "lux",
      "dataType": "double",
      "formatter": {
        "scale": 0,
        "shift": 0,
        "swap": false
      }
    }
  ]
}
```

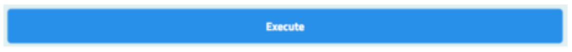
3.1.2 [OPTIONAL] Create a Sensor Type Using the API

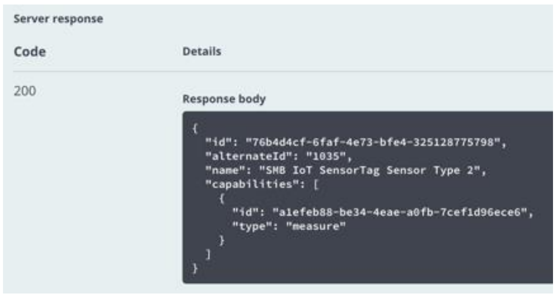
In the following a sensor type is created. The previously created capability is assigned to the sensor type.

Explanation	Screenshot
1. Open the **Internet of Things API Documentation** and choose **Authorize.**	 Internet of Things Service API Documentation [Base URL: /iot/core/api/v1/] iot-service-api.yaml Terms of service Schemes: HTTPS Authorize
2. Choose the POST request entry **/sensorTypes**	 POST /sensorTypes Creates a new sensor type
3. Choose **Try it out**.	Try it out

4. Copy and replace the entire sample **JSON string** from the right which defines a sensor type with the **name "**SMB IoT SensorTag Sensor Type 2" and one property.
 	
Replace the "CAPBILITY_ID" with the **capability id** created previously.
	{
	"name":"SMB IoT SensorTag Sensor Type 2",
	"capabilities":
	[{
	"id":" CAPBILITY_ID ",
	"type":"measure"
	}]
	}

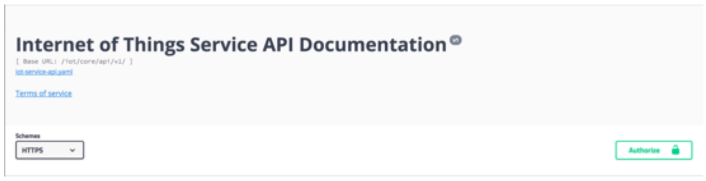
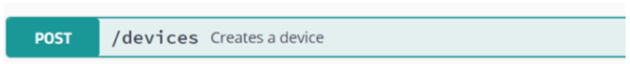
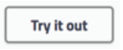
5. Paste the **JSON string** in the body field for POST request.	 Name: body * required (body)
	Description: Specification of the sensor type that will be created.
	Example Value | Model
	{
	"name":"SMB IoT SensorTag Sensor Type 2",
	"capabilities":[
	{
	"id":"a1efeb88-be34-4eae-a0fb-7cef1d96ece6",
	"type":"measure"
	}
	]
	}

6. Choose **Execute**.	 Execute

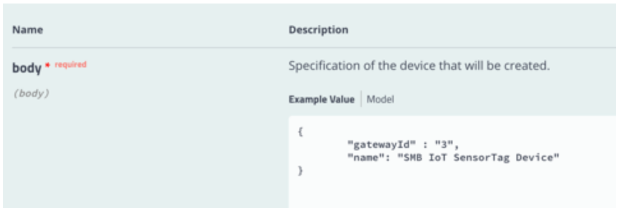
7. Scroll to the **response body.** In case of success the response code is 200 and the body contains all information of the created sensor type.
 	
Note down the ID of the sensor type.
	 Server response
	Code: 200
	Details: Response body
	{
	"id": "76b4d4cf-6faf-4e73-bfe4-325128775798",
	"alternateId": "1035",
	"name": "SMB IoT SensorTag Sensor Type 2",
	"capabilities": [
	{
	"id": "a1efeb88-be34-4eae-a0fb-7cef1d96ece6",
	"type": "measure"
	}
	]
	}

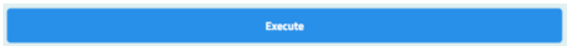
3.1.3 [OPTIONAL] Create a Device Using the API

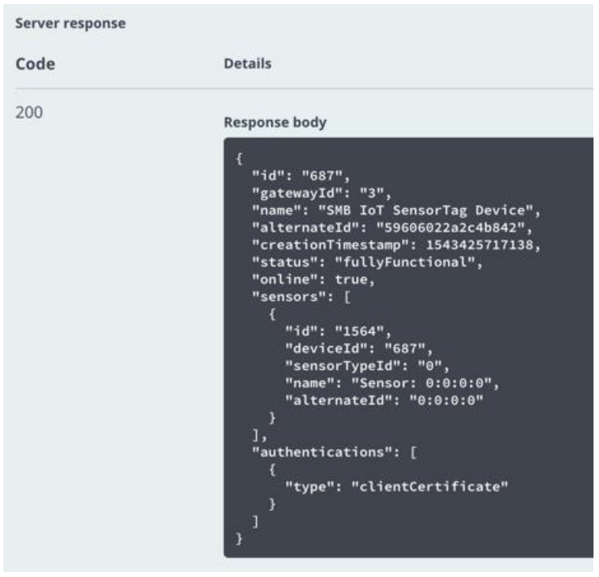
In the following a device is created. The device entity will be assigned to one specific gateway.

Explanation	Screenshot
1. Open the **Internet of Things API Documentation** and choose **Authorize.**	 Internet of Things Service API Documentation [Base URL: /iot/core/api/v1/] iot-service-api.yaml Terms of service Schemes: HTTPS
2. Choose the POST request entry **/devices** (Creates a new device).	
3. Choose **Try it out**.	

4. Copy the sample **JSON string** from the right which defines a device with the **name "*SMB IoT SensorTag Device*"** which is assigned to the gateway with the ID 3 (REST).	{
	"gatewayId" : "3",
	"name": "SMB IoT SensorTag Device"
	}

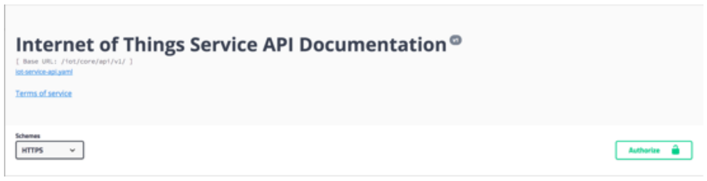
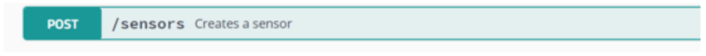
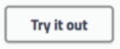
5. Paste the **JSON string** in the body field for POST request.	 Name: body * required (body)
	Description: Specification of the device that will be created.
	Example Value | Model
	`{`
	`"gatewayId" : "3",`
	`"name": "SMB IoT SensorTag Device"`
	`}`

6. Choose **Execute**.	

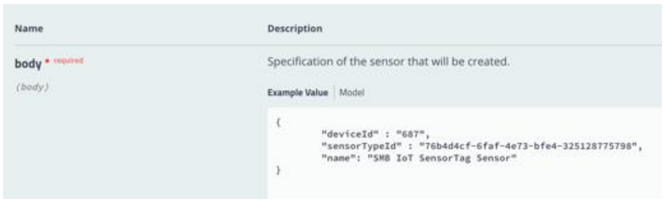
7. Scroll to the **response body.** In case of success the response code is 200 and the body contains all information of the created device.
 	
Note down the **ID** of the Device
	 Server response
	Code: 200
	Details: Response body
	`{`
	`"id": "687",`
	`"gatewayId": "3",`
	`"name": "SMB IoT SensorTag Device",`
	`"alternateId": "59606022a2c4b842",`
	`"creationTimestamp": 1543425717138,`
	`"status": "fullyFunctional",`
	`"online": true,`
	`"sensors": [`
	`{`
	`"id": "1564",`
	`"deviceId": "687",`
	`"sensorTypeId": "0",`
	`"name": "Sensor: 0:0:0:0",`
	`"alternateId": "0:0:0:0"`
	`}`
	`],`
	`"authentications": [`
	`{`
	`"type": "clientCertificate"`
	`}`
	`]`
	`}`

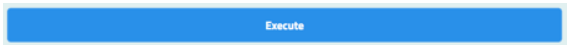
3.1.4 *[OPTIONAL] Create a Sensor Using the API*

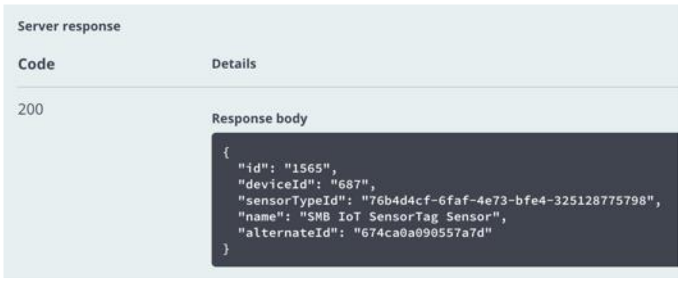
In the following a sensor is created. The sensor will be assigned to the previously created device.

Explanation	Screenshot
1. Open the **Internet of Things API Documentation** and choose **Authorize.**	 Internet of Things Service API Documentation [Base URL: /iot/core/api/v1/] iot-service-api.yaml Terms of service Schemes: HTTPS
2. Choose the POST request entry **/sensors** (Creates a new sensor).	 POST /sensors Creates a sensor
3. Choose **Try it out**.	

4. Copy and replace the entire sample **JSON string** from the right which defines a sensor with the **name “***SMB IoT SensorTag Sensor”.*
 	
Replace the “*DEVICE_ID”* with the **device id** of the previously created device (Step 45)
 	
Replace the “*SENSOR_TYPE_ID”* with the **sensor type id** of the previously created sensor type. (Step 38)
	{
	"deviceId" : "DEVICE_ID",
	"sensorTypeId" : "SENSOR_TYPE_ID",
	"name": " *SMB IoT SensorTag Sensor* "
	}

5. Paste the **JSON string** in the body field for POST request.	 Name: body * required (body)
	Description: Specification of the sensor that will be created.
	Example Value | Model
	{
	"deviceId" : "687",
	"sensorTypeId" : "76b4d4cf-6faf-4e73-bfe4-325128775798",
	"name": "SMB IoT SensorTag Sensor"
	}

6. Choose **Execute**.	

7. Scroll to the **response body.** In case of success the response code is 200 and the body contains all information of the created sensor.
 	
Note down the **ID** of the sensor.
	 Server response
	Code: 200
	Details: Response body
	{
	"id": "1565",
	"deviceId": "687",
	"sensorTypeId": "76b4d4cf-6faf-4e73-bfe4-325128775798",
	"name": "SMB IoT SensorTag Sensor",
	"alternateId": "674ca0a090557a7d"
	}

www.sap.com/contactsap

Run Simple