

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. The left sidebar has a dark theme with white text. The 'Services' option is highlighted with a blue background. The main content area shows a breadcrumb navigation: Home [Europe (Rot) - Trial] / Europe (Rot) - Trial / i344095trial. Below this, it says 'Subaccount: i344095trial - Services' and 'Filtered: 1 of 39'. A search bar is present. Under the 'Internet of Things' section, there is a card for 'Internet of Things' with a 'Not enabled' status, a 'Develop, customize, and operate IoT business applications in the cloud.' description, and a small globe icon.

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

The screenshot shows the 'Internet of Things - Overview' page in the SAP Cloud Platform Cockpit. The left sidebar is visible. The main content area shows a breadcrumb navigation: Home [Europe (Rot) - Trial] / Europe (Rot) - Trial / i344095trial / Internet of Things. Below this, it says 'Service: Internet of Things - Overview' and 'Not enabled'. There is a large 'Enable' button. A 'Service Description' section provides a brief overview of the service. At the bottom, there is a 'Take Action' section with 'Configure Service' and 'Go to Service' buttons.

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).

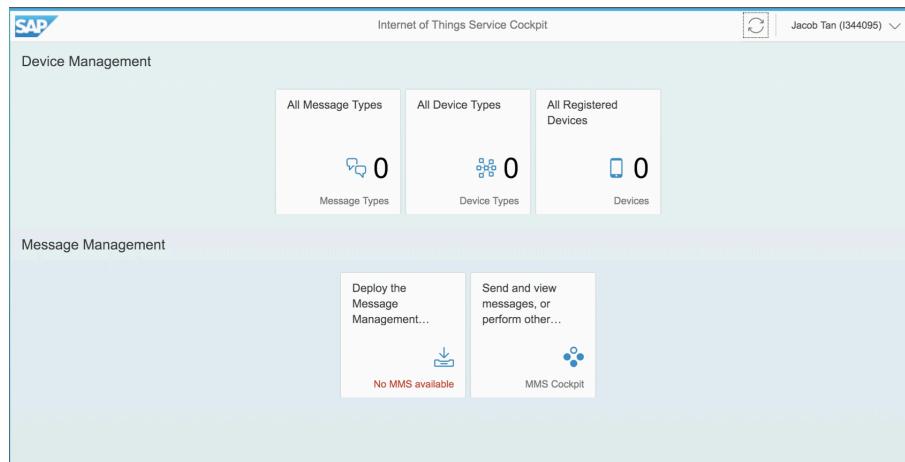
Deploying the Message Management Service (MMS)

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**”.

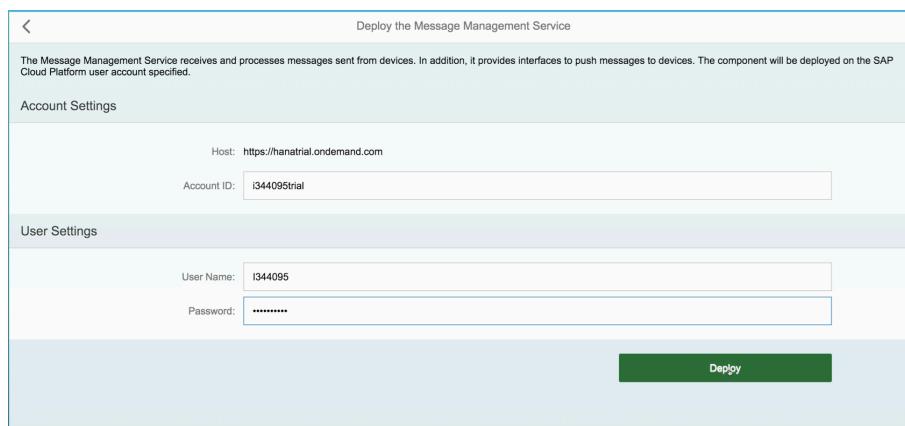
Click on “**Deploy Message Management Service**” on the bottom left tile.



The screenshot shows the SAP Internet of Things Service Cockpit interface. At the top, there are three main sections: Device Management, Message Management, and Analytics. The Device Management section contains three tiles: All Message Types (0), All Device Types (0), and All Registered Devices (0). The Message Management section contains two tiles: 'Deploy the Message Management...' (No MMS available) and 'Send and view messages, or perform other...' (MMS Cockpit). The Analytics section is partially visible at the bottom.

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.

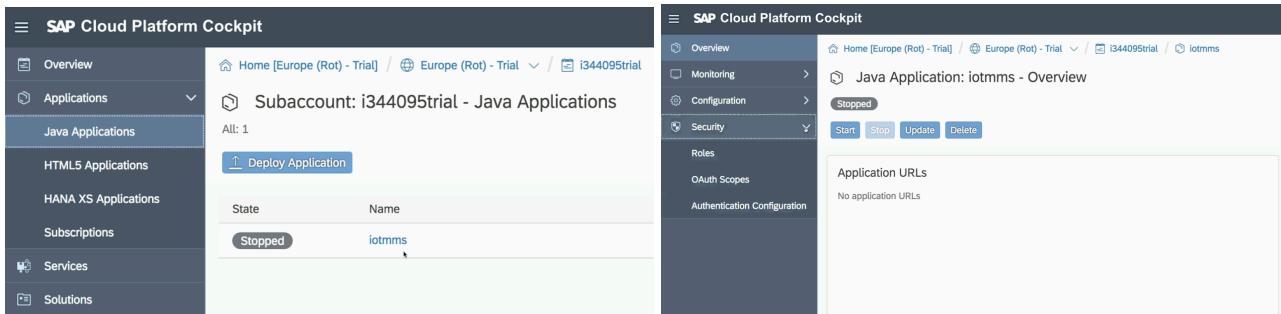


The screenshot shows a deployment dialog box titled “Deploy the Message Management Service”. It includes a note about the service receiving and processing messages from devices. The “Account Settings” section contains fields for “Host” (set to “https://hanatrial.ondemand.com”) and “Account ID” (set to “i344095trial”). The “User Settings” section contains fields for “User Name” (set to “i344095”) and “Password” (redacted). A large green “Deploy” button is located at the bottom right.

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

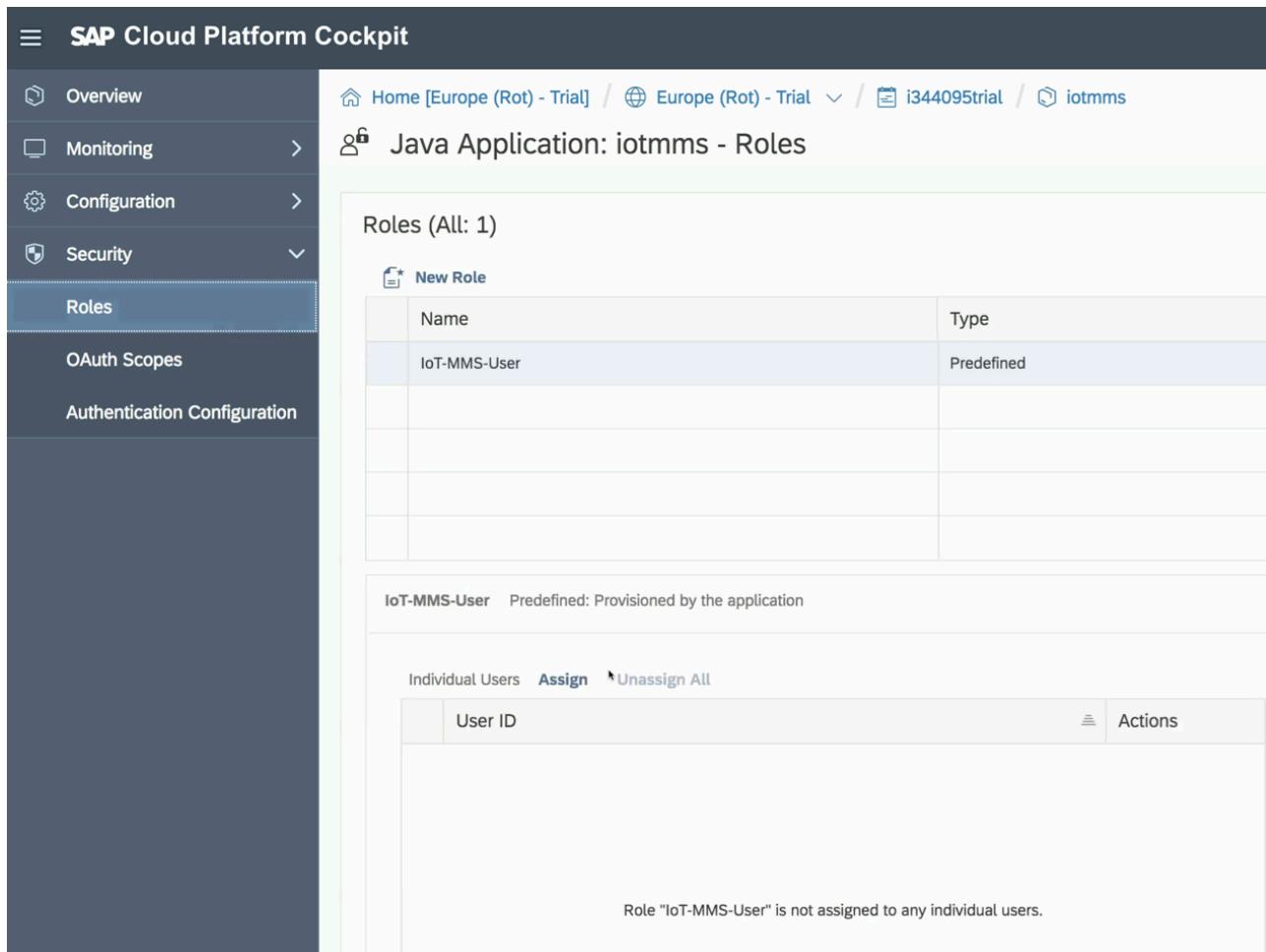
From the SAP Cloud Platform Cockpit, select **Java Applications** and Click on **iotmms**. This will take us to the Java Application Overview screen.



The left screenshot shows the SAP Cloud Platform Cockpit interface with the Java Applications menu item selected. A table lists one application: 'Subaccount: i344095trial - Java Applications' with a state of 'Stopped' and name 'iotmms'. The right screenshot shows the Java Application Overview for 'iotmms'. The left sidebar has 'Security > Roles' selected. The main area shows the application status as 'Stopped' with buttons for Start, Stop, Update, and Delete. Below that is a section for Application URLs, which is currently empty.

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 with the Roles menu item selected. The main area displays a table titled 'Roles (All: 1)' with one entry: 'Name' (IoT-MMS-User) and 'Type' (Predefined). Below the table, a note states: 'IoT-MMS-User Predefined: Provisioned by the application'. At the bottom, there is a section for 'Individual Users' with an 'Assign' button and a table for selecting a User ID. A message at the bottom right says: 'Role "IoT-MMS-User" is not assigned to any individual users.'

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. On the left, there's a sidebar with navigation links: Overview, Monitoring, Configuration, Security (with Roles selected), OAuth Scopes, and Authentication Configuration. The main content area is titled 'Java Application: iotmms - Roles'. It displays a table of roles with one entry: 'Name' IoT-MMS-User and 'Type' Predefined. A modal dialog box is overlaid on the screen, titled 'Assign role "IoT-MMS-User" to user'. It contains a 'User ID' input field with the value 'i344095' and two buttons at the bottom: 'Assign' and 'Cancel'.

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 create a Database Schema to bind it with the iotmms Java Application.

Click on your **SCP Neo Account** (xXXXXXtrial) on the breadcrumbs to go back to the main screen.

This screenshot shows the SAP Cloud Platform Cockpit interface again, but from a different perspective. The sidebar and breadcrumb navigation are identical to the previous screenshot. The main content area is now empty, showing only the title 'Java Application: iotmms - Roles'.

STEP 2: CREATING HANA DB SCHEMA FOR IOTMMS

A database schema is required to host your IoT device sensor data in SAP Cloud Platform. In this step, we will be creating a HANA trial database schema.

Note: The trial database schema will be **stopped automatically after 12 hours** to free up resources. If no further activity, it will be **deleted after 7 days**.

Go to **SAP HANA / SAP ASE > Database & Schemas > Click on New**

The screenshot shows the SAP Cloud Platform Cockpit interface. The left sidebar has a dark theme with the following navigation items: Overview, Applications, Services, Solutions, SAP HANA / SAP ASE (with a dropdown arrow), Database Systems, Databases & Schemas (which is highlighted in blue), and Service Requests. The main content area is titled 'Subaccount: i344095trial - Databases & Schemas'. It shows a table with one row, labeled 'All: 0'. A blue 'New' button is located above the table. The table columns are 'DB/Schema ID' and 'Type'.

Enter Database & Schema details. Click **Create**.

Note: Remember your SYSTEM user Password

The screenshot shows the 'New Database/Schema' creation form. The title is 'New Database/Schema'. The form fields are as follows:

- *Database ID:
- Database System:
- *SYSTEM User Password:
- *Confirm Password:
- Configure a User for SHINE.: OFF (OFF is selected)
- Parameters:
 - Web Access: ON
 - DP Server: ON

At the bottom right are 'Create' and 'Cancel' buttons.

It will take about **5 – 10 mins** for the Database Schema to be created.

The screenshot shows the SAP Cloud Platform Cockpit interface. On the left, there is a navigation menu with options: Overview, Data Source Bindings, Permissions, and Events. The Events option is selected. The main content area is titled "Database: iotmmsdb - Events". It displays a table of events with columns: Time, ID, Event, and Description. There are 17 entries in the log, all of which are successful (indicated by green checkmarks). Some of the entries include: "Tenant DB creation finished successfully (created and configured)", "HTTPS access from Internet enabled", "Web dispatcher configuration added successfully", "Database started successfully", and "Database creation started".

Once created successfully, Click on **Data Source Bindings** on the Left menu.

The screenshot shows the SAP Cloud Platform Cockpit interface. The left menu is visible with the "Data Source Bindings" option selected. The main content area is titled "Database: iotmmsdb - Overview". It shows a status message: "STARTED" followed by a note that it will be stopped in 11 hour(s) and 58 minute(s). Below this are three buttons: "Configure", "Stop", and "Delete". A large box labeled "Details" contains the following information: Type: Database, Database System: HANA MDC (<trial>), Version: 1.00.122.08.1490178281, Bindings: 0 available. Under "Administration Tools", it lists SAP HANA Cockpit, SAP HANA Web-Based Development Workbench, and SAP HANA Interactive Education (SHINE).

Bind the Database Schema to your iotmms Java Application
Click on **New Binding** > (By default) Select **iotmms** > Enter **SYSTEM** credentials > **Save**

The screenshot shows the SAP Cloud Platform Cockpit interface. The left menu is visible with the "Data Source Bindings" option selected. The main content area is titled "Database: iotmmsdb - Data Source Bindings". It shows a message: "All: 0" and a button: "New Binding...". Below this is a table with columns: Data Source, Java Application, and Custom Logon. The table currently has one row: "No data source bindings.". A modal dialog box titled "New Binding" is open. It contains fields for: *Data Source: <default>, *Java Application: iotmms, Custom Logon:, *Database User: SYSTEM, and *Password: (a masked string). There is also a checked checkbox: Verify credentials. At the bottom of the dialog are "Save" and "Cancel" buttons.

Click on your **SCP Neo Account** (xXXXXXtrial) on the breadcrumbs to go back to the main screen.

Select **Applications > Java Applications > iotmms**

The screenshot shows the SAP Cloud Platform Cockpit interface. On the left, there is a navigation sidebar with various options like Overview, Applications, Java Applications (which is selected), HTML5 Applications, HANA XS Applications, Subscriptions, Services, Solutions, SAP HANA / SAP ASE, Database Systems, Databases & Schemas, and Service Requests. The main content area is titled "Subaccount: i344095trial - Java Applications". It shows a single application entry: "iotmms" with a state of "Stopped". There is a "Deploy Application" button at the top.

(Optional) Under **Data Source Bindings**, Check that you have the **iotmmsdb** bound.

The screenshot shows the SAP Cloud Platform Cockpit interface. The navigation sidebar includes Overview, Monitoring, Configuration (selected), Destinations, Data Source Bindings (selected), Security, Roles, OAuth Scopes, and Authentication Configuration. The main content area is titled "Java Application: iotmms - Data Source Bindings". It shows one binding entry: "iotmms" with a DB/Schema ID of "iotmmsdb". There is a "New Binding" button at the top.

Start your iotmms Java Application.

Within a few minutes it should be started.

The screenshot displays two side-by-side views of the SAP Cloud Platform Cockpit Java Application overview. Both views show the same application details: "iotmms" is listed under "Java Application: iotmms - Overview". The left view shows a status of "Starting" with buttons for "Start additional process", "Stop", "Update", and "Delete". The right view shows a status of "Started" with the same buttons. Both screens also show the "Application URLs" as <https://iotmms344095trial.hanatrial.ondemand.com/>.

Congratulations, now you've successfully **(1)** Created a iotmms Java Application to support your IoT scenarios & **(2)** Bind it to a HANA Database in SAP Cloud Platform! Next, we will add our IoT device (SensorTag) into the IoT Service Cockpit.

STEP 3: 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 "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: 4 items (represented by a speech bubble icon)
- All Device Types: 3 items (represented by a gear icon)
- All Registered Devices: 3 items (represented by a smartphone icon)

Message Management: This section contains two cards:

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

At the bottom left, there is a notification icon with a count of 1. The footer contains links to 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 search bar and 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' (disabled) and 'Fields' (selected). Below the tabs is a table with columns: Position, Name, Type, and Optional Settings. The table has 9 rows, each representing a field: timestamp (date), dev (string), accx (string), accy (string), accz (string), alt (string), long (string), lat (string), and lux (string). At the bottom of the table is a summary row: '9 fields added'. At the very bottom of the interface is a dark footer bar with three buttons: a '+' sign, a 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 Services Cockpit interface. On the left, there is a sidebar with a back arrow icon, a search bar, and a refresh button. Below the search bar, there are three entries:

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

On the right, the main area is titled "Message Type" and displays the details for the "B1IoT SensorTag Message Type". It shows the ID "2de0e4992cd7b86ad115" and two buttons: "Information" (with an "i" icon) and "Fields" (with a magnifying glass icon). The "Information" button is underlined, indicating it is selected.

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 Services Cockpit interface. At the top, there is a header with the SAP logo, the title "Internet of Things Service Cockpit", a refresh button, and a user profile "Jacob Tan (P1558868588)".

The main area is divided into sections:

- Device Management**: Contains three tiles:
 - All Message Types: 5 Message Types
 - All Device Types: 4 Device Types
 - All Registered Devices: 3 Devices
- Message Management**: Contains two tiles:
 - Redeploy the Message Management: Up to Date
 - Send and view messages, or perform other: MMS Cockpit

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 'Device Types' creation interface. 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 is titled 'Create Device Type' and has two tabs: '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, '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 warning icon with a '1' indicating one pending action.

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

Information

*Name: B1 IoT SensorTag Device Type

Further Details: Enter a URL to call up further details

Message Types

Assignment Name Message Type Direction

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 "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:** Shows 5 message types. The icon is a speech bubble with two dots. Below it says "Message Types".
- All Device Types:** Shows 4 device types. The icon is a gear-like shape. Below it says "Device Types".
- All Registered Devices:** Shows 3 registered devices. The icon is a smartphone. Below it says "Devices".

Message Management:

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

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).

Devices

Create Device

Information

Name: B1 IoT SensorTag Device

Device Type: B1 IoT SensorTag Device Type

Further Details: Enter a URL to call up further details

Custom Attributes

Add Custom Attribute

Key	Value
No data	

+

! 1

Create 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.

Devices

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

Device

B1 IoT SensorTag Device

Information Authentication

ID: 7d6a8683-e692-4c3f-aba2-31ddbf41bd0

Device Type: B1 IoT SensorTag Device Type

Authentication Type: OAuth

OAuth Access Token

New OAuth access token for device "B1 IoT SensorTag Device" created.
Token: ada6a44d4840efe9ba5ba8e1fdfcae5a

Close OK

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's a sidebar titled "Devices" with a search bar and a refresh icon. Below the search bar, there are four device entries:

- B1 IoT SensorTag Device (selected, highlighted in blue)
- DevelopmentTessel
- DeviceSimulator
- iOS Device

For the selected device, the main panel has a title "B1 IoT SensorTag Device". Below the title are two circular icons: "Information" (blue) and "Authentication" (light blue). The "Information" tab is active. To its right, detailed device information is displayed:

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

At the bottom of the main panel, there are three buttons: "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 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 tiles:

- All Message Types: Shows 5 message types, represented by a speech bubble icon.
- All Device Types: Shows 4 device types, represented by a gear icon.
- All Registered Devices: Shows 4 registered devices, represented by a smartphone icon.

Message Management: This section contains two tiles:

- Update the Message Management: Shows a download icon and the text "New Version: 2.56.0".
- Send and view messages, or perform other: Shows a gear icon and the 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 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, represented by a cube icon.
- Push Messages to Devices: Push Service, represented by a bar chart icon.
- Configure the Message Management Service: Configuration, represented by a gear icon.
- Configure Processing Service Mappings: Configuration, represented by a gear icon.
- View Registered Devices And Device Types: IoT Service Cockpit, represented by a hand icon.

Data Services: This section contains four tiles:

- Messaging through HTTP: HTTP API, represented by a gear icon.
- Messaging through MQTT over TCP: MQTT TCP API, represented by a gear icon.
- Messaging through WebSockets: WebSocket API, represented by a gear icon.
- Messaging through MQTT over WS: MQTT WebSocket API, represented by a gear icon.

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 text box with the following content:

```
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.
```

Below the description is a "Send Message" section. It contains two input fields: "Data Endpoint" and "Message".

Data Endpoint: `https://iotmmsp1558868588trial.hanatrial.ondemand.com/com.sap.iotservices.mms/v1/api/http/data/d000-e000-v000-i000-c000-e001`

Message: `{"mode":"sync","messageType":"m0t0y0p0e1","messages":[{"sensor":"sensor1","value":20,"timestamp":1413191650}]}`

At the bottom right of the message section is a "Send" button with a gear 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://iotmms<your-i/s/pXXXXX>trial.hanatrial.ondemand.com/com.sap.iotservices.mms/v1/api/http/data/<replace-this-part-with-your-device-id>>

Replace the red parts with your values.

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. Replace those in red to your values setup previously.

POST

<https://iotmms<i/s/pXXXXXX>trial.hanatrial.ondemand.com/com.sap.iotservices.mms/v1/api/http/data/<replace-this-part-with-your-device-id>>

Headers

Content-Type: application/json

Authorization: Bearer <Your OAuth Access Token>

Body:

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

The screenshot shows the Postman interface with the following details:

- Request URL:** https://iotmmsp1558868588trial.hanatrial.ondemand.com/com.sap.iotservices.mms/v1/api/http/data/7d6a8683-e692-4c3f-aba2-31ddbf41bd0
- Headers:**
 - Content-Type: application/json
 - Authorization: Bearer ada6a44d4840ef9ba5ba8e1fdfcae5a
- Body:**
 - form-data
 - x-www-form-urlencoded
 - raw
 - binary
 - JSON (application/json)

```
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: }
```

Response:

The screenshot shows the Postman interface with the following details:

- Status:** 200 OK
- Time:** 3056 ms
- Size:** 299 B
- Body:**
 - Pretty
 - Raw
 - Preview
 - JSON

```
1: {  
2:   "msg": "1 message(s) received from device [7d6a8683-e692-4c3f-aba2-31ddbf41bd0]"  
3: }
```

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 interface. At the top, there are four icons: a blue circle with an 'i', a blue circle with a signal, a blue circle with a location pin, and a blue circle with an exclamation mark. Below these is a dropdown menu labeled "Sensor Data" with a "Connected" status indicator. A "Stream to SCP" button is visible, along with a timer set to "2s". The main area displays sensor data for "Movement" (X-axis: -0.0478515625, Y-axis: 0.029296875, Z-axis: -0.9912109375) and an "Accelerometer Sensor". It also shows "Thing Position" and "Light Intensity" (138 lux).

The right screenshot shows the configuration screen for streaming to SAP Cloud Platform IoT. It includes fields for "URL" (https://iotmmsp1558868588trial.hanatrial.ondem...), "Mode" (sync), "Message Type" (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" (application/json) and "Authorization" (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

URL

To send data to an endpoint using the HTTP API, a device needs to send an HTTP POST request to the URL `https://<host>/com.sap.iotservices.mms/<version>/api/http/data/<device id>`, where `<version>` is the current API version 'v1' for the MMS in this exercise and `<device id>` is the identifier for the device sending the data.

Message Type

After a device has been successfully registered within RDMS and has been assigned to a certain device type, the device can start sending messages that conform to a message type with ID. This message contains the fields: `timestamp`, `accx/y/z`, `lat`/`long`/`alt` & `lux`. The fields are part of the respective message type definition.

Content Type (Header)

One of the Header parameters of a typical web service call that defines this body payload (message) is of JSON format.

Mode

Using the field 'mode' you can control the data sending mode, depending on the value of the field:

- `sync` - synchronous message delivery, request is blocked until message is successfully processed. We will use `sync` for this exercise.
- `async` - asynchronous message delivery, no acknowledgment. If the request has been validated successfully and processing continues, the server responds with code 202 - Accepted.

Message

These messages are generated by B1 IoT app (predefined in the code level) to follow the similar format of that which you have defined in your `MessageType` in SCP. It contains fields & sensor data value that will eventually be sent as JSON in HTTP body.

Authorization (Header)

For the HTTP API, the token needs to be added as a request header to all requests. The header has the following schema:
`Authorization: Bearer <oauth_token>`

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*

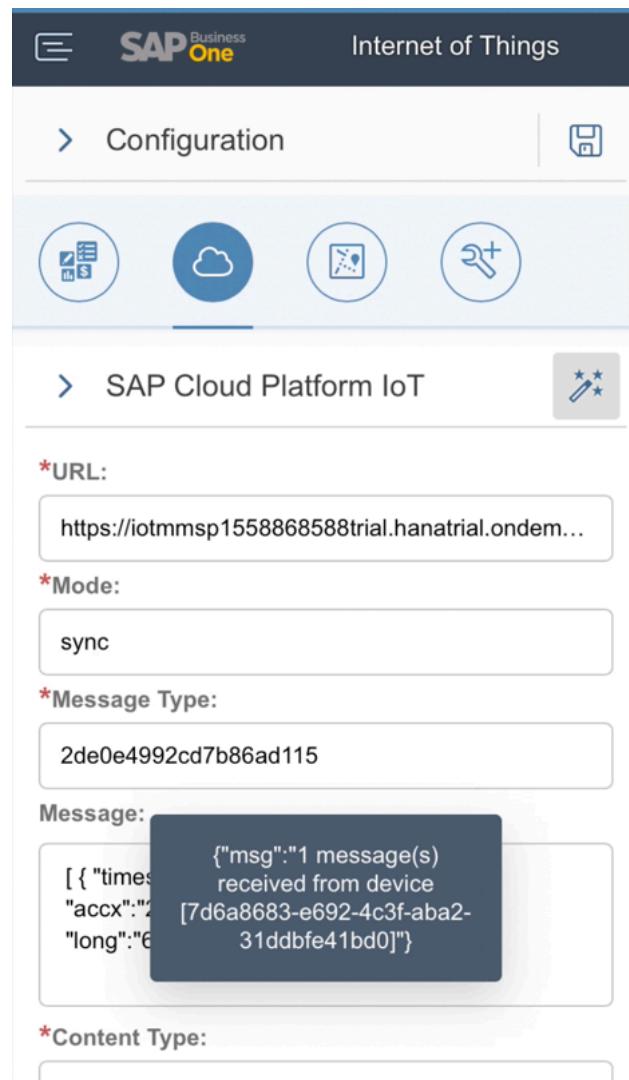
Header

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 IoT Service Cockpit selected. The top navigation bar displays "Internet of Things Service Cockpit" and the user "Jacob Tan (P1558868588)". The main area is divided into two main sections: "Device Management" and "Message Management".

- Device Management:** Contains three tiles:
 - "All Message Types" (5): Shows a speech bubble icon and the number 5.
 - "All Device Types" (4): Shows a gear icon and the number 4.
 - "All Registered Devices" (3): Shows a smartphone icon and the number 3.
- Message Management:** Contains two tiles:
 - "Redeploy the Message Management" (Up to Date): Shows a download icon and the status "Up to Date".
 - "Send and view messages, or perform other" (MMS Cockpit): Shows a message icon and the label "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. The top navigation bar displays "Message Management Service Cockpit" and the user "Jacob Tan (P1558868588)".

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_5XBN2L7P45ONSUN1JN9XX7ALV 1 >

 T_IOT_3F676714E256C6F959C2
NEO_5XBN2L7P45ONSUN1JN9XX7ALV 1708 >

 T_IOT_A1D6DBB3B5BF0C68025A
NEO_5XBN2L7P45ONSUN1JN9XX7ALV 8 >

 T_IOT_ACKSTORE
NEO_5XBN2L7P45ONSUN1JN9XX7ALV 2 >

 T_IOT_CONFIG
NEO_5XBN2L7P45ONSUN1JN9XX7ALV 67 >

 T_IOT_HTTP_PUSH
NEO_5XBN2L7P45ONSUN1JN9XX7ALV >

 T_IOT_MONITOR_LOG
NEO_5XBN2L7P45ONSUN1JN9XX7ALV 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_5XBN2L7P45ONSUN1JN9XX7ALV.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 (+08)	Wed May 13 2015 01:05:13 (+08)	1	2	3	4	5	6	7	8



Application Data



Table



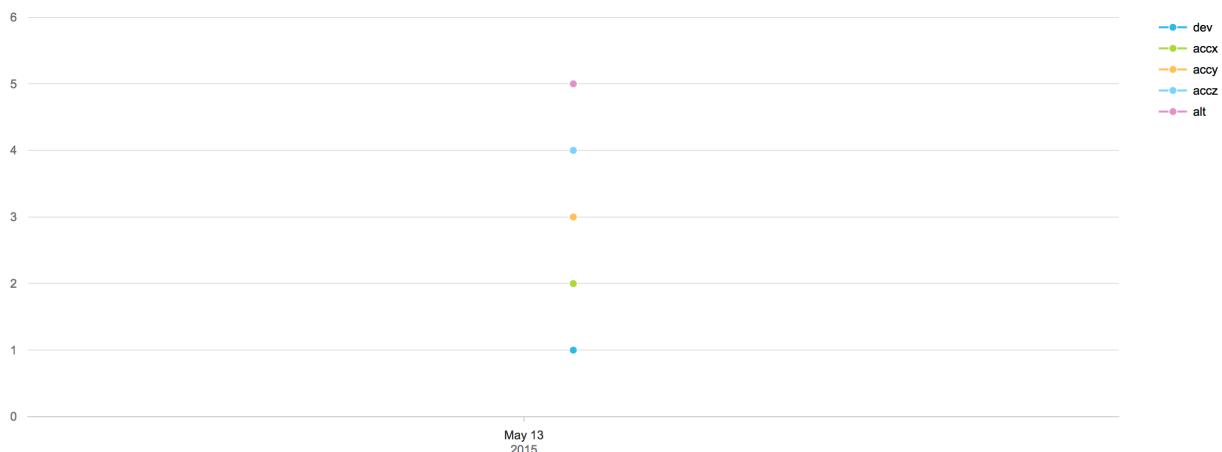
Chart

▼ 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.

