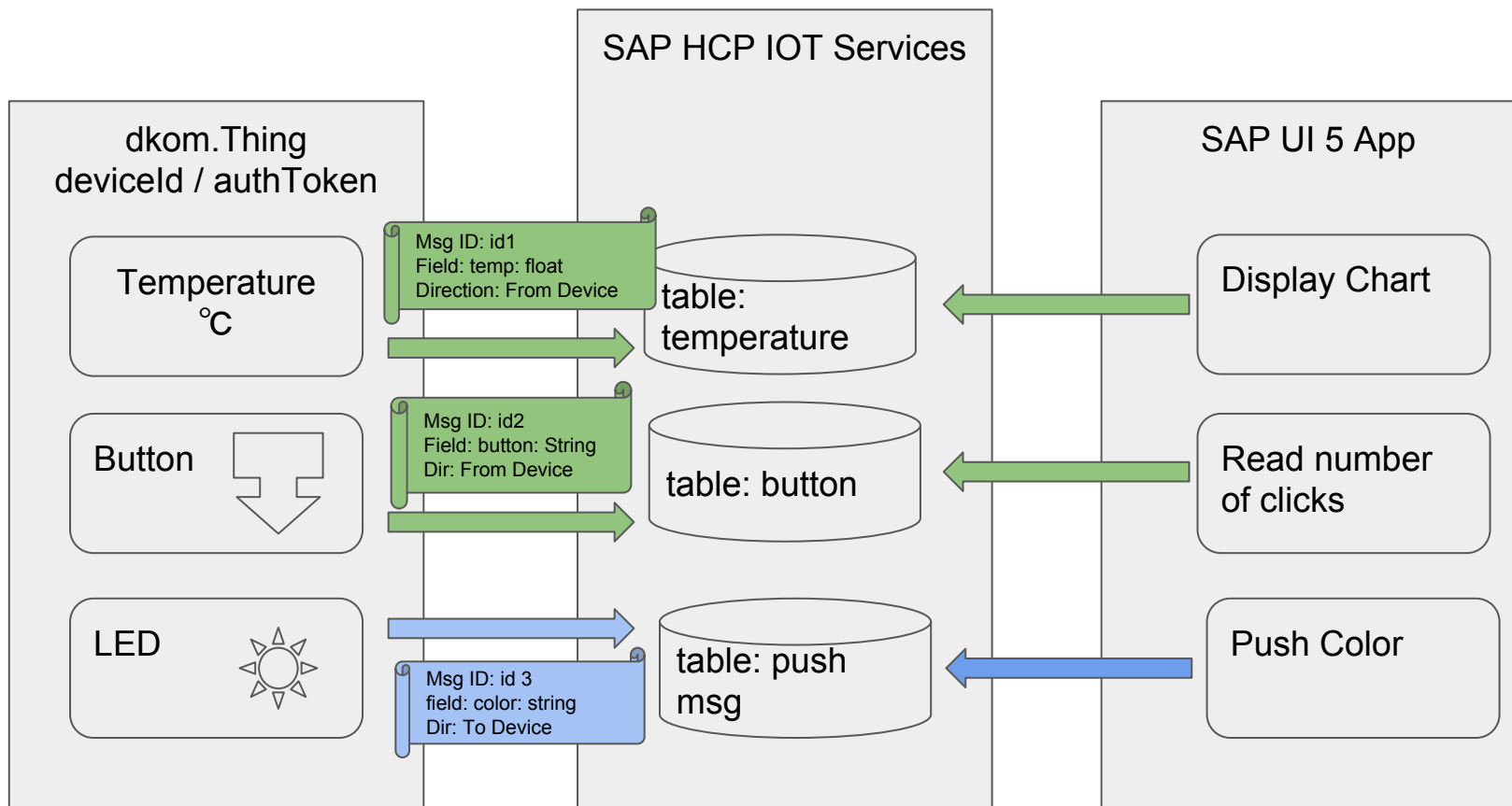


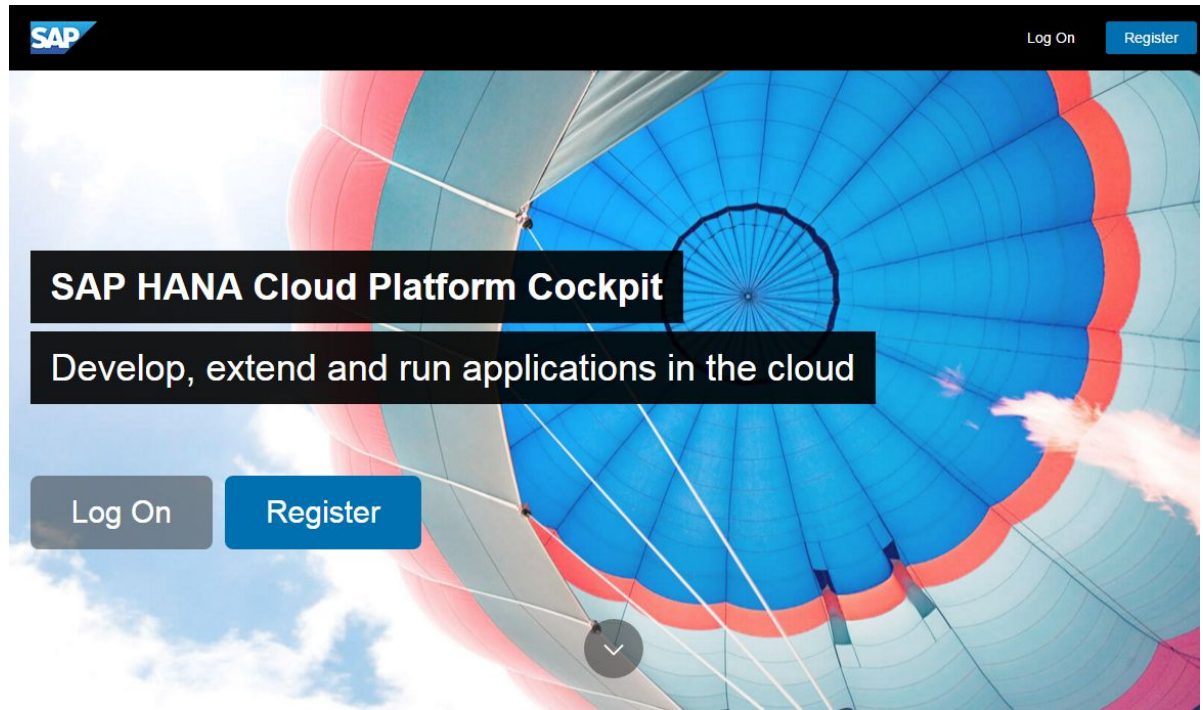
# Fun with IOT, HCP and SAP UI5

Vladimir Savchenko



# Create (or Reuse) an account on HCP Trial

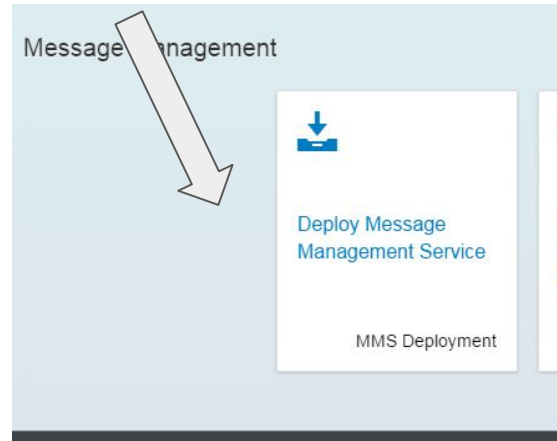
- URL: <https://account.hanatrial.ondemand.com/>



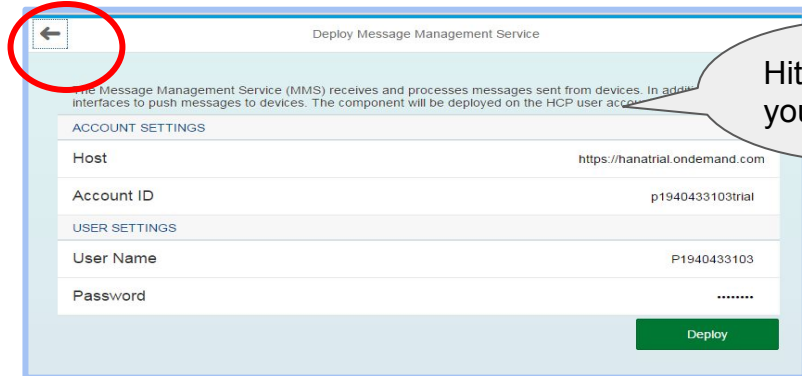
# Enable HCP Internet of Things Services



- at the end - select “Go To Service
- when it opens - click the “Deploy Message Management Service
- Then enter your account password.
- Hit Deploy
- It takes 1-2 minutes to start the app



# Create Devices and Messages in IOT Cockpit



Deploy Message Management Service

The Message Management Service (MMS) receives and processes messages sent from devices. In addition, it provides interfaces to push messages to devices. The component will be deployed on the HCP user account.

ACCOUNT SETTINGS

Host `https://hanatrial.ondemand.com`

Account ID `p1940433103trial`

USER SETTINGS

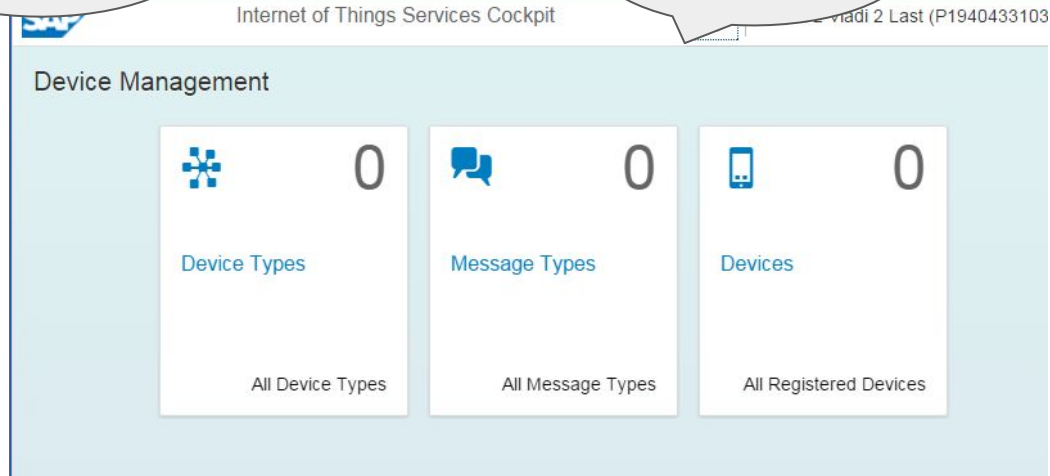
User Name `P1940433103`

Password `*****`

Deploy

Hit Back if  
you are here

You have to  
arrive here



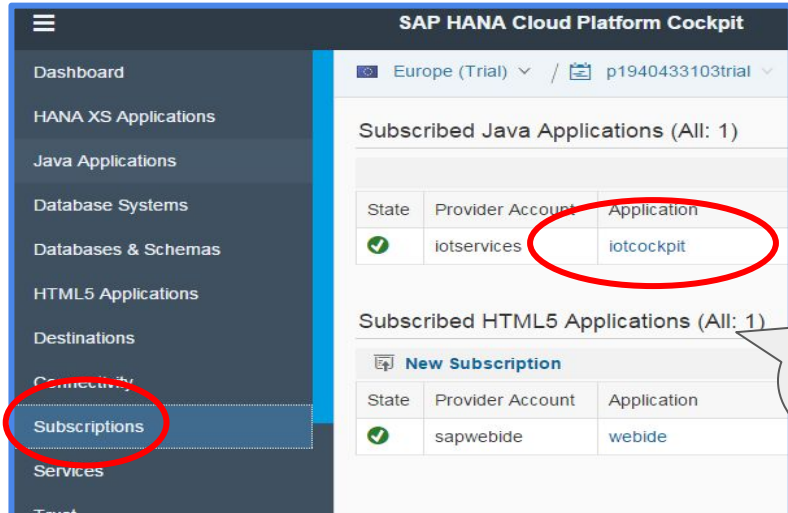
Internet of Things Services Cockpit

Device Management

Device Types 0  
All Device Types

Message Types 0  
All Message Types

Devices 0  
All Registered Devices



SAP HANA Cloud Platform Cockpit

Europe (Trial) / p1940433103trial

Subscribed Java Applications (All: 1)

State	Provider Account	Application
✓	iotservices	iotcockpit

Subscribed HTML5 Applications (All: 1)

New Subscription

State	Provider Account	Application
✓	sapwebide	webide

If not:  
Go To Subscriptions  
→ iotcockpit and the  
open the Application  
URL

# Create new Device Type

The screenshot shows the 'New Device Type' form in the SAP IoT Services Cockpit. The form is divided into two main sections: 'Information' and 'Further Details'. The 'Information' section contains a required field '\*Name:' with the value 'dkomThingDeviceType'. The 'Further Details' section contains a field for 'Enter a URL to call up further details'. The form is displayed on a mobile device interface, with a search bar and a 'No data' message on the left. At the bottom, there is a navigation bar with a plus icon, a trash icon, a warning icon with the number 1, and 'Cancel' and 'Create' buttons. A red box highlights the 'Create' button.

Device Types

Search

No data

New Device Type

Information

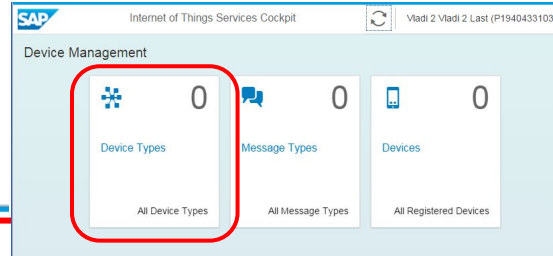
\*Name:

dkomThingDeviceType

Further Details:

Enter a URL to call up further details

Cancel Create



Note: None of the artefacts about to be created is editable for now. So in case you make a mistake, you have to start from scratch. So have patience :)

Choose a name. It is not important. E.g.:

**dkomThingDeviceType**

Then click [**Create**] button

# Add Messages

Message Types

No data

New Message Type

Information

\*Name: dkomThing\_Temp\_Outbound

\*Device Type: dkomThingDeviceType

\*Direction: From Device

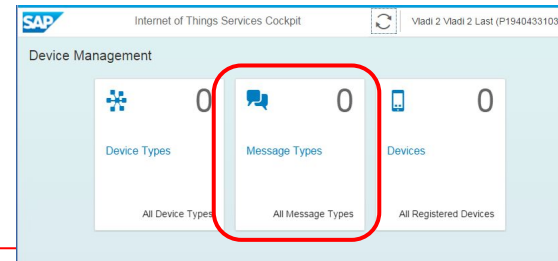
Fields

+ Add Field

Position	Name	Type
1	temp	float

Cancel



Create



- Create the 3 messages for the communication. Make sure to enter correct values for:
  - Message Type Name - is only for reference
  - DeviceType (only if you have more than one)
  - Direction
  - Field Name / Type
    - remove auto-generated fields
- Message Details
  - Temperature
    - Msg Type: **dkom\_Thing\_Temp\_Out**
    - Direction: **From Device**
    - Field Name: **temp**
    - Field Type: **float**
  - Button
    - Msg Name: **dkom\_Thing\_Button\_Out**
    - Direction: **From Device**
    - Field Name: **button**
    - Field Type: **String**
  - Led Color
    - Msg Name: **dkom\_Thing\_LED\_In**
    - Direction: **To Device**
    - Field Name: **color**
    - Field Type: **String**

# Messages Result

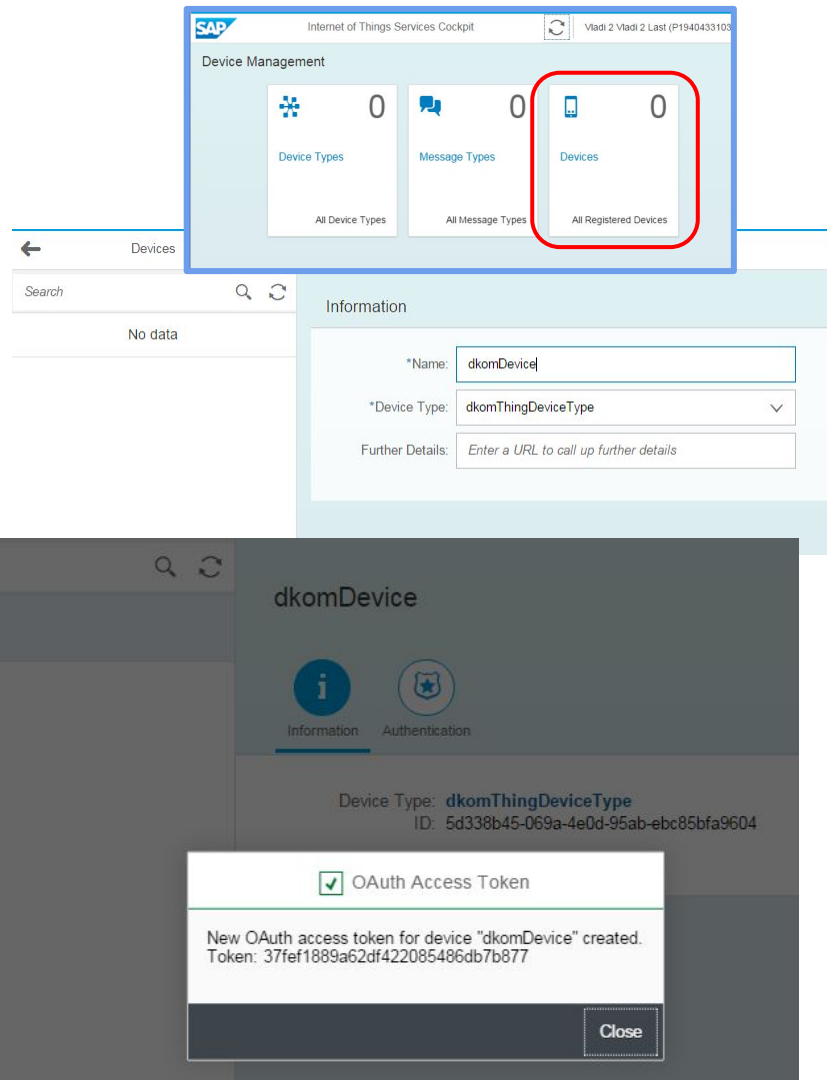
- This is how it should look like at the end.
- Note that each Message has an ID.
- Copy them down somewhere, as you will need them in 10 minutes
  - Button: xxxx
  - Led: yyyy
  - Temp: zzzz

← Message Types	Message Type
<div>Search 🔍 ↺</div> <div>dkomThing_Button_Out</div> <div>dkomThing_LED_In</div> <div>dkomThing_Temp_Outbound</div>	<div>dkomThing_Button_Out</div> <div><div> Information</div><div> Fields</div></div> <div>Device Type: dkomThingDeviceType Direction: From Device ID: fa298c8241991759ef4f</div>



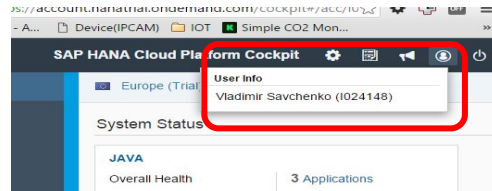
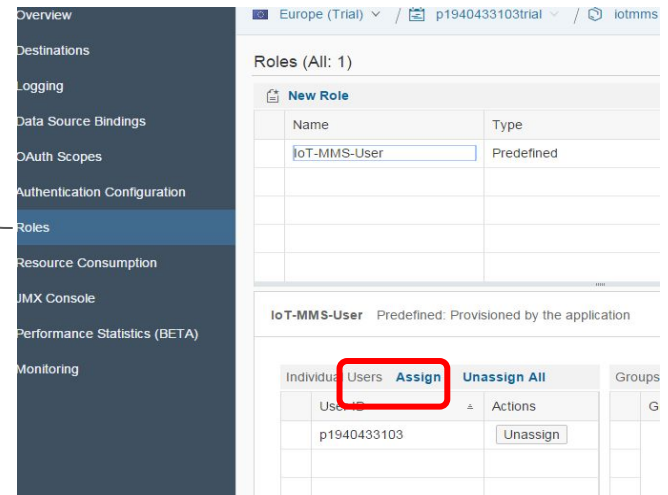
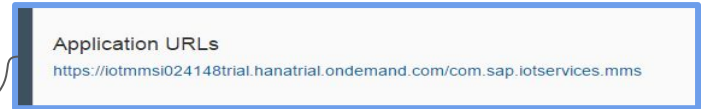
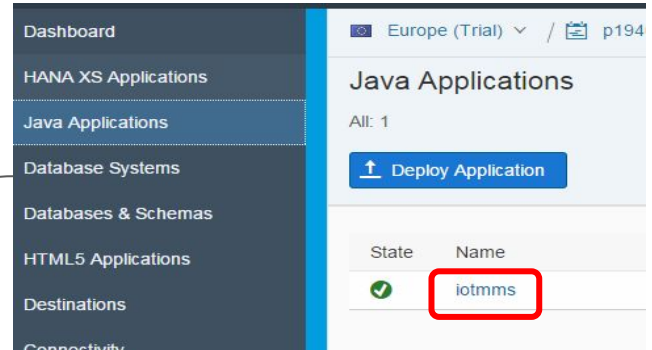
# Create The Device

- Create a new device.
- Name (not important) E.g. : **dkomDevice**
- The Device is generated. It has an ID and OAuth Token.
- The OAuth Token is important and visible only once.
- Copy both the DeviceID and the OAuthToken somewhere for reference
- 
- *In case you forget the OAuth Token - simply go to the "Authentication" Tab on this screen and you will be able to invalidate the current token and generate a new one*



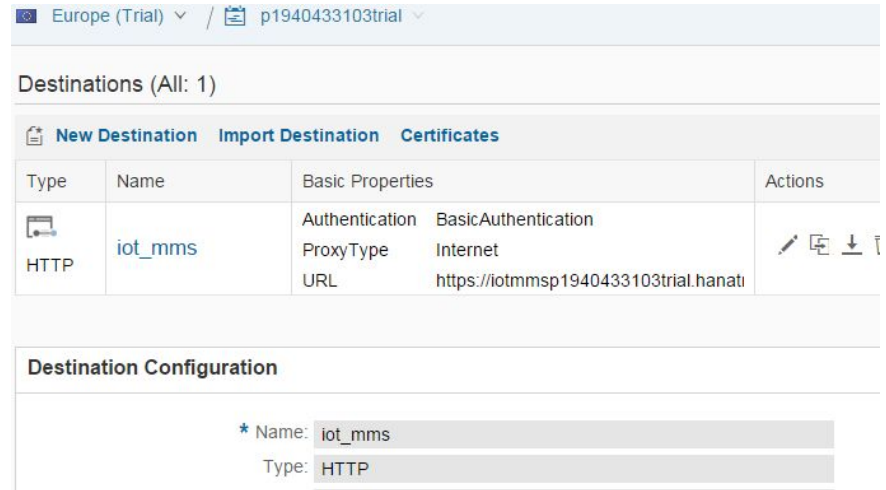
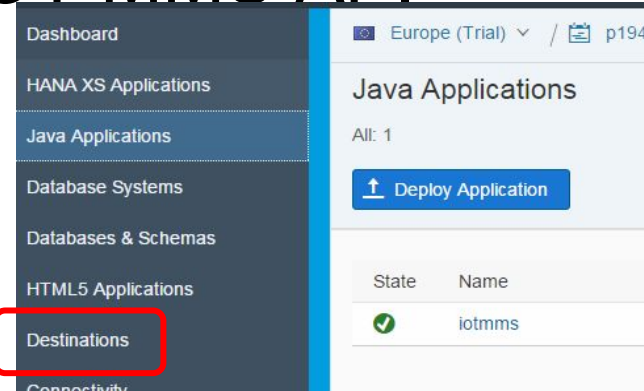
# Allow your user to call the IOT MMS API

- By default no user is assigned the role to allow calling the IOT MMS API to push messages to devices. Most of the operations are accomplished via the OAuth token. Just pushing the message to the device still requires Basic Authentication and a user.
- Open “Account Cockpit → Java Applications → iotmms”
- Copy the Application URL somewhere
- On the next screen, go to the “Roles” tab
- Click Assign
- And add your P/I/D user (See below where to take it from, especially for P users)



# Create Destination to the IOT MMS API

- To access the IOT MMS API from a HTML5 App, we need to create a destination and then add a route in the HTML5 App to this Destination.
- To create the destination: Go to Account Cockpit - > Destinations. And create a new one with following properties
- Name: `iot_mms`
- Type: HTTP
- URL: Copy the Application URL from the previous step
- Authentication: Basic Auth
- User/Pass: Your Account credentials



# Start the WebIDE

Start WebIDE from the  
Account Cockpit →  
Subscriptions Tab

**SAP HANA Cloud Platform Cockpit**

Europe (Trial) / p1940433103trial / webide

**Overview**

Roles

Logging

**Active Version**

Active Application Version: 1.24.3-24022016132009

Application URL: <https://webide-p1940433103trial.dispatcher.hanatrial.ondemand.com>

**SAP HANA Cloud Platform Cockpit**

Europe (Trial) / p1940433103trial

**Subscribed Java Applications (All: 1)**

State	Provider Account	Application	St
✓	iotervices	iotcockpit	25

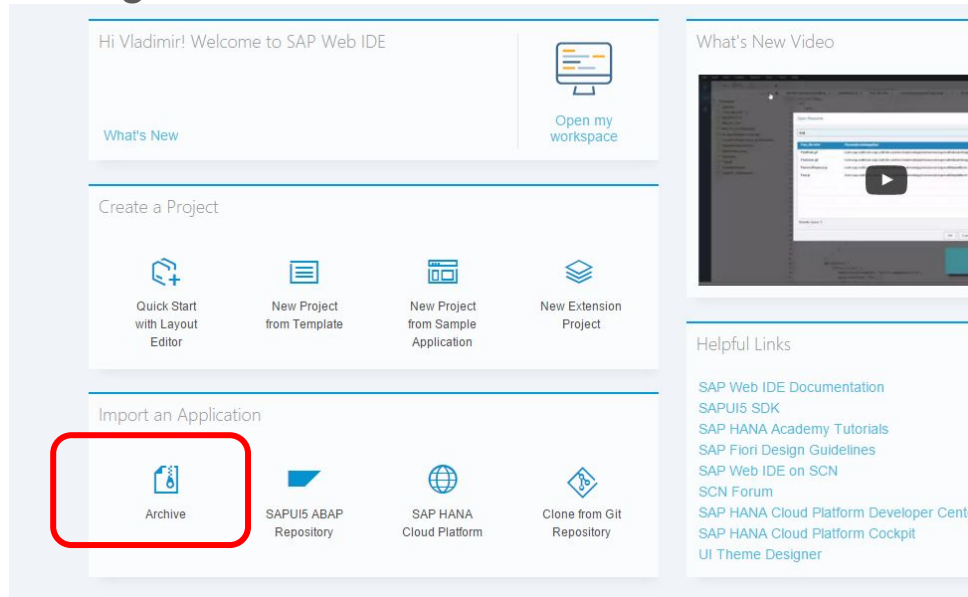
**Subscribed HTML5 Applications (All: 1)**

**New Subscription**

State	Provider Account	Application	Na
✓	sapwebide	webide	we

# Import the Sample app into Web Ide

- Download the Sample app from here [https://github.com/vlast3k/vThingCO2/raw/master/files/DKomSofia\\_IOT\\_Sample\\_App.zip](https://github.com/vlast3k/vThingCO2/raw/master/files/DKomSofia_IOT_Sample_App.zip)
- Import it from the Web IDE Home Page
- It should look like this:



# Configure the Web App

- open the file “main.js” and on the top there are 5 variables that you need to edit according to your device and message ids

```
1  var deviceId = "8f9b541b-cf97-45f4-be7e-c42521d18fb6";  
2  var token = "3abd57fe85f5353178911d8872dc153b";  
3  var msgIdLED = "759c55b23a3985597db3";  
4  var msgIdButton= "df3e2e4d1584f9ac20cd";  
5  var msgIdTemp = "8099277fb913f6159bd3";  
6
```

# Attach the dkom.Thing and connect it to Wifi

- Install the USB Drivers from here <http://www.arduinoed.eu/files/windows8/CH341SER.zip>, for Windows 7 and 8 and [http://kig.re/downloads/CH34x\\_Install.zip](http://kig.re/downloads/CH34x_Install.zip) for Mac
- Connect the VThing with the USB Cable. It should light up **Green**
- Install and Start the Chrome Configuration Utility (and chrome if you do not have it) from there: <https://chrome.google.com/webstore/detail/vair-co2-monitor-configur/kiangganloipimjbgolfjihaknkifhcn> (press “**Add to Chrome**”)
- When it loads it should automatically find the device. Now enter the Wifi connection details and press “**Set Wifi**”
- In few seconds it should show the IP in the log at the bottom

SSID: DKOM  
Pass: <<< empty >>>



The screenshot shows a web interface for configuring a device. At the top left, a green button labeled 'Connected' is visible. To its right, there are links for 'Home Page', 'Install USB Driver', and 'Like the Facebook Page, Follow the Twitter or Watch the Github Page a new version'. Below this, a section titled 'Setup: Wifi' contains a form. The form has two input fields: 'SSID' and 'Pass'. The 'SSID' field is currently empty, and the 'Pass' field is also empty. To the right of the 'Pass' field is a 'Set Wifi' button. Below these fields, there is an 'Update Interval (sec)' field with the value '20' and a 'Set' button. A note below the interval field states '(default is 120 sec)'. At the bottom of the interface, a 'Reporting' section shows 'SAP HCP Internet of Things'.

Connected

Home Page  
Install USB Driver  
Like the Facebook Page, Follow the Twitter or Watch the Github Page a new version

Setup: Wifi

SSID  Pass  Set Wifi

Update Interval (sec) 20 Set  
(default is 120 sec)

Reporting: SAP HCP Internet of Things

# Connect the dkom.Thing to HCP IOT Service

- Expand the “Reporting: SAP HCP Internet Of Things”
- Enter the values
  - SAP HCP IOT Host
    - Take **ONLY** the **HOST** part of the Application URL you already stored
- For the remaining - add the corresponding values
  - *The Message ID for the LED is not necessary*
- Press [Set]. And wait few seconds until the log says “Configuration Stored”
- → now the device will start to send Temperature updates each 15 seconds

Reporting: SAP HCP Internet of Things

**SAP HCP IOT Host:**

iotmmsi024148trial.hanatrial.ondemand.com

**Device Id:**

d912f6c1673acea6cbfe62bb52a6aa2

**Token:**

d912f6c1673acea6cbfe62bb52a6aa2

**Message ID (Temperature):**

fbf9c0ac6c3089c39e0a

**Message ID (Button):**

4a56c4c8a304895f3ed1

Set

[More Info](#)



# That's it for now !

- For digging deeper into those aspects you can join the IOT training that will be conducted on premise
- Also for programming the actual thing and not using the pre-made firmware
- and also - how to measure real ambient temperature and not the heat coming from the LED and device itself