Experience the power of IBM IBM IoT Platform

Hands-On Workshop

ATTENTION :
Certaines copies d’écrans
et certains menus ne sont pas à jour

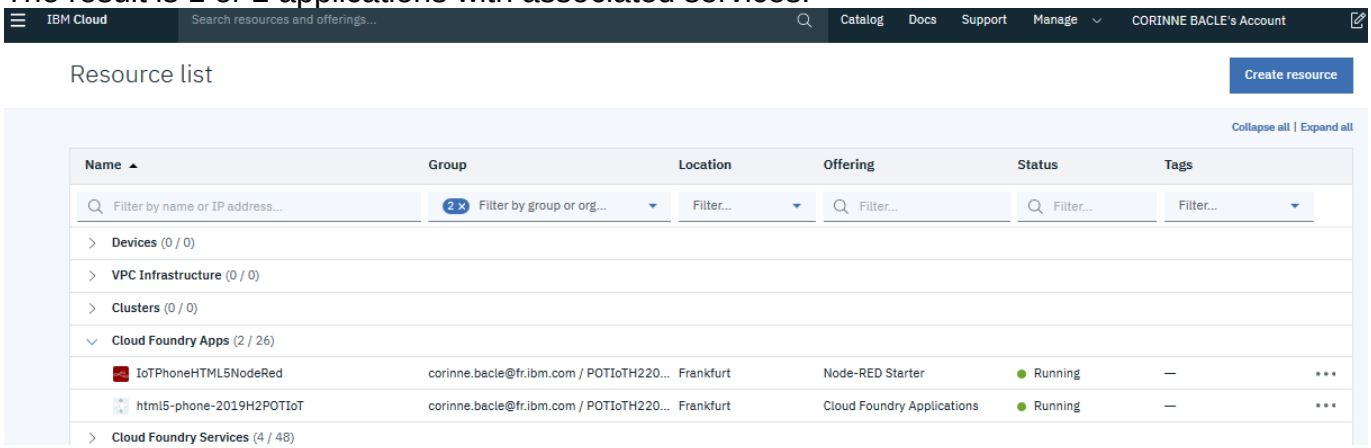
Octobre 2020 Corinne Bacle

Table of Contents

I. Prerequisites

Your IBM Cloud environment must be configured to have the IoT Phone application and the Node-RED Starter depending on your IBM Cloud account constraints.

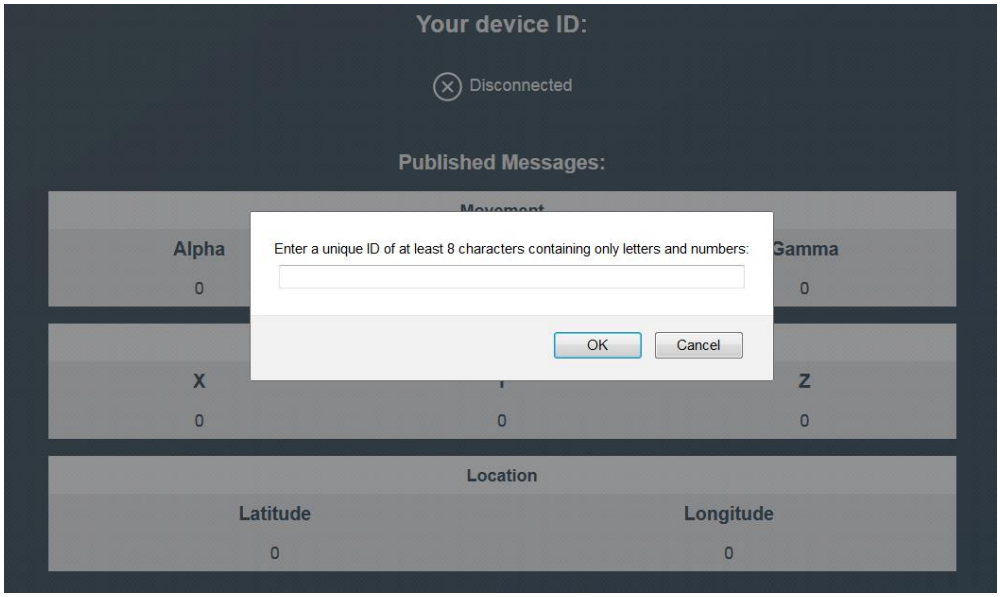
This has been done during the previous step of the POT.
The result is 1 or 2 applications with associated services:



II. Explore data coming from the IoTPhone application

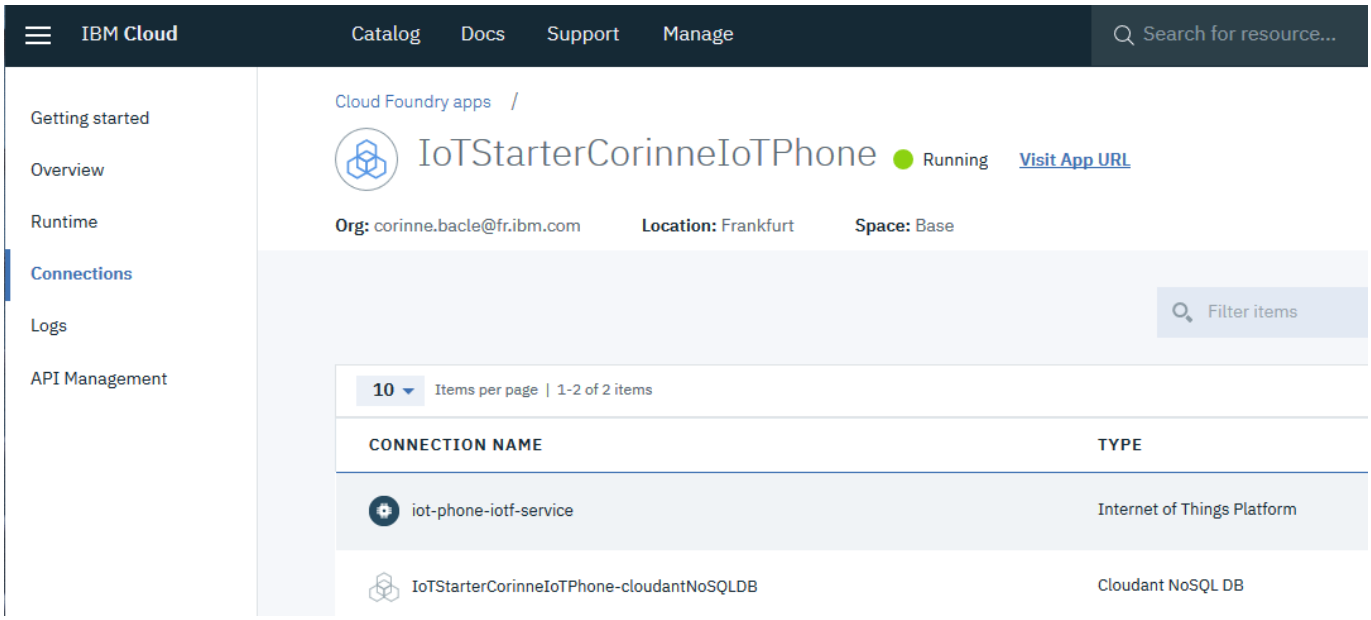
1. Go to the IoTPhone application (click on the application name) and Click on "Visit App URL" in the first architecture, use the demonstration application in the second architecture

2. Open this link on your phone and enter a unique ID:

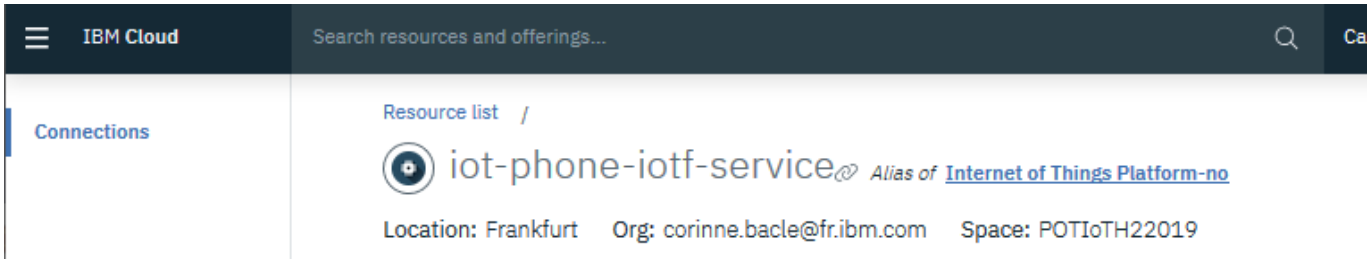


Let it run 10 second (the time to save the link) and refresh the tab to stop data transfer. The application generates a lot of messages that we don't need now.

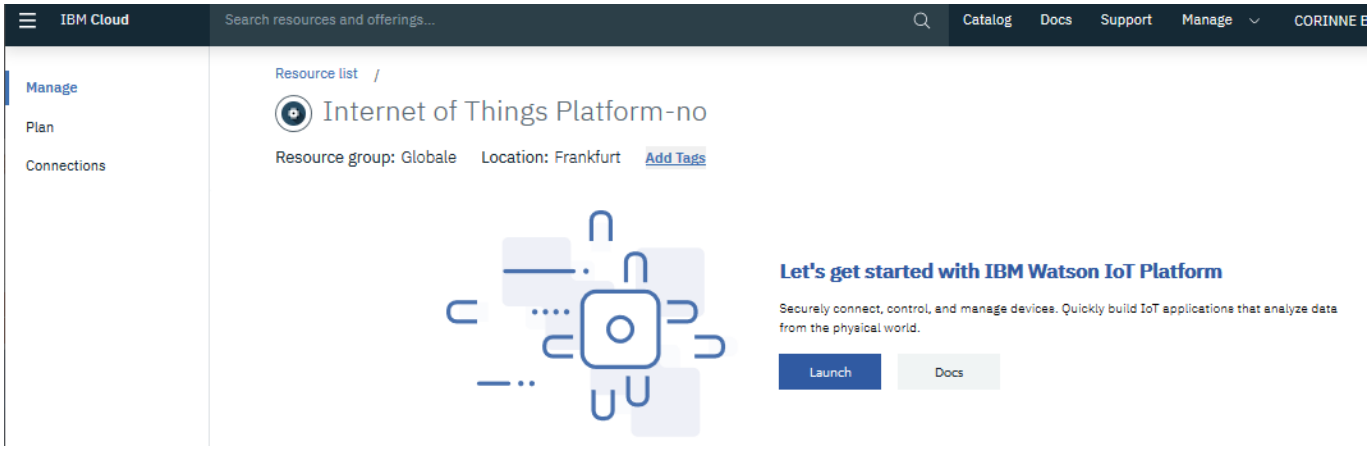
3. Go back the PC interface, in the IoT Platform service tab, go to Connections, Click on the associated Internet Of Things Platform service



4. Click on the alias of link:

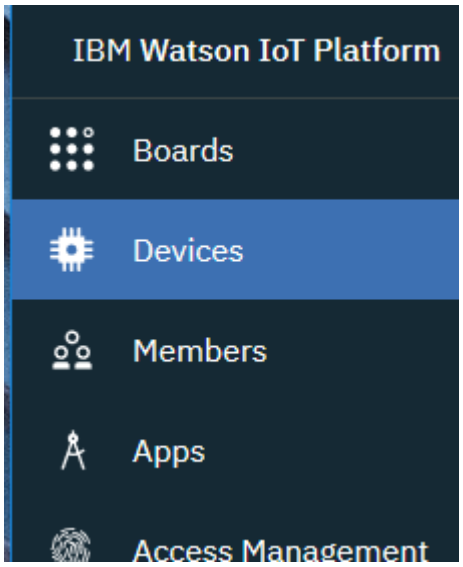


5. Select Manage then Click on “Launch”

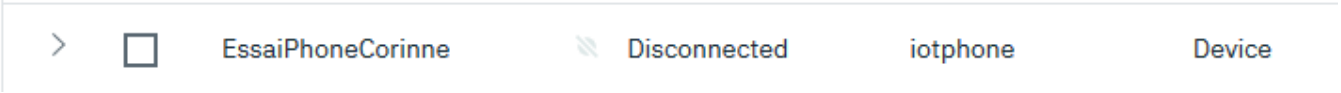


Tip : use a private web explorer windows in case of problems.

6. On the left, Click on the device icon (second icon) menu:

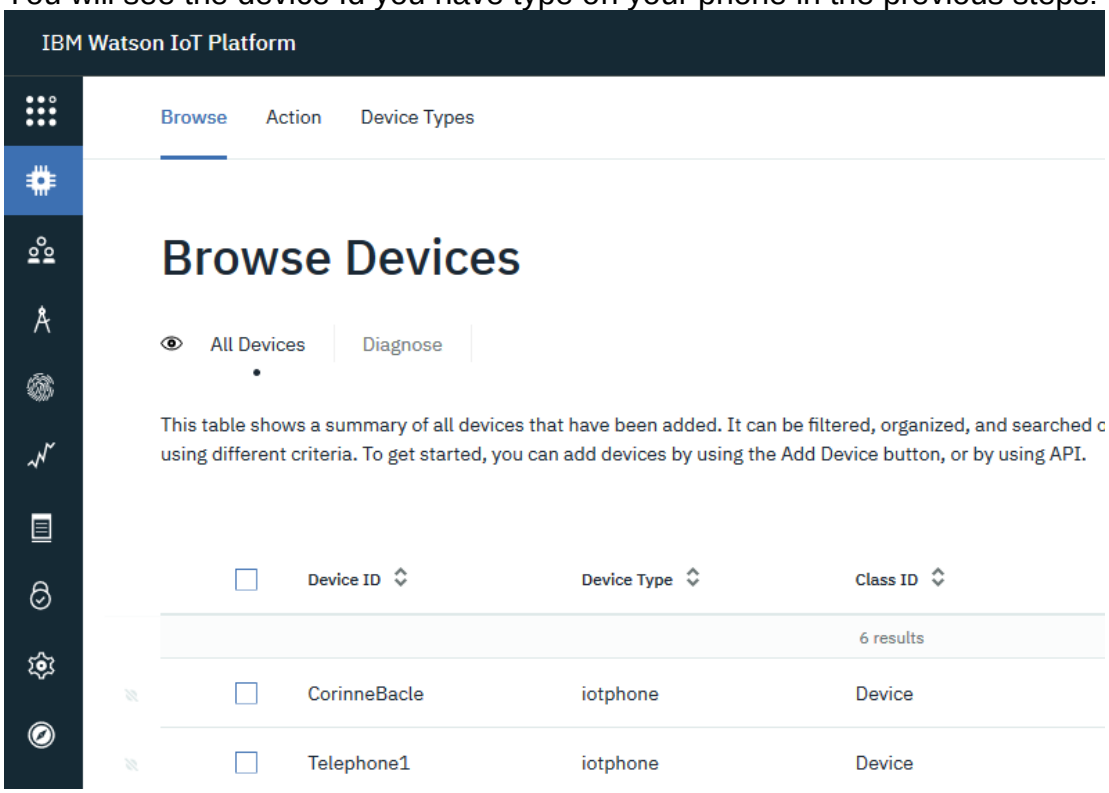


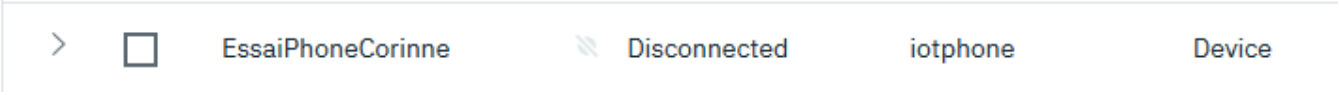
During the first test you made, a device type “iotphone” and a device were created:



7. Click to “Browse”

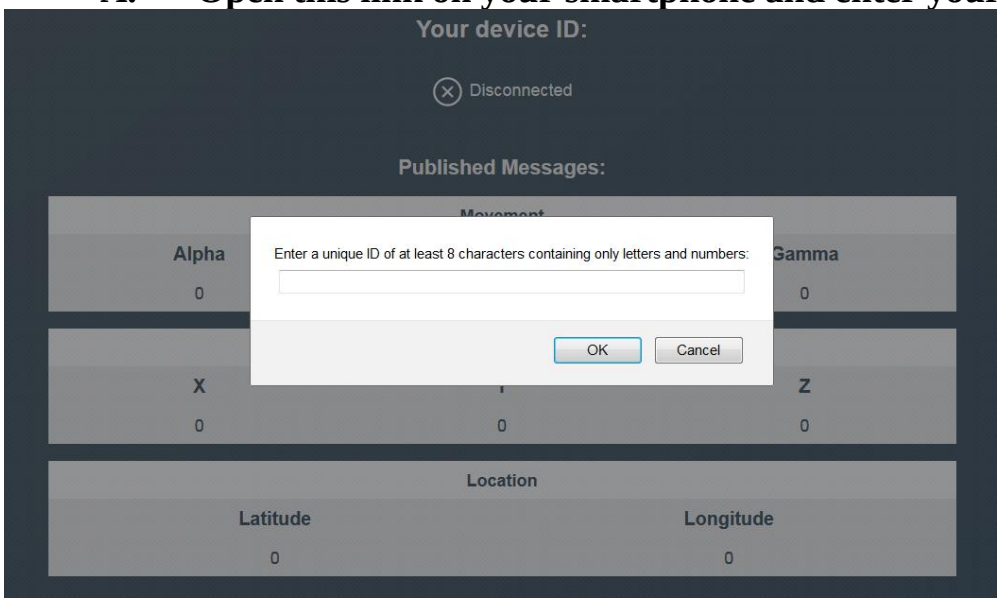
The IoTPhone application automatically create devices and save device credentials in the associated Cloudant service. This application has specific access rights to do so.
You will see the device Id you have type on your phone in the previous steps:



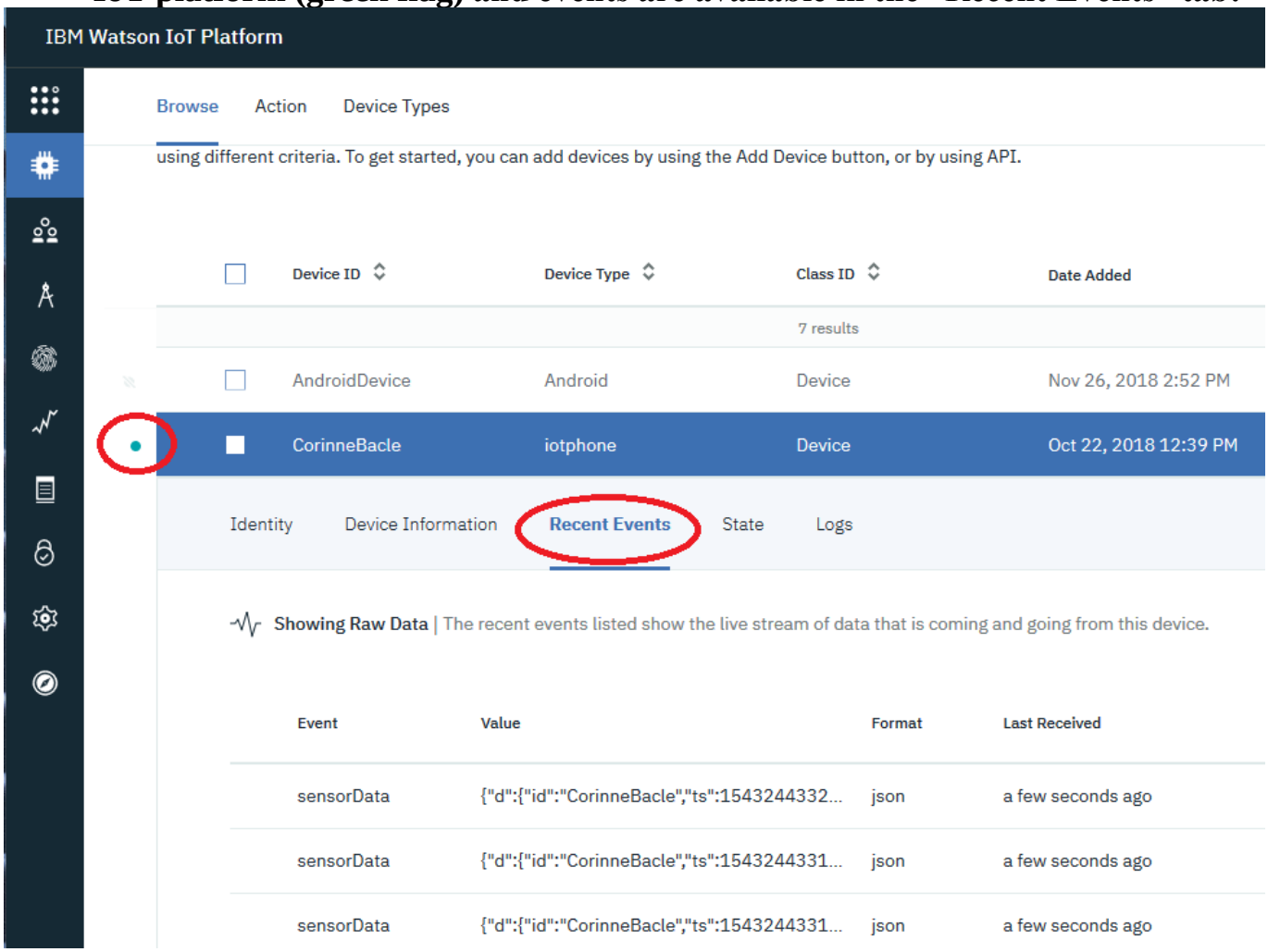


III. Use data coming from your IoT SmartPhone

A. Open this link on your smartphone and enter your unique ID:



B. Notice that your device is recognized as a connected device in the IBM IoT platform (green flag) and events are available in the "Recent Events" tab:

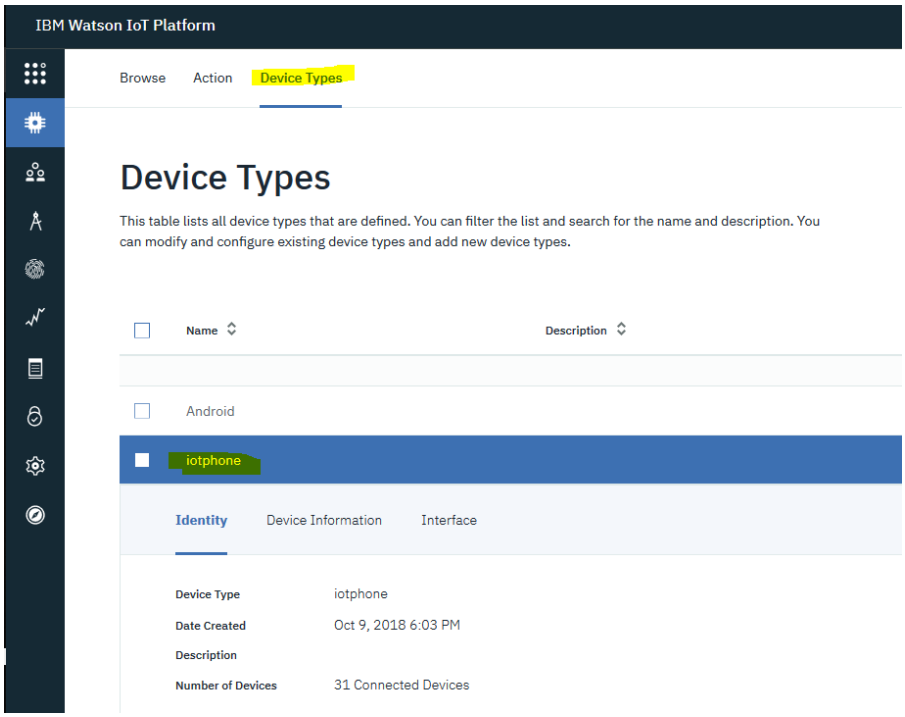


C. Create Physical Interface

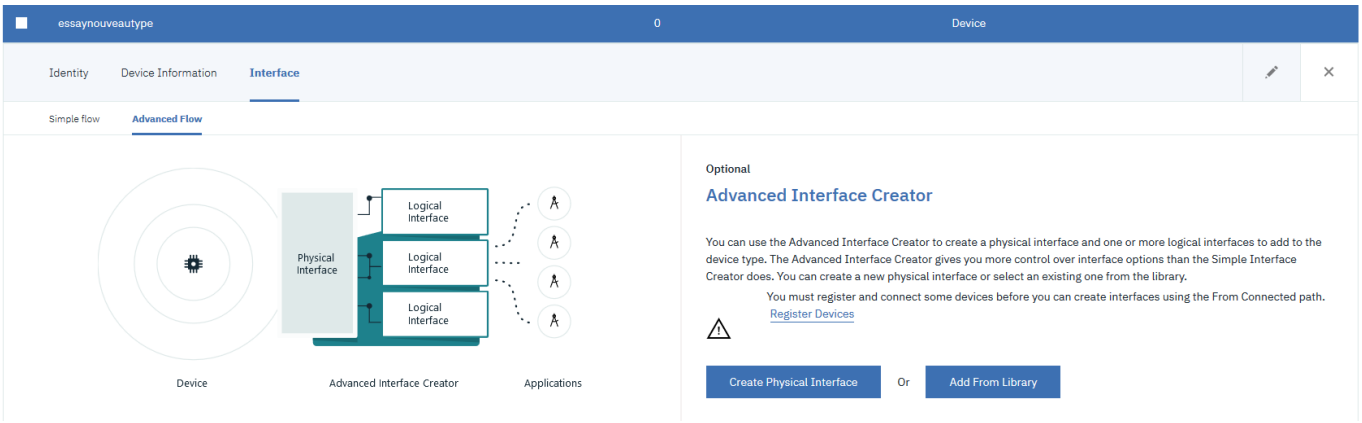
This is part of the data management feature of IoTP , see https://www.ibm.com/support/knowledgecenter/SSQP8H/iot/platform/GA_information_management/ga_im_definitions.html for details.

To use Watson™ IoT Platform features such as dashboard, you must create a physical interface to map device data to user friendly properties names, set the data units for the properties, and specify a message type to use with the schema.

- From the main navigation menu, click **Devices**.
- Click **Device Types** and select the device type that you want to create an interface for: iotphone.
- View the device type information and click **Interface**.

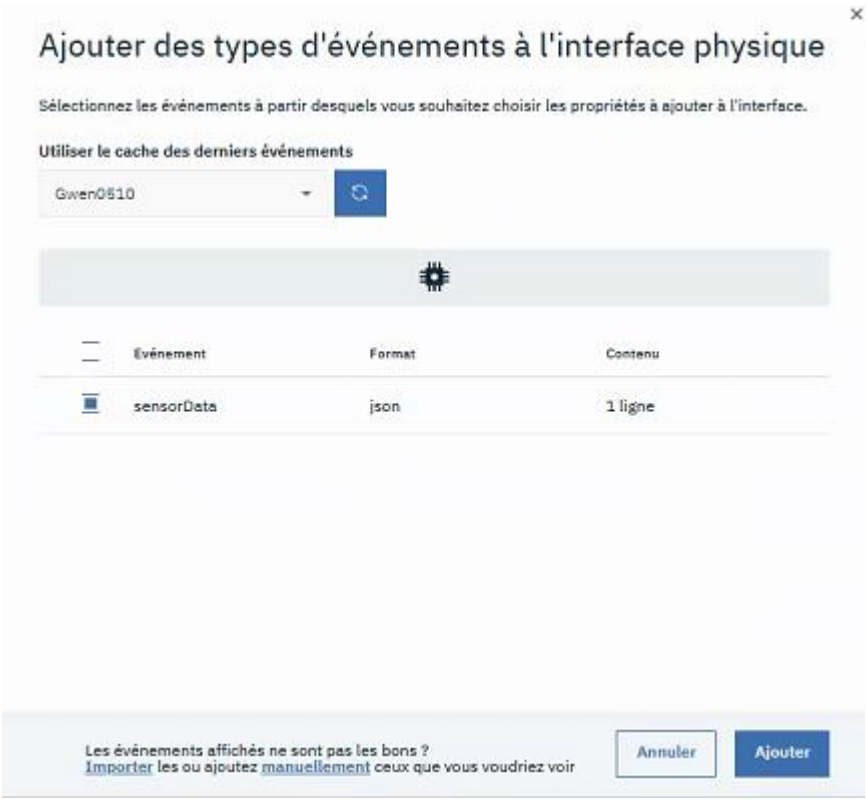


- Click **Advanced Flow**.



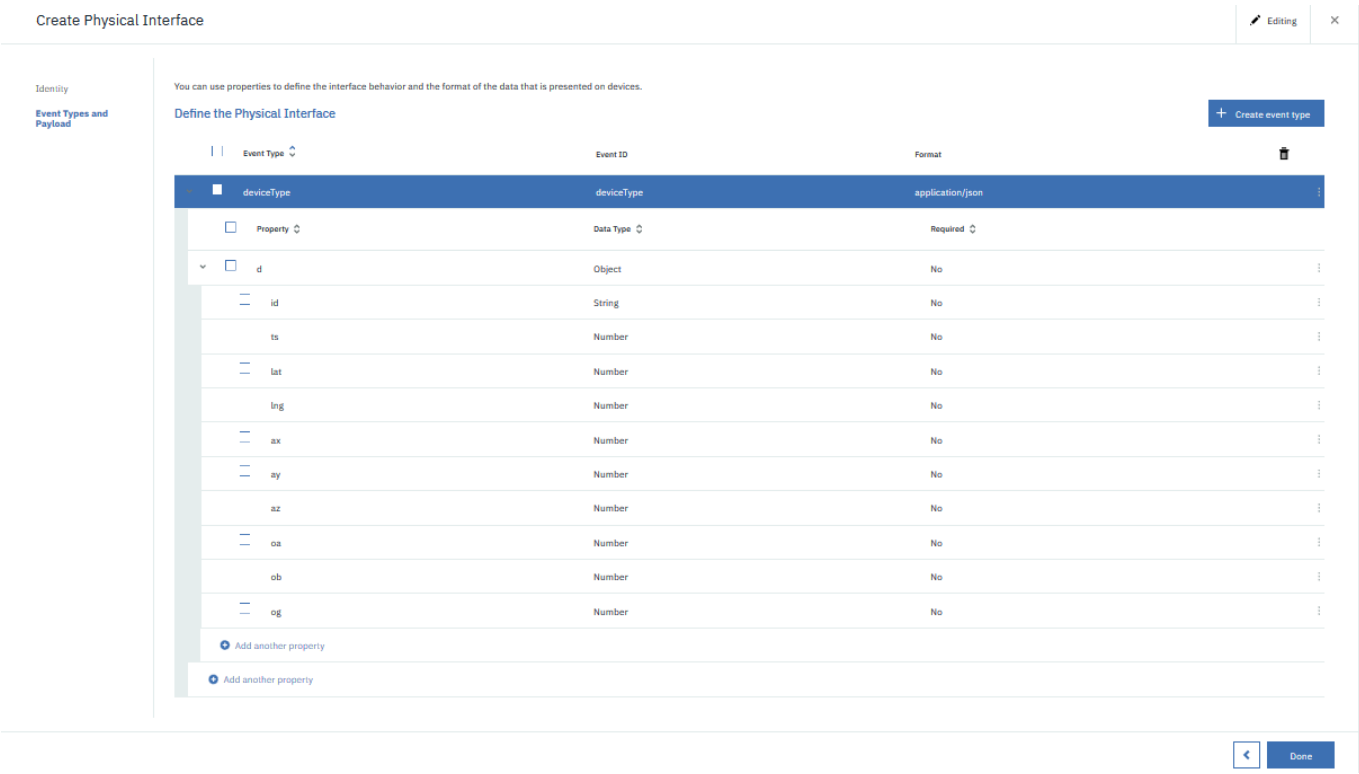
- Click **Create Physical Interface**.
- Click **Next** then **Create event Type** to start adding events and properties to the physical interface. (Click **Use last event cache** if the device is not connected).
- Select “sensorData

- Run the device interface (html5 web page) with your device Id, select deviceType then OK.
- Warning : use zom- if the bottom menu is not visible:

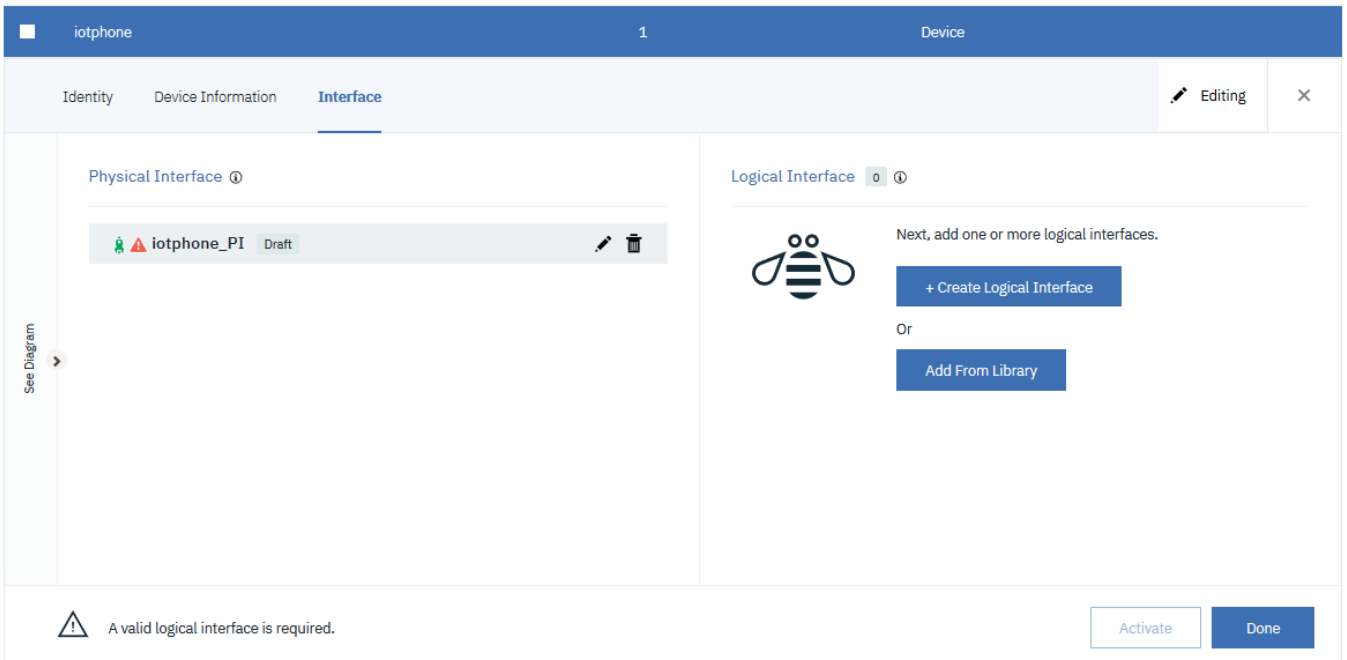


Select "**Add**"

a. All properties of the device event are automatically import. The system listens for active events for connected devices of the selected device type.
b. Add more properties if you want to test.



- Click **Done**. The physical interface is created.

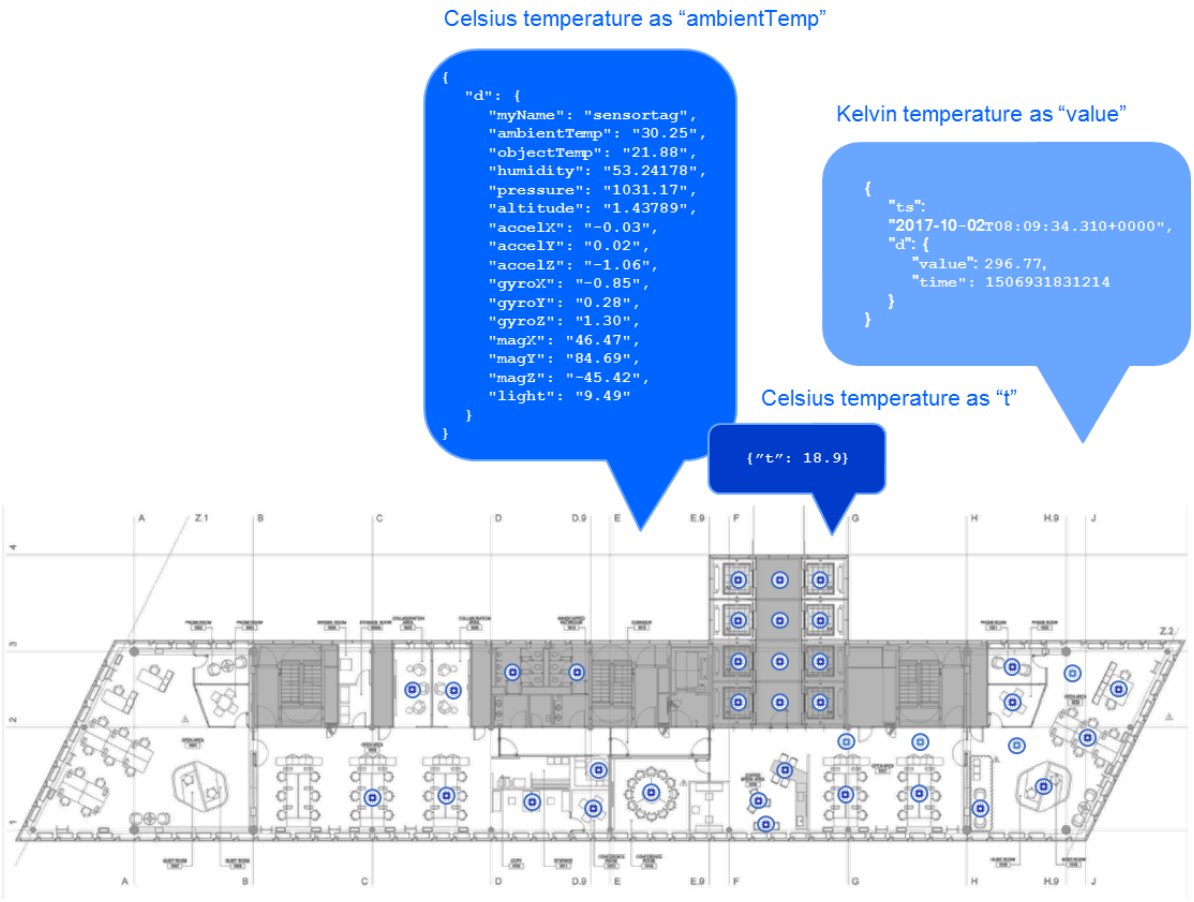


D. Create Logical Interface

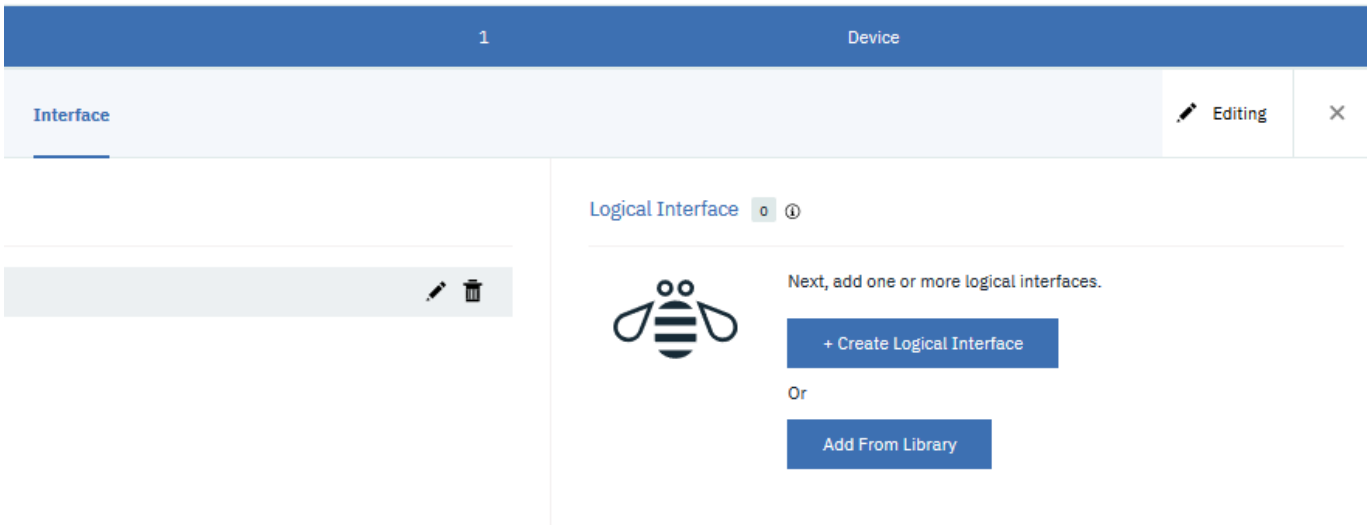
Transform & Cleanse

Use devices schemas and logical interfaces to insulate applications from variability across device types, sensor models, variants and versions

Example: Several different models and brands of temperature sensor represented by a single common logical interface



- In the device type interface, select “**create logical interface**”:

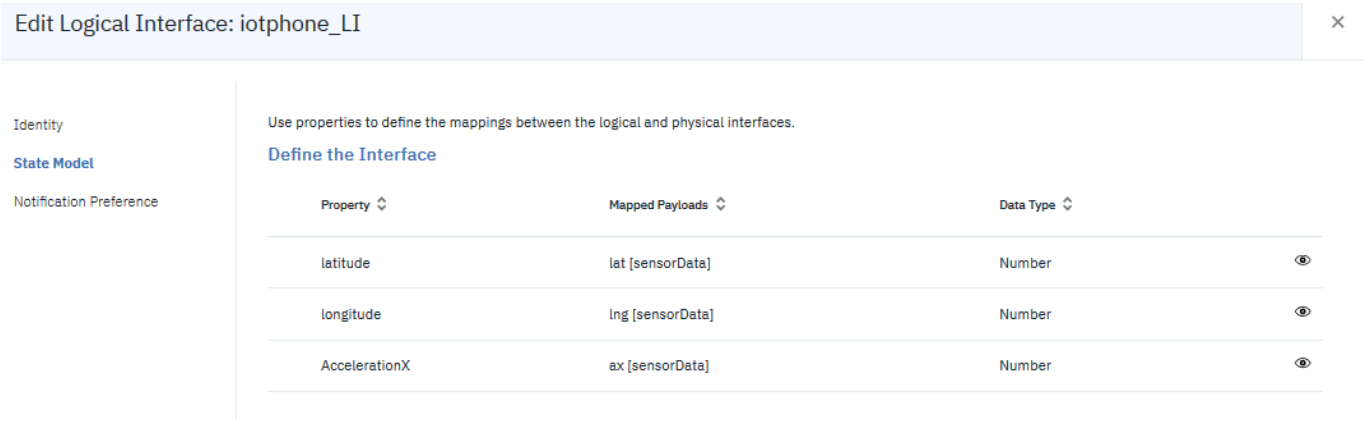


  - Select a name
  - Click **Next**
  - Click **add properties**

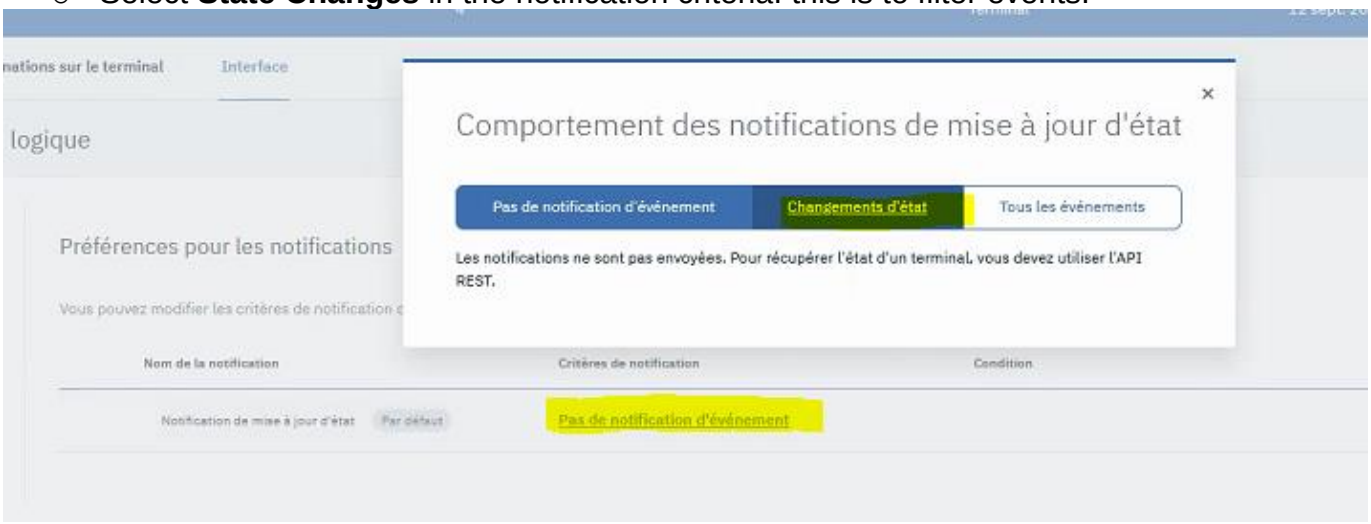
Then, using the exact mapping and names :

  - Select d => lat , name it **Latitude**
  - Select d=> lng, name it **Longitude**
  - Select d=> ax, name it **AccelerationX**
  - Select d=>ob, name it **OrientationB**

Nb : look at possible mapping calculation directly in the interface.

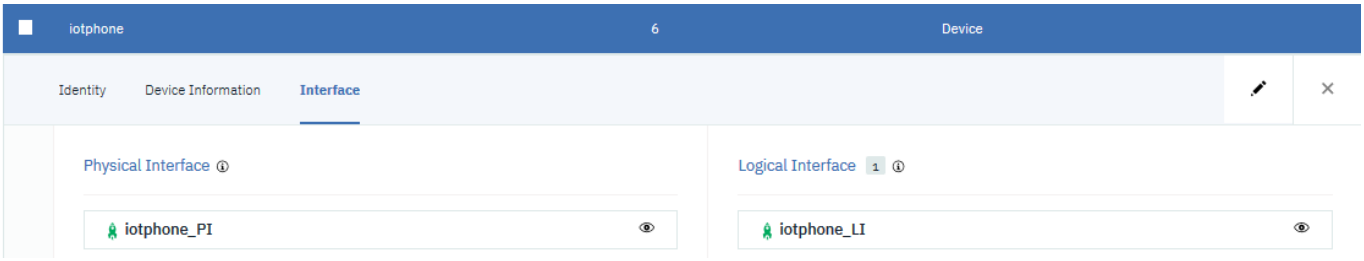


- Click **next**
- Select **State Changes** in the notification criteria: this is to filter events.

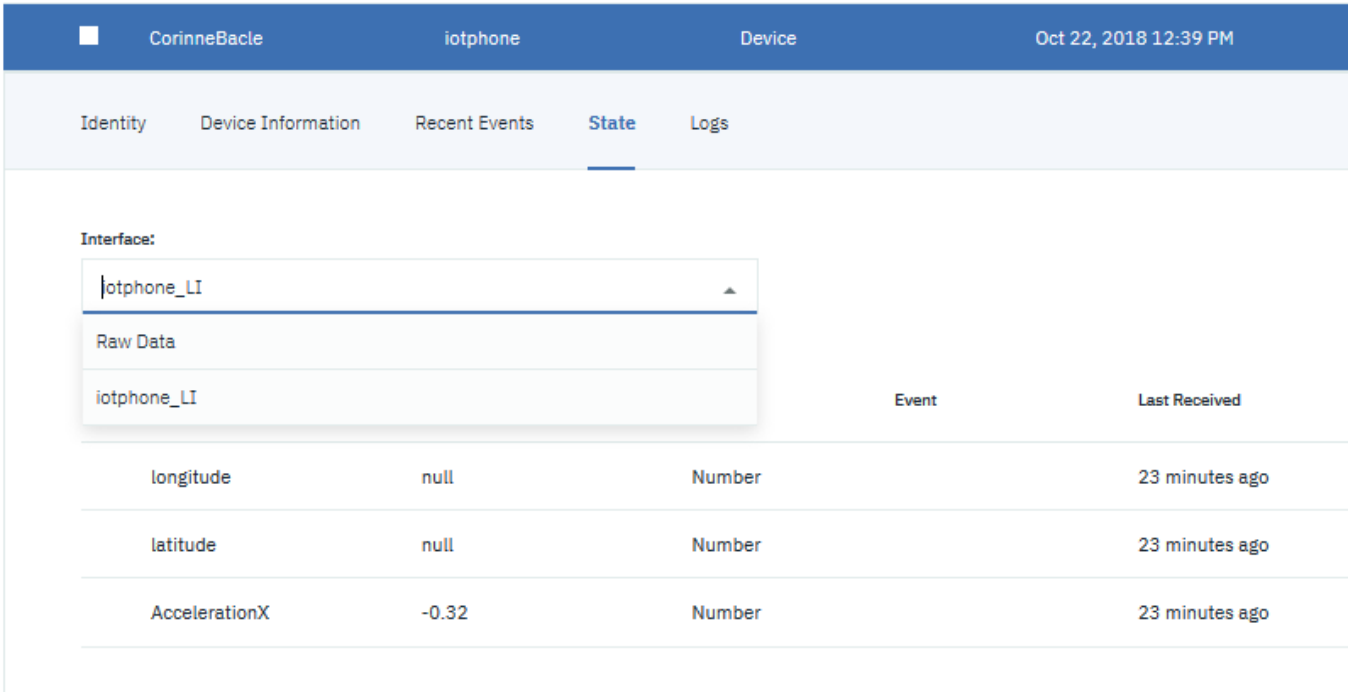


- Click **Apply** then **Done**
- Activate interfaces : Click **Activate** in the Interface, then Deploy and Done

Interfaces are now active:



As a result, you can see the "**State**" of your device, based on Raw Data or logical interfaces:

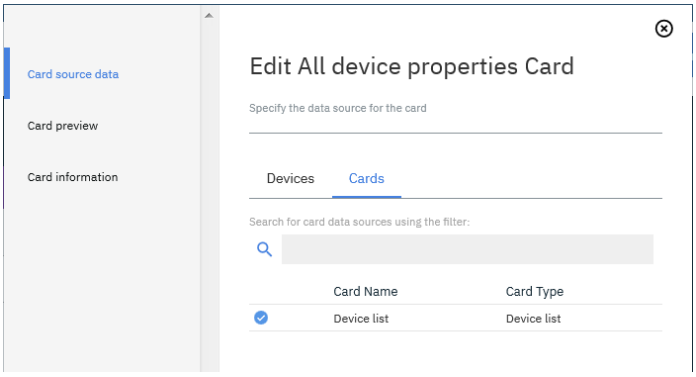


A logical interface can be associated to an API Key to filter data and to control the data format.

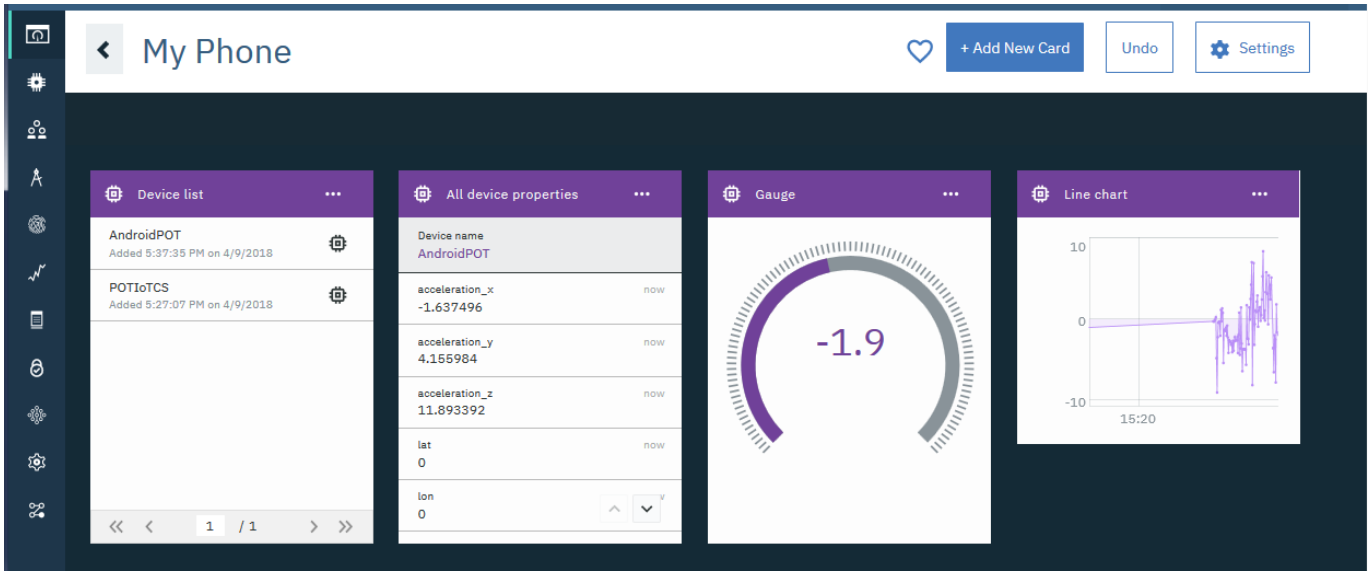
E. Visualize in IoTP Dashoard

- Open the Dashboard menu:

- Create a board to contain the cards for your devices.
    - If the All Boards, page is not already displayed, select **Boards** from the IBM IoT Platform dashboard left menu, and then click **Create New Board**.
    - **Enter a name** for the board (for example, My Phone) and click **Next**.
    - On the next page, click **Create**.
- Click the board that you just created to open it.
- Click ***Add New Card***, select ***Device List***, add the card.
- Click ***Add New Card***, select ***All Device Properties***, select Card source data : Cards/Device List, add the card.



In this configuration, to display all properties of a device, you must first select the device in the device list.
Test Gauge, Line Chart etc. The selection of event and device properties is based on the physical interface defined before.



F. Create a board to display location in a map

Create a board and cards to display device data in the IBM IoT Platform dashboard map.

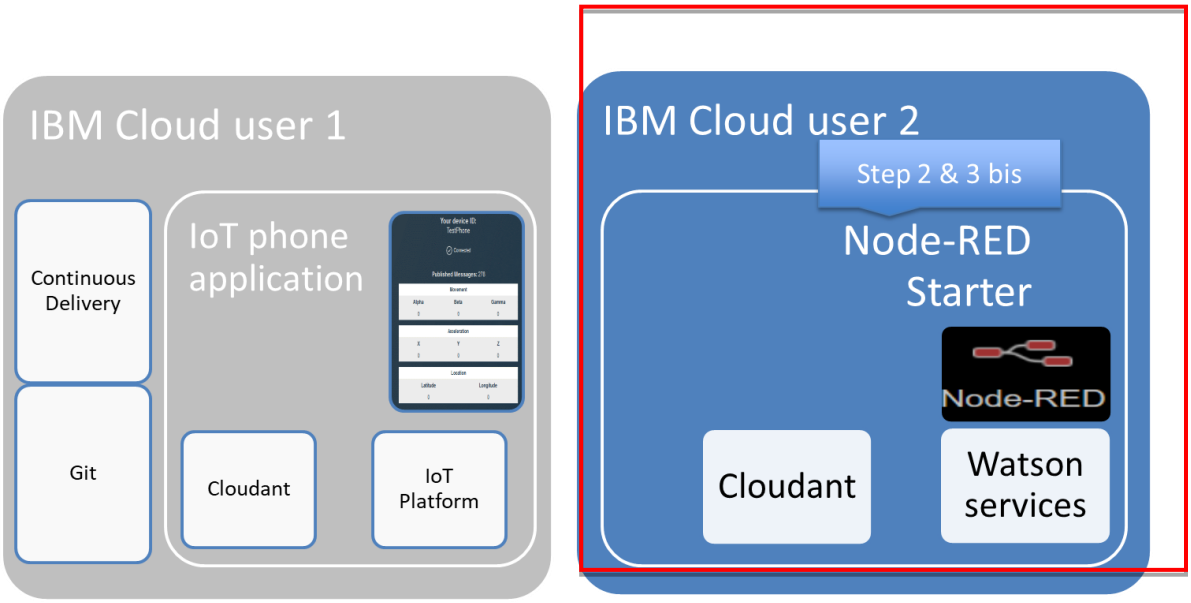
1. Go to your board
2. Click **Add New Card**, and then select the **Device list** card type, which is located in the Devices section / show more (if not done in previous steps).
3. Select your device from the list, then click **Next**.
4. Click **Add New Card**, and then select the **Device Map** card type, which is located in the Devices section / show more.
5. Select your device list from the list, then click **Next**.
6. Select Data point from your device type, for longitude and latitude
7. Click Next
8. In the Card Preview page, select **M** as the size, and click **Next**.
9. In the Card Information page, change the name of the card to **Device Map Location** and click **Submit**. The location card map appears on the dashboard and shows the live latitude and longitude of the device.

IV. Use your NodeRed application to create and generate an alert

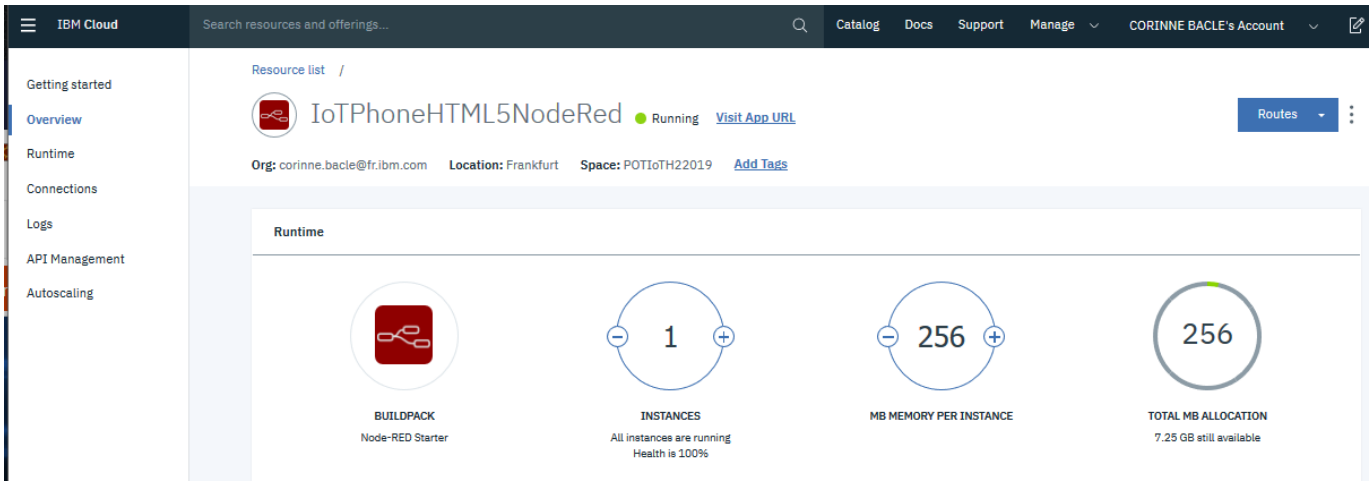
A. Visualize in Node-RED

Go back to your application « IBM Cloud User 2 »

1. Go back to your application in IBM Cloud, you are now going to collect your data from your app.



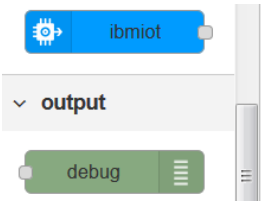
Go back to your Node-RED Starter app overview:



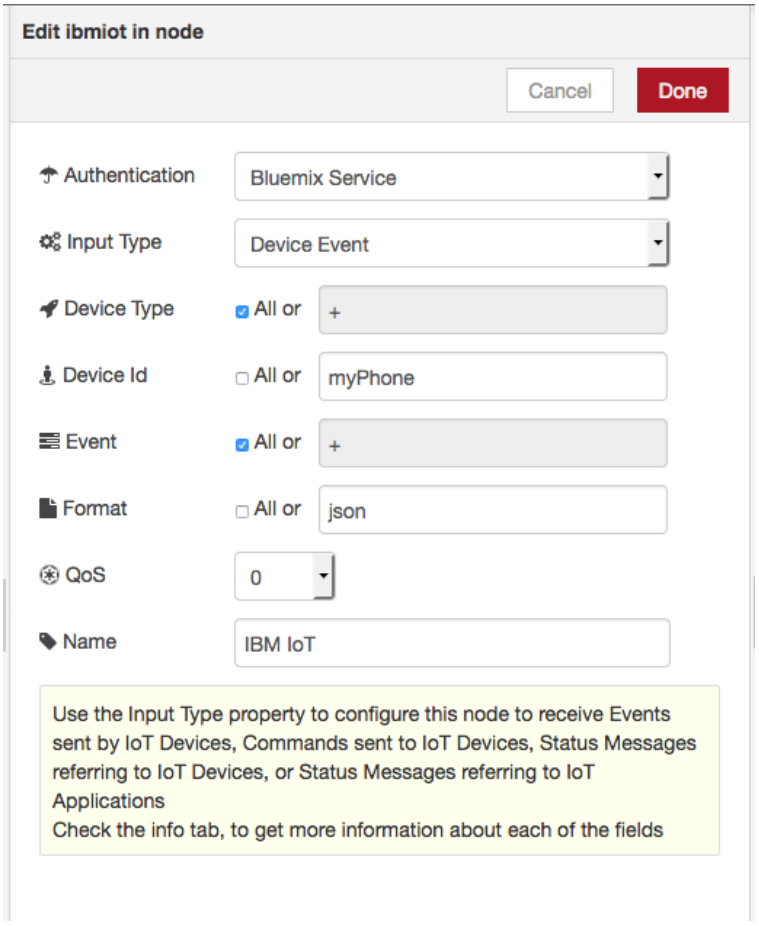
2. Click on Visit App URL to access the Node-RED web editor:

3. Click on "Go to your Node-RED flow editor"

This app you have created in IBM Cloud provides a browser-based editor (Node-RED) that makes it easy to wire together flows that can be deployed to the runtime. In the case of IoT, Node-RED is powerful to quickly test all the possibilities that IBM Cloud offers with different kind of services. Your Node-RED app has a public URL like any web app but there is a way to add a user/password to secure your workspace. Directives are in the annex.

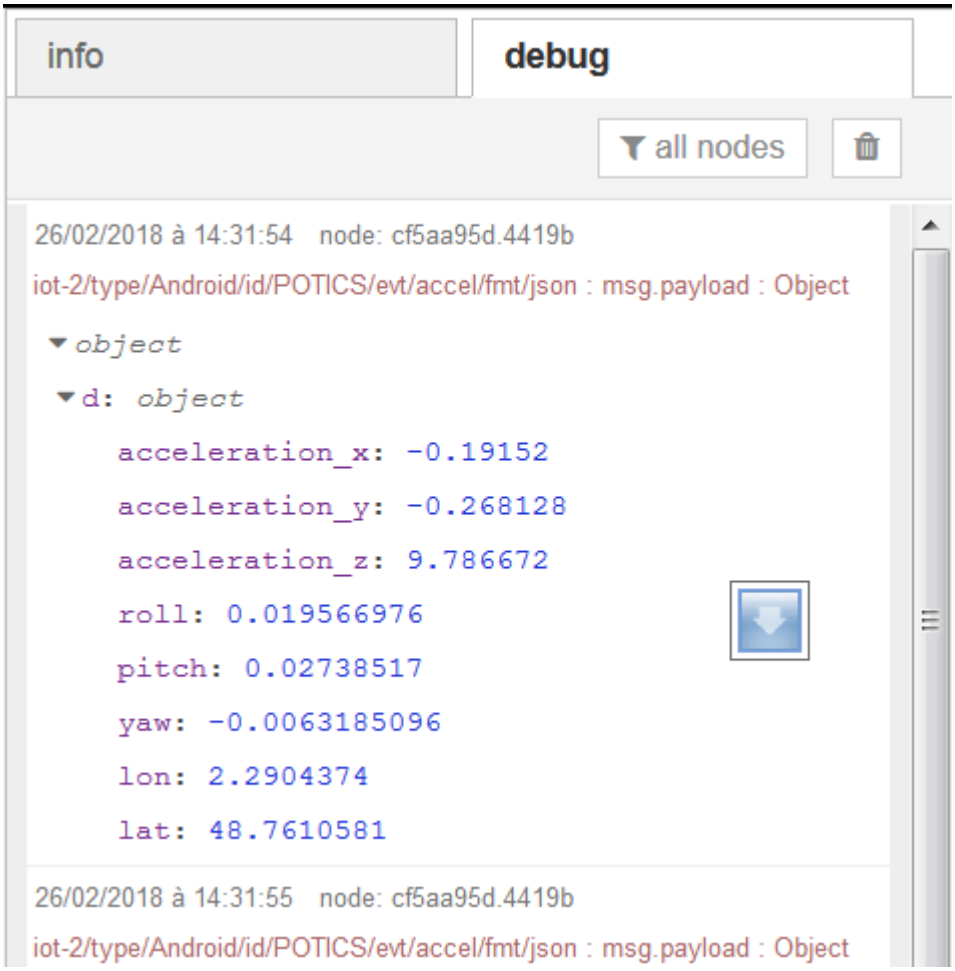


4. Then, double click on the IBM IoT node and fill your device id:



5. Click “Done” and deploy your app by clicking up and right on the “Deploy” button (The button turns grey which means the flow is deployed)

6. Then select the debug tab to visualize your data:



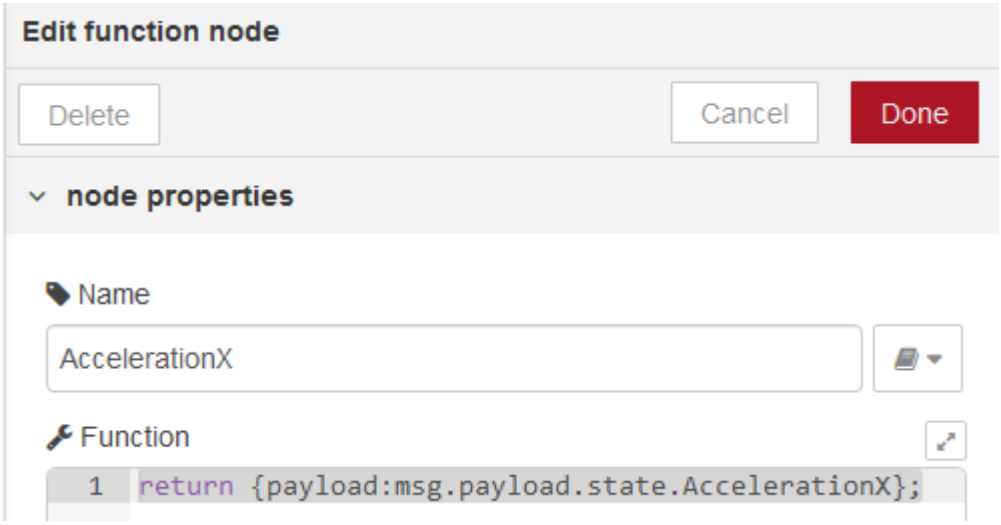
Node IBM IoT

You are receiving data via MQTT protocol in JSON format.

- Select “Device State Event” in the input type of the IBM IoT Node
- Deploy
- Compare the result in the debut tab
- The “Device State Event” use the logical interface format.
- Close

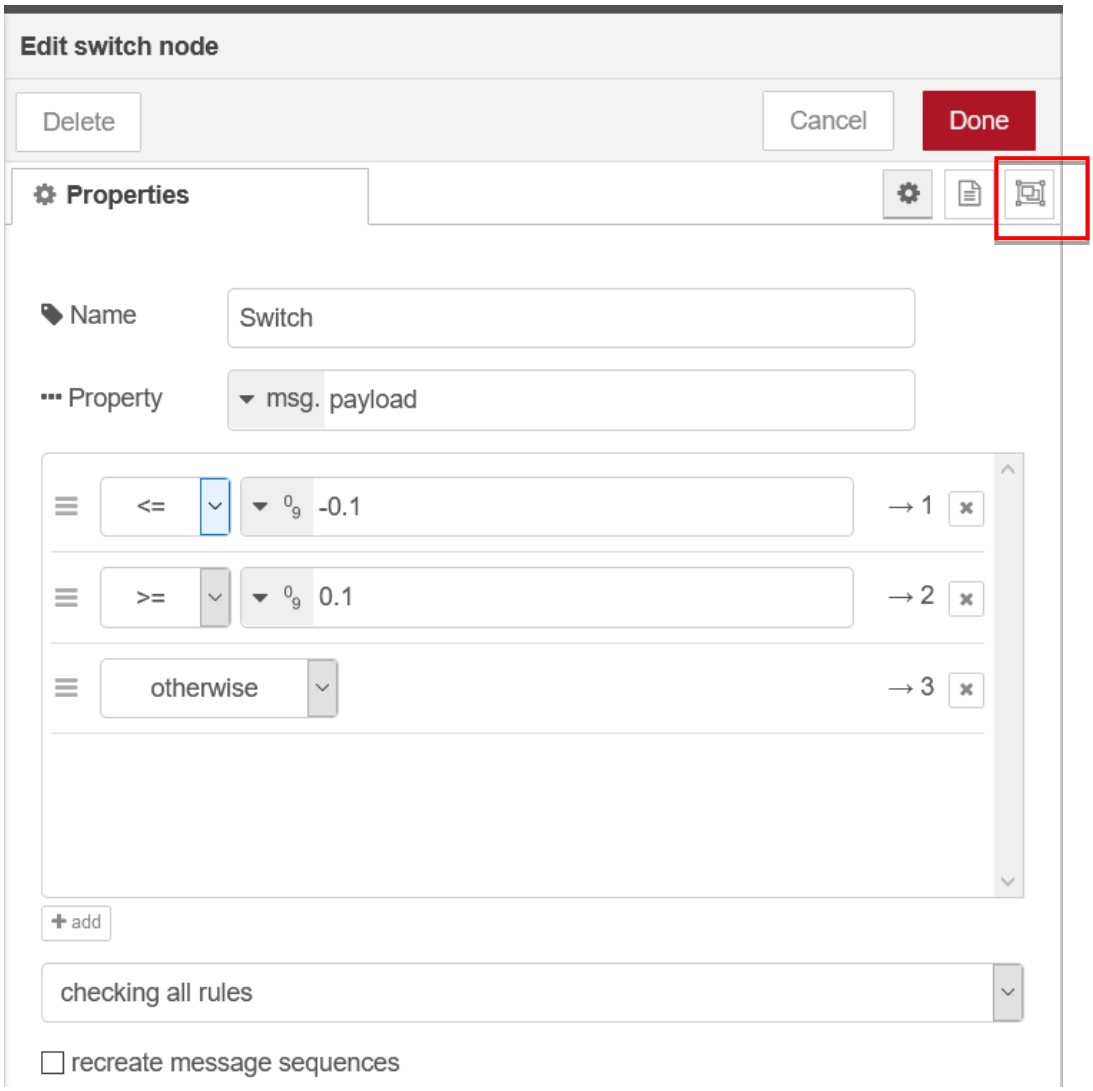
Add a New Node to Get the Acceleration Value

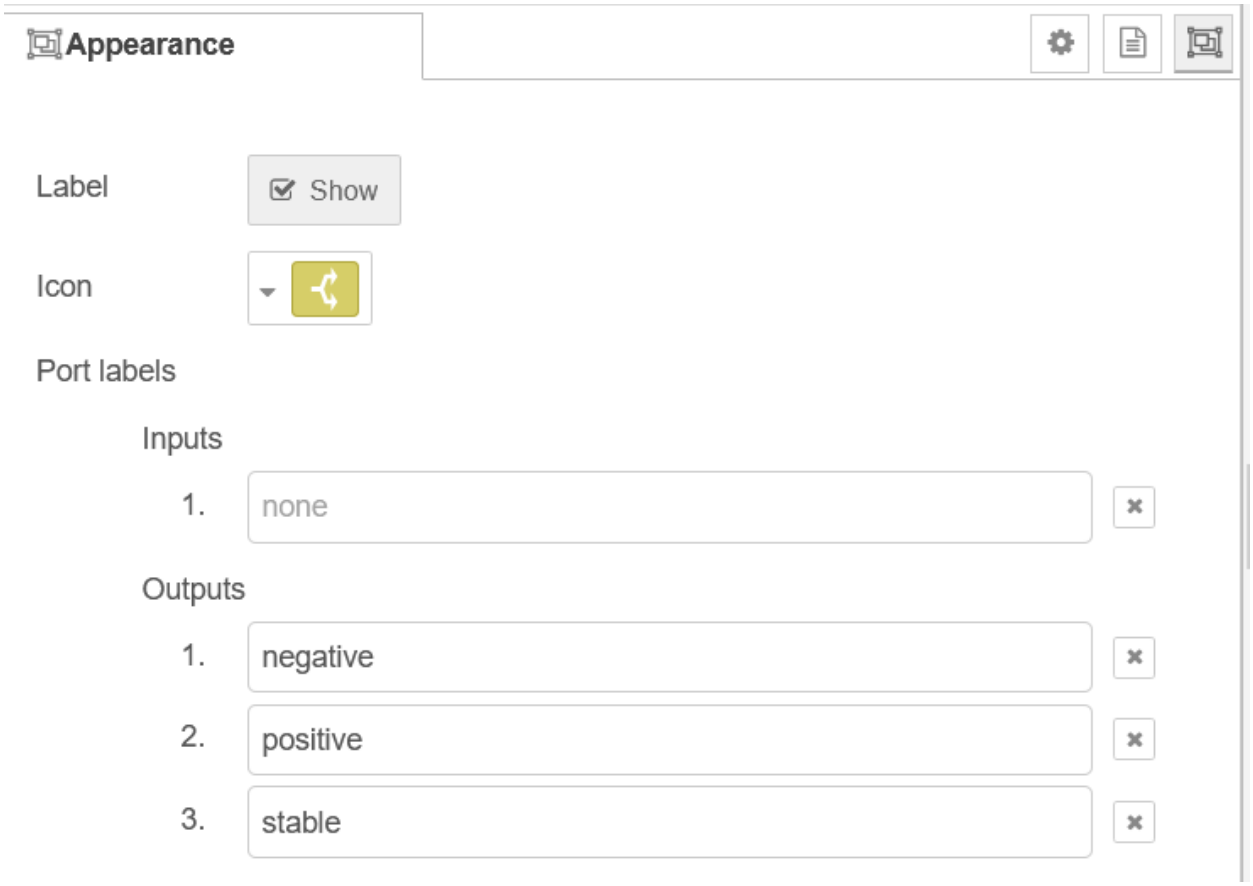
- Add a function node with the content :
  return {payload:msg.payload.state.AccelerationX};
  (use the format you have defined in the logical interface)



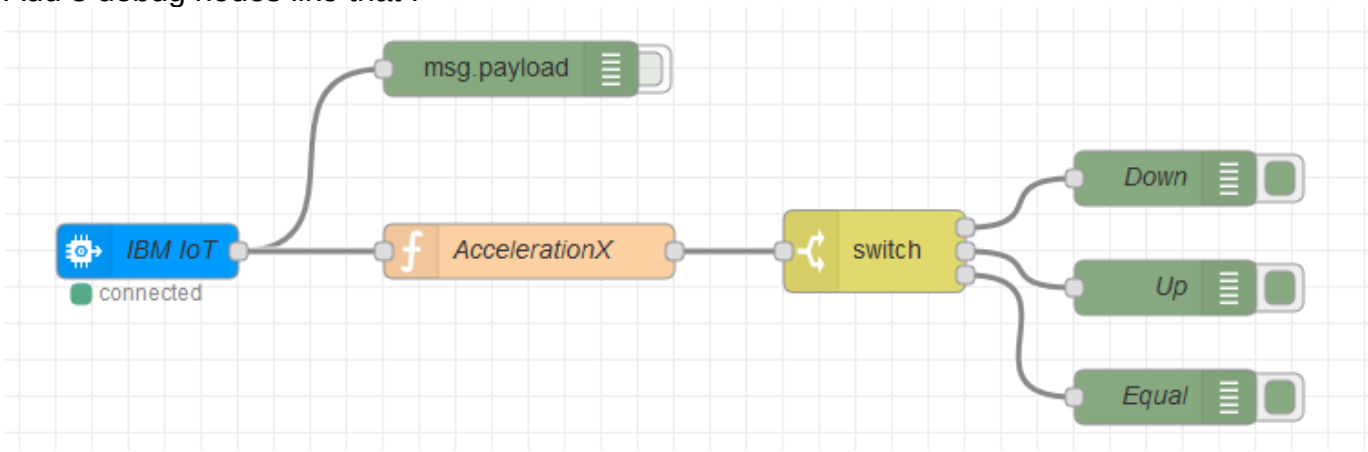
Add a New Switch Node 

- Add a switch function and see screen capture to configure it.don't forget to use numbers.





- Add 3 debug nodes like that :



- Click Deploy
- Run your app
- See the result in the debug tab

What you have done so far:

- You have created an app in IBM Cloud using the IoT Boilerplate
- You have registered our device in IoT Watson Platform org
- You have connected your smartphone to your IoT Watson platform org
- You have checked that we are receiving the data from the smartphone in your IBM Cloud app

What you will do next :

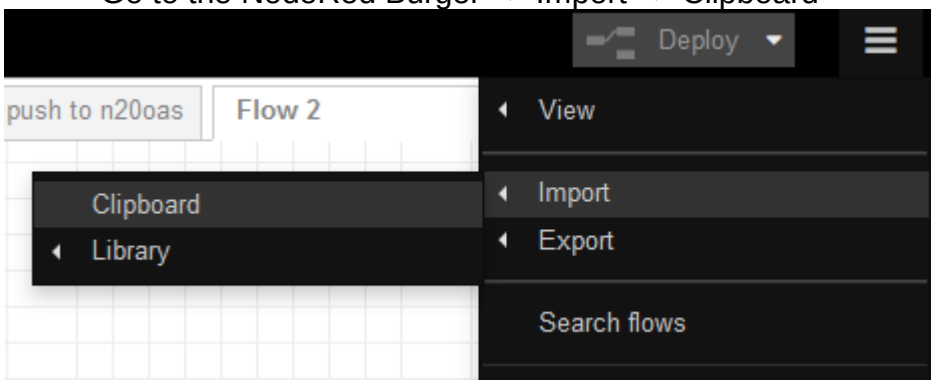
- Create an alert
- Send a text to create a vocal alert
- Store the data in a Cloudant DB

B. Smartphone interaction with an application: Create an alert

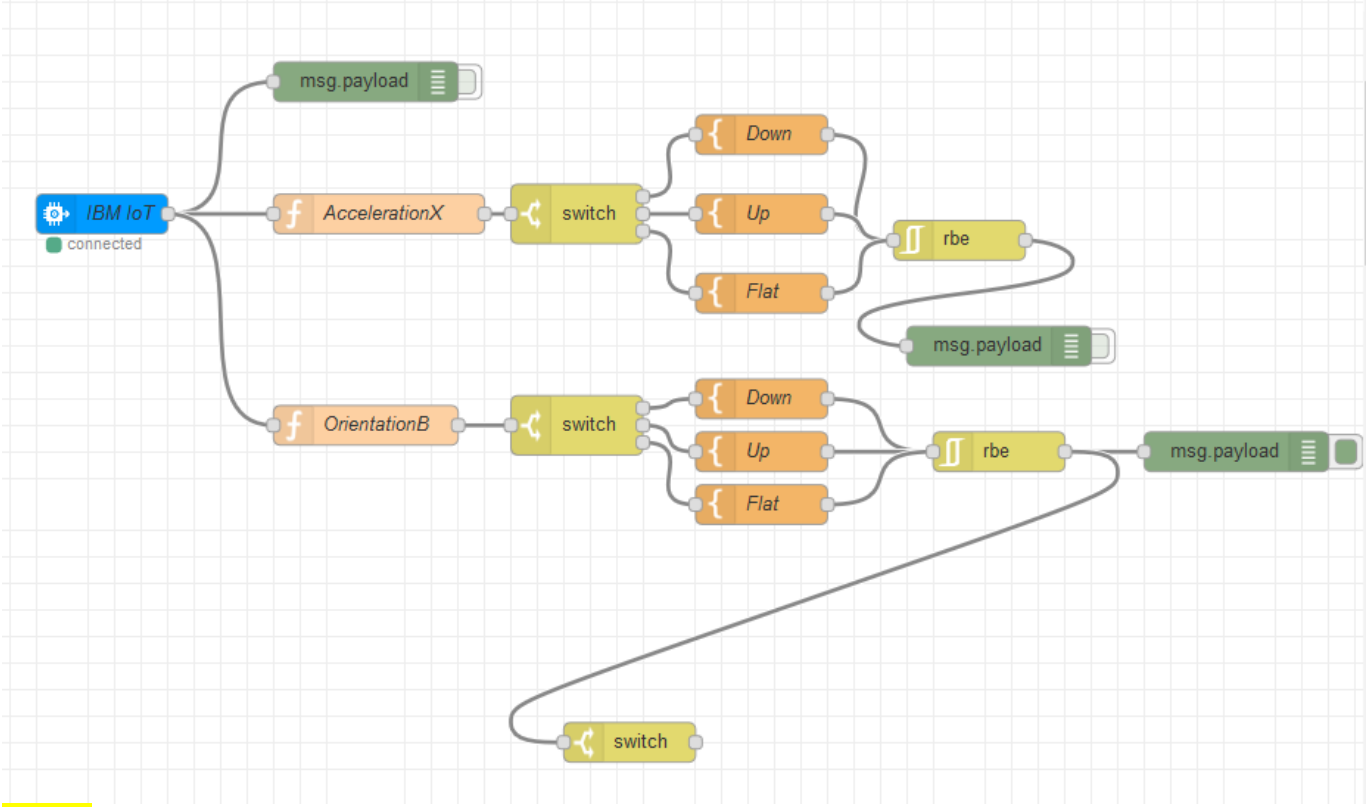
Préparation : aller chercher le fichier CheckFlat.txt dans le repository GitHub
Pour rappel : https://github.com/CorinneBacle/html5-phone
Répertoire Node-Red-Text

1. Import in a new flow the content of the file named: CheckFlat.txt

- Open the txt file
- Select all then copy to the clipboard
- Go to the NodeRed Burger => Import => Clipboard



- Select “in a new flow” => the txt import in a new flow, so not necessary to create a new flow first
- Past your clipboard content
- Click Import
- Then deploy
- Result



TIPS :

- Penser à reconfigurer l'authentification sur le nœud IBM IOT
- Refaire également les debugs

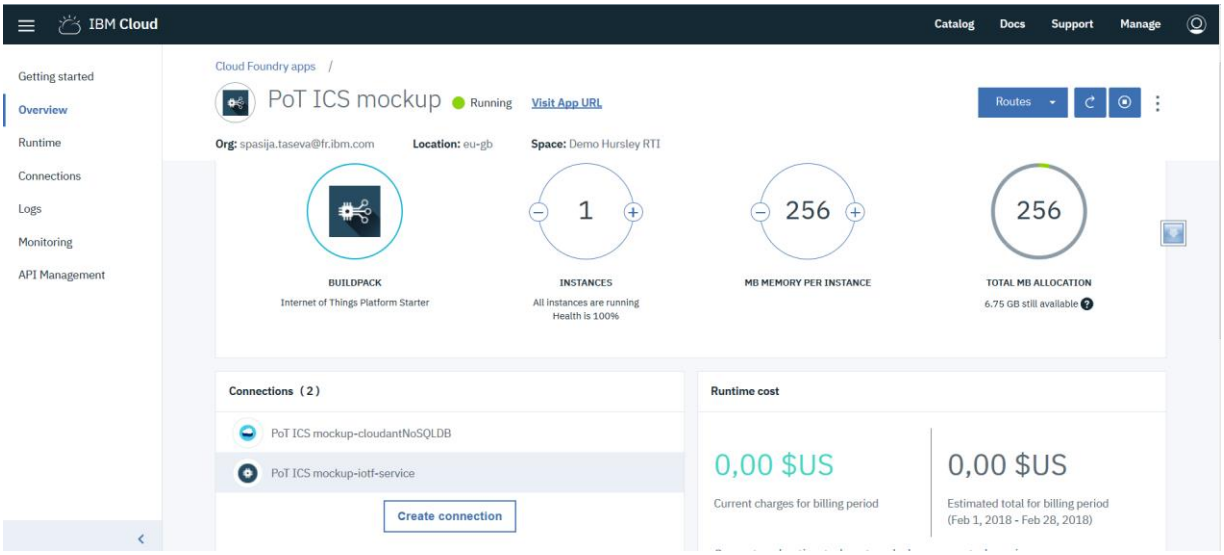
-
- Run your web interface
- See the result in the debug tab

2. Use the Watson Text To Speech service to alert when the phone is flat

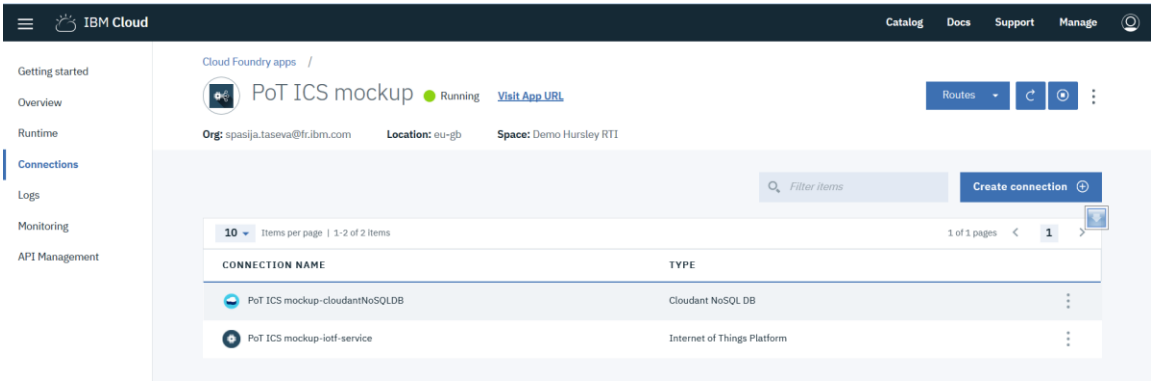
You are now going to use the Text To Speech service to play an audio alert in a web page using a web socket.

IBM CLOUD / user1

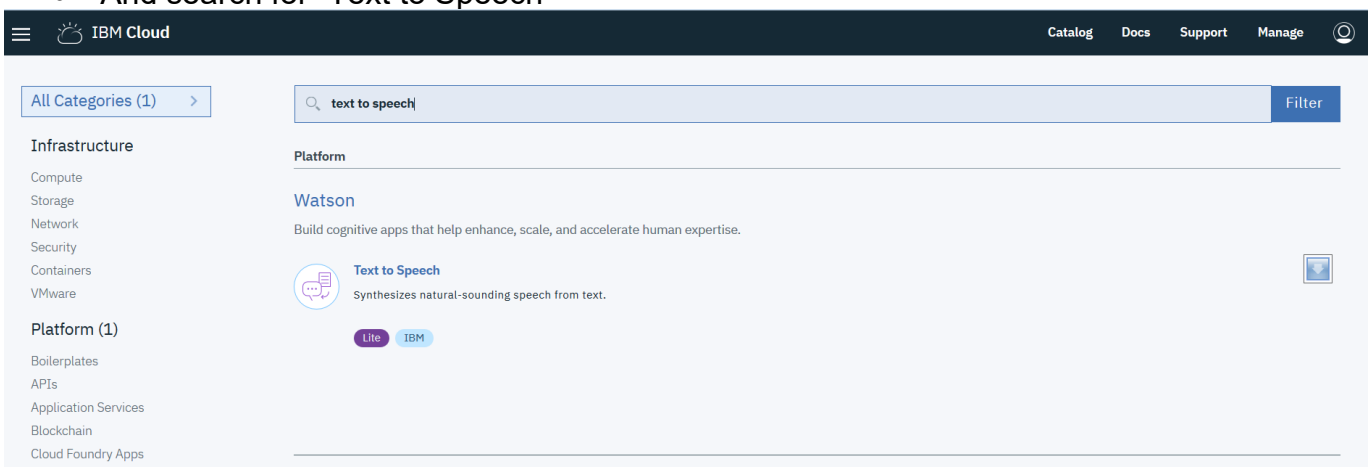
- Go back to IBM Cloud and click on the "Connections" tab:



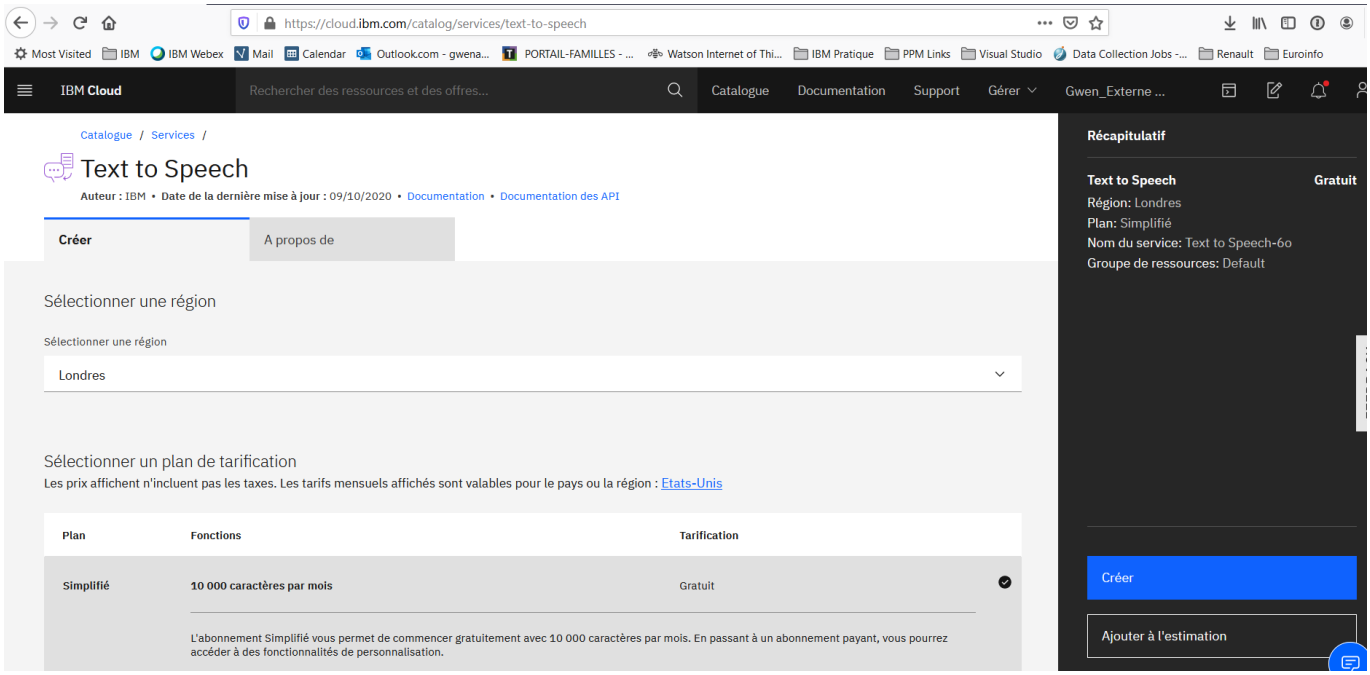
- Click on Catalog:



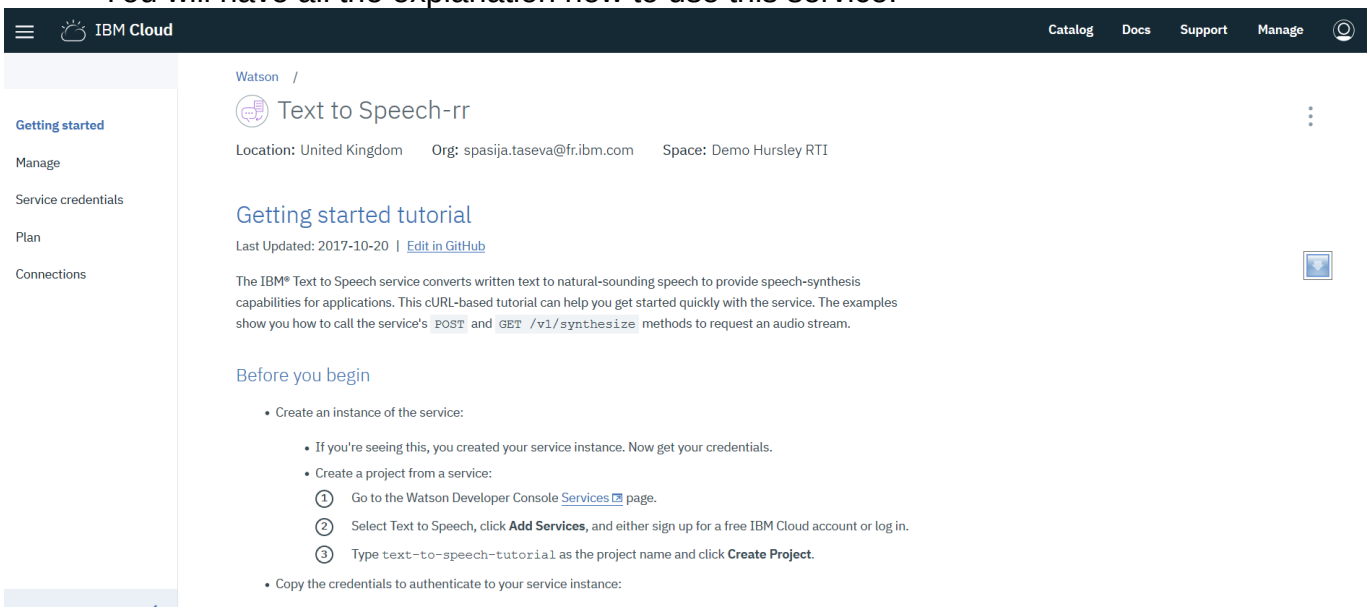
- And search for “Text to Speech”



- Click on Create



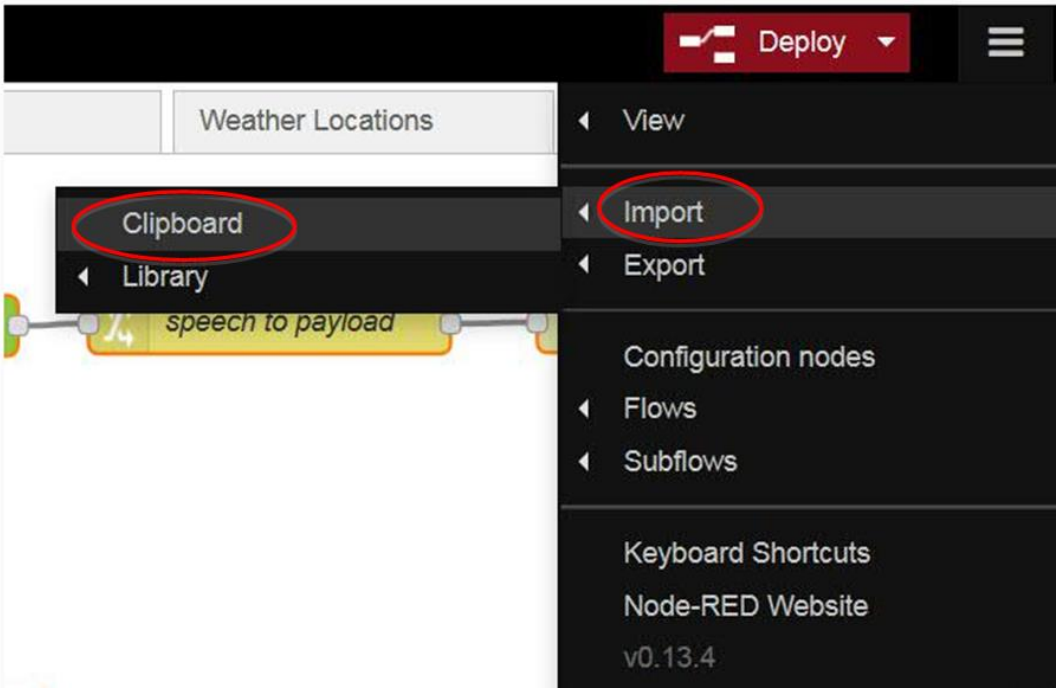
- You will have all the explanation how to use this service:



- Go to Service Credentials
- Click New Credential
- Select Auto Generate
- Select View credentials and keep it in a separate file for the next phase.

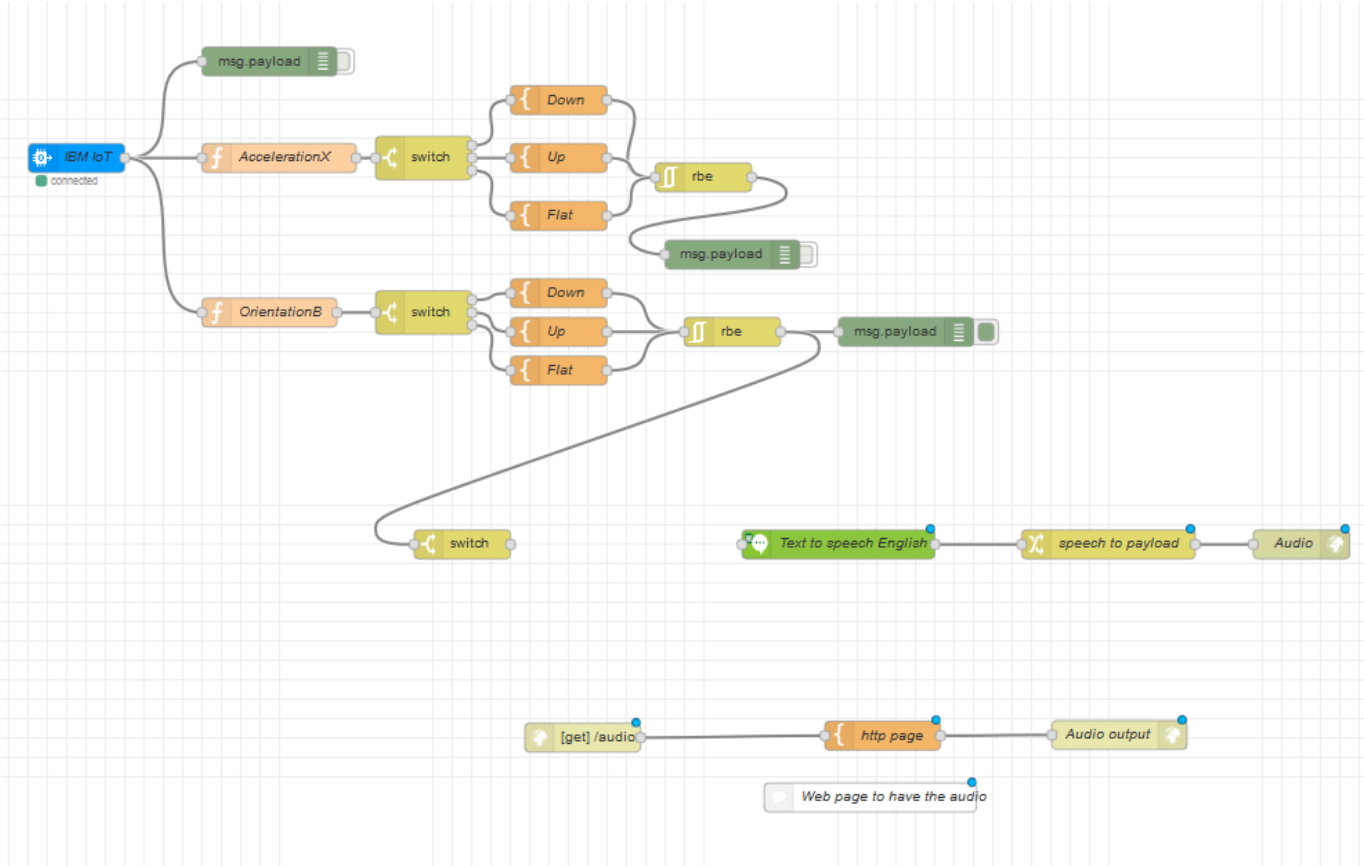
Note : in case of problem, stop then tart the application again.

- Go to your NodeRed application
- Import



- Import in the current flow the content of the file named: **TextToSpeechNodered.txt**

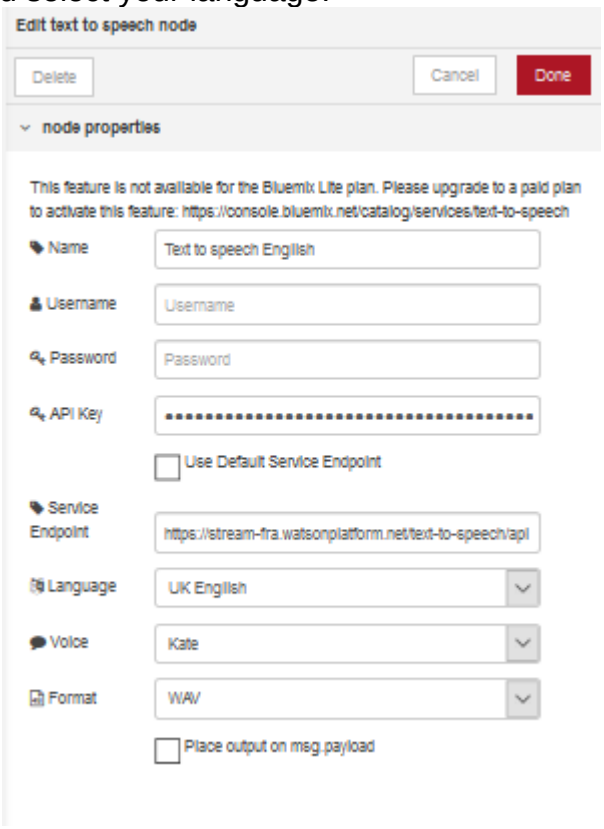
Note ; if it create a new flow, cut and past it to the current flow before any deployment.
Result :



- Add a template between the switch and the text to speech with a text like :
  “Hey the phone is {{payload}} now, good job!”
- Change the switch node to have == Flat

- Connect the switch node to the function and the function node to the text to speech

- Edit the Text to Speech node, add the API Key you have saved before in a separate file, add the service endpoint and select your language:



- For test you can add an Debug node after IOT input node.

- Click to **Deploy**.

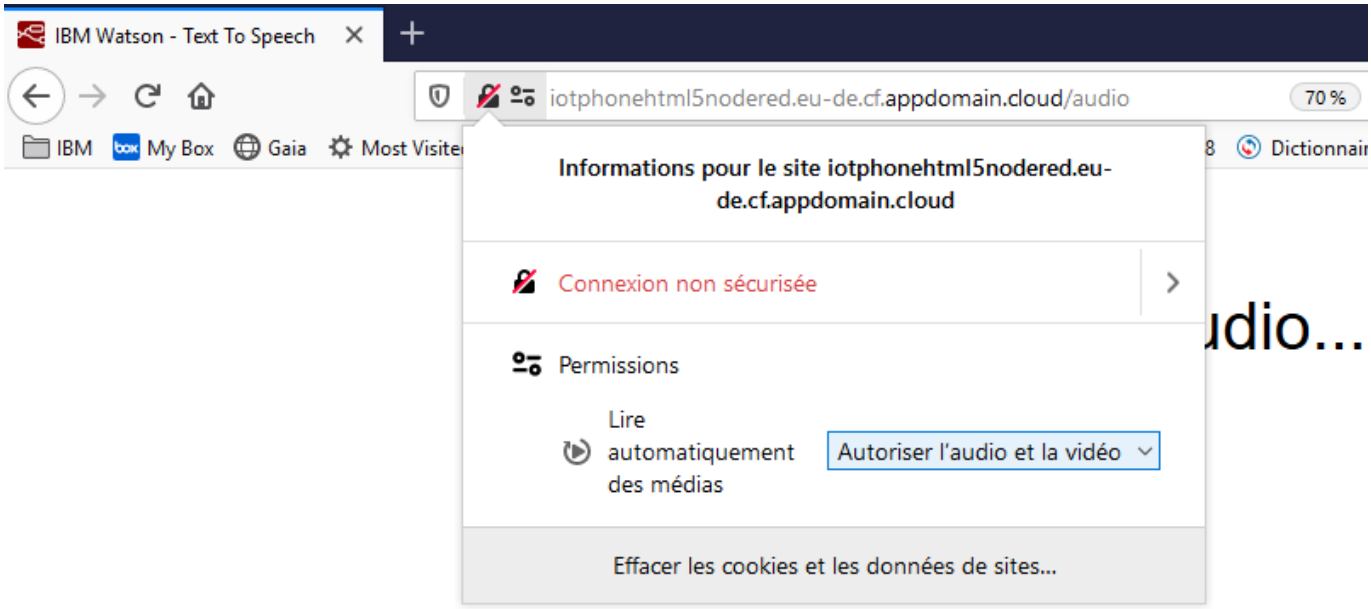
- Browse [url from nodered for your applicationname.cloud] /audio

N.B: url from nodered without ‘s’ for http://

- Turn your laptop volume up.

Waiting for audio...

- Try to have your phone flat.Then verify the audio result.
- In case of problem : check permissions:

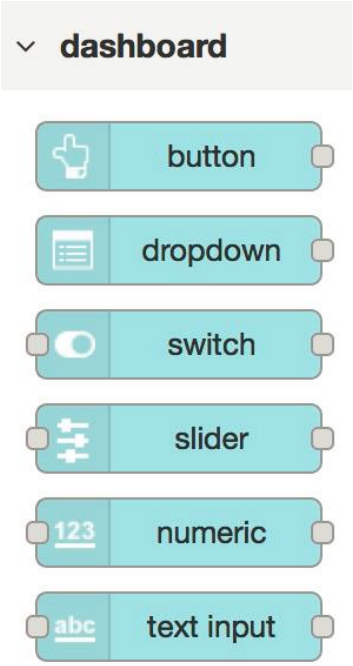


V. Create a dashboard application in Node-RED

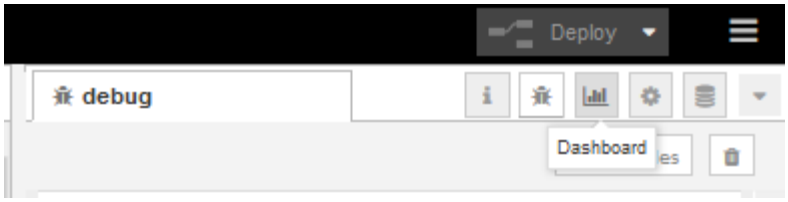
A. Node-RED Dashboarding capability:

We need it to be able to display elements in the next step

1. Note dashboard nodes on the palette:

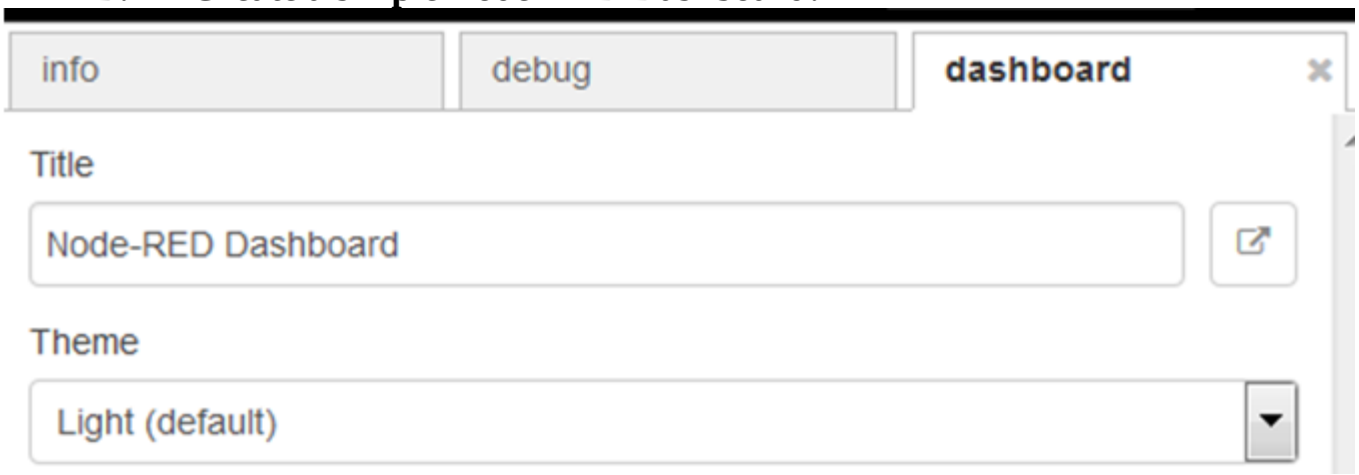


2. Note also that there is a **dashboard** tab in the right-hand sidebar:



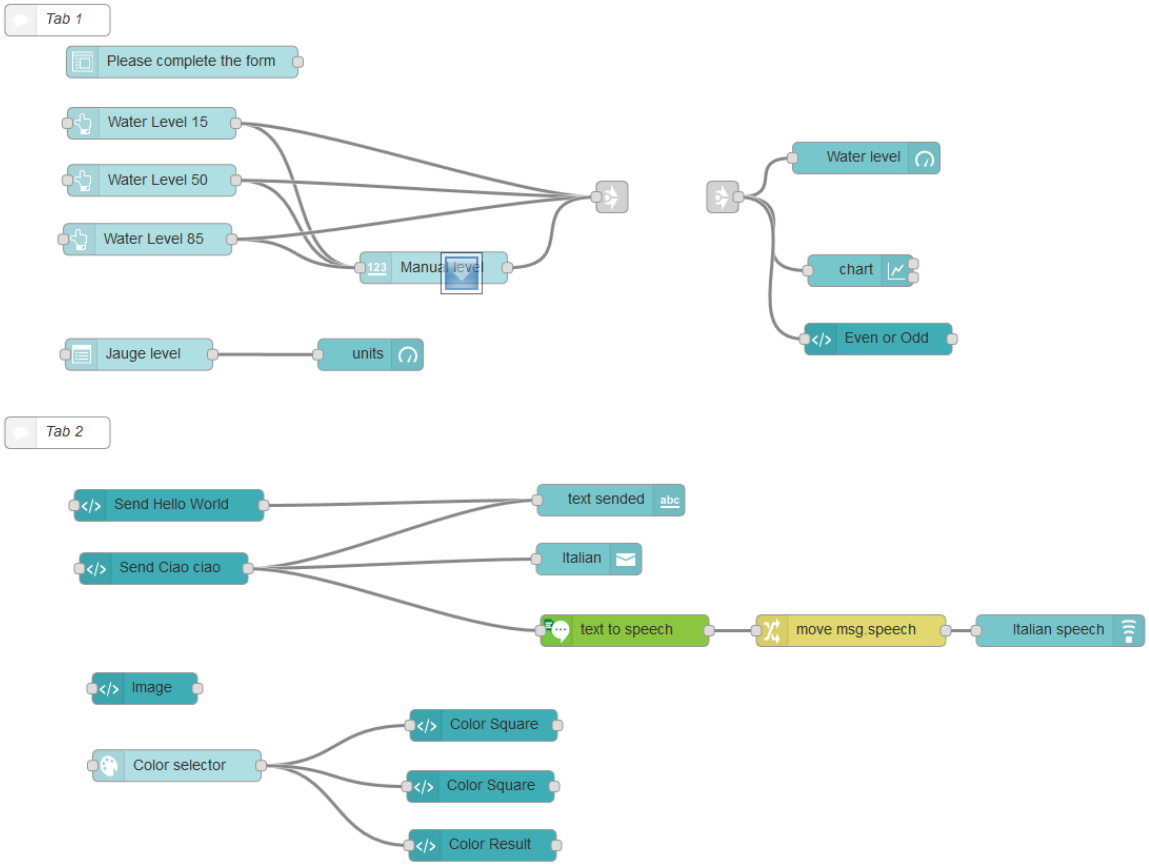
TIP: This dashboard tab may be used to add new tabs, menus etc to the visualization dashboard. There are also two available themes by default – light and dark.

B. Create a simple Node-RED Dashboard:



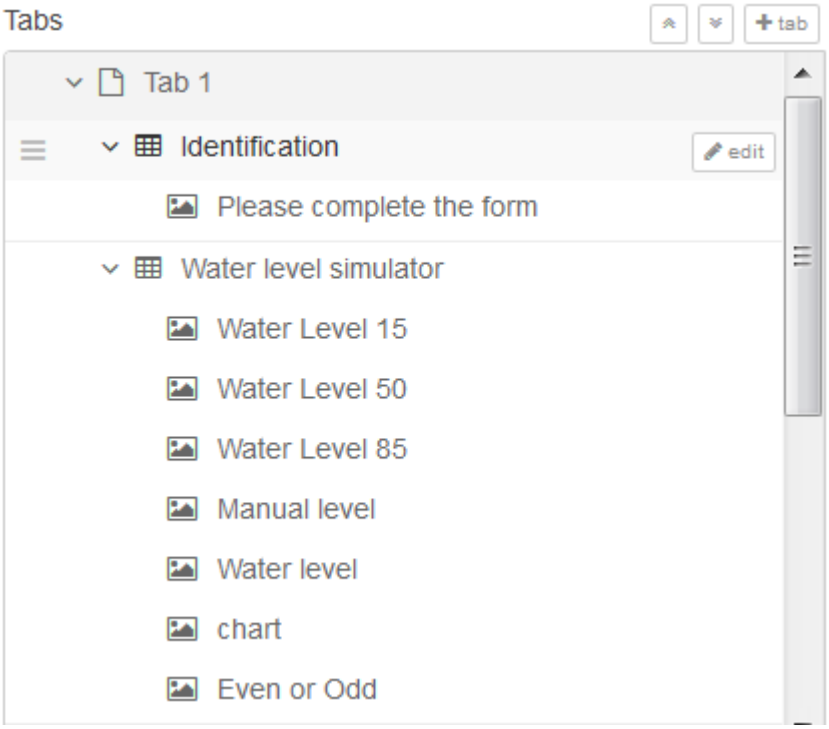
Please import in a new Node-RED tab the file named: NodeRED Dashboard Sample.txt

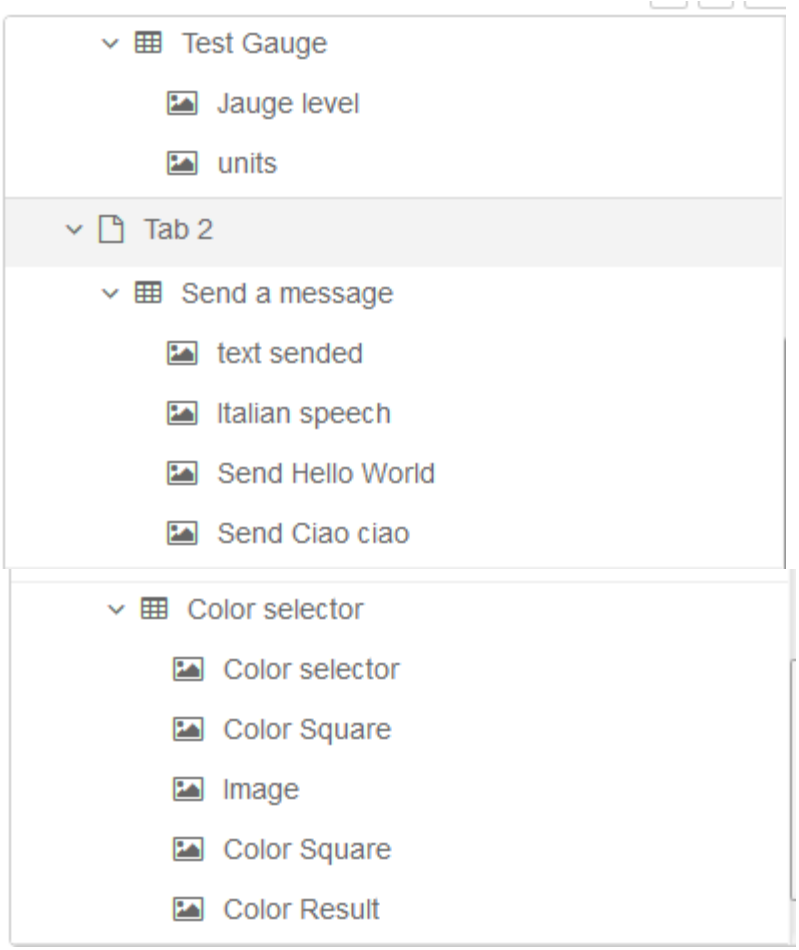
1. Result in Node-RED:



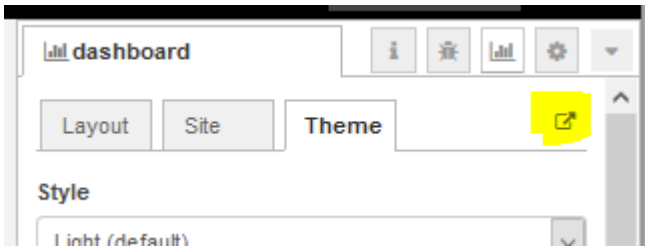
Configure the TextToSpeech with the API key. Keep it in Italian.

2. Configure the dashboard:

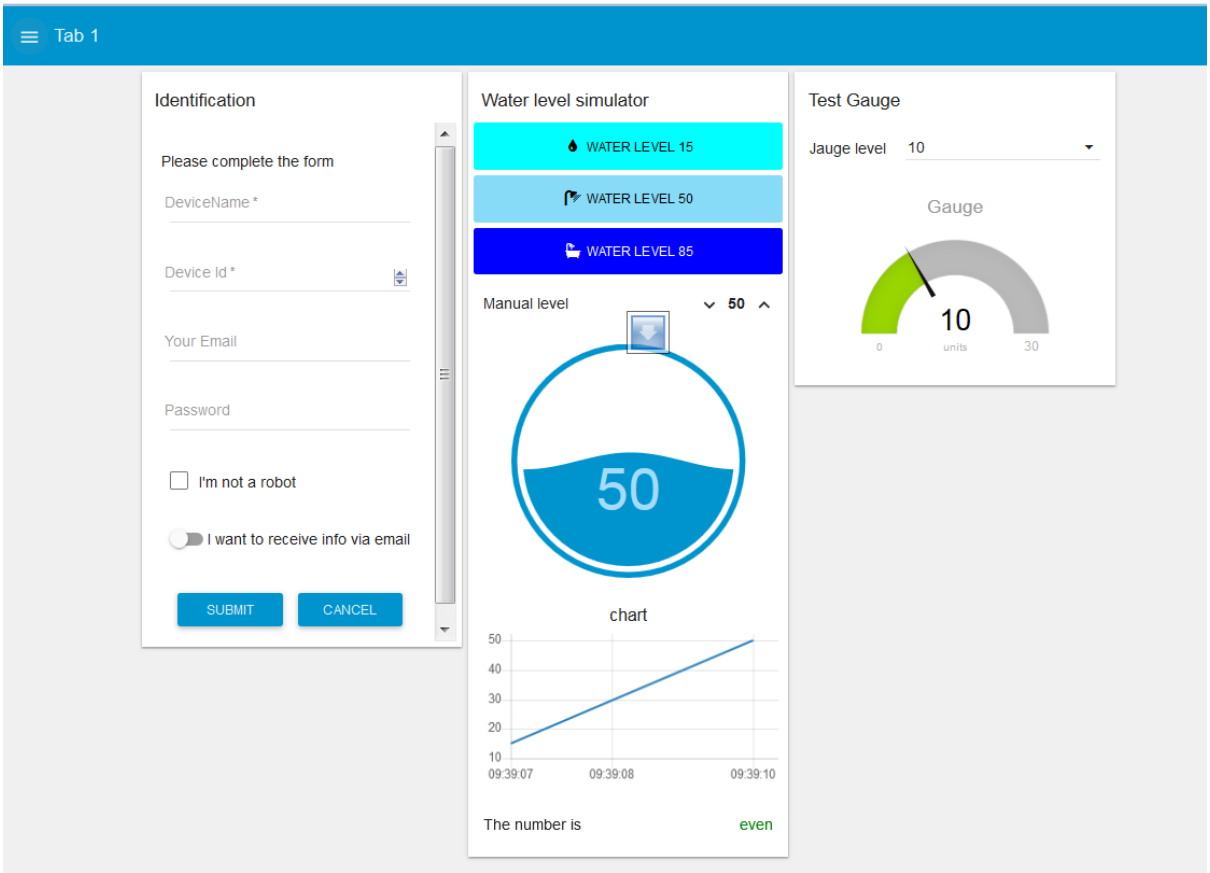


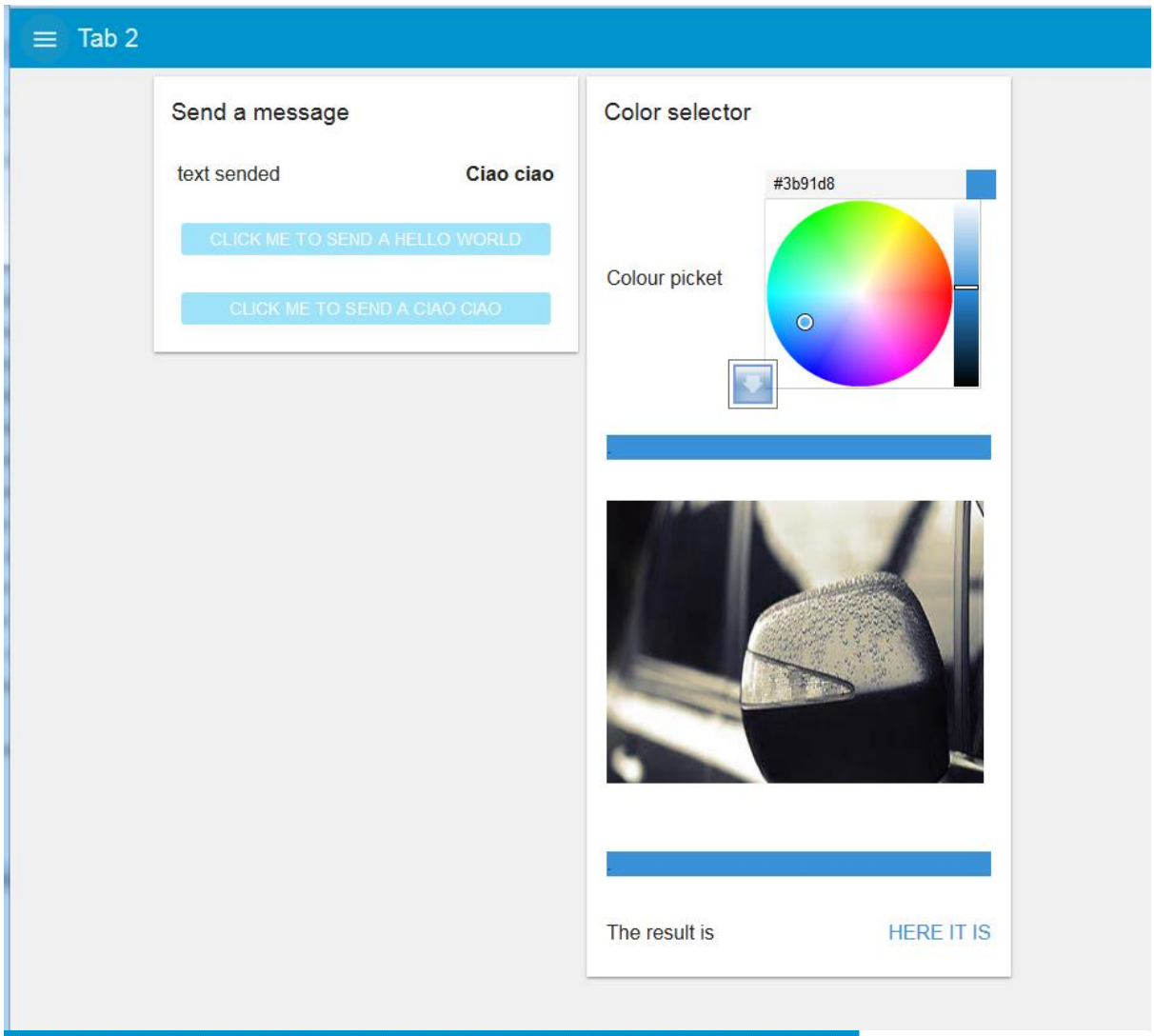


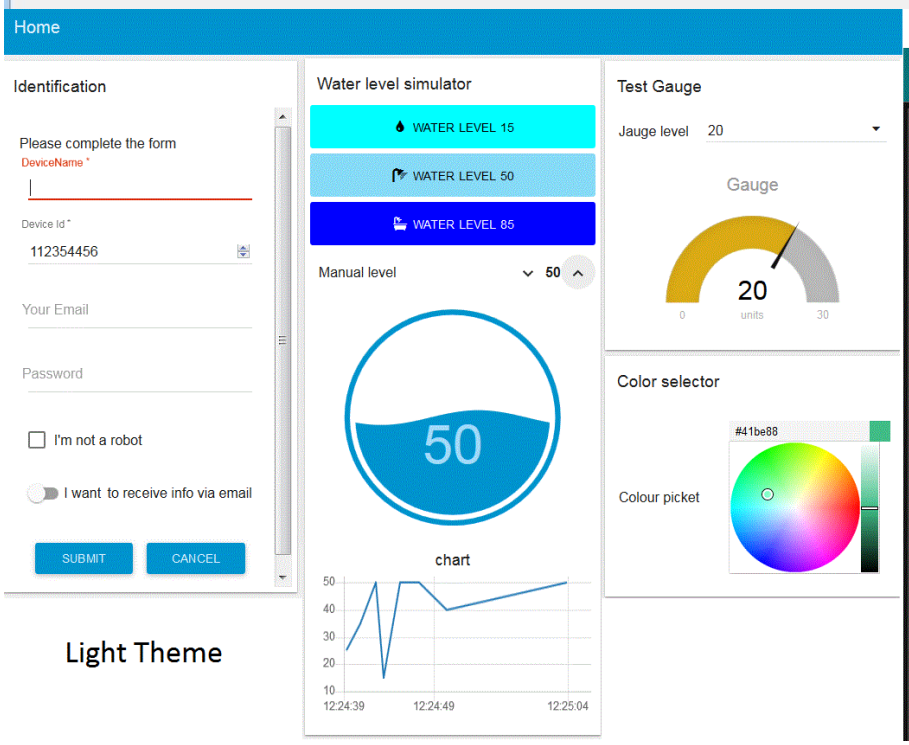
3. Open the dashboard :

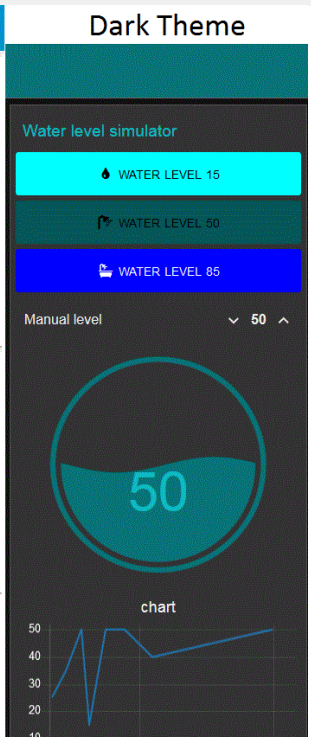


4. Result in the dashboard:



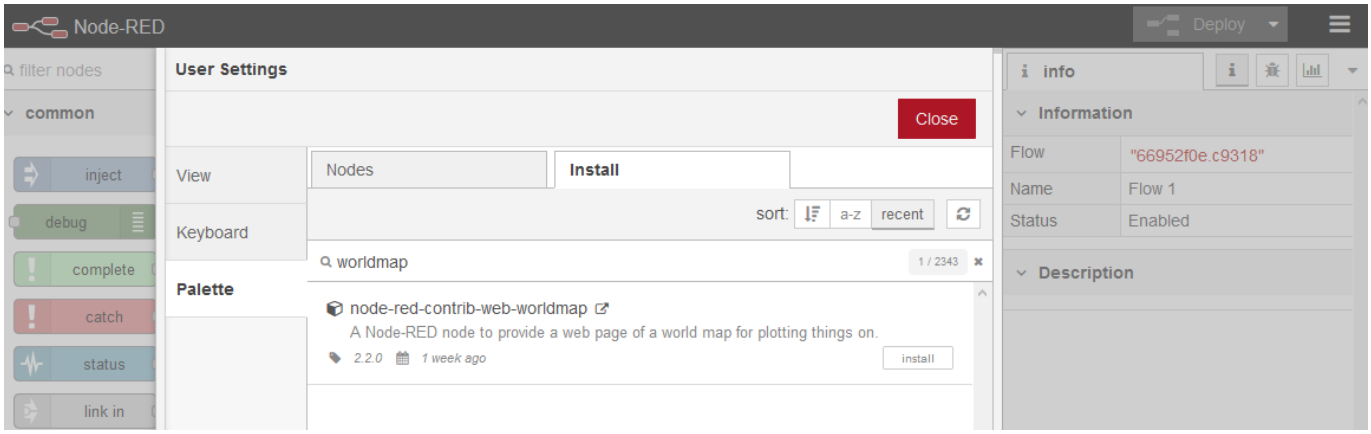

Tab 2
Send a message
text sended
Ciao ciao
CLICK ME TO SEND A HELLO WORLD
CLICK ME TO SEND A CIAO CIAO
Color selector
#3b91d8
Colour picket
The result is
HERE IT IS


Home
Identification
Please complete the form
DeviceName *
Device Id *
112354456
Your Email
Password
I'm not a robot
I want to receive info via email
SUBMIT
CANCEL
Water level simulator
WATER LEVEL 15
WATER LEVEL 50
WATER LEVEL 85
Manual level
50
chart
Test Gauge
Jauge level
20
Gauge
Color selector
#41be88
Colour picket
Light Theme

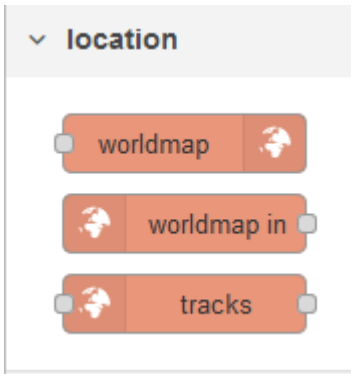

Dark Theme
Water level simulator
WATER LEVEL 15
WATER LEVEL 50
WATER LEVEL 85
Manual level
50
chart

C. Test worldmap

Add the worldmap palette:



New nodes :



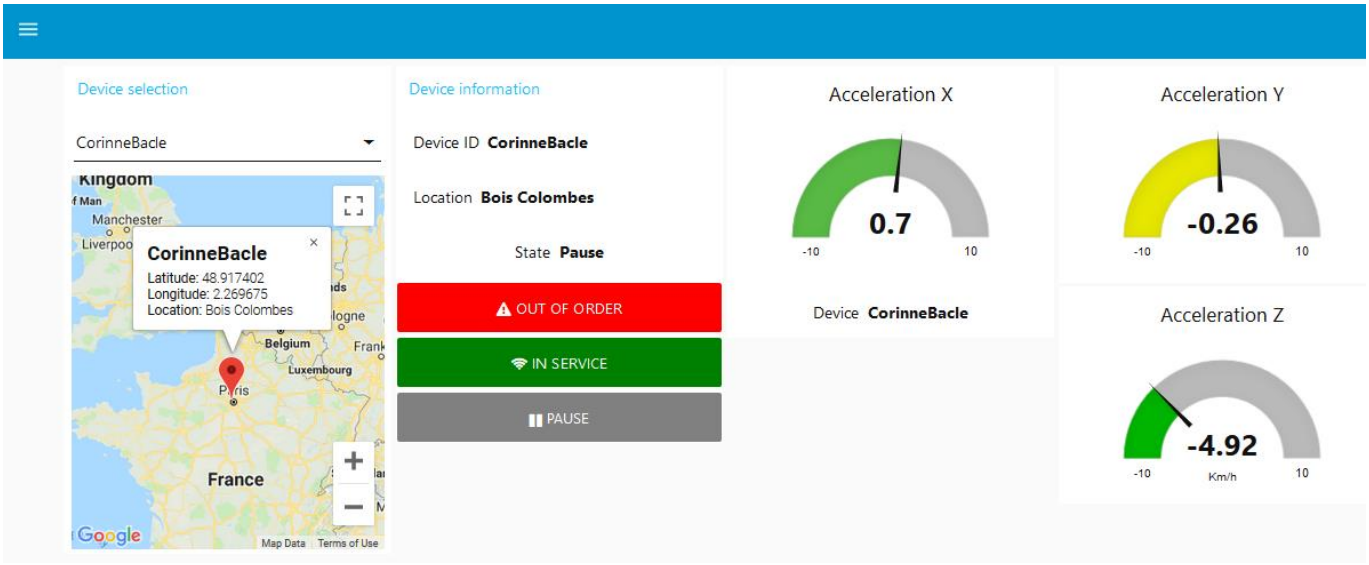
https://<YourApplName>.eu-gb.cf.appdomain.cloud/worldmap/ (see the link in the information of worldmap nodes.
Readme is available here: https://www.npmjs.com/package/node-red-contrib-web-worldmap

D. Create a dashboard to edit device meta data, display a map and display messages

Please import in a new Node-RED tab the file named: NodeRED Dashboard Device Edition.txt in a new tab.
Mettre à jour la partie fonction Status en metadata et dans le format information
Mettre à jour pour utiliser WorldMap de NodeRed

Update device manager nodes to select your device type if needed (iotphone).
In the IoT Platform, edit the device descriptive location of your iotphone device, and add metadata :

```
{
        "status": "0",
        "longitude": 2.269675,
        "latitude": 48.917402
}
```



E. Use metadatas to create a dashboard to display devices in a table and in a map

Please import in a new Node-RED tab the file named: NodeRED Dashboard Device table.txt
Update device manager nodes to select your device type if needed (Android).
In the IoT Platform, edit the device descriptive location of your **Android** device, and add metadata

```
{
        "status": "0",
        "longitude": 2.269675,
        "latitude": 48.917402,
        "transmission_status": 0
}
```

You can add more Android devices with different latitude/longitude and transmission-status to 0 or 1 or 2 to test the result.

Update the second buildHtml node with your googlemap API key.
Result:

Devices

Device	Device Location	Longitude	Latitude	Status
POTIoTCS	Ici			0
ThePhone	Tour Eiffel	48.858093	2.294694	1

REFRESH

CLEAR