



# **Experience the power of IBM Watson IoT Platform**

## **Hands-On Workshop**

## Table of Contents

I.	Prerequisites.....	3
II.	Explore data coming from the IoTPhone application .....	3
III.	Use data coming from your IoT SmartPhone.....	10
	A. Open this link on your smartphone and enter your unique ID: .....	10
	B. Notice that your device is recognized as a connected device in the Watson IoT platform (green flag) and events are available in the “Recent Events” tab: .....	11
	C. Create Physical Interface .....	11
	D. Create Logical Interface .....	14
	E. Visualize in IoT Dashboard.....	17
	F. Create a board to display location in a map.....	17
IV.	Use your NodeRed application to create and generate an alert.....	18
	A. Visualize in Node-RED .....	18
	B. Smartphone interaction with an application: Create an alert.....	23
V.	Directly in IoT Platform : Create an alert when your smartphone falls down .....	29
	A. About rules and actions: .....	29
	B. Understanding rules.....	29
	C. Configuring notifications .....	30
	D. Node-RED Dashboarding capability: .....	30
	E. Configure a rule: .....	31
VI.	Store your data in a Cloudant NoSQL DB.....	36
	A. Solution embedded in the IoT Platform: (To Be Updated) .....	36
	B. Storage using Node-RED: .....	36
VII.	Create a dashboard application in Node-RED .....	40
	A. Create a simple Node-RED Dashboard: .....	40
	B. Test worldmap .....	44
	C. Create a dashboard to edit device meta data, display a map and display messages....	44
	D. Use metadatas to create a dashboard to display devices in a table and in a map .....	46
VIII.	Annex 1 : Secure your Node-RED workspace: .....	47
IX.	Annex 2 : Defining a simulated device using Watson IoT Platform .....	48
	A. Activate device simulator: .....	48
	B. Create a simulated device: .....	48
X.	Annex 3: sample API usage : device management automation using Node-RED .....	49
XI.	Annex 4 : RLAC usage (Ressource-Level Access Control) .....	49

## I. Prerequisites

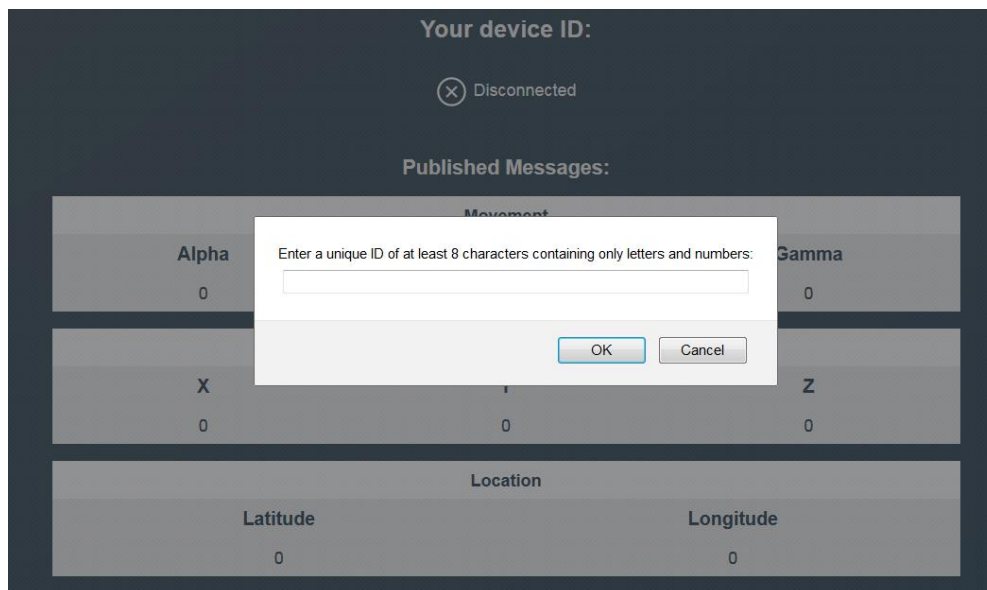
Your IBM Cloud environment must be configured to have the IoT Phone application and the Node-RED Starter depending on your IBM Cloud account constraints.

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

Name	Group	Location	Offering	Status	Tags
Filter by name or IP address... Filter by group or org... Filter... Filter... Filter... Filter...					
Devices (0 / 0)					
VPC Infrastructure (0 / 0)					
Clusters (0 / 0)					
Cloud Foundry Apps (2 / 26)					
IoTPhoneHTML5NodeRed	corinne.bacile@fr.ibm.com / POTIoT220...	Frankfurt	Node-RED Starter	Running	...
html5-phone-2019H2POTIoT	corinne.bacile@fr.ibm.com / POTIoT220...	Frankfurt	Cloud Foundry Applications	Running	...
Cloud Foundry Services (4 / 48)					

## II. Explore data coming from the IoTPhone application

- 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
- 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.

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

IBM Cloud Catalog Docs Support Manage Search for resource...

Cloud Foundry apps /

**IoTStarterCorinneIoTPhone** ● Running [Visit App URL](#)

Org: corinne.bacle@fr.ibm.com Location: Frankfurt Space: Base

Filter items

10 Items per page | 1-2 of 2 items

CONNECTION NAME	TYPE
iot-phone-iotf-service	Internet of Things Platform
IoTStarterCorinneIoTPhone-cloudantNoSQLDB	Cloudant NoSQL DB

4. Click on the alias of link:

IBM Cloud Search resources and offerings...

Connections

Resource list /

**iot-phone-iotf-service** Alias of [Internet of Things Platform-no](#)

Location: Frankfurt Org: corinne.bacle@fr.ibm.com Space: POTIoTH22019

5. Select Manage then Click on "Launch"

IBM Cloud Search resources and offerings... Catalog Docs Support Manage CORINNE B

Manage

Plan

Connections

Resource list /

**Internet of Things Platform-no**

Resource group: Globale Location: Frankfurt [Add Tags](#)

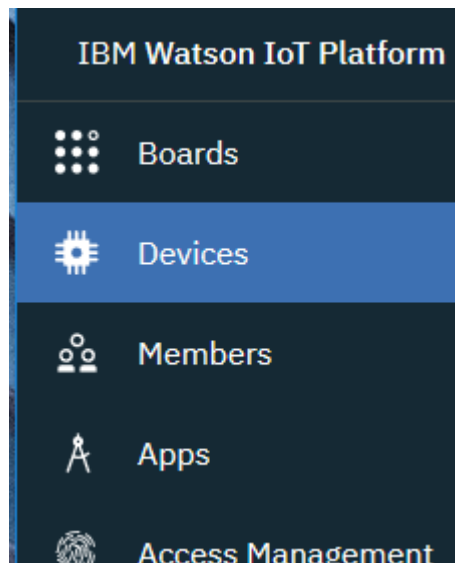
**Let's get started with IBM Watson IoT Platform**

Securely connect, control, and manage devices. Quickly build IoT applications that analyze data from the physical world.

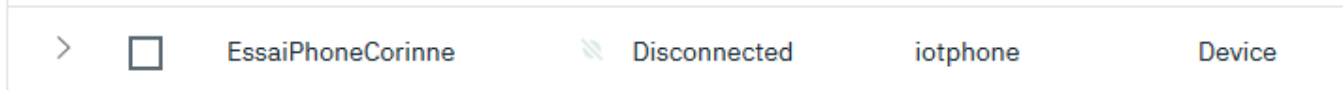
[Launch](#) [Docs](#)

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 on “Device Type”

IBM Watson IoT Platform

corinne.bacle@fr.ibm.com  
ID: ju2tvu

Browse Action **Device Types** + Add Device Type

## Device Types

This table lists all device types that are defined. You can filter the list and search for the name and description. You can modify and configure existing device types and add new device types.

Type the name to search for

Name	Description	Number of Devices	Class ID	
1 result				
iotphone		6	Device	

The “iotphone” device type is automatically created by the IoTPhone application.

8. Test a manual device creation

- Click on “Add Device Type”
- Choose “Device” and put a name for your Device:

## Select Type

Device types group devices that have similar characteristics, such as model number, firmware version, or location. Give the device type a unique name and a description that identifies characteristics that are shared by devices of this type.

Type

Device

Or

Gateway

Name

iPhone

The device type name is used to identify the device type uniquely and uses a restricted set of characters to make it suitable for API use.

Description

Name it “**Android**” (case sensitive). You don’t need to add any description. Click “Next” then “Done”.

You just added a new device type: Android.

Note that you can add Metadata.

### 9. 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:

IBM Watson IoT Platform

Browse Action Device Types

## Browse Devices

☐ All Devices
 ☐ Diagnose

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

<input type="checkbox"/>	Device ID ↕	Device Type ↕	Class ID ↕
6 results			
<input type="checkbox"/>	CorinneBacle	iotphone	Device
<input type="checkbox"/>	Telephone1	iotphone	Device

> ☐ EssaiPhoneCorinne ☐ Disconnected iotphone Device

## 10. Add manually your own Android device :

- Click on “Add Device” on the right, select Android as device type
- Give a name to your device (it will be your device id), click Next:

[Browse](#) [Action](#) [Device Types](#) [Interfaces](#)

Add Device

Identity Device Information Security Summary

Select a device type for the device that you are adding and give the device a unique ID.

Device Type Android

Device ID AndroidCorinnePhone

Cancel Next

- Click Next, you don't need any metadata.
- For the security part, it is recommended to you to provide a simple token (between 8 and 36 characters long and should contain a mix of lower and upper-case letters, numbers and symbols). If you skip this, a token will be automatically generated but this one won't be easy to use for the next steps of this hands-on. Click Next.
- A summary of your device details appears. Copy all these information in a text editor or as a screenshot. The token is unrecoverable.

# Device Drilldown - AndroidCorinnePhone

Device Credentials

Connection Information

Recent Events

State

Device Information

Metadata

Extension Configuration

Diagnostics

## Device Credentials

You registered your device to the organization. Add these credentials. Once the device is connected, you can navigate to view connection and event data.

Organization ID	eo9v4x
Device Type	Android
Device ID	AndroidCorinnePhone
Authentication Method	use-token-auth
Authentication Token	AndroidCorinnePhone

11. You have now 2 devices and 2 device types.

>	<input type="checkbox"/>	EssaiPhoneCorinne	Disconnected	iotphone	Device
>	<input type="checkbox"/>	AndroidCorinnePhone	Disconnected	Android	Device

12. Now see how to use connection security

- In the Security menu on the left, change security settings to accept non SSL connections for Android devices:

IBM Watson IoT Platform

QUICKSTART

SERVICE STATUS

DOCUMENTATION

BLOG

corinne.bacte@r.ibm.com  
ID: (IoTCS POT 3pyhsy)

Policies

## Policies

You can configure policies to enhance connection security and control access to the server from devices.

Connection Security

Configure the security level for device connection.

Blacklist

Disabled

- The select “Add Custom rules”, select scope “Android” and Select “**TLS Optional**”



← Back

Cancel Save

Default TLS with Token Authentication Refresh compliance 7 devices

### Custom Rules

You can define custom connection rules for specific device types. Custom rules overwrite the default rule for the specified device types. The predicted compliance value is updated to reflect the default settings and the custom settings.

Add Custom Rule

<input type="checkbox"/> Scope	Security Level	Predicted Compliance	# of Devices	
<input type="checkbox"/> Android	TLS Optional	Refresh compliance	0	

- Then click Save.

#### What you have done so far:

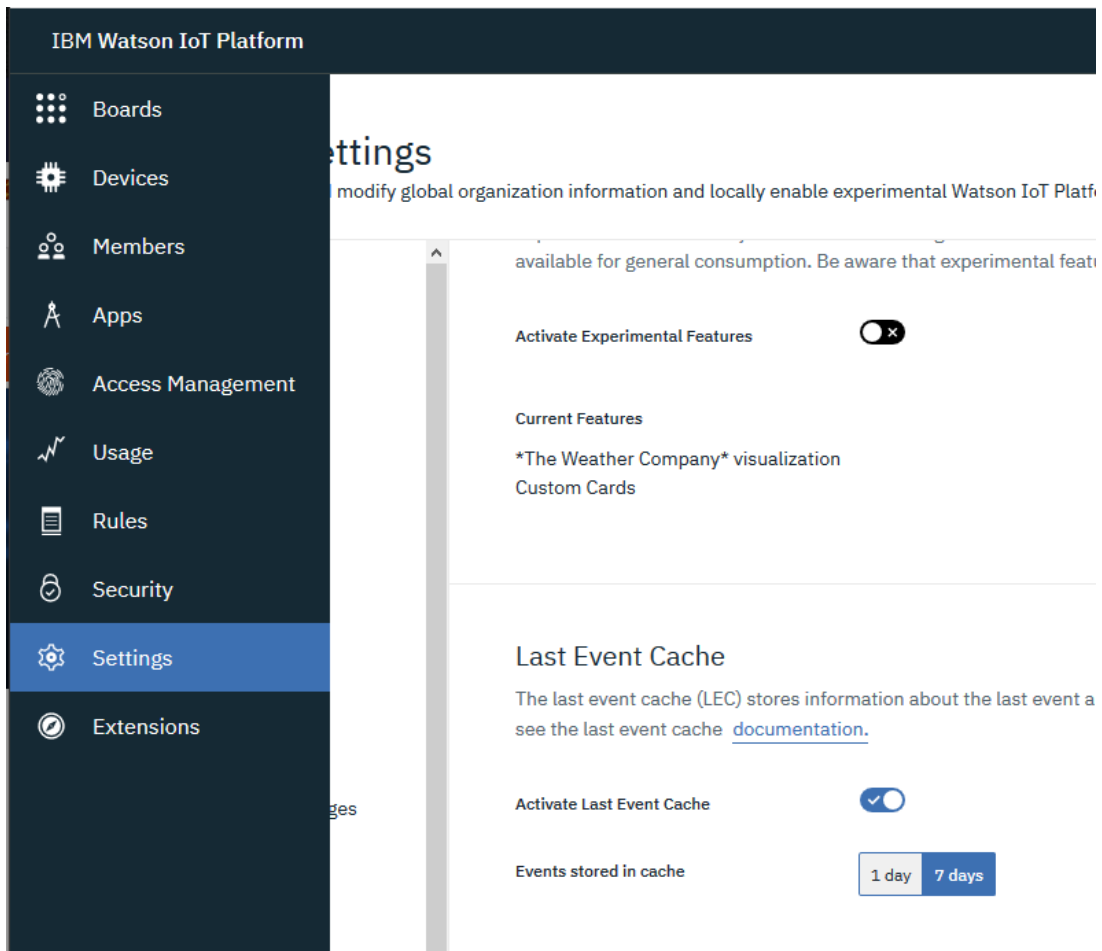
- You have created an IoTPhone web app in IBM Cloud to transform your phone to a IoT device
- You have created a Node-RED starter web app in IBM Cloud
- You have registered 2 devices in the Watson IoT Platform based on iotphone and Android types.
- You have create a custom security rule for Android device type.

#### What you will do next:

- Connect your device to your Watson IoT organization and access your device data in your IoT Starter web app
- Leverage your sensor data with Node-RED

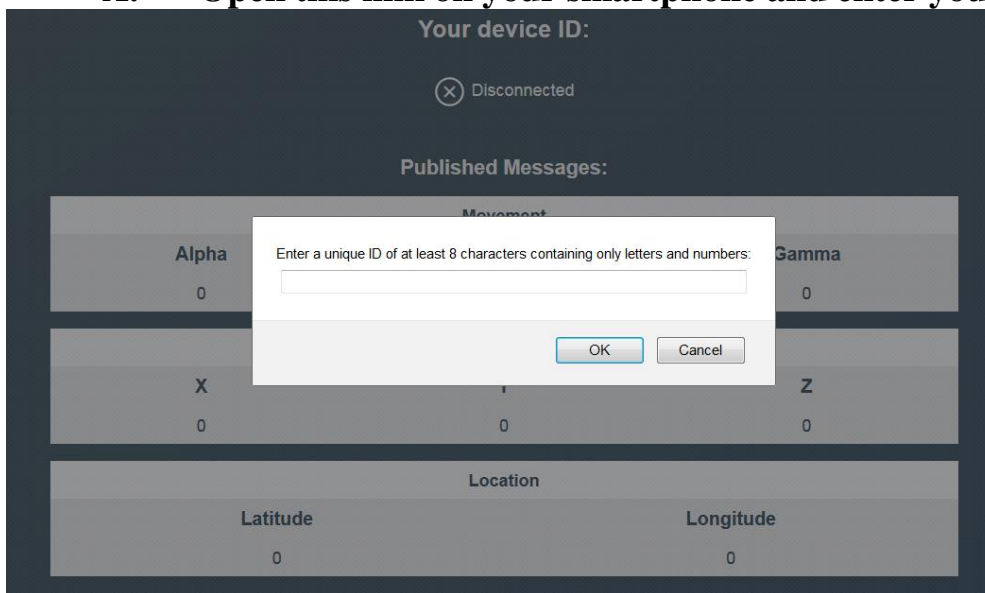
13. Now configure the Last Event Cache (if not already done)

- In the Settings menu on the left, Activate the Last event Cache feature:

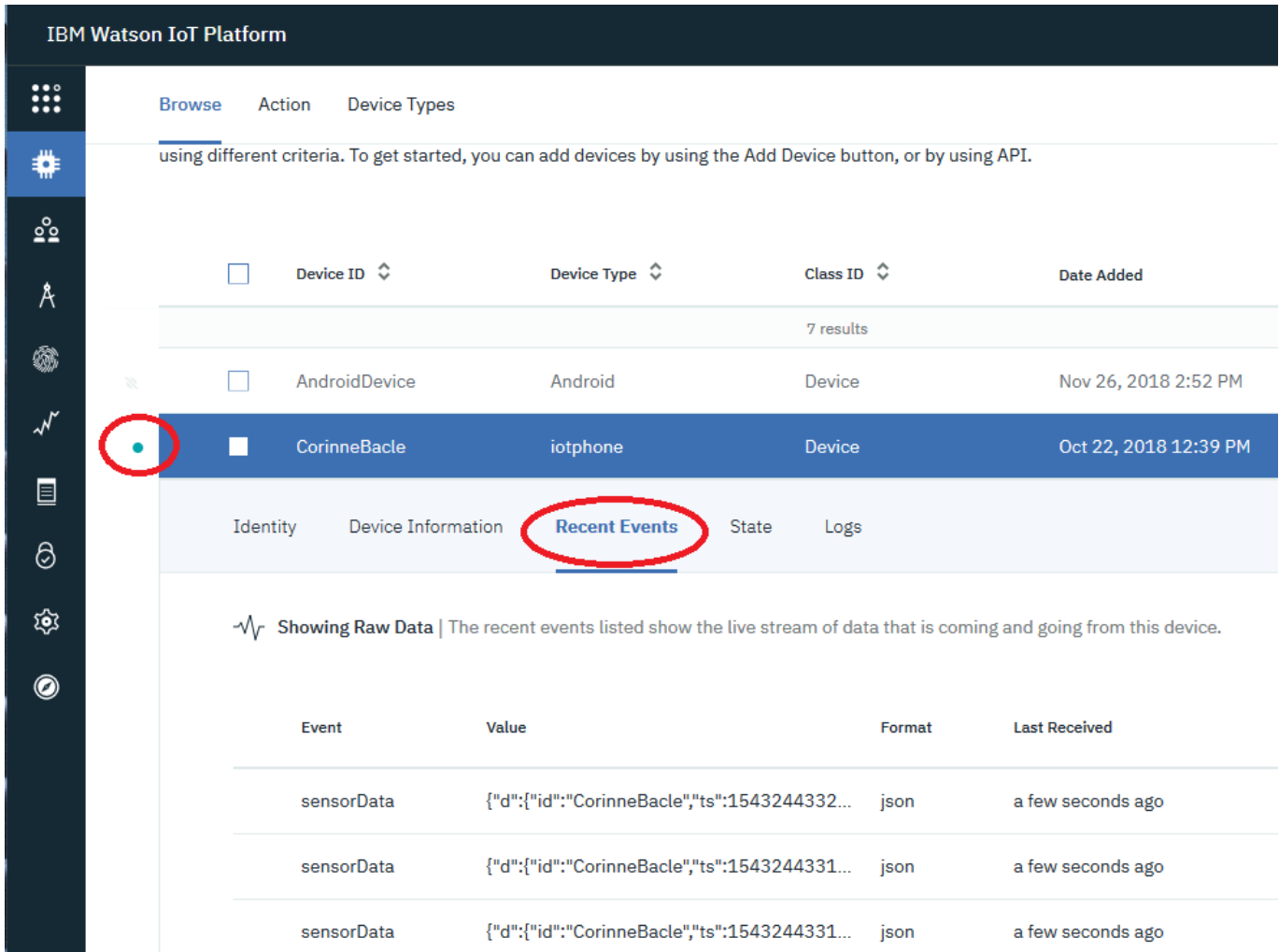


### III. Use data coming from your IoT SmartPhone

#### 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 Watson IoT platform (green flag) and events are available in the “Recent Events” tab:**



The screenshot shows the IBM Watson IoT Platform interface. On the left is a dark navigation bar with various icons. The main content area has a header with 'Browse', 'Action', and 'Device Types'. Below this is a table of devices. The first device is 'CorinneBacle' of type 'iotphone', which is highlighted in blue and has a green status flag circled in red. Below the device list, the 'Recent Events' tab is selected and circled in red. This tab shows a stream of events for the selected device, with columns for Event, Value, Format, and Last Received. The events are 'sensorData' events in 'json' format, received 'a few seconds ago'.

Device ID	Device Type	Class ID	Date Added
7 results			
AndroidDevice	Android	Device	Nov 26, 2018 2:52 PM
CorinneBacle	iotphone	Device	Oct 22, 2018 12:39 PM

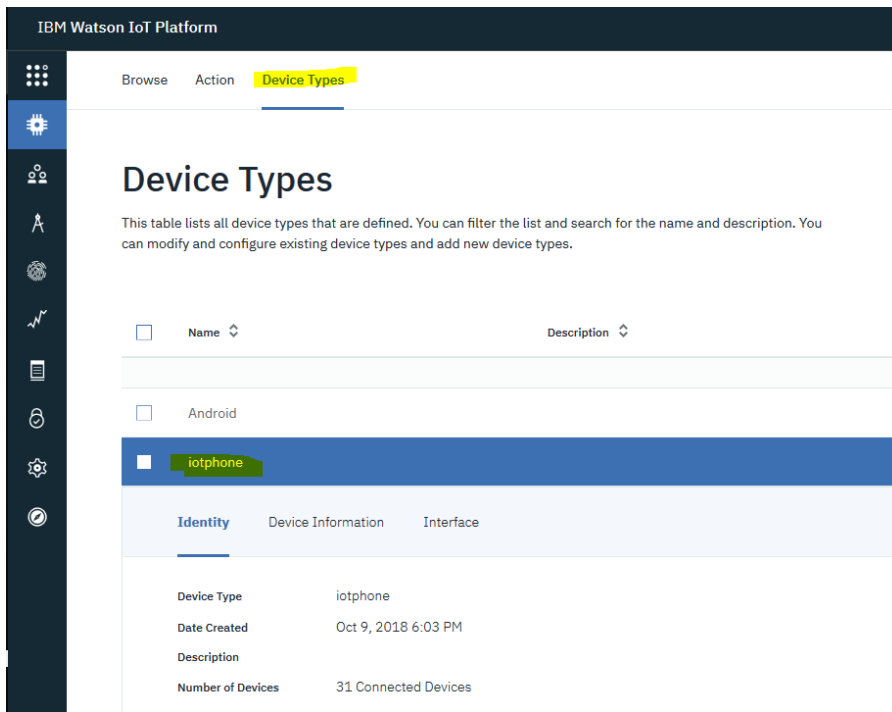
Event	Value	Format	Last Received
sensorData	{"d":{"id":"CorinneBacle","ts":1543244332...	json	a few seconds ago
sensorData	{"d":{"id":"CorinneBacle","ts":1543244331...	json	a few seconds ago
sensorData	{"d":{"id":"CorinneBacle","ts":1543244331...	json	a few seconds ago

**C. Create Physical Interface**

This is part of the data management feature of IoT Platform, see [https://www.ibm.com/support/knowledgecenter/SSQP8H/iot/platform/GA\\_information\\_management/ga\\_im\\_definitions.html](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**.



IBM Watson IoT Platform

Browse Action **Device Types**

## Device Types

This table lists all device types that are defined. You can filter the list and search for the name and description. You can modify and configure existing device types and add new device types.

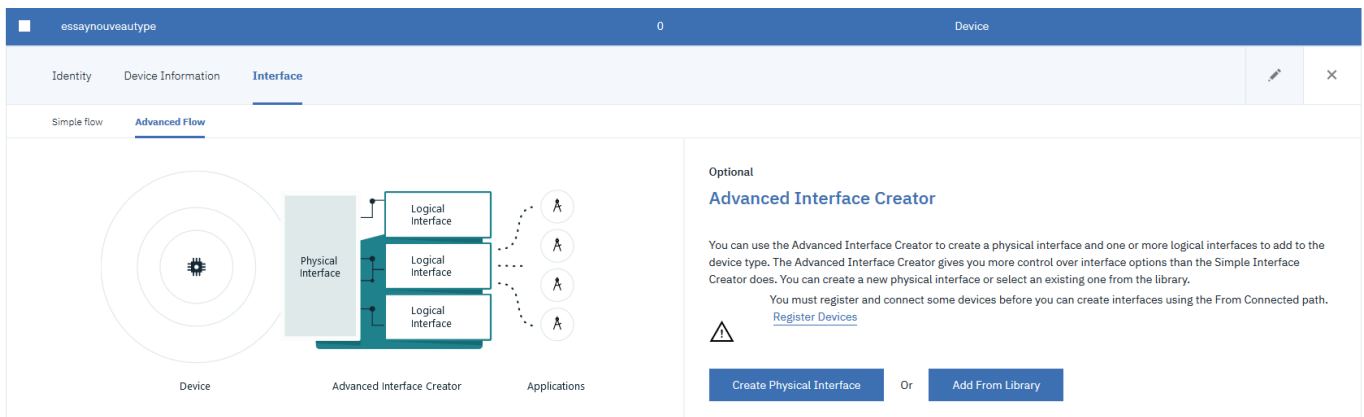
Name	Description
Android	
<b>iotphone</b>	

**iotphone**

Identity Device Information Interface

Device Type	iotphone
Date Created	Oct 9, 2018 6:03 PM
Description	
Number of Devices	31 Connected Devices

- Click **Advanced Flow**.



essaynouveau type 0 Device

Identity Device Information **Interface**

Simple flow **Advanced Flow**

Device

Physical Interface

Logical Interface

Logical Interface

Logical Interface

Applications

Optional

### Advanced Interface Creator

You can use the Advanced Interface Creator to create a physical interface and one or more logical interfaces to add to the device type. The Advanced Interface Creator gives you more control over interface options than the Simple Interface Creator does. You can create a new physical interface or select an existing one from the library.

You must register and connect some devices before you can create interfaces using the From Connected path.

[Register Devices](#)

Create Physical Interface Or Add From Library

- 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).
- 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:

## Add Event Types to Physical Interface

Select the events from which you want to choose the properties to add to the interface.

Use Last Event Cache

15C9999C44BB

Not seeing the right events?  
[Import](#) them or add the ones you'd like to see [manually](#).

Cancel
Add

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

Create Physical Interface

Editing
X

Identity
Event Types and Payload

You can use properties to define the interface behavior and the format of the data that is presented on devices.

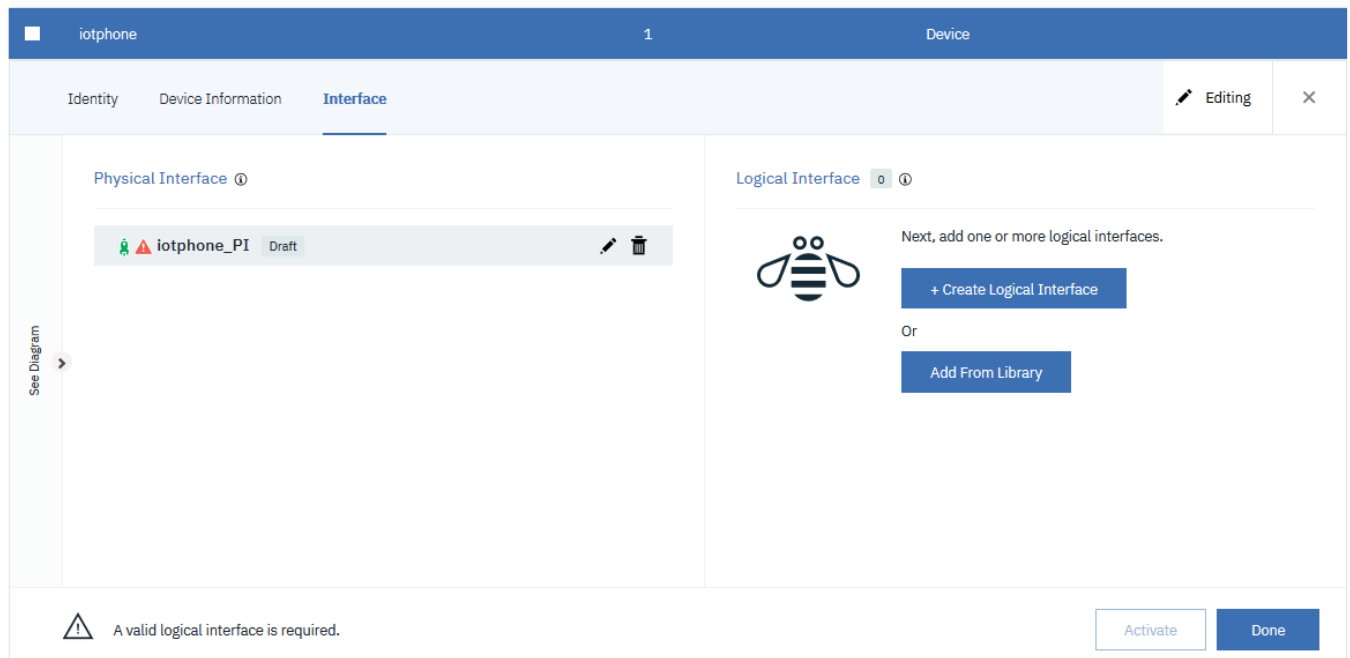
### Define the Physical Interface

+ Create event type

Event Type	Event ID	Format
device type	device type	application/json
Property	Data Type	Required
<div> d </div>	Object	No
<div> id </div>	String	No
<div> ts </div>	Number	No
<div> lat </div>	Number	No
<div> lng </div>	Number	No
<div> ax </div>	Number	No
<div> ay </div>	Number	No
<div> az </div>	Number	No
<div> oa </div>	Number	No
<div> ob </div>	Number	No
<div> og </div>	Number	No
Add another property		
Add another property		

Done

- 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

Celsius temperature as "ambientTemp"

```
{
  "d": {
    "myName": "sensortag",
    "ambientTemp": "30.25",
    "objectTemp": "21.88",
    "humidity": "53.24178",
    "pressure": "1031.17",
    "altitude": "1.43789",
    "accelX": "-0.03",
    "accelY": "0.02",
    "accelZ": "-1.06",
    "gyroX": "-0.85",
    "gyroY": "0.28",
    "gyroZ": "1.30",
    "magX": "46.47",
    "magY": "84.69",
    "magZ": "-45.42",
    "light": "9.49"
  }
}
```

Kelvin temperature as "value"

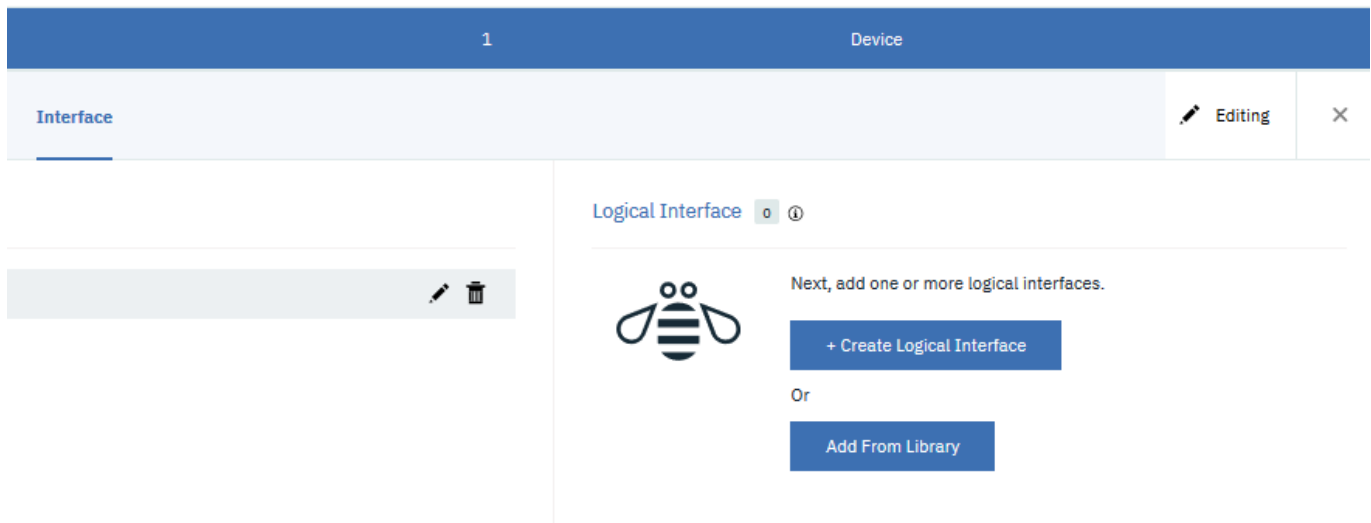
```
{
  "ts":
    "2017-10-02T08:09:34.310+0000",
  "d": {
    "value": 296.77,
    "time": 1506931831214
  }
}
```

Celsius temperature as "t"

{"t": 18.9}



In the device type interface, select "create logical interface":



- Select a name
  - 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.

## Edit Logical Interface: iotphone\_LI



Identity

State Model

Notification Preference

Use properties to define the mappings between the logical and physical interfaces.

## Define the Interface

Property ↕	Mapped Payloads ↕	Data Type ↕	
latitude	lat [sensorData]	Number	
longitude	lng [sensorData]	Number	
AccelerationX	ax [sensorData]	Number	

- 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

✓ The interfaces are valid, but the device type is not yet deployed.

Activate

Done

Interfaces are now active:

iotphone

6

Device

Identity

Device Information

Interface

✎

✕

Physical Interface ①

iotphone\_PI

Logical Interface 1 ①

iotphone\_LI

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

CorinneBacle

iotphone

Device

Oct 22, 2018 12:39 PM

Identity

Device Information

Recent Events

State

Logs

Interface:

iotphone\_LI

Raw Data

iotphone\_LI

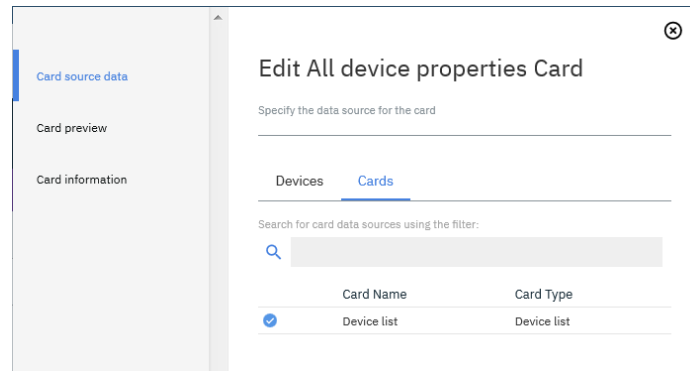
			Event	Last Received
longitude	null	Number		23 minutes ago
latitude	null	Number		23 minutes ago
AccelerationX	-0.32	Number		23 minutes ago

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



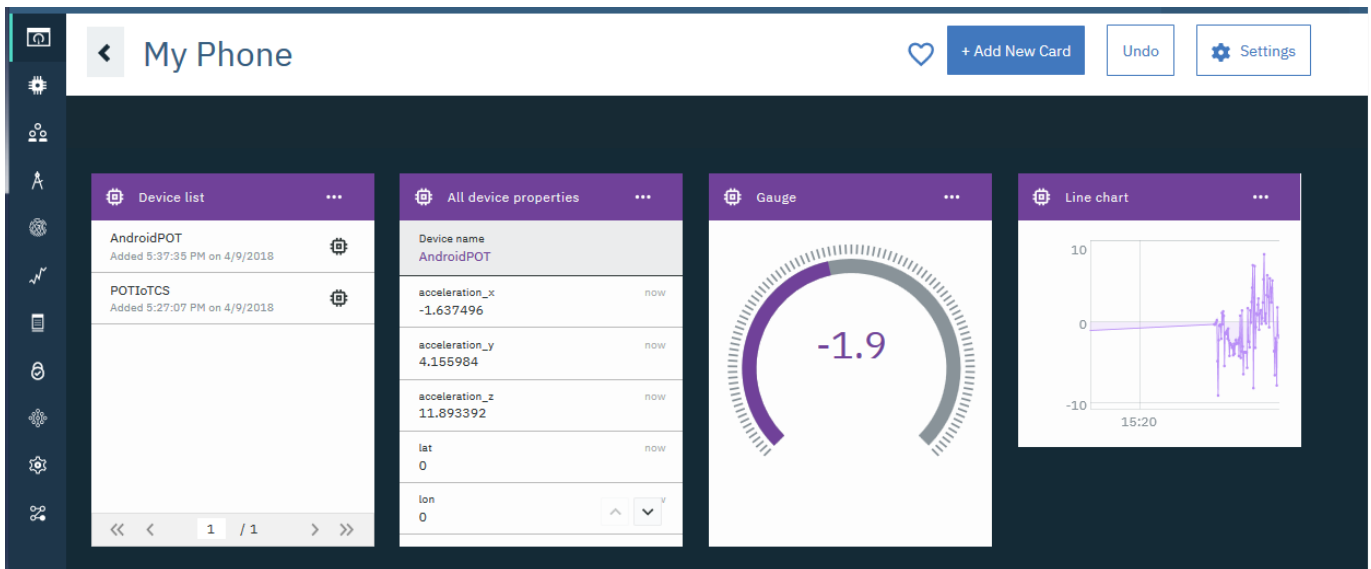
## E. Visualize in IoTP Dashboard

- Create a board to contain the cards for your devices.
  - If the All Boards, page is not already displayed, select **Boards** from the Watson 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 Watson 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.
3. Select your device from the list, then click **Next**.

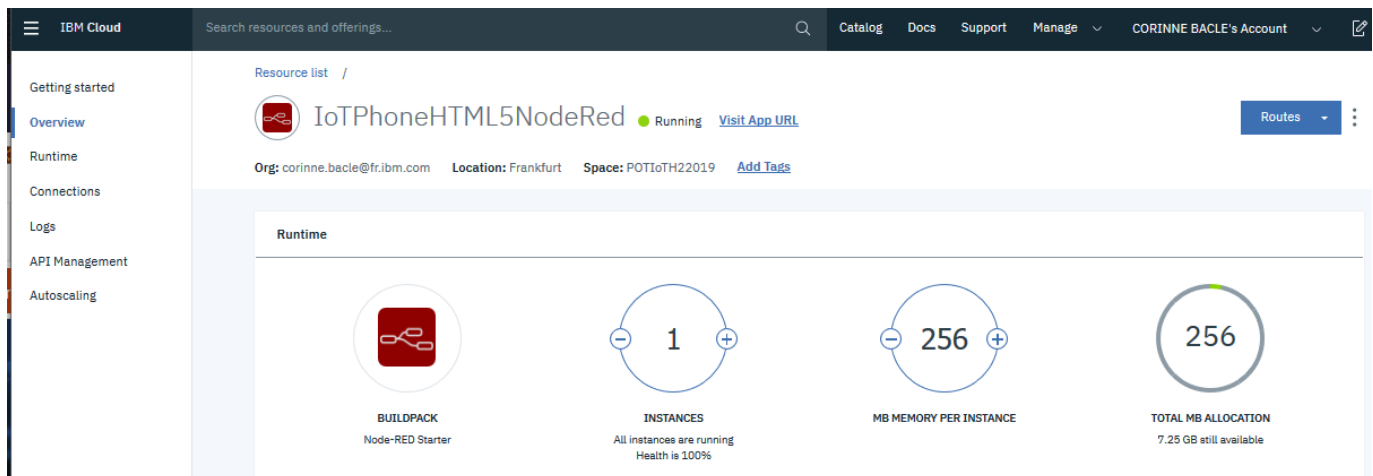
4. Click **Add New Card**, and then select the **Device Map** card type, which is located in the Devices section / show more.
5. Select your device list from the list, then click **Next**.
6. Select Data point from your device type, for longitude and latitude
7. Click Next
8. In the Card Preview page, 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

1. Go back to your application in IBM Cloud (in your browser, the page you have bookmarked at the beginning), you are now going to collect your data from your app.

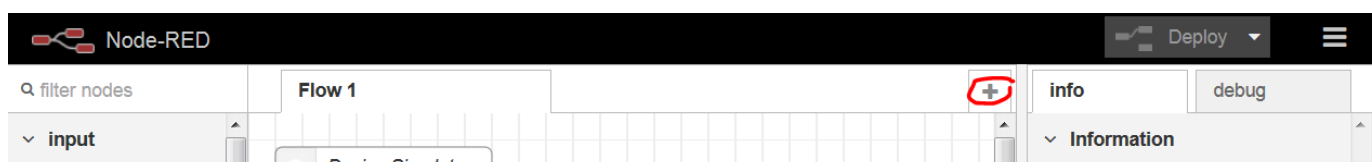
Go back to your Node-RED Starter app overview:



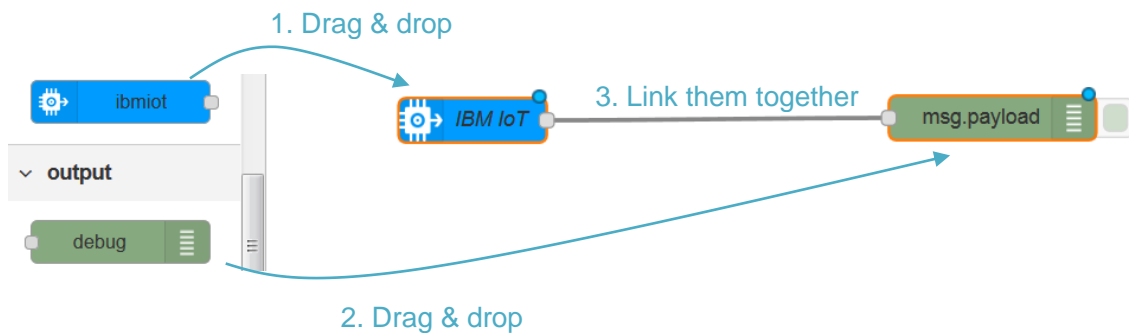
2. Click on Visit App URL to access the Node-RED web editor:
3. Click on “Go to your Node-RED flow editor”

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 create a **new** flow (or use the one created in the init phase)



5. Drag and drop the following nodes:



6. Then, double click on the IBM IoT node to configure it with the “Bluemix service” authentication and fill your device id:

Edit ibmiot in node

Cancel
Done

Authentication
Bluemix Service

Input Type
Device Event

Device Type
☒ All or +

Device Id
☐ All or myPhone

Event
☒ All or +

Format
☐ All or json

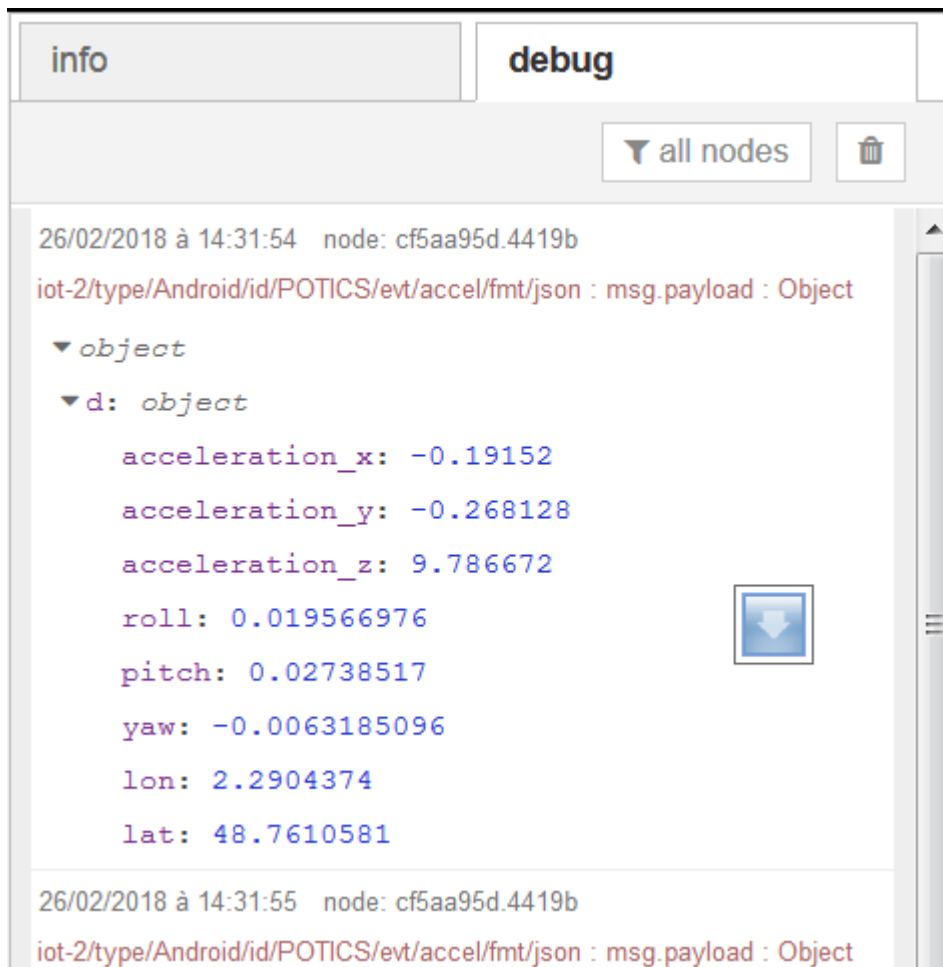
QoS
0

Name
IBM IoT

Use the Input Type property to configure this node to receive Events sent by IoT Devices, Commands sent to IoT Devices, Status Messages referring to IoT Devices, or Status Messages referring to IoT Applications

Check the info tab, to get more information about each of the fields

7. 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)
8. Then select the debug tab to visualize your data:



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 debug tab
- The “Device State Event” use the logical interface format.
- Add a function node with the content :  
 return {payload:msg.payload.state.AccelerationX};  
 (use the format you have defined in the logical interface)

### Edit function node

▼ node properties

Name

AccelerationX

Function

```
1 return {payload:msg.payload.state.AccelerationX};
```

- Add a switch function and see screen capture to configure it. don't forget to use numbers.
- Nb: Screen capture are not in the last version of the switch note.

### Edit switch node

Properties


Name

Property ▼ msg. payload

≡	<	▼ <sup>0</sup> / <sub>9</sub> -0.1	→ 1	✕
≡	>	▼ <sup>0</sup> / <sub>9</sub> 0.1	→ 2	✕
≡	otherwise ▼		→ 3	✕


**Appearance**

Label ☒ Show


Icon 


Port labels


Inputs

1.  

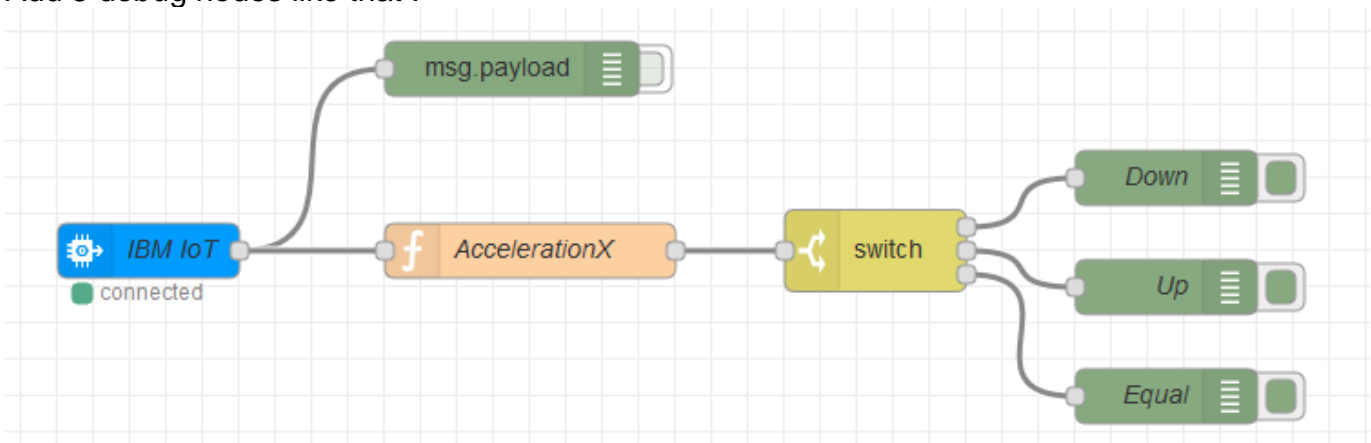
Outputs

1.  

2.  

3.  

- 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

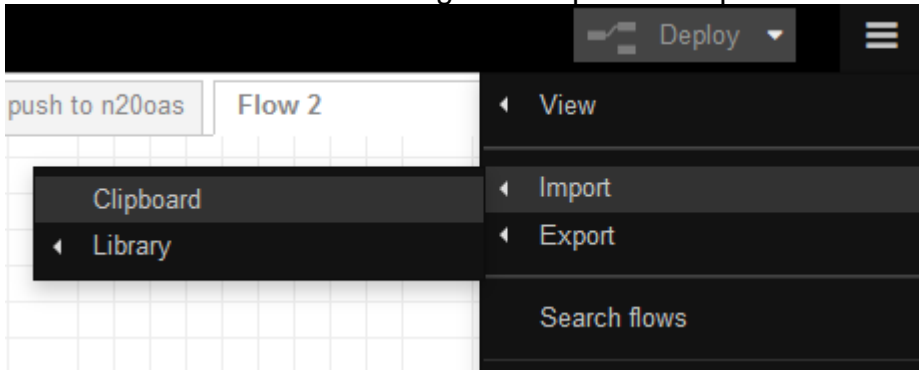
#### What you will do next :

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

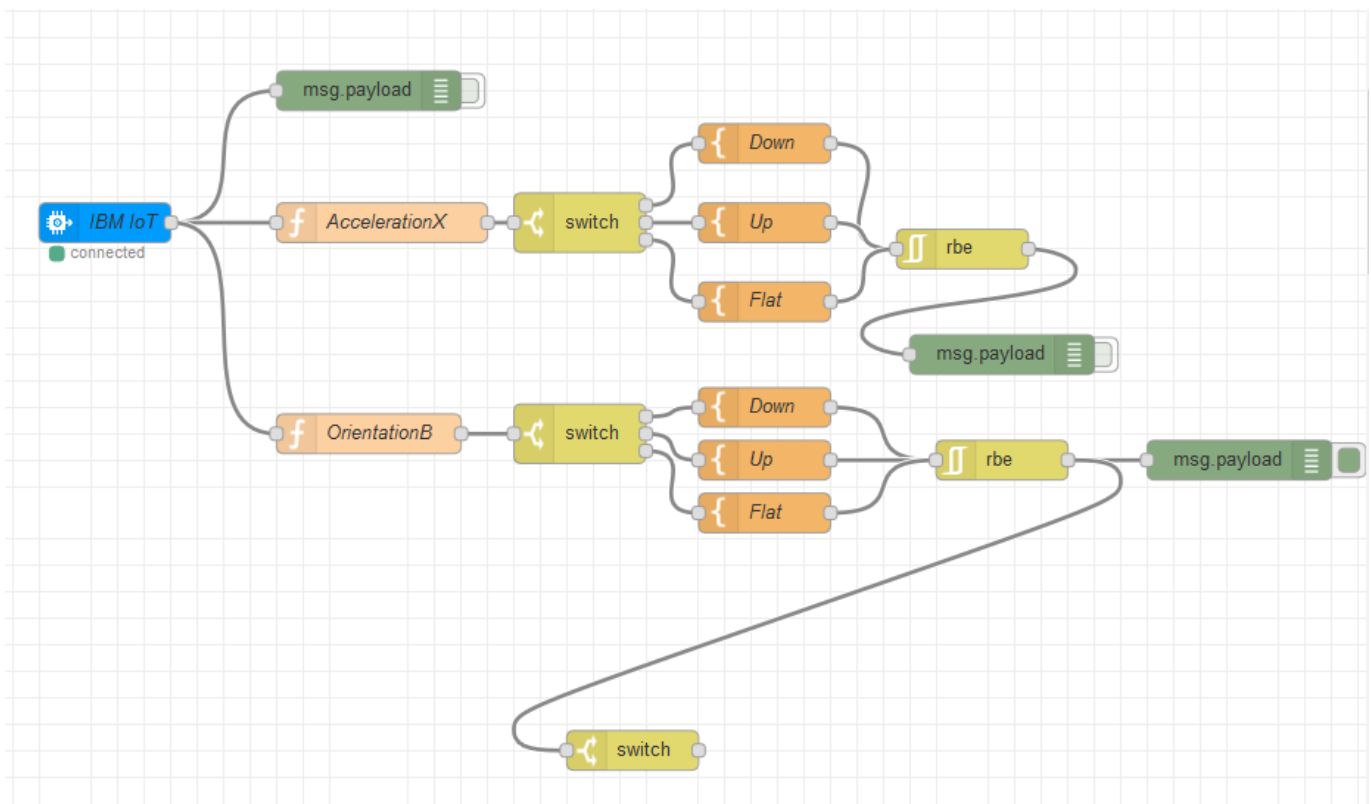
## B. Smartphone interaction with an application: Create an alert

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

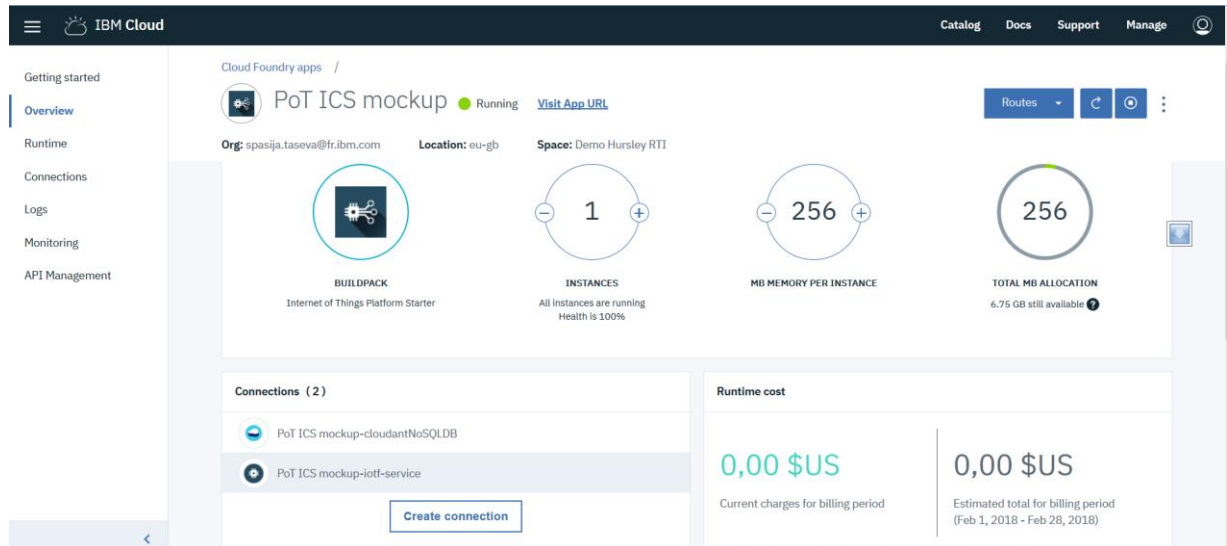


- Run your web interface
- See the result in the debug tab

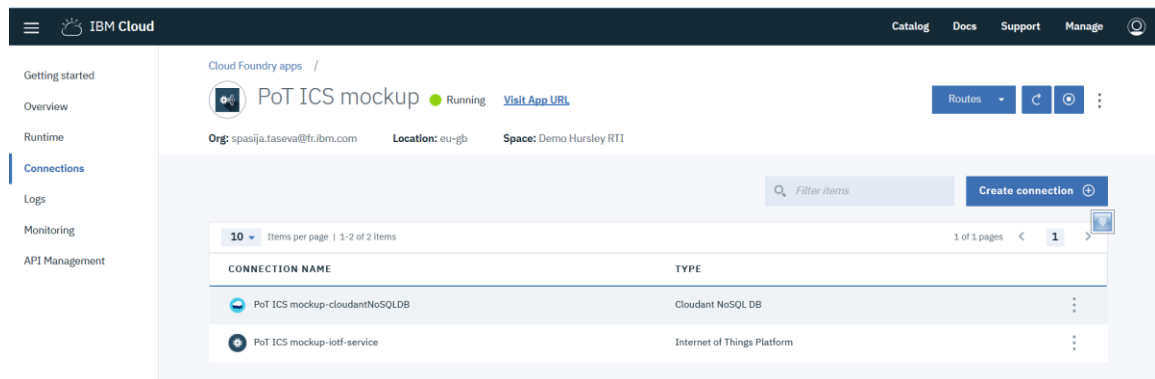
## 2. Use the Watson Text To Speech service to alert when the phone is flat

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

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



- Click on Catalog:



- And search for “Text to Speech”



The screenshot shows the IBM Cloud Catalog interface. At the top, there's a navigation bar with 'Catalog', 'Docs', 'Support', and 'Manage'. Below this, a search bar contains 'text to speech'. On the left, a sidebar lists categories like 'Infrastructure' and 'Platform (1)'. The main content area shows the 'Text to Speech' service card, which includes a description: 'Synthesizes natural-sounding speech from text.' and a 'Create' button.

- Click on Create

This screenshot shows the configuration page for the 'Text to Speech' service. It includes a 'Service name' field set to 'Text to Speech-rr'. Below this, there are three dropdown menus: 'Choose a region/location to deploy in:' (set to 'United Kingdom'), 'Choose an organization:' (set to 'spasija.taseva@fr.ibm.com'), and 'Choose a space:' (set to 'Demo Hursley RTI'). A 'Create' button is at the bottom right. On the left, there's a 'View Docs' link and a table with service details: AUTHOR (IBM), PUBLISHED (02/22/2018), TYPE (Service), and LOCATION (Sydney, Germany, United Kingdom, US South).

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

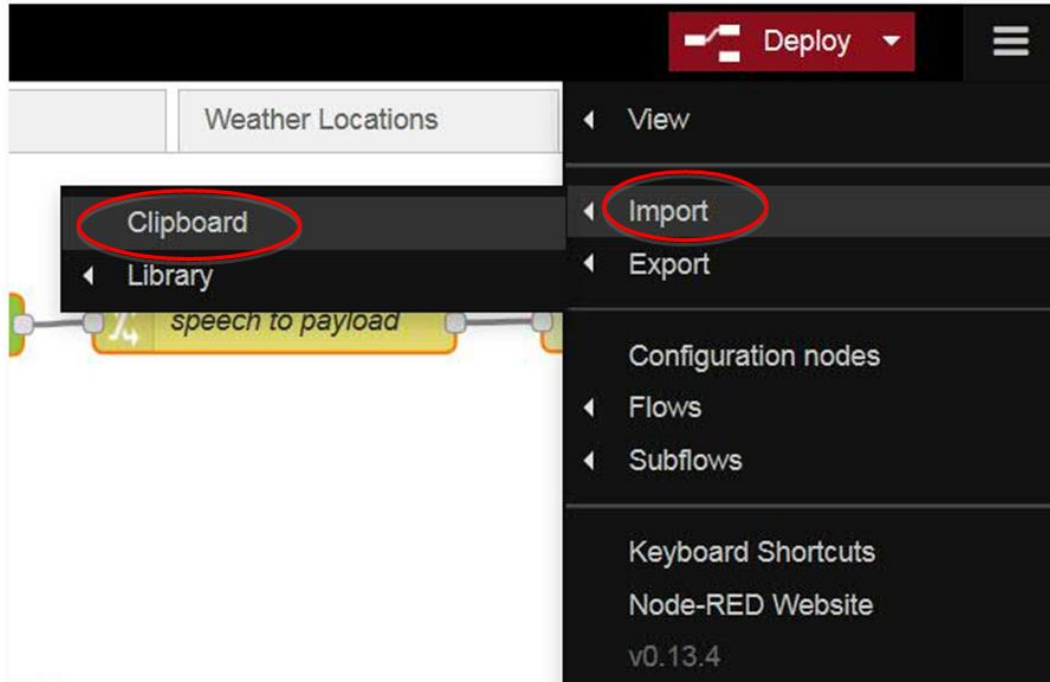
This screenshot shows the 'Getting started' tutorial page for the 'Text to Speech' service. The page title is 'Text to Speech-rr' and it shows the location, organization, and space. The tutorial is dated 'Last Updated: 2017-10-20'. It explains that the service converts written text to natural-sounding speech. The 'Before you begin' section lists steps: 1. Create an instance of the service. 2. If you're seeing this, you created your service instance. Now get your credentials. 3. Create a project from a service. 4. Go to the Watson Developer Console Services page. 5. Select Text to Speech, click Add Services, and either sign up for a free IBM Cloud account or log in. 6. Type text-to-speech-tutorial as the project name and click Create Project. 7. Copy the credentials to authenticate to your service instance.

- 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 start the application again.

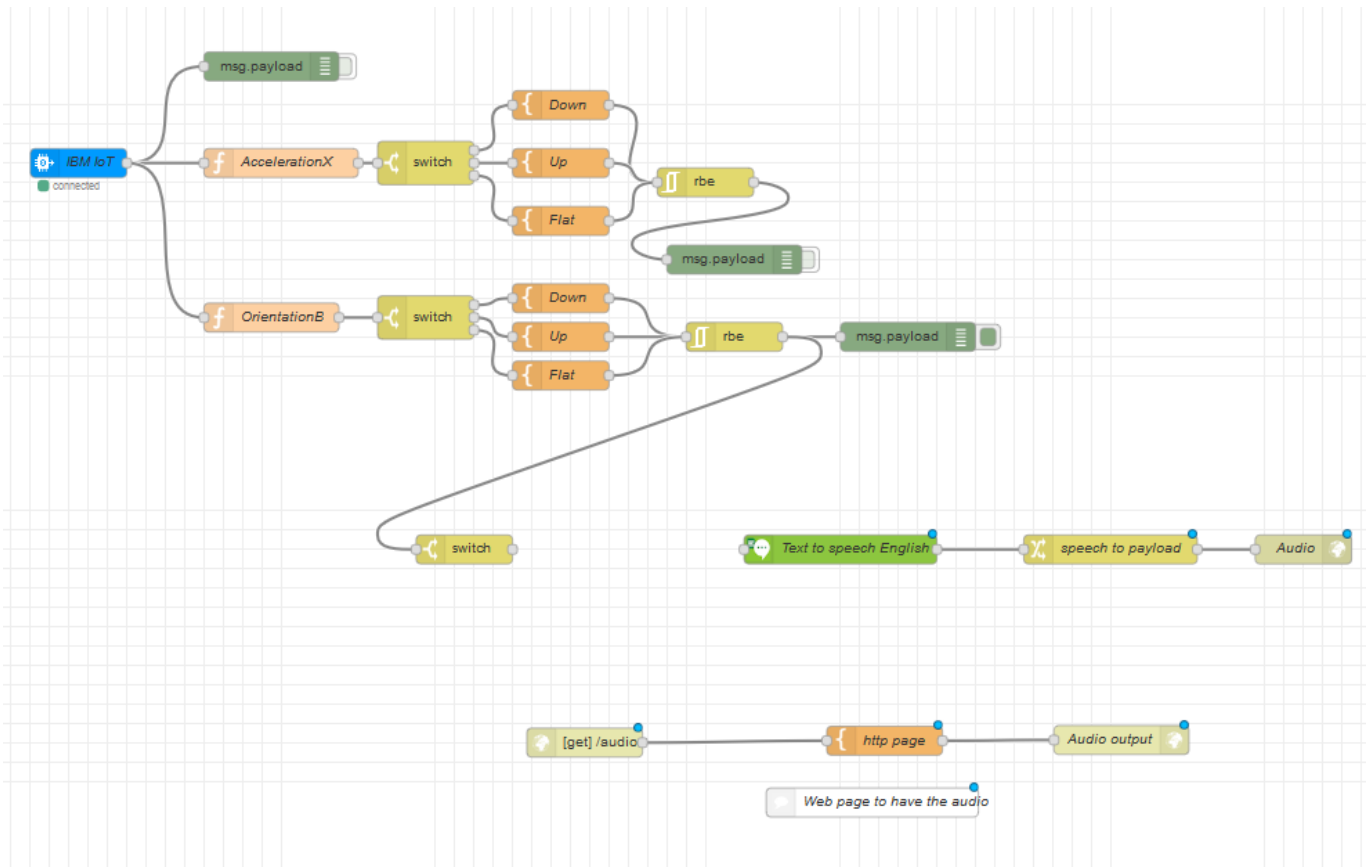
- Go to your NodeRed application
- Import



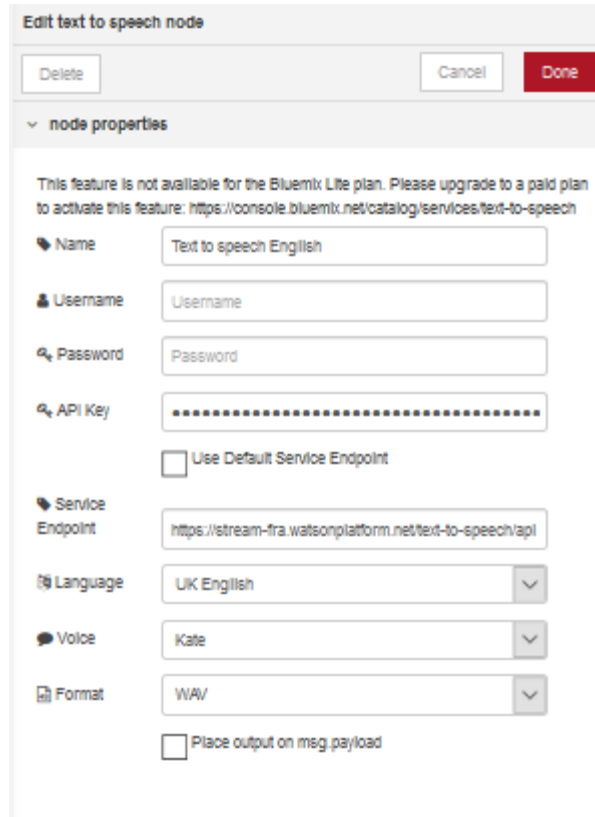
- Import in the current flow the content of the file named: **TextToSpeechNodered.txt**

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:



**Edit text to speech node**

Delete Cancel Done

▼ node properties

This feature is not available for the Bluemix Lite plan. Please upgrade to a paid plan to activate this feature: <https://console.bluemix.net/catalog/services/text-to-speech>

Name Text to speech English

Username Username

Password Password

API Key .....

☐ Use Default Service Endpoint

Service Endpoint <https://stream-fra.watsonplatform.net/text-to-speech/api>

Language UK English ▼

Voice Kate ▼

Format WAV ▼

☐ Place output on msg.payload

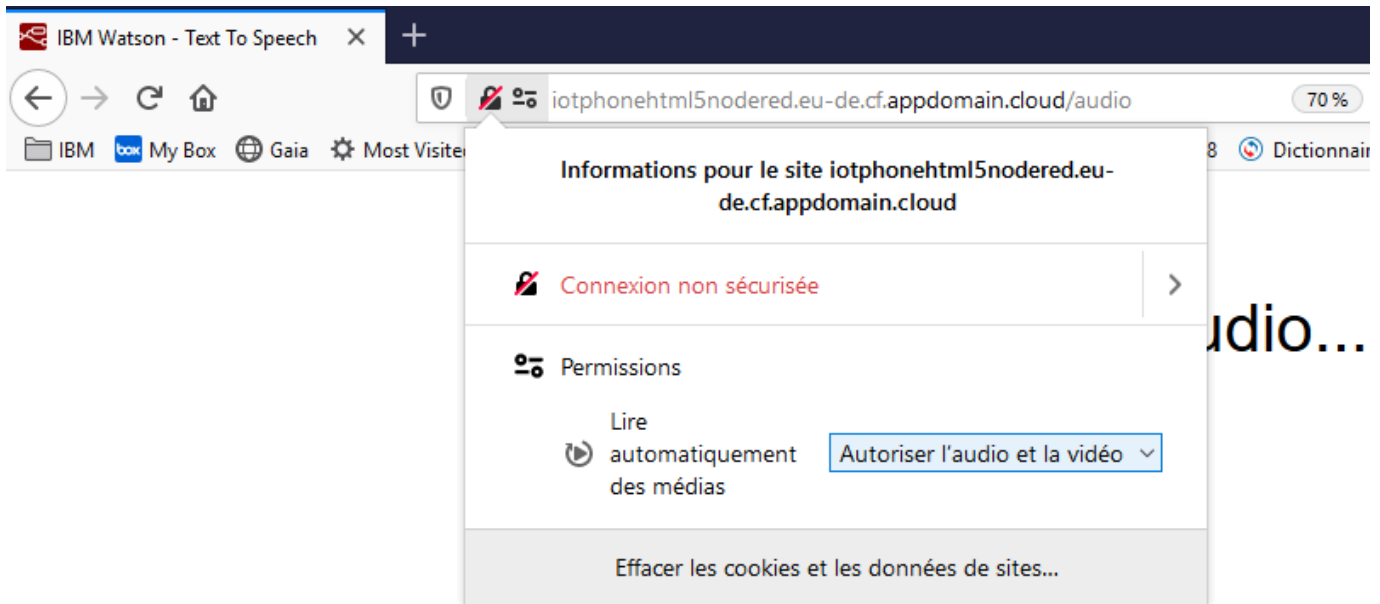
- For test you can add an Debug node after IOT input node.
- Click to **Deploy**.
- Browse [url from nodered for your applicationname.cloud] /audio

**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. Directly in IoT Platform : Create an alert when your smartphone falls down

Note : This chapter will be updated as soon as the graphical interface will be available in the free IoT platform plan.

### A. About rules and actions:

See

[https://www.ibm.com/support/knowledgecenter/SSQP8H/iot/platform/information\\_management/im\\_rules.html](https://www.ibm.com/support/knowledgecenter/SSQP8H/iot/platform/information_management/im_rules.html) for details.

With IBM Watson™ IoT Platform Service you can set up rules that trigger when an event that is received by Watson IoT Platform Service causes a change to the device state.

Embedded rules are condition-based decision points that match real-time device data with predefined threshold values or other property data to trigger the rule if a condition is met.

With embedded rules, you specify the conditions that trigger a rule. You can then set up an action in response to the trigger, for example, sending an alert to a user's device and an email to an administrator, when the temperature of your device spikes.

### B. Understanding rules

Rules are associated with a [logical interface](#) and are written against the [device state](#). You can associate one or more rules with a logical interface.

Rules get evaluated when a device event that is received by Watson IoT Platform could affect the device state that is defined by a logical interface. This is called "event-driven evaluation".

Each rule must have a **name** and a **condition** parameter defined. The **condition** parameter must conform to the following criteria:

- The parameter must be an expression.
- The parameter must evaluate to a Boolean value of **true** or **false**. If the condition expression evaluates to **true** an MQTT message is published on an MQTT topic.

### C. Configuring notifications

You can configure rules to set a notification strategy in which you define conditions determining the timing and frequency of notifications. This enables you to ensure that notifications are sent to alert users when data falls outside of normal ranges and conditions, while controlling the number of alerts that are triggered for a rule over a period of time. For example, you can set it up so that when the temperature of the device spikes for a specified amount of time, an alert is sent to the dashboard on a user's device, and an email is sent to the administrator.

Notifications can be sent every time the conditions specified in a rule are met, only the first time they are met, when they are met a certain number of times, or when they are met for a certain duration.

The following table shows the notification conditions that you can specify when you configure rules:

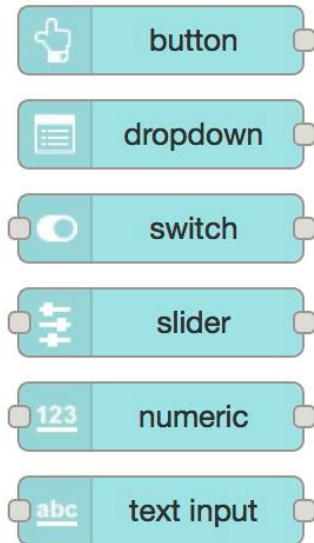
Condition name	Condition value
Default	The default notification condition triggers a notification event every time the rule is validated as true.
Becomes True	The rule is configured to trigger a notification the first time the rule is validated as true. Subsequent messages that are also validated as true do not trigger notifications. This is reset when the conditions are no longer met so that the rule is triggered the next time they are.
X in Y	This condition triggers a notification event if the rule is validated as true <i>x</i> number of times in <i>y</i> amount of time, measured in <i>days/hours/minutes/custom value</i> . For example, the rule can be configured to trigger only once if the conditions are met four times in 30 minutes. The device sends one new message every five minutes. At noon, the temperature initially exceeds 90 degrees, which meets the condition. The conditional trigger counter is started, but the rule is not yet triggered. After 15 minutes and three more messages that indicate that the temperature exceeded 90 degrees were received, the rule is triggered. The rule is then not triggered for another 15 minutes regardless of the temperature.
Persist	The rule is configured to trigger a notification event if the rule persists as true for the specified amount of time, such as 60 seconds or two days. The time interval starts when the conditions are initially met.

### D. Node-RED Dashboarding capability:

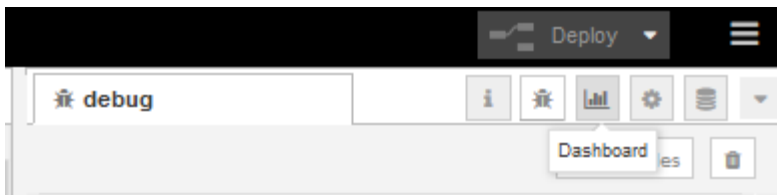
We need it to be able to display alerts in the next step

1. Note dashboard nodes on the palette:

## ▼ dashboard



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.

## E. Configure a rule:

In the current version, the graphical interface is not available to create rules.

We will use Node-RED to create a rule.

### 1. Set a AccelerationMax value per device

You might want to use variables so that you do not need to hard-code values in your rule condition expressions and can set a different threshold value for each device instance.

We will add a metadata called AccelerationMax to the iotphone device type, then set the value for your device.

- Go to Devices => device types
- Open iotphone device type.
- Select Device information section.
- Edit Metadata
- Add :

```
{
  "AccelerationMax": 10
}
```

- Save

Each new device will inherit of this new metadata property.

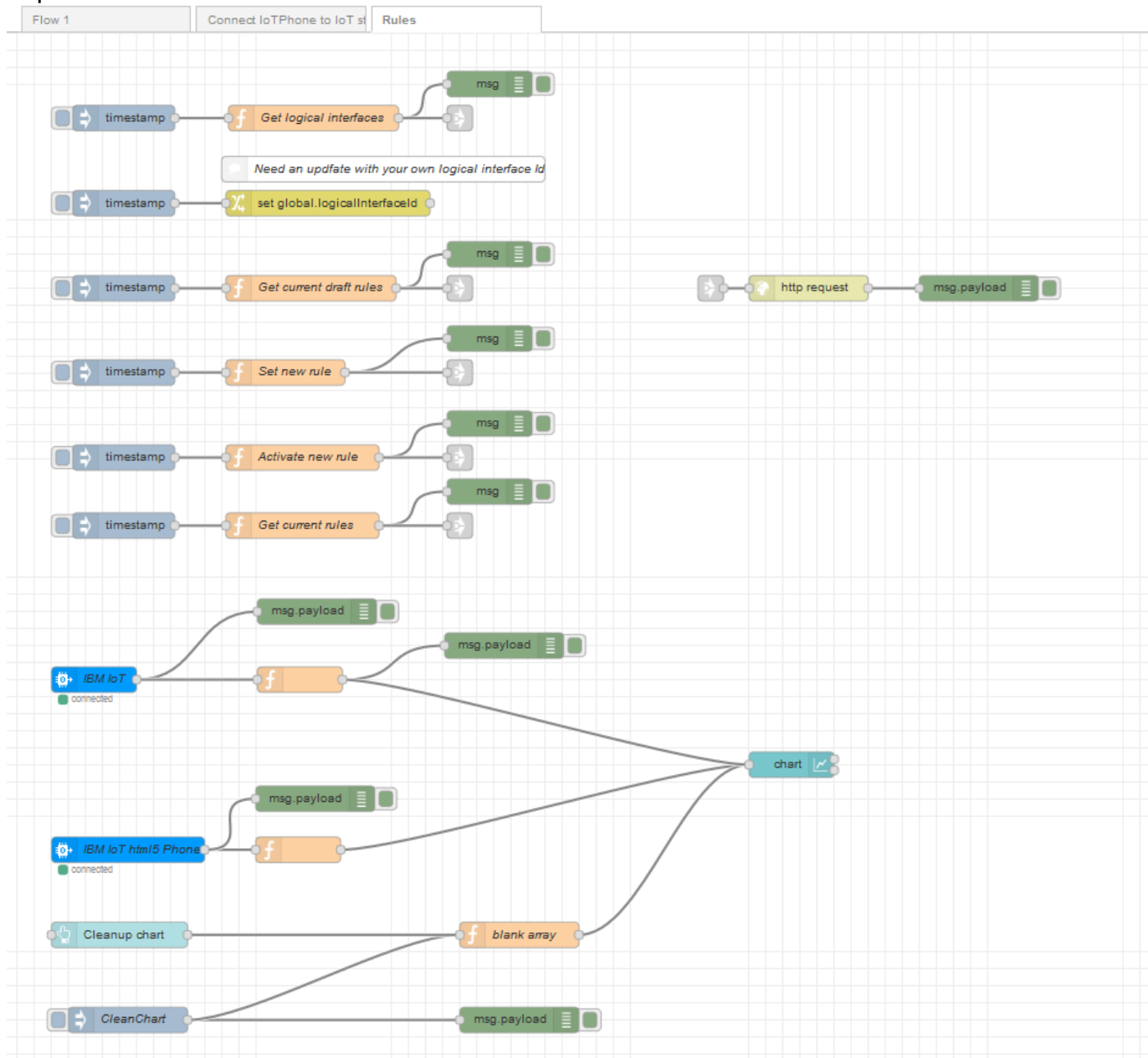
- For your existing device, do the same: edit metadata property and add the **AccelerationMax**. Adapt the value to your phone.

## 2. Generate an API Key

- Go to Apps
- Select **Generate API Key**
- Description: ApiKeyNodeRedGet
- Role : Operations Application
- Generate Key
- Keep/backup the result in a file.

## 3. Create a new tab in NodeRed to create the IoT rule and display results

Import the “NodeRedRulesDefinition and dashboard in NodeRed.txt” in NodeRed





## 4. Update the http request node with your API username and token

**Edit http request node**

Delete Cancel Done

▼ node properties

Method - set by msg.method -

URL http://

☐ Enable secure (SSL/TLS) connection

☒ Use basic authentication

Username a-ff0h6y-hxf19xfvbr

Password .....

Return a parsed JSON object

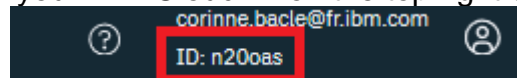
Name Name

## 5. Update the logical interface Id

Update function flows like “Get logical interface” http link with your own organization Id.

Retrieve the Org ID of your Watson IOT Platform instance:

The Org ID is indicated below your IBM Cloud ID on the top right of the browser:



For each JS Function node, please retrieve your Org ID and update the URL accordingly:

msg.url=<https://YOURORGID.internetofthings.ibmcloud.com/api/v0002/logicalinterfaces>";

=> à mettre à jour avec une variable

Get the logical interface id : Click on the first timestamp in the left, get the id of your iotphone logical interface in the debug tab :

15/01/2019 à 16:06:08 node: 641d10ed.5a457

msg: Object

```
{_msgid: "fc0505aa.0a4f98", topic: "",
payload: 1547564767574, url:
"https://ff0h6y.internetofthing...", method:
"GET" }
```

15/01/2019 à 16:06:09 node: 2517ccd4.5bf84c

msg.payload: Object

▼ object

▼ results: array[1]

▼ 0: object

```
version: "active"
created: "2019-01-14T15:28:56Z"
createdBy: "corinne.bacle@fr.ibm.com"
updated: "2019-01-14T17:08:03Z"
updatedBy: "a-ff0h6y-hxf19xfvbr"
name: "iotphone_LI"
description: ""
id: "5c3caab8c5d1250028de0b6b"
schemaId: "5c3caab7c5d1250028de0b6a"
```

▶ refs: object

▼ meta: object

total\_rows: 1

Update the “set global.logicalInterfaceId” node with your id:

Edit change node

Delete

Cancel

Done

▼ node properties

Name

Name

Rules

Set

▼ global. logicalInterfaceId

to

▼ a z 5c3caab8c5d1250028de0b6b

## 6. Set the rule and check the result

Click on the timestamp nodes to :

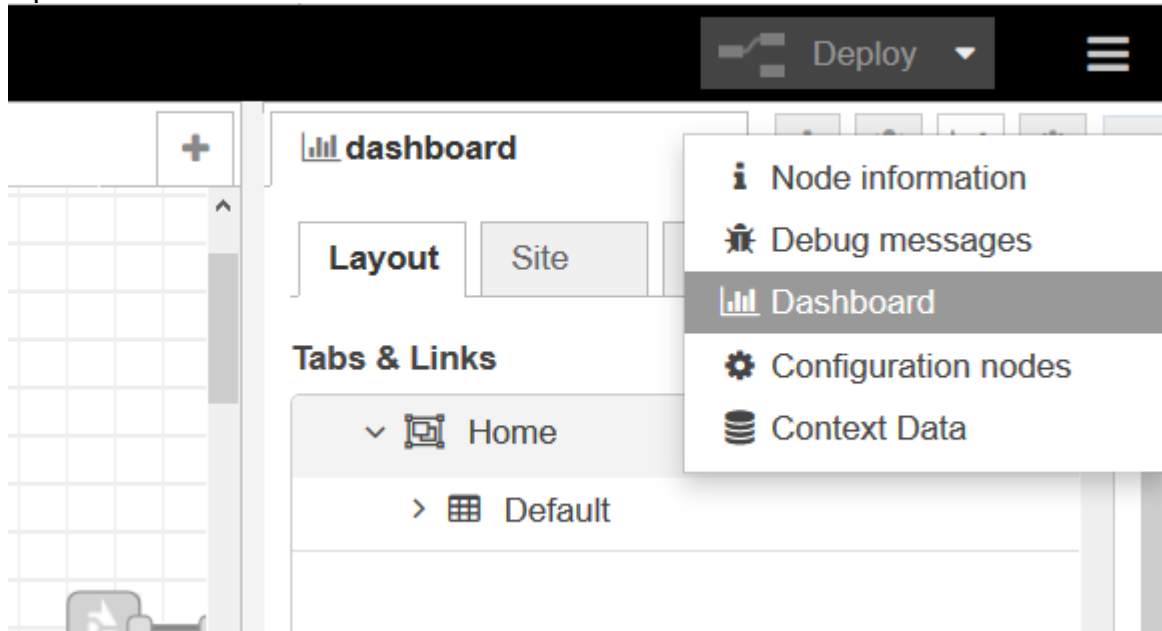
- Get current draft rules and check there is no rules
- Set new rule (you can edit the content to see how it's configure)
- Activate new rule
- Get current rules and check the rule is active

Note that the notification strategy is every-time.

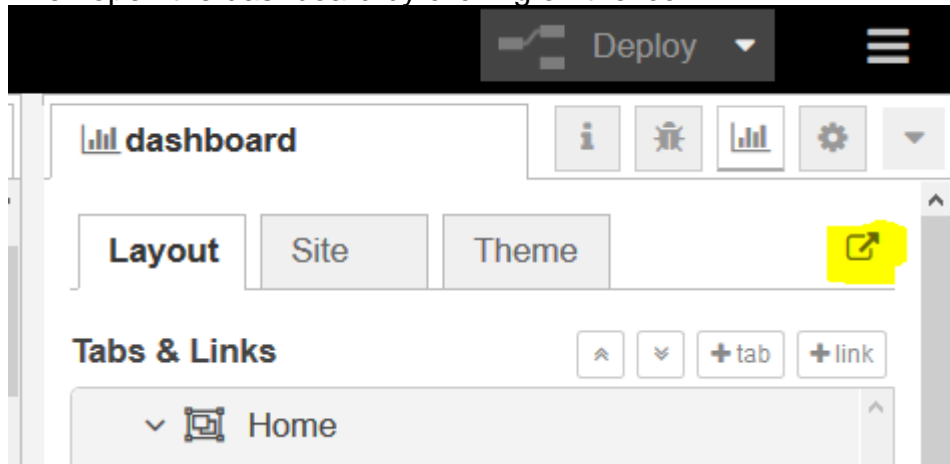
7. Check the result in the new dashboard

Update the phone id in the « input IoT » node if needed.

Open the dashboard tab:



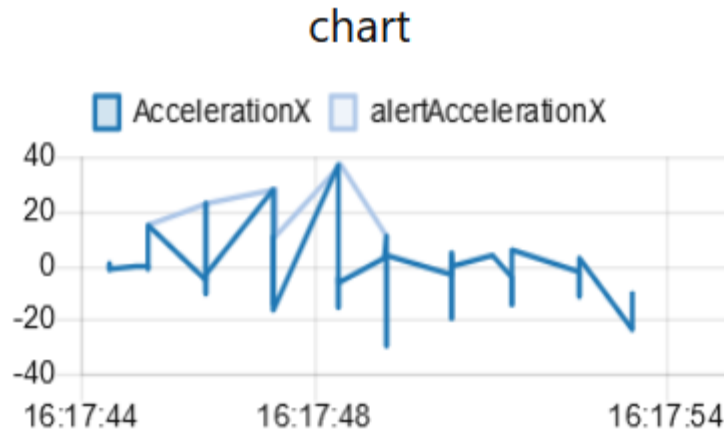
Then open the dashboard by clicking on the icon :



Connect your phone and see the result:

## Default

## CLEANUP CHART



## VI. Store your data in a Cloudant NoSQL DB

Note that the storage in Cloudant is available by default in the IoT Connection Service solution.

### A. Solution embedded in the IoT Platform: (To Be Updated)

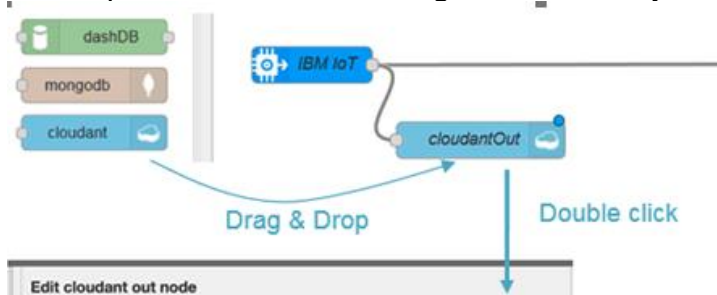
This is available only through API, not done during this workshop. See the documentation in the “Extensions” left menu.

In this case all events coming from devices are stored directly in the Cloudant database, you can also decide which events are stored depending on device type and event types.

In the next step we will see how to control what is stored using Node-RED.

### B. Storage using Node-RED:

- Go back to your Node Red browser window:
  - Drag & drop a Cloudant **Out** node (only one entry point on the left)
  - Set it up as shown below and give a name to your database.



### Edit cloudant out node

Service

pot160317-cloudantNoSQLDB

Database

myiotdata

Operation

insert

☐ Only store msg.payload object?

Name

Name

Click Done and then Deploy your flow.

You can access the DB user interface from Bluemix:

- Go back to your application overview in Bluemix (browser tab), open the “Cloudant NoSQL DB” service

The screenshot shows the IBM Bluemix Cloud Foundry Apps overview page for application **pot160317**. The application status is **Running**. The left sidebar contains navigation links: Dashboard, Getting started, Overview (selected), Runtime, Connections, Logs, and Monitoring. The main content area displays the following information:

- BUILDPACK:** SDK for Node.js™
- INSTANCES:** All instances are running, Health is 100%
- MB MEMORY PER INSTANCE:** (Not specified)
- TOTAL MB ALLOCATION:** 5.5 GB still available
- Connections (3):**
  - pot160317-cloudantNoSQLDB
  - pot160317-iotf-service
  - Text to Speech-I4
- Runtime cost:**
  - 0,15 \$US** (Current charges for billing period)
  - 0,15 \$US** (Estimated total for billing period 03/01-03/31)

Buttons for **Connect new** and **Connect existing** are visible at the bottom of the connections list.

and launch the interface:

← Pot160317

pot160317-cloudantNoSQLDB

Manage Service Credentials Plan Connections

Cloudant NoSQL DB

LAUNCH

3. Check the number of records is increasing (refresh the browser):


Name	Size	# of Docs	Update Seq	Actions
myiotdata	78.6 KB	151	1	[Icons]
nodered	20.8 KB	4	3	[Icons]

You are now storing data from your device inside a Cloudant DB.  
You can see your database content by clicking on the database name.

{ } JSON Table ☐ Include Docs Document ID Options API

id "49af1ea4b95f34299a28f0965d48d6b6"

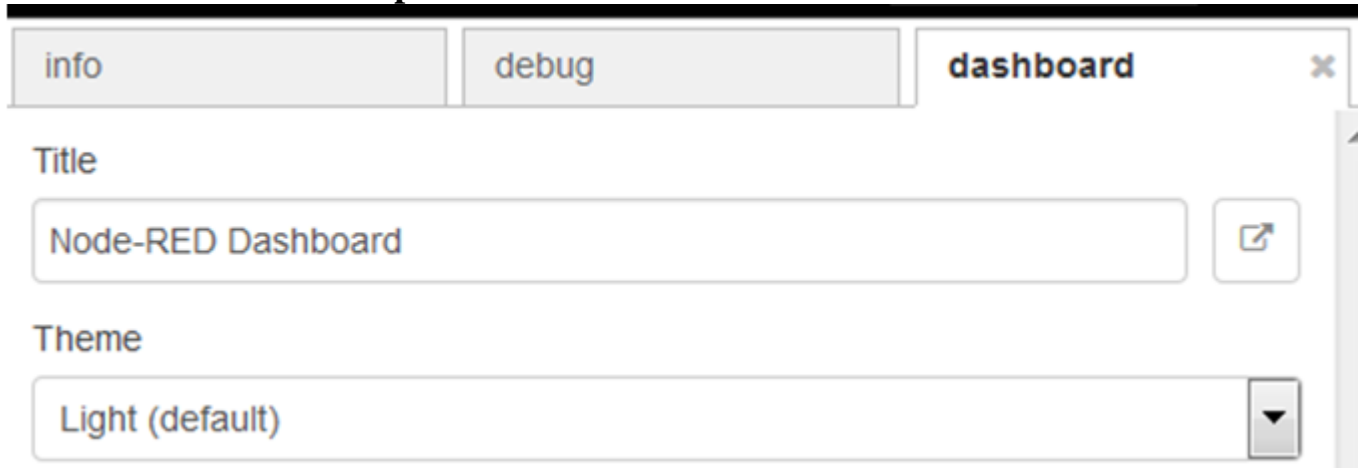
```
{
  "_id": "49af1ea4b95f34299a28f0965d48d6b6",
  "_rev": "1-218a3fc3201d2e459eb98cc5f27c1820",
  "value": {
    "rev": "1-218a3fc3201d2e459eb98cc5f27c1820"
  },
  "key": "49af1ea4b95f34299a28f0965d48d6b6"
}
```

 Save Changes [Cancel](#)

```
1 {
2   "_id": "49af1ea4b95f34299a28f0965d48d6b6",
3   "_rev": "1-218a3fc3201d2e459eb98cc5f27c1820",
4   "topic": "iot-2/type/Android/id/myPhone/evt/accel/fmt/json",
5   "payload": {
6     "d": {
7       "acceleration_x": -0.37828064,
8       "acceleration_y": 0.39263916,
9       "acceleration_z": 9.718063,
10      "roll": 0.038905874,
11      "pitch": -0.040350538,
12      "yaw": 0.000091552734,
13      "lon": 7.1858891,
14      "lat": 43.6697837
15    }
16  },
17  "deviceId": "myPhone",
18  "deviceType": "Android",
19  "eventType": "accel",
20  "format": "json",
21  "msgid": "97d2321e.682dd"
22 }
```

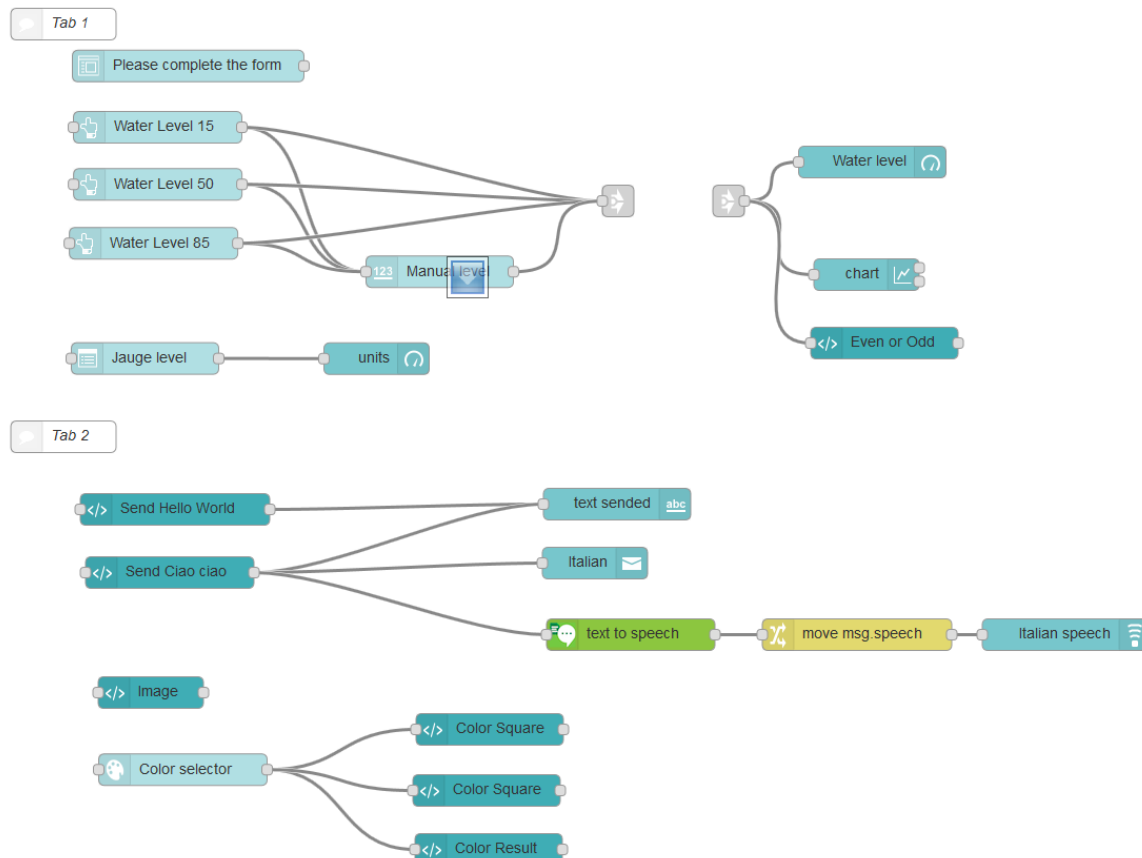
## VII. Create a dashboard application in Node-RED

### A. Create a simple Node-RED Dashboard:



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

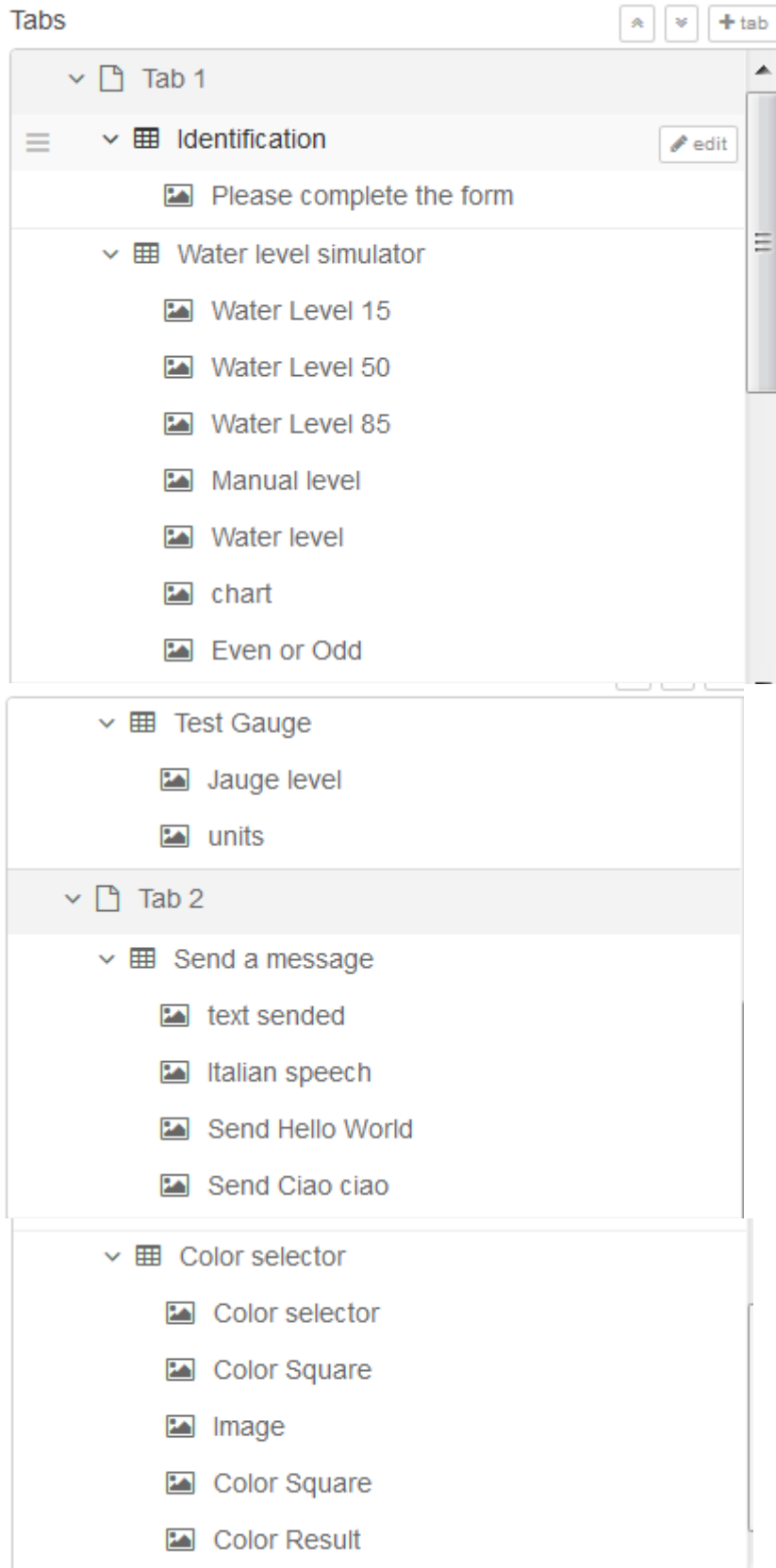
#### 1. Result in Node-RED:



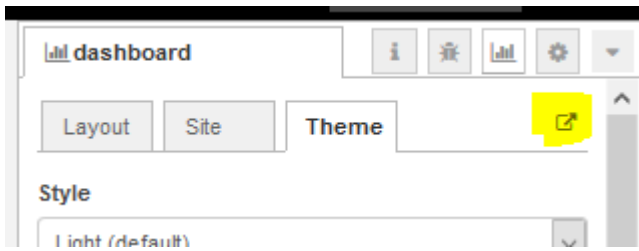
Configure the TextToSpeech with the API key. Keep it in Italian.



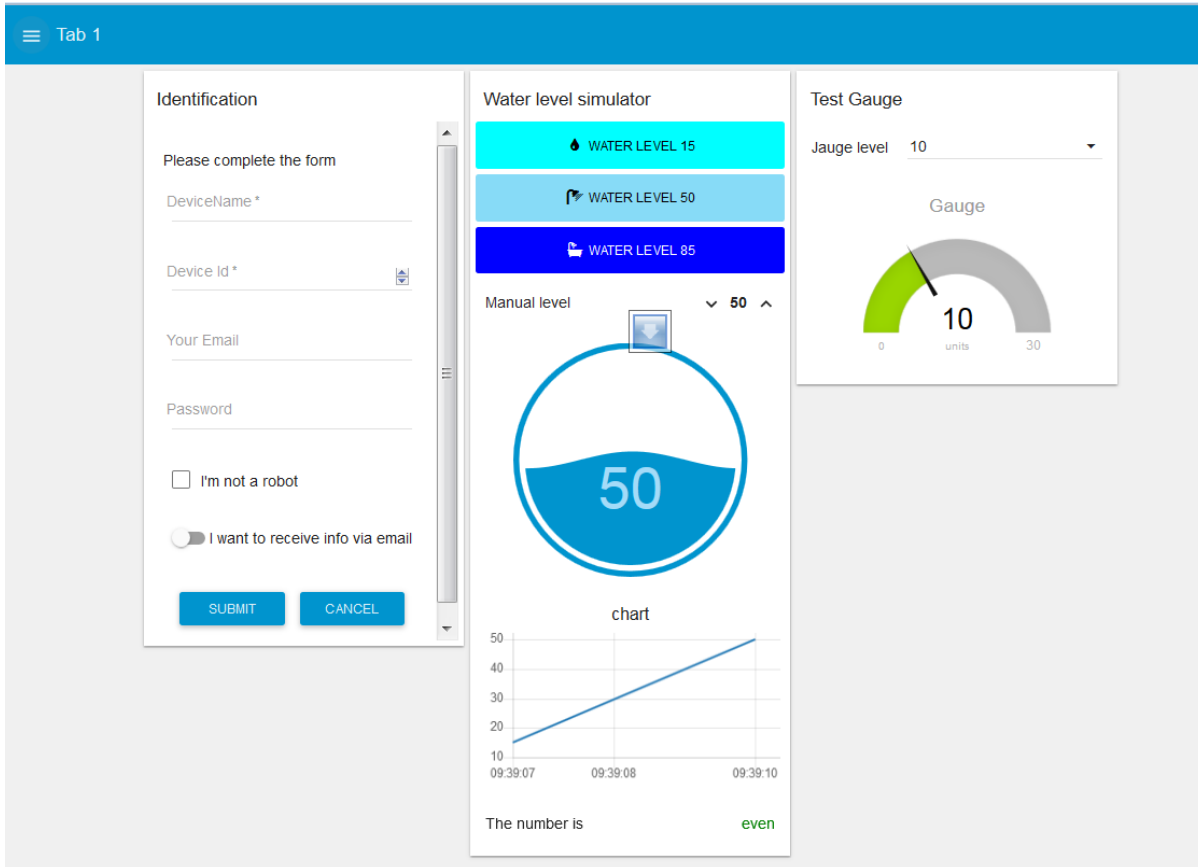
## 2. Configure the dashboard:



## 3. Open the dashboard :



## 4. Result in the dashboard:



Tab 2

Send a message

text sended

Ciao ciao

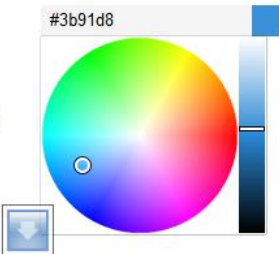
CLICK ME TO SEND A HELLO WORLD


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

50

chart

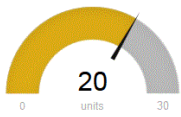


Test Gauge

Gauge level

20

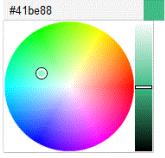
Gauge



Color selector

#41be98

Colour picket



Dark Theme

Water level simulator

WATER LEVEL 15

WATER LEVEL 50


WATER LEVEL 85

Manual level

50

50

chart

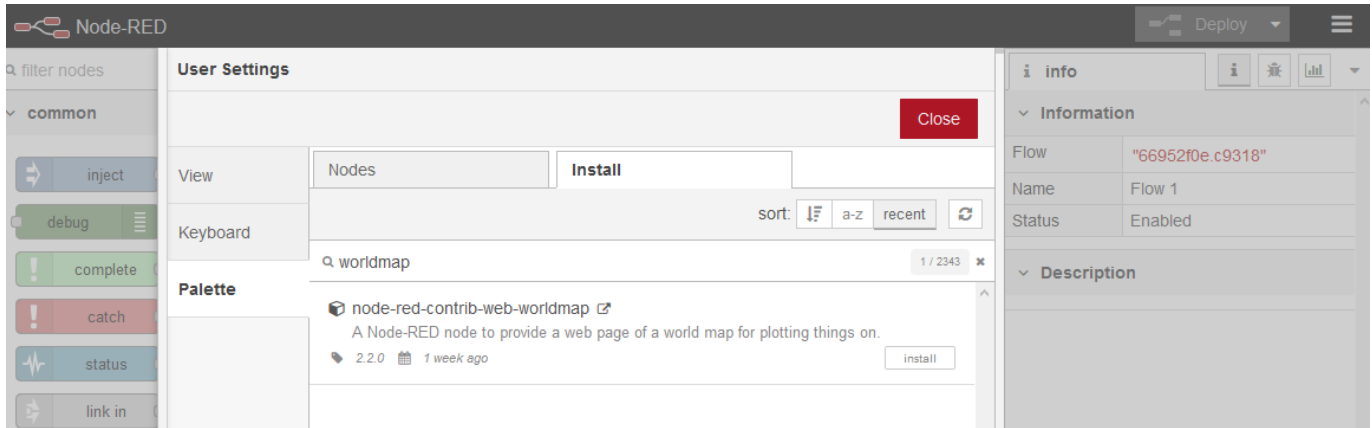


Light Theme

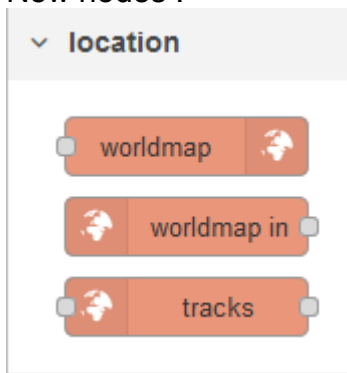
Page | 43

## B. Test worldmap

Add the worldmap palette:



New nodes :



<https://<YourAppName>.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>

## C. 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
}
```


Update the buildHtml node with your googlemap API key:




### Edit template node


Delete

Cancel


Done


 Properties




 Name

buildHtml





 Property


▼ msg. template

 Template


Syntax Highlight: mustache

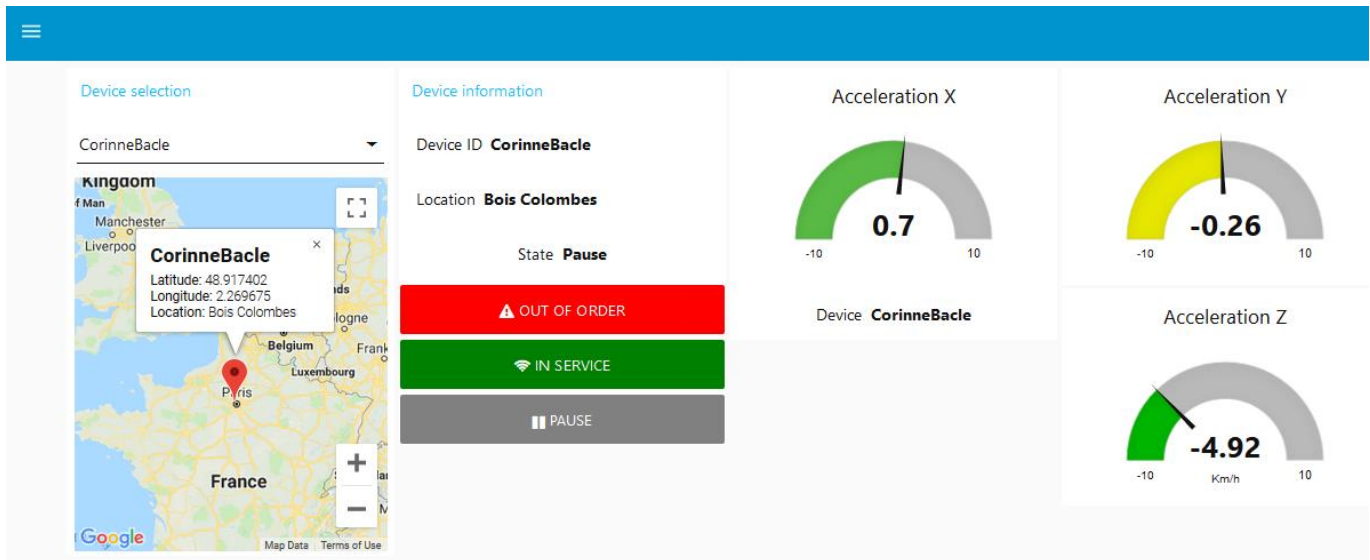


```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5       #map {
6         height: 500px;
7         width: 100%;
8       }
9     </style>
10  </head>
11  <body>
12    <div id="map"></div>
13    <script>
14      {{{payload.js}}}
15    </script>
16    <script async defer
17      src="https://maps.googleapis.com/maps/api/js?ke
18    </script>
19  </body>
20 </html>
```

 Format

Mustache template





#### D. Use metadata 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
<a href="#">POTIoTCS</a>	Ici			0
<a href="#">ThePhone</a>	Tour Eiffel	48.858093	2.294694	1

REFRESH

CLEAR



## VIII. Annex 1 : Secure your Node-RED workspace:

[http://\[yourappname\].mybluemix.net/#password-protecting](http://[yourappname].mybluemix.net/#password-protecting)

Select: Runtime and Environment variables

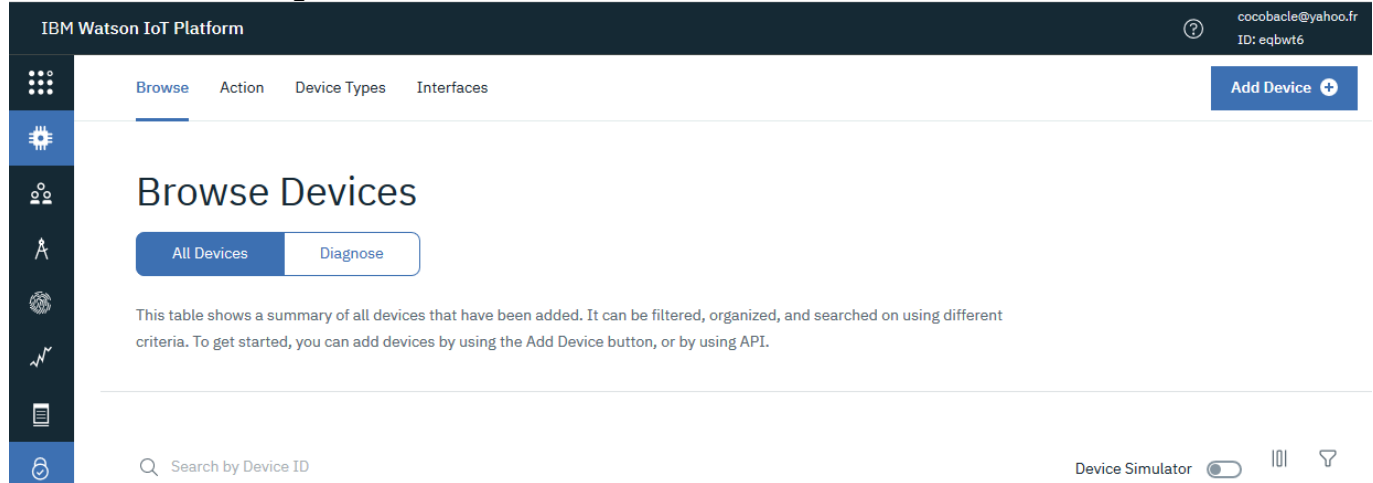
The environment variables you can set are:

- `NODE_RED_USERNAME` - the username to secure the editor with
- `NODE_RED_PASSWORD` - the password to secure the editor with
- `NODE_RED_GUEST_USER` - set to `true` to allow anonymous users to have read-only access to the editor

## IX. Annex 2 : Defining a simulated device using Watson IoT Platform

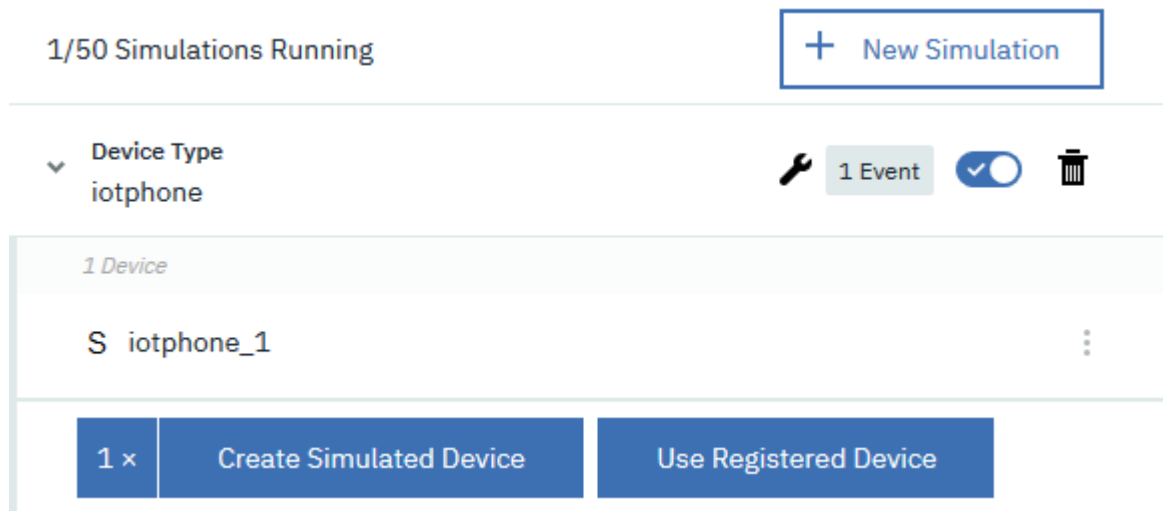
### A. Activate device simulator:

In the IoT Platform, go to the devices tab, then click on Device Simulator



### B. Create a simulated device:

- Click on the “0 simulations running” on the bottom of the page.
- Click on Create simulation
- Select a device type
- Change the default event if you want
- Click on save
- Click on “Create Simulated Device” or “Use Registeres Device”
- Then activate the simulation :

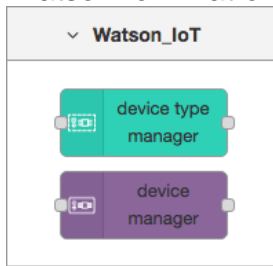


Check the result on Node-RED.  
Don't forget to disactivate the simulation.



## X. Annex 3: sample API usage : device management automation using Node-RED

There is a collection of Node-RED nodes to perform device and gateway operations using the Watson IoT Platform.



The “Device type manager” node is available for delete, retrieve single device type or update operations.

- GetAll
- Create
- Delete
- Get
- Update

The “Device manager” node lets you perform actions on devices that are connected to IBM Watson IoT Platform.

- GetAll
- Create
- Get
- Update
- Delete
- GetLoc
- UpdateLoc
- GetDm

See <https://flows.nodered.org/node/node-red-contrib-ibm-wiotp-device-ops> for details.

It's then possible to combine this nodes with the “csv” node to create multiple devices for example. But for Bulk create/delete, you can also pass the input as an array in msg.payload.

**Please import in a new flow the content of the file named: Manage device.txt**

Test the different options: Get all device types, a new device creation, device update, transform device data as a message.

Note: Update with iPhone device type if needed.

## XI. Annex 4 : RLAC usage (Ressource-Level Access Control)

You can use IBM Watson™ IoT Platform groups to grant members and API keys access to specific devices. After you create a group and add devices to it, you add members and API keys to the group and assign them roles within the group. The combination of roles and groups determines which devices the users and API keys can access, and the actions that they can perform on the devices.

You can manage groups by using the Watson IoT Platform dashboard user interface or by using the Watson IoT Platform access control APIs.

For more information about access control and groups, and for instructions on using Watson IoT Platform access control APIs to manage groups, see [Resource-level access control overview](#).

1. From your Watson IoT Platform dashboard (your bookmark), select **Access Management** in the left navigation bar.
2. Select the **Groups** tab and view the list of groups.
3. Click **Add Group**.
4. In the **Add Group** window, enter the group name and an optional description.
5. Click **Next**.
6. Click **Add Devices to Group**.
7. Select the devices that you want to add to the group and then click **Done**.
8. Click **Finish** to create the group. You can edit the group by add more devices or removing devices, and you can add more groups.

After you create a group and add devices to it, you add members and API keys to the group and assign them roles within the group. The combination of roles and groups determines which devices the users and API keys can access, and the actions that they can perform on the devices.

To test the result, using the RLAC API in the Node-RED interface:

Add other POT participants to your IoT platform (menu “Members”)

Add also more devices.

Create an API Key (menu APPS, Generate API Key with “Operations Application” role, save the details.

Import the RLAC flow from the file: RLAC.txt

In all http request node:

update the <https://<yourOrgId>.internetofthings.ibmcloud.com> link

Select “Use basic authentication”, use your previously generated API Key for username and password.

Click on Deploy.

Click on the “Click to have the group Id” inject node. Look at your group uid in the debug tab.

Update this group uid in the “Update the group uid in the topic here”

Update group uid at the end of the payload of the next input node and change

[Spasija.taseva@fr.ibm.com](mailto:Spasija.taseva@fr.ibm.com) with one of your users.

Click on Deploy.

Then click on the second injection node to verify your device in the group.

Click on the 3rd to assign a group to your user?

Click on the 4th to verify the result.

Ask the user to connect to your IoT platform to verify devices he has access to.