How to connect your HummBox Device to your IBM Bluemix. A five minutes installation to show your sensing data on real time dashboard

GreenCityZen: who are we:

GreenCityZen is an eco-startup that designs and sells technology solutions for the environmental measurement, addressed to its environmental industrial customers and the smart and sustainable cities. Green CityZen develops the Humm solution " IoT for the environment " an innovative solu-tion for the management and control of environmental sensors fleets, cost effective, scala-ble, and natively interoperable.

What is HummBox device:

The HummBox is a multiple sensor connected device provided by **GreenCityZen.**
HummBox provides advanced IoT monitoring solution of soil moisture and temperature. Its low cost, low power and easy scalability allow to address new fields of applications such as rainwater management performance in smart cities, irrigation precisions and decision for golfs, green areas and agriculture.

Requirements:

An IBM Bluemix Account.

HummBox Device ID

HummBox Device Token (HummboxGCZDevice_ID exp HummboxGCZ1BC02)

Skill level:

Beginner: This recipe is done for GreenCityZen customers.

Overview:

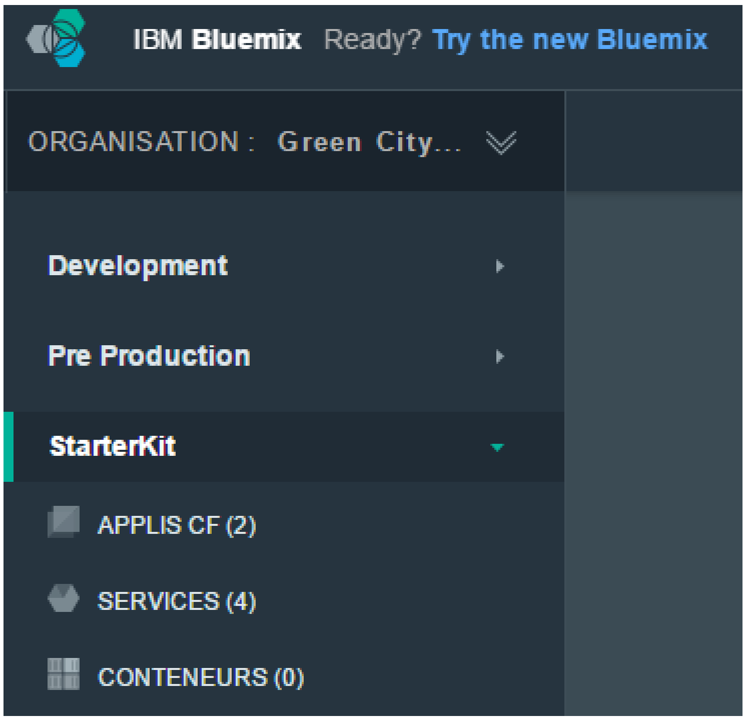
In order to manage and treat efficiently the data sent by your HummBox device, IBM Bluemix offers you a several service that can be used to explore and analyze your results.

This recipe will show how to create your own **boilerplate** « **Kit de démarrage Internet of Things Platform** » application.

Step 1: Create your own application and add the needed services:

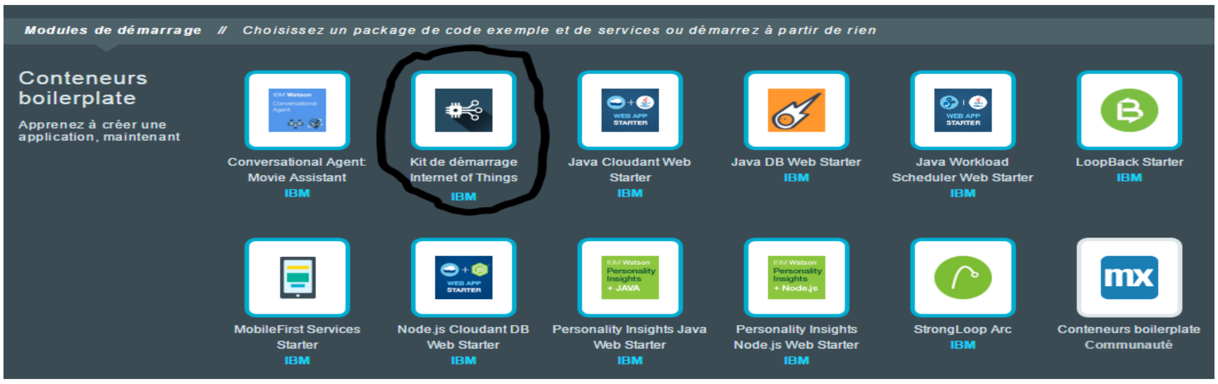
1-Login in your Bluemix account.

2-Go to **your** space, under **your** Organization.



3-From the dashboard, create a new application.

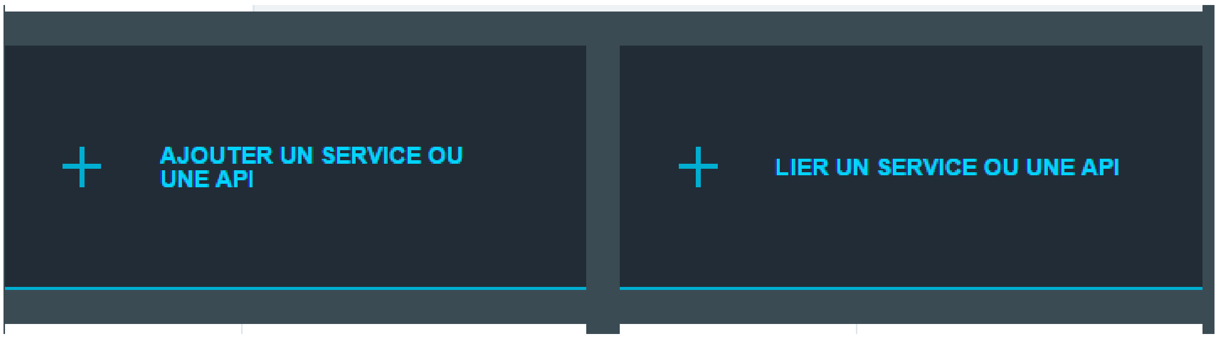
4-Select the **Internet of Things Foundation boilerplate**.


Modules de démarrage // Choisissez un package de code exemple et de services ou démarrez à partir de rien
Conteneurs boilerplate
Apprenez à créer une application, maintenant
Conversational Agent: Movie Assistant IBM
Kit de démarrage Internet of Things IBM
Java Cloudant Web Starter IBM
Java DB Web Starter IBM
Java Workload Scheduler Web Starter IBM
LoopBack Starter IBM
MobileFirst Services Starter IBM
Node.js Cloudant DB Web Starter IBM
Personality Insights Java Web Starter IBM
Personality Insights Node.js Web Starter IBM
StrongLoop Arc IBM
Conteneurs boilerplate Communauté

5-The next page shows details for the application. On the right side of the page, provide a name for the application.

After creating the application, the application dashboard will load and your application will be automatically started.

6- Access your application in your Dashboard then click on « **add new service or API**

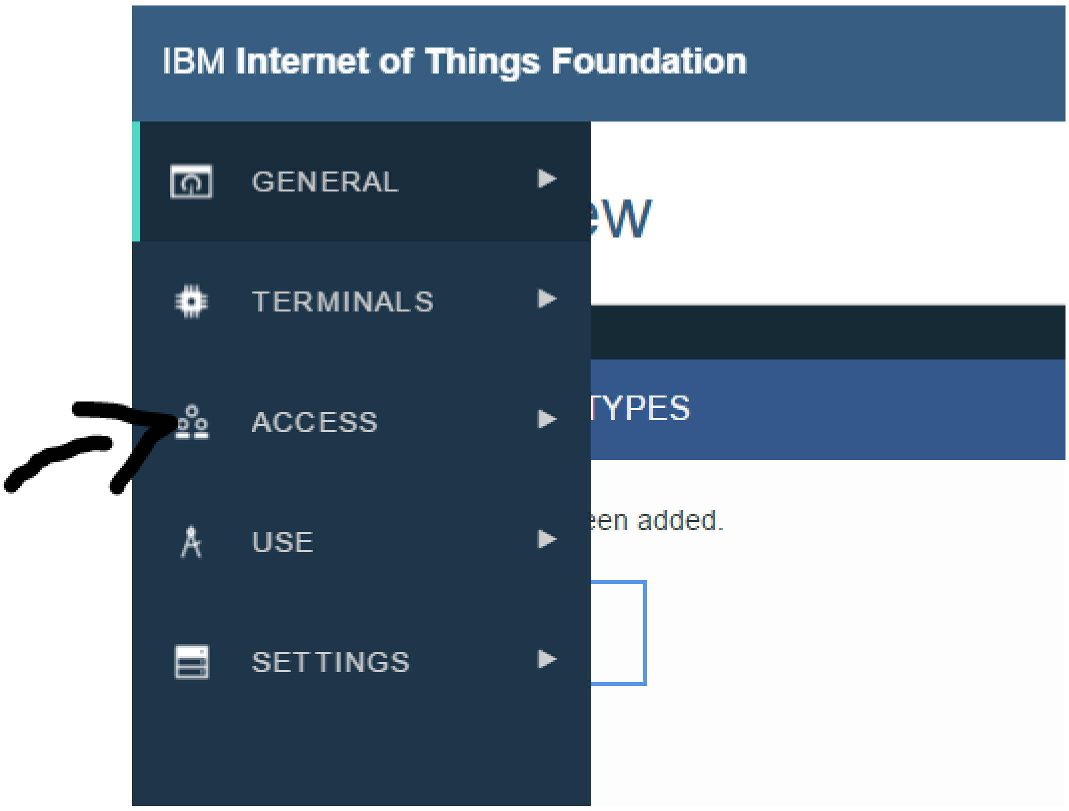


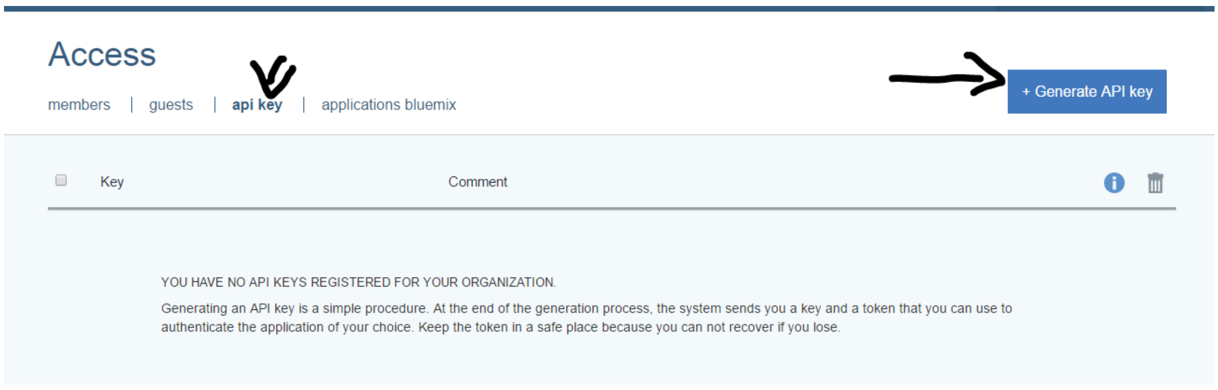
Step 2: Configure IOTF service:

1- Now click on **the Internet of Things** service in the application dashboard. You will get the configuration page for the service.

2- Click the '**launch**' button to open the Internet of **Things Foundation dashboard**.

3- Go to **Access** and press **API key**, and generate a new one

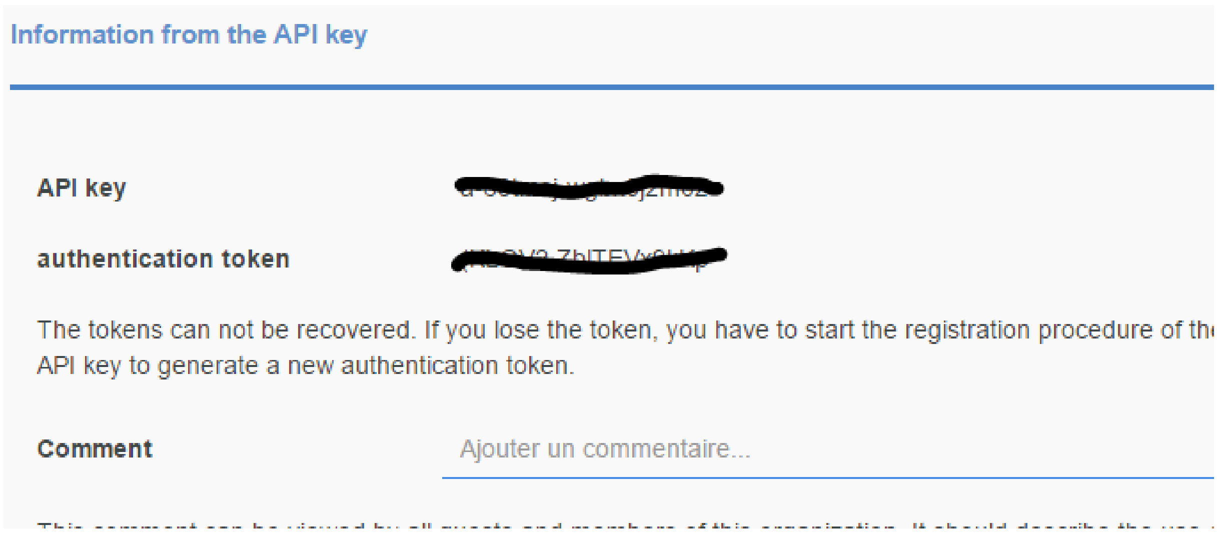




YOU HAVE NO API KEYS REGISTERED FOR YOUR ORGANIZATION.

Generating an API key is a simple procedure. At the end of the generation process, the system sends you a key and a token that you can use to authenticate the application of your choice. Keep the token in a safe place because you can not recover if you lose.

4- Then you will get an **API key** and an **authentication token**


Information from the API key

API key [redacted]

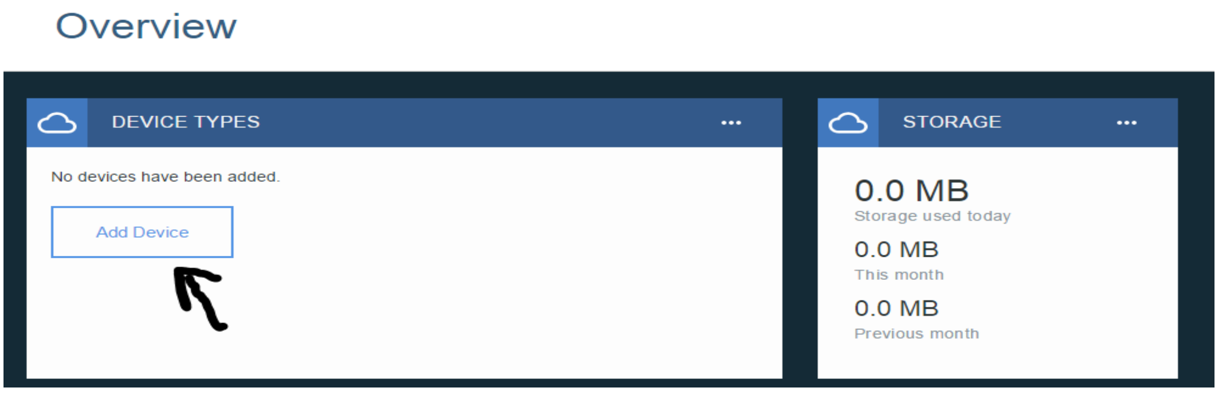
authentication token [redacted]

The tokens can not be recovered. If you lose the token, you have to start the registration procedure of the API key to generate a new authentication token.

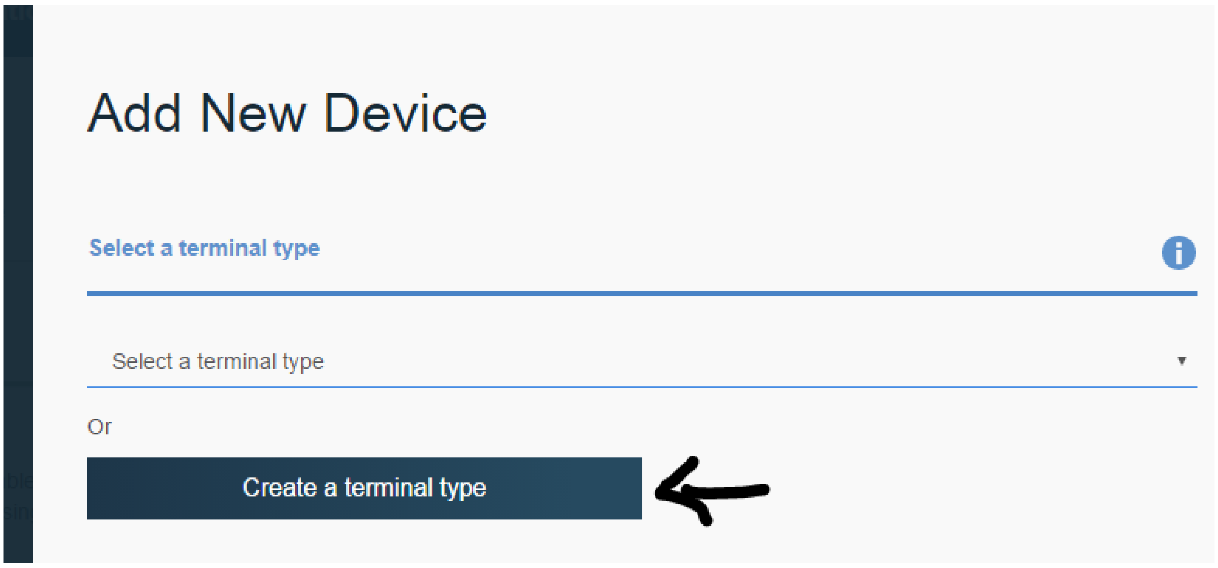
Comment

Be careful: you have to save the API Key and the authentication token somewhere because you won't have the access to them again.
Also you have to send to GCZ administers your organization ID.

5- Now you will have to declare a new device type :
5.1 -Go to **General**
5.2 -Click **Add a new Device**

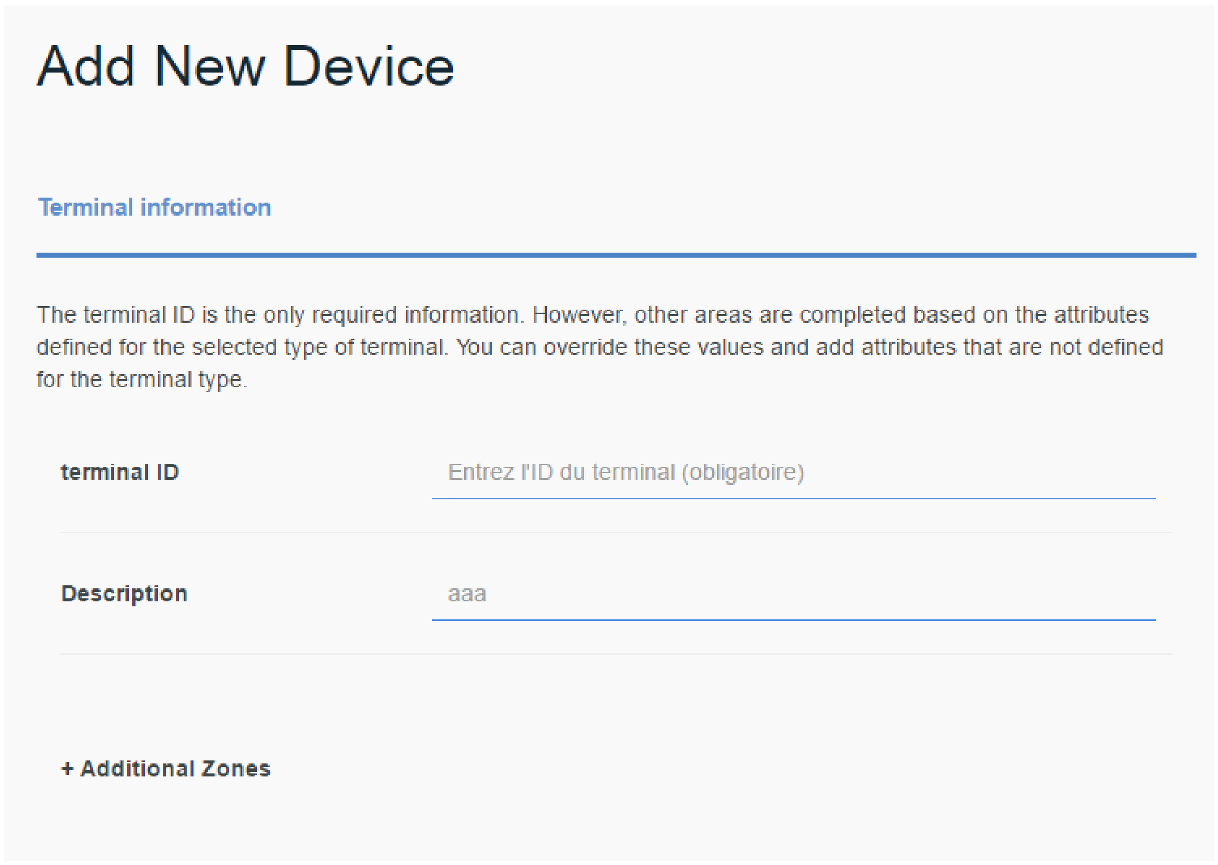


5.3 -Click **Create a Terminal Type**



5.4 -Enter the name of the type given by GreenCityZen (SmartSoil)

6- Now let's add your new device:
6.1- Go to **General**
6.2- Click on **Add Device** and press next
6.2- Choose the device type that you created and press next
6.3- Fill the terminal ID (given by GreenCityZen)



Add New Device

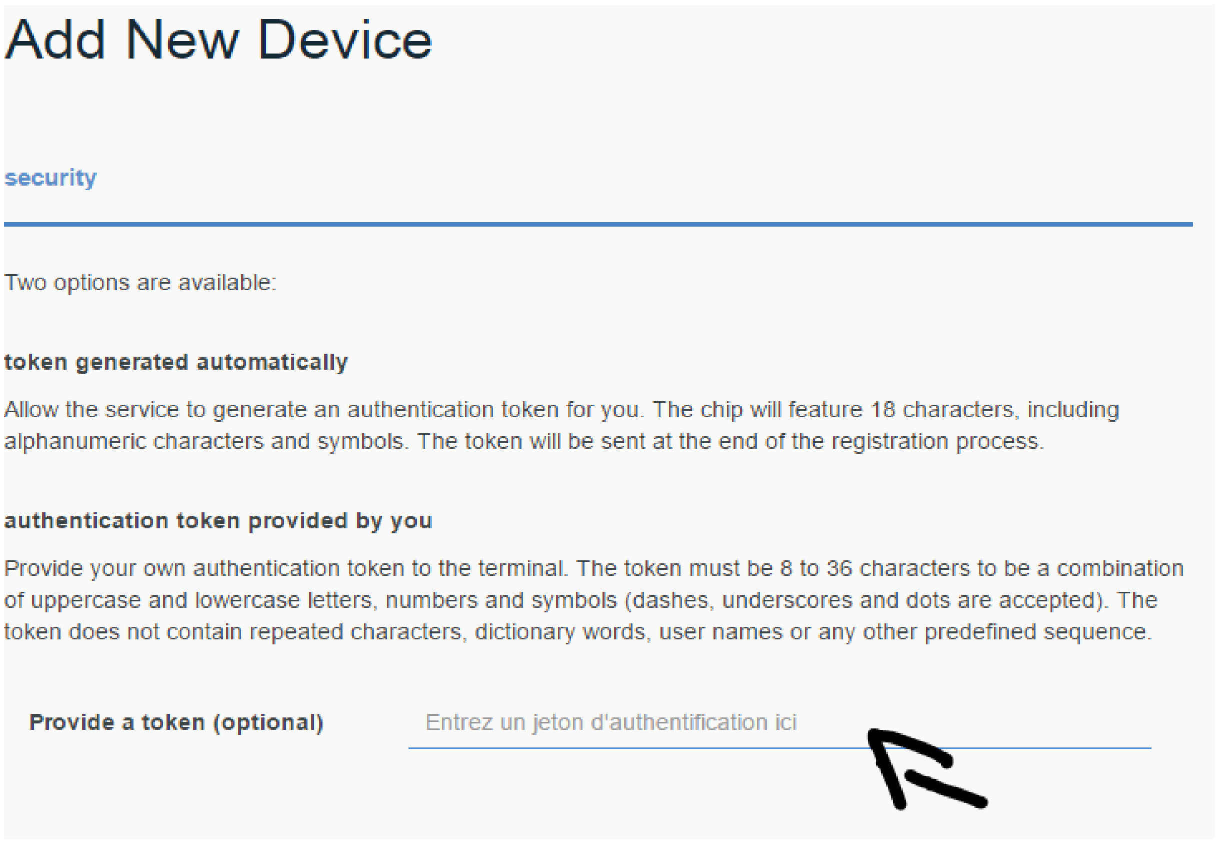
Terminal information

The terminal ID is the only required information. However, other areas are completed based on the attributes defined for the selected type of terminal. You can override these values and add attributes that are not defined for the terminal type.

terminal ID	Entrez l'ID du terminal (obligatoire)
Description	aaa

+ Additional Zones

6.4- Fill the device token and continue.



Add New Device

security

Two options are available:

token generated automatically

Allow the service to generate an authentication token for you. The chip will feature 18 characters, including alphanumeric characters and symbols. The token will be sent at the end of the registration process.

authentication token provided by you

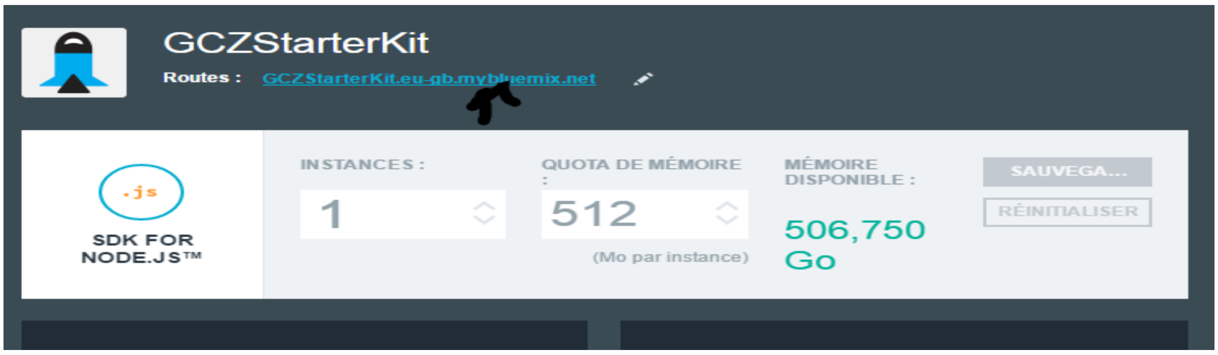
Provide your own authentication token to the terminal. The token must be 8 to 36 characters to be a combination of uppercase and lowercase letters, numbers and symbols (dashes, underscores and dots are accepted). The token does not contain repeated characters, dictionary words, user names or any other predefined sequence.

Provide a token (optional)	Entrez un jeton d'authentification ici

6.6- Finally you will get a summary of your terminal description.

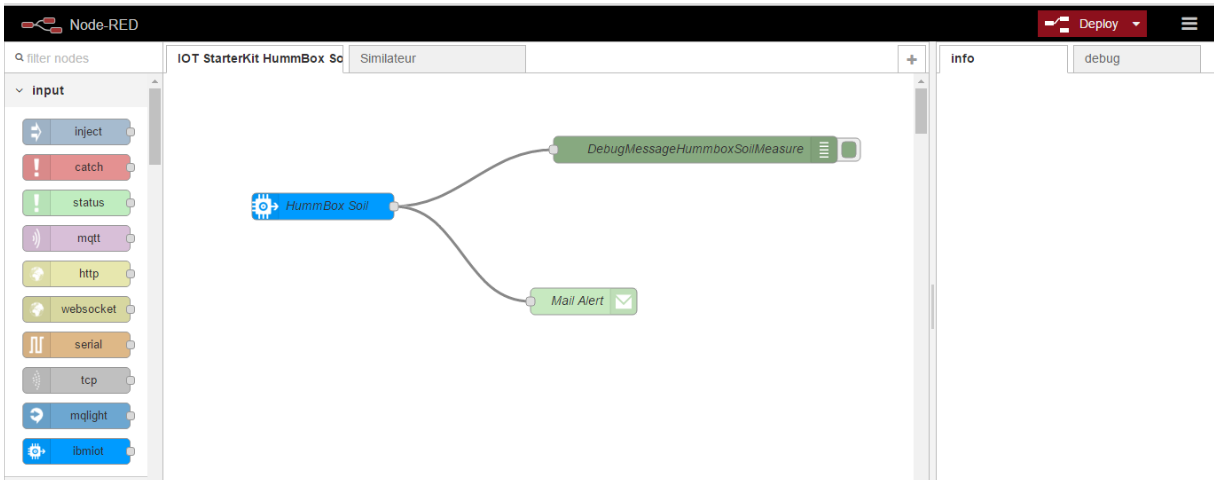
Step 4: Play with Node Red:

1- Click on the link in your Dashboard

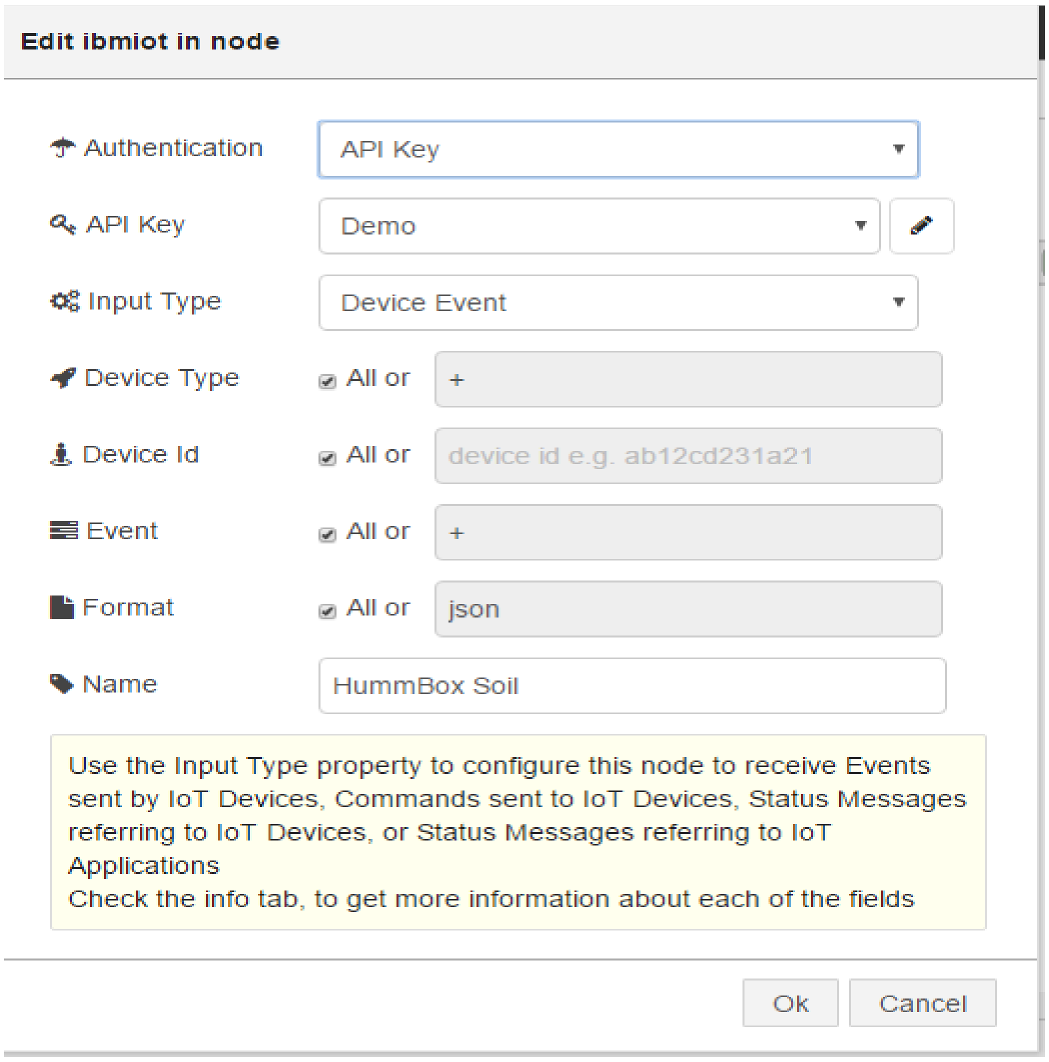


2- Click on ***Go to your Node-Red flow editor,*** and then you will get Node-Red dashboard.

3- We will propose you a basic configuration to edit your device data :



4- First step add an **ibmiot** input node onto the canvas. Double click on the node to edit the configuration. Set the following properties:



Edit ibmiot in node

Authentication	API Key
API Key	Demo
Input Type	Device Event
Device Type	✓ All or +
Device Id	✓ All or device id e.g. ab12cd231a21
Event	✓ All or +
Format	✓ All or json
Name	HummBox Soil

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

Authentication: API Key

Input Type: Device Event

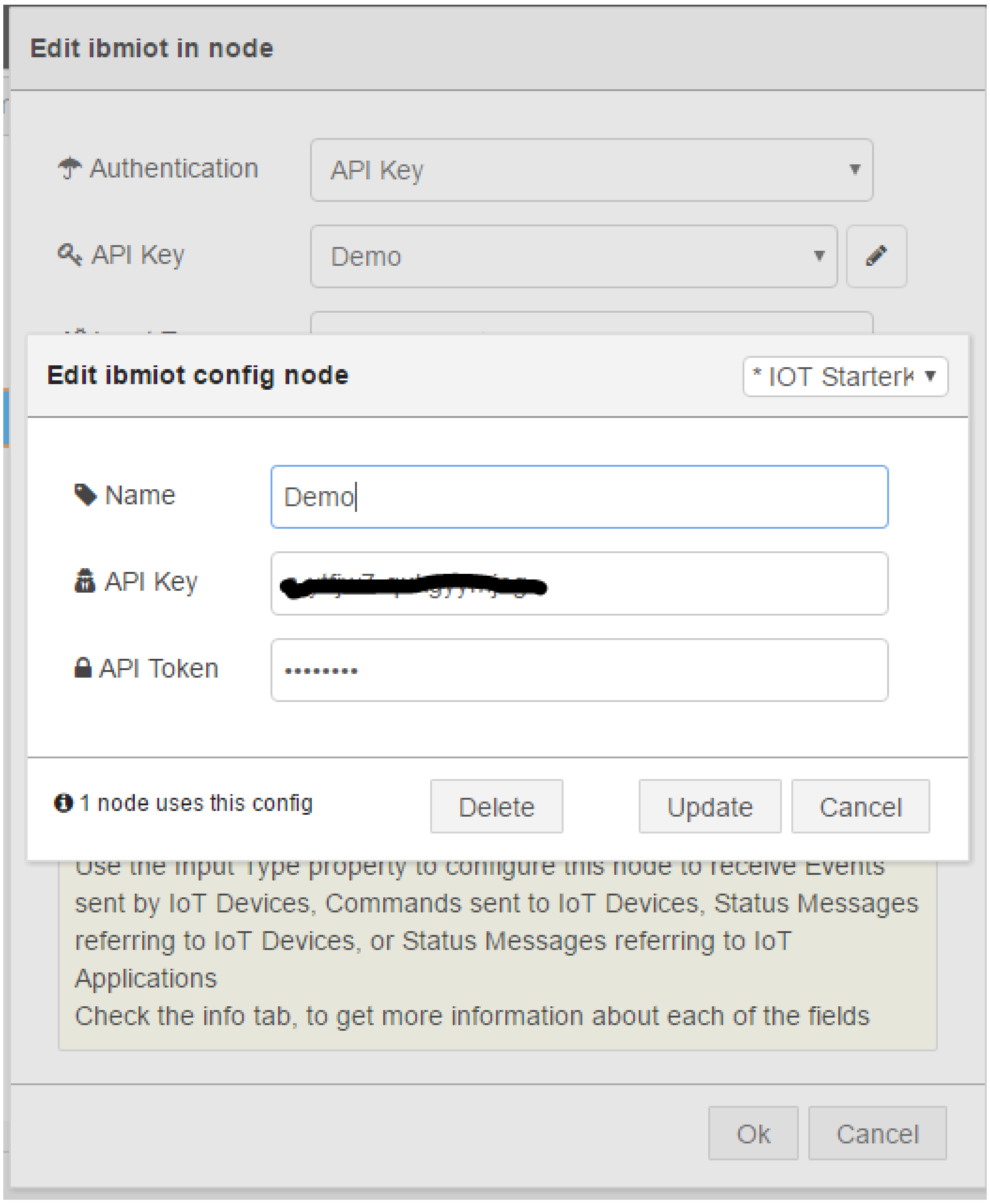
Device Type: All

Device ID: All

Event: All

Format: All

Then Click on the pen to edit the API key and insert the API you get before.



Edit ibmiot in node

Authentication: API Key

API Key: Demo

Edit ibmiot config node

* IOT Starterk

Name: Demo

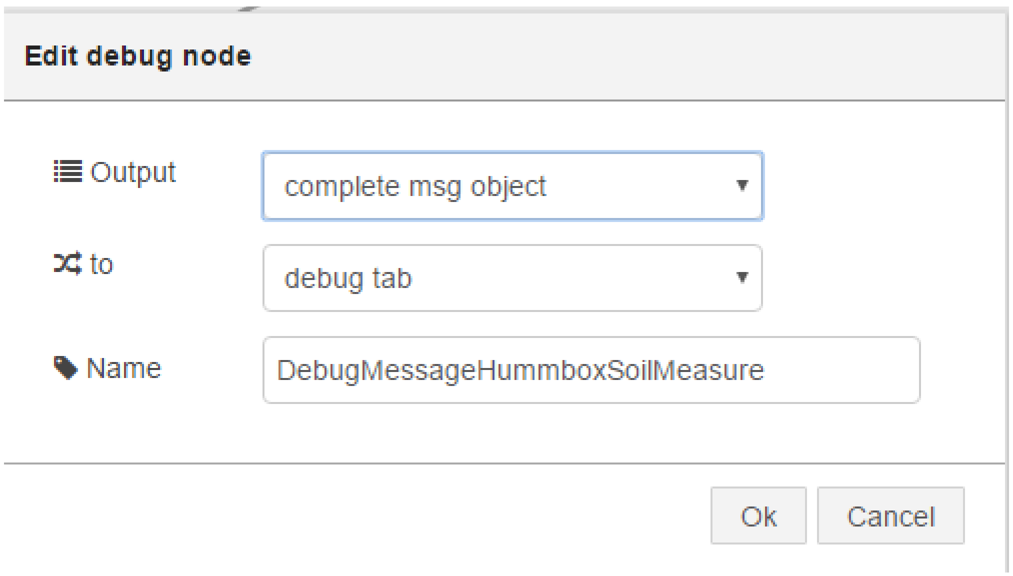
API Key: [redacted]

API Token: •••••••••

1 node uses this config

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

5- Drag a **debug node** and wire it to the ibmiot input node, and configure it like shown below:



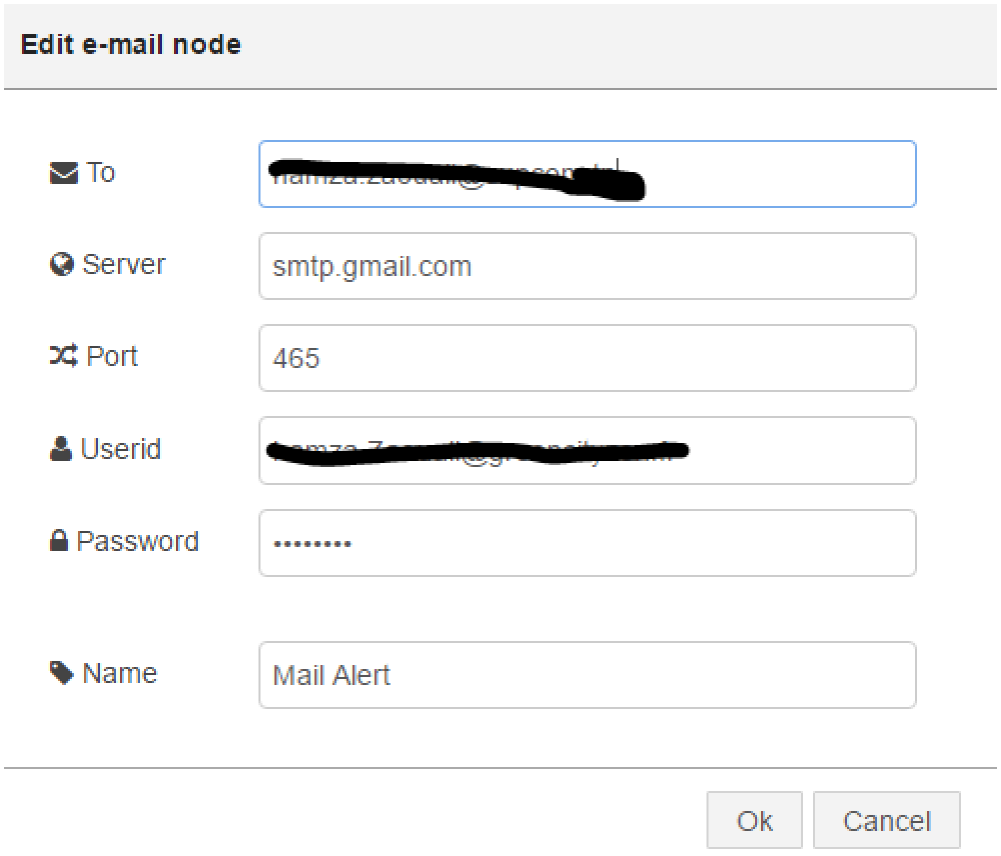
Edit debug node

Output: complete msg object

to: debug tab

Name: DebugMessageHummboxSoilMeasure

6- In order to get your data by email, Click on **e-mail node** and wire it to the ibmiot input node, and configure it like shown below:



Edit e-mail node

To: [redacted]

Server: smtp.gmail.com

Port: 465

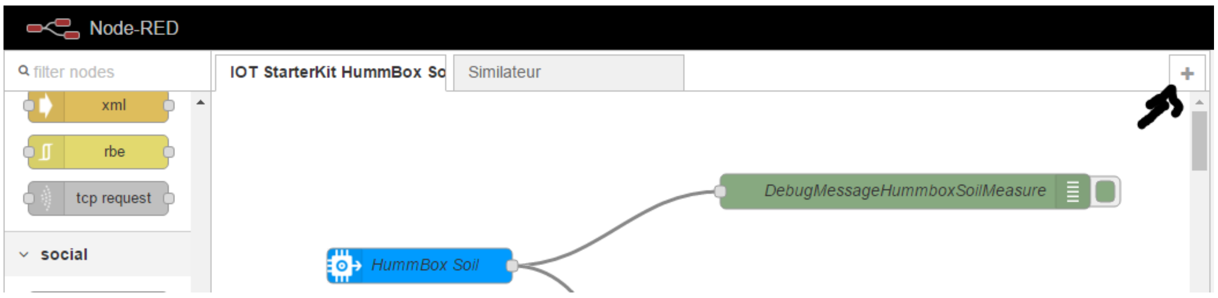
Userid: [redacted]

Password: ••••••••

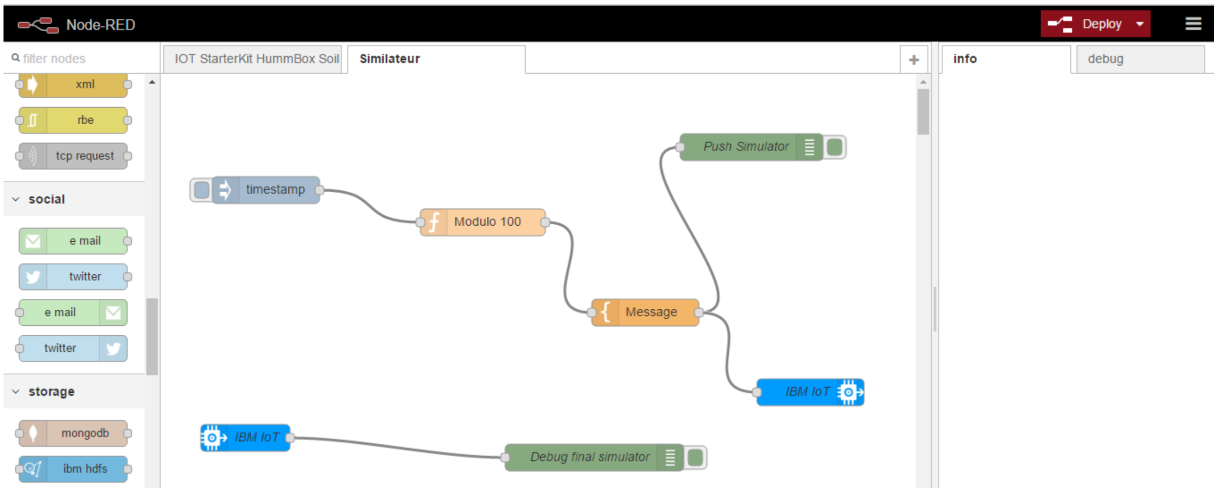
Name: Mail Alert

The second part on configuring our Node-Red, we will create a simple simulator

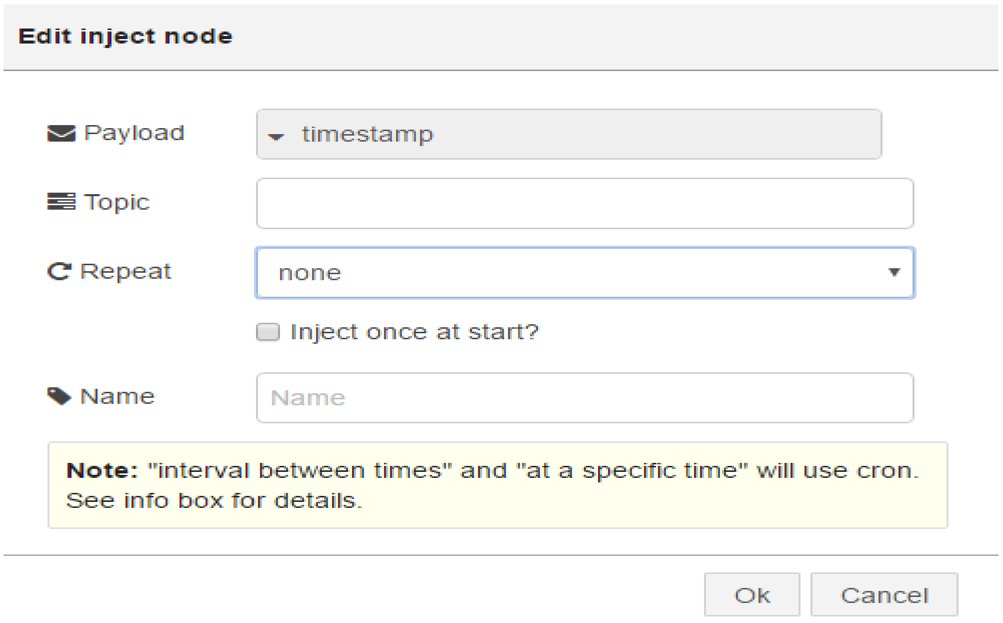
1- Click on « **+** » in your dashboard to add a new flow :



2- We will propose you an example of a simple simulation:



3- First add **an inject node** and double click on it to configure it like shown below :

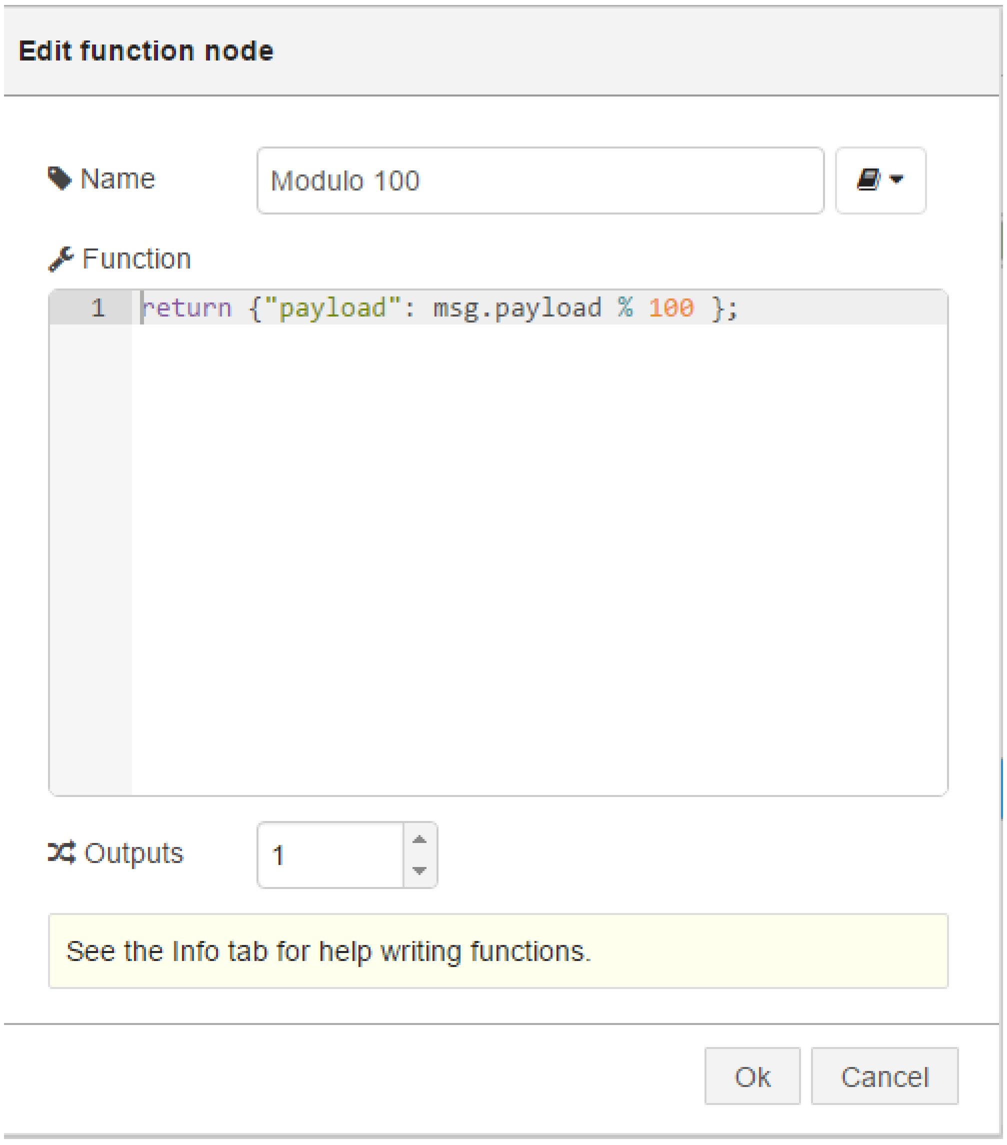


Edit inject node

Payload	timestamp
Topic	
Repeat	none
	Inject once at start?
Name	Name

Note: "interval between times" and "at a specific time" will use cron. See info box for details.

4- Add a **function node,** wire it to the inject node and configure it :



Edit function node

Name: Modulo 100

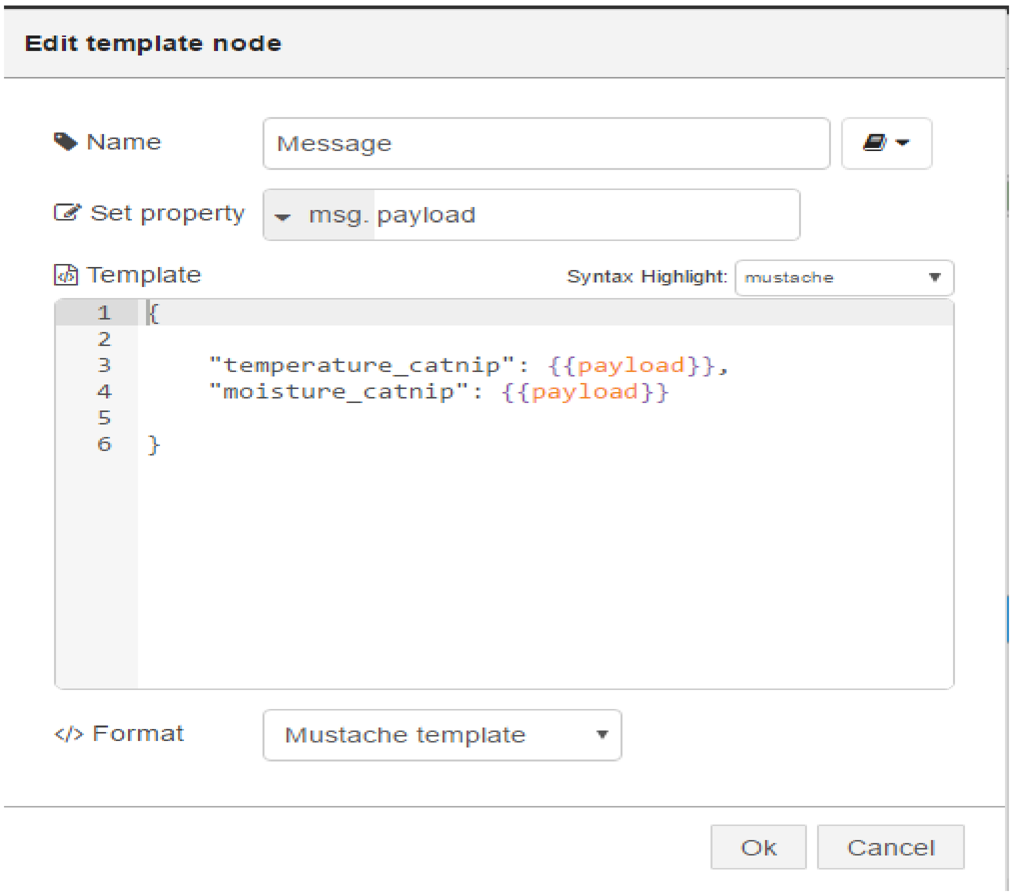
Function:

```
return {"payload": msg.payload % 100 };
```

Outputs: 1

See the Info tab for help writing functions.

5- Add a **template node,** wire it to the function node and configure it :



Edit template node

Name: Message

Set property: msg. payload

Template — Syntax Highlight: mustache

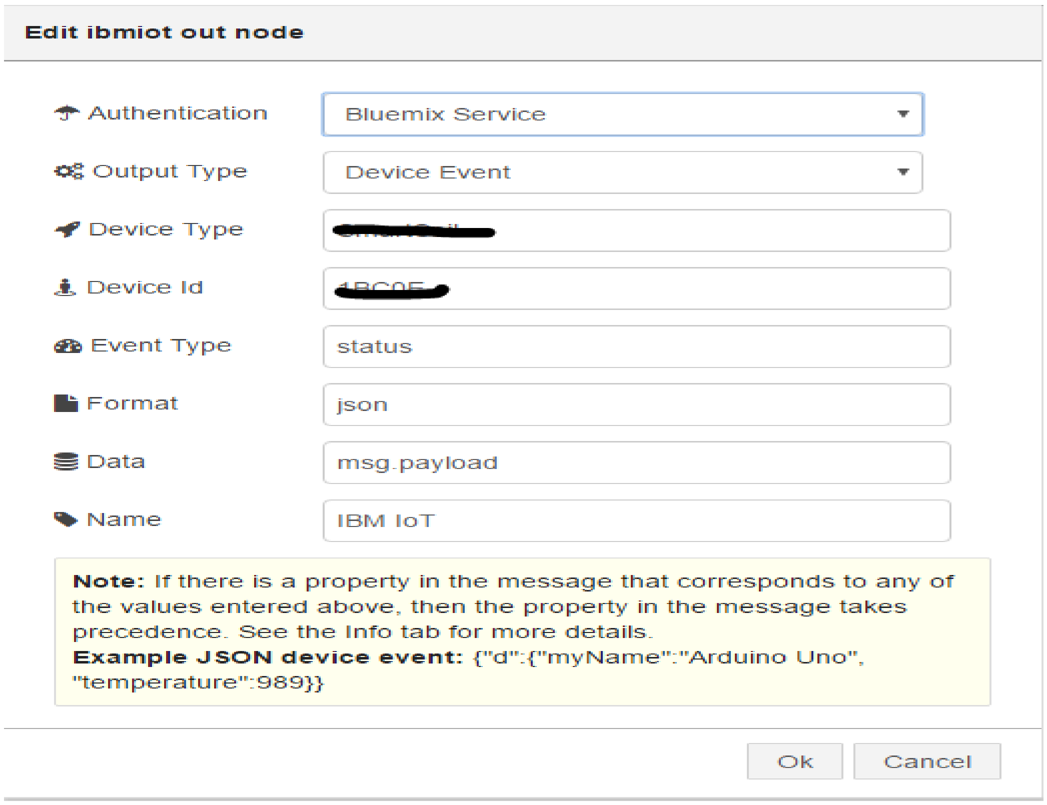
```
{

    "temperature_catnip": {{payload}},
    "moisture_catnip": {{payload}}

}
```

Format: Mustache template

6- Add an **ibmiot output,** wire it to the template node and double click on it to do the configuration :



Edit ibmiot out node

Authentication: Bluemix Service

Output Type: Device Event

Device Type: [redacted]

Device Id: [redacted]

Event Type: status

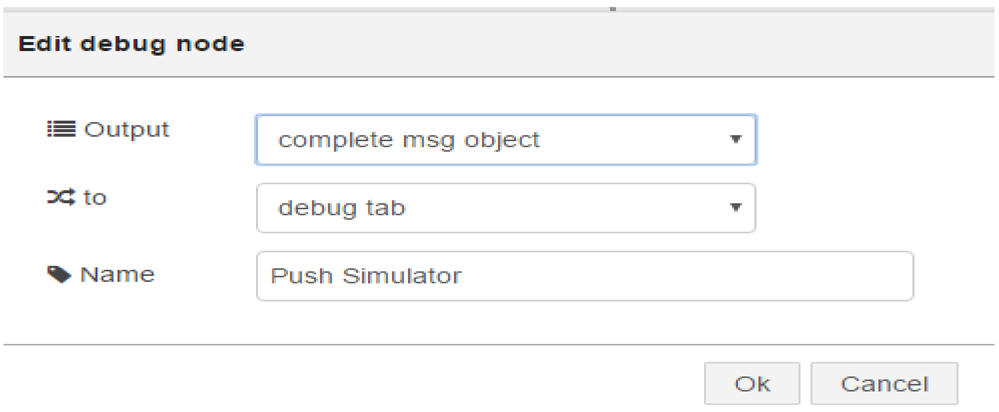
Format: json

Data: msg.payload

Name: IBM IoT

Note: If there is a property in the message that corresponds to any of the values entered above, then the property in the message takes precedence. See the Info tab for more details.
Example JSON device event: {"d":{"myName":"Arduino Uno", "temperature":989}}

7- Add a **debug node,** wire it to the **template** node and configure it :

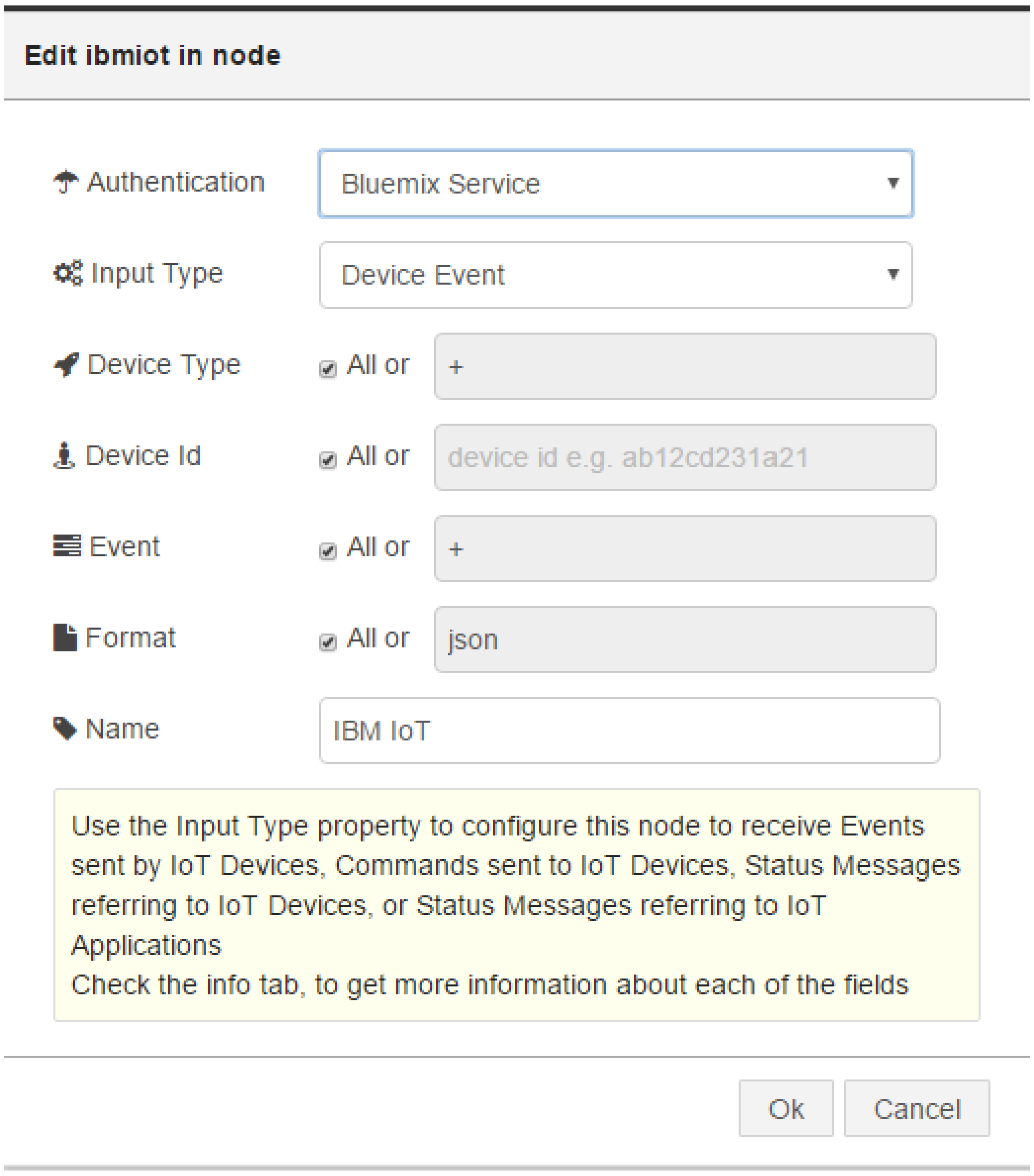

Edit debug node

Output: complete msg object

to: debug tab

Name: Push Simulator

8- Add an **ibmiot input node :**


Edit ibmiot in node

Authentication: Bluemix Service

Input Type: Device Event

Device Type: ✓ All or +

Device Id: ✓ All or device id e.g. ab12cd231a21

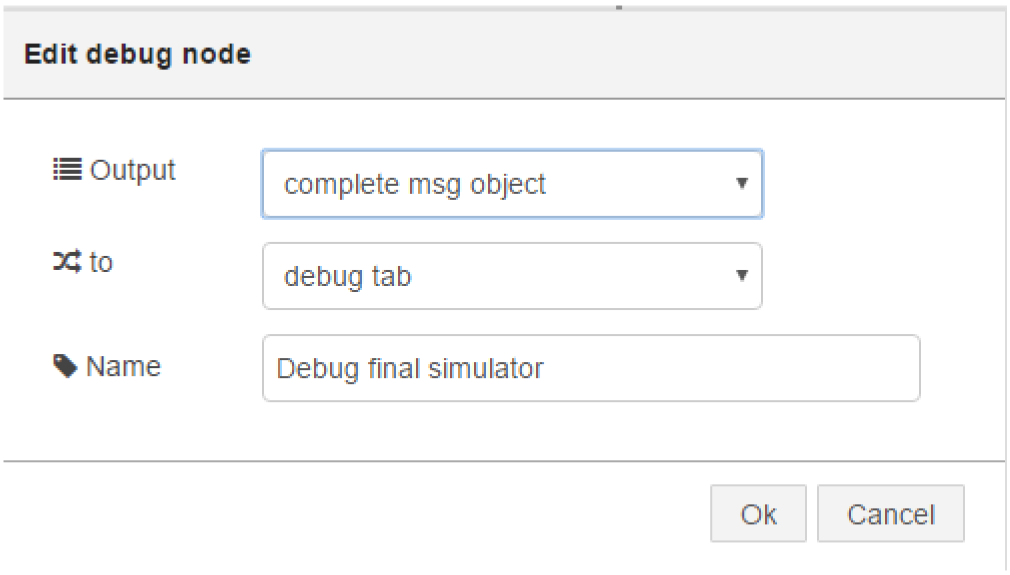
Event: ✓ All or +

Format: ✓ All or json

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

9- Finally add another **debug node :**


Edit debug node

Output: complete msg object

to: debug tab

Name: Debug final simulator

And to see your work click on deploy, then debug to see your data.