



## Bluemix en Pratique

### Exercice 2 - Utilisation de l'internet des objets (IoT)

**Version:**

1

**Modification:**

28-juin-15

**Propriétaire:**

Philippe THOMAS for Devoxx

## Table des matières

---

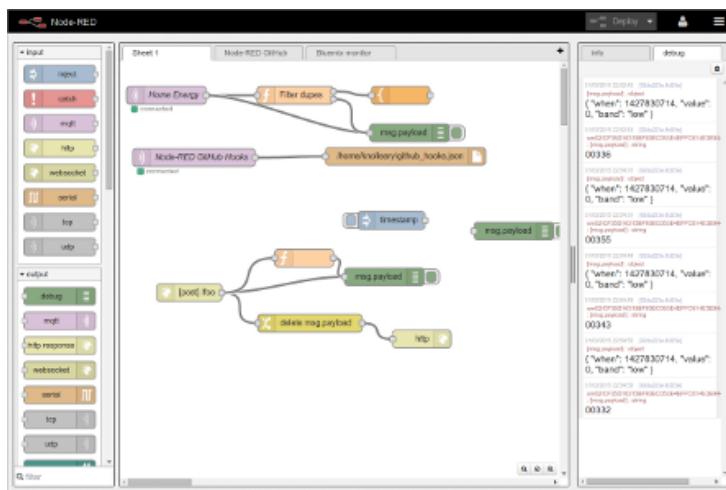
Bluemix en Pratique .....	1
Exercice 2 - Utilisation de l'internet des objets (IoT) .....	1
Exercice 2.a – Création d'une application Node-RED.....	4
Exercice 2.b – Création d'un simulateur de température et d'humidité .....	7
Exercice 2.c – Création d'un flux Node-RED simple .....	9
Exercice 2.d – Création d'un flux complexe.....	12
Exercice 2.e – Enregistrement des données dans une base.....	15

IBM Bluemix propose une grande quantité de services (plus de 100 aujourd’hui). Parmi ces services et ces applications, il en existe certains qui répondent à la problématique de collecte de données sur une grande échelle, avec des protocoles variés et sur des appareils qui peuvent être sophistiqués ou très basiques. L’internet des objets (ou Internet of Things) correspond à ce domaine de collecte et permet de fournir des données pour le Big Data & Analytics.

Dans cet exercice, nous allons voir comment une application Node.js spécialisée peut de façon simple servir à collecter des données, à les transformer puis à les stocker en vue d’une analyse ultérieure.

IBM fournit deux dispositifs dans le cadre de Bluemix :

- une application baptisée Node-RED (<http://nodered.org>) qui est une application open source écrite en Node.js. Cette application peut être utilisée telle quelle pour permettre une collecte efficace de données dans l’internet des objets. De nombreux protocoles en entrée, en sortie ou des fonctions de transformation sont supportés. Cette application est fournie sous la forme d’un boilerplate dans Bluemix.



- Un service IoT qui permet l’interfaçage avec des équipements et des appareils divers. IBM a créé dans cet esprit une fondation : Internet of Things Foundation (<https://internetofthings.ibmcloud.com/>) afin de réunir toutes les informations nécessaires au développement d’applications sur ce sujet (on parle également de recettes – recipes).

Dans cet exercice 2, nous allons donc utiliser une application Node-RED et des pseudo devices qui gèrent la température.

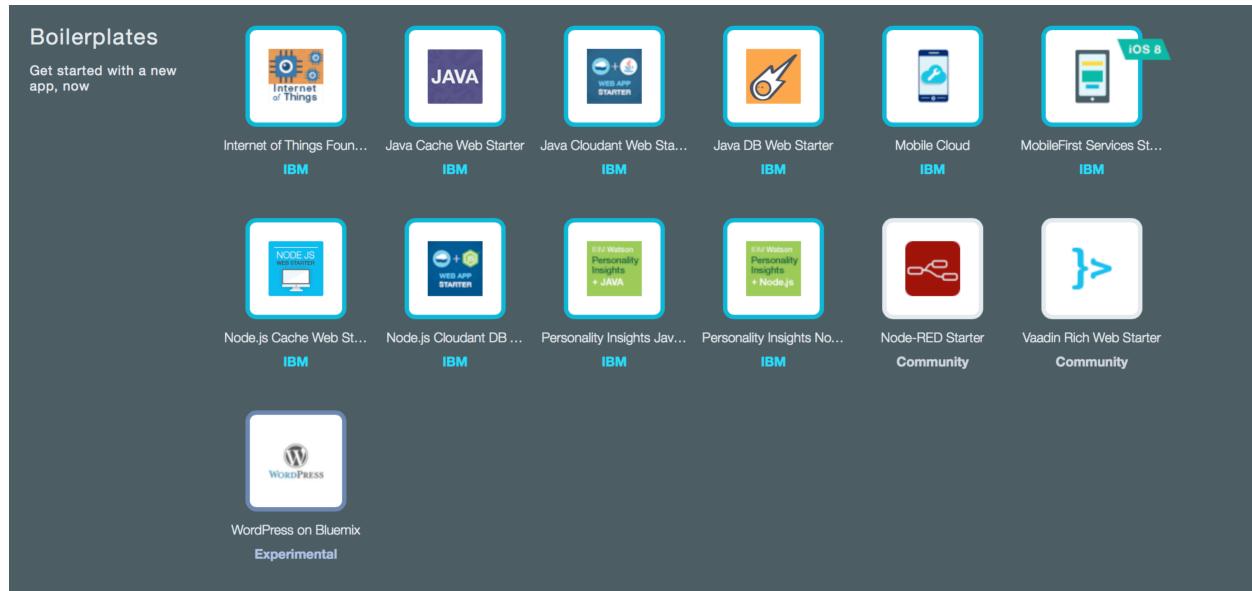
## Exercice 2.a – Création d'une application Node-RED

Cet exercice vous montrera comment déployer une application Node-RED avec la console Bluemix.

Dans votre navigateur, suivez le lien suivant:

<http://bluemix.net>

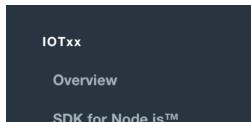
Une fois connecté à Bluemix, ouvrir le catalogue (en haut dans le menu) puis chercher Node-RED starter dans la section Boilerplate :



Cliquer sur Node-RED starter et remplir les champs en utilisant les initiales à la place de XX :

On constatera que ce boilerplate contient une application Node.js (avec le code Node-RED), une base de données No SQL dans le cloud et un composant de monitoring.

Cliquer sur ‘CREATE’ afin de créer et de lancer cette application. Puis cliquer sur ‘Overview’ sur le menu à gauche :



Après une minute environ, l’application doit être lancée (rond vert coché à droite).

The screenshot shows the IOTxx application interface. At the top, it says 'IOTxx' and 'Routes: IOTxx.mybluemix.net'. Below this, there are sections for 'INSTANCES' (set to 1), 'MEMORY QUOTA' (512 MB), and 'AVAILABLE MEMORY' (1.500 GB). There are 'SAVE' and 'RESET' buttons. To the right, there's an 'APP HEALTH' section with a green checkmark and 'Your app is running.' status, and a 'RESTART' button. Below that is an 'ACTIVITY LOG' section with the following entries:

- 4/2/15 2:24 PM philibert.thomas.7@gmail.com started IOTxx app
- 4/2/15 2:23 PM philibert.thomas.7@gmail.com updated IOTxx app
  - changed routes
- 4/2/15 2:23 PM philibert.thomas.7@gmail.com created IOTxx app

Pour se connecter à l’application, utiliser le lien bleu (Routes) en dessous du titre de l’application. L’écran suivant doit apparaître :

The screenshot shows the Node-RED in BlueMix application editor. The title bar says 'Node-RED in BlueMix'. The main area is titled 'Node-RED in BlueMix' and describes it as 'A visual tool for wiring the Internet of Things'. Below this is a diagram of a flow with several nodes connected by wires. A red button at the bottom center says 'Go to your Node-RED flow editor'. At the bottom left, there is information about Node-RED and its version in the BlueMix environment. At the bottom right, there are links to learn how to password-protect the instance and customize Node-RED.

Node-RED provides a browser-based editor that makes it easy to wire together flows that can be deployed to the runtime in a single-click.

The version running here has been customised for the BlueMix cloud environment.

More information about Node-RED, including documentation, can be found at [nodered.org](http://nodered.org).

Go to your Node-RED flow editor

[Learn how to password-protect your instance](#)

[Learn how to customise Node-RED](#)

Puis cliquer sur le bouton rouge au centre pour obtenir l’écran suivant :



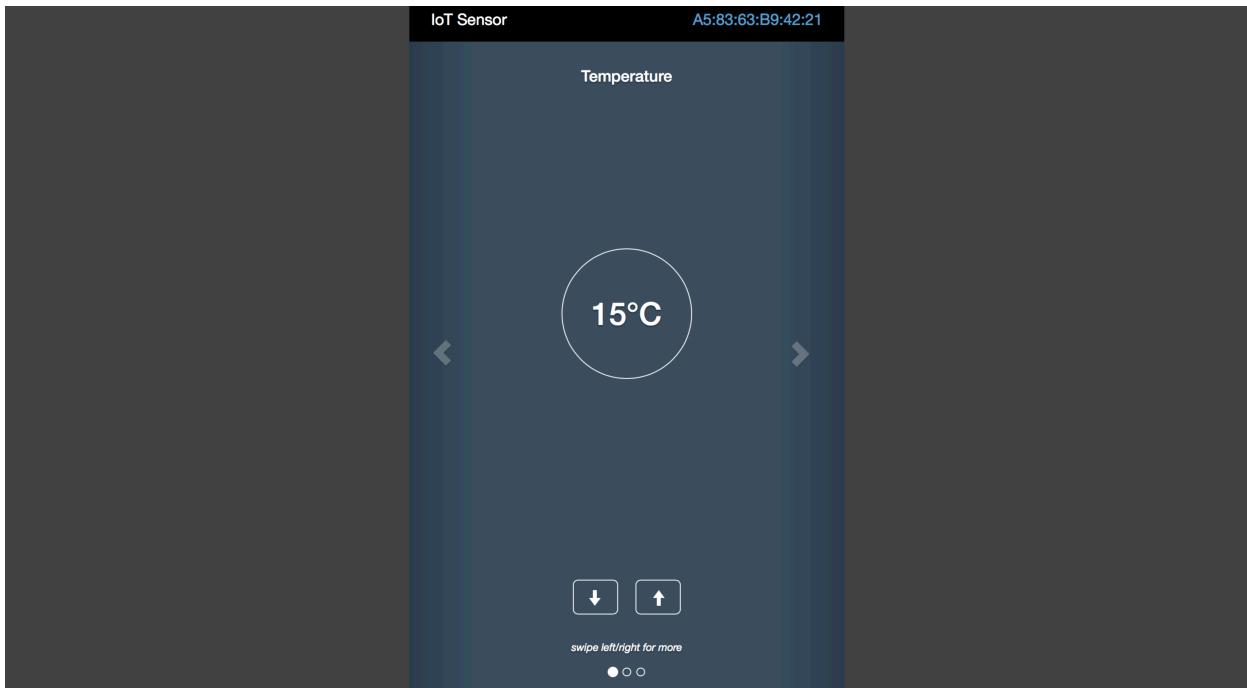
L'application est opérationnelle.

## Exercice 2.b – Création d'un simulateur de température et d'humidité

Nous allons maintenant créer un simulateur de température et d'humidité pour notre exercice.

Rendez vous sur le site suivant en utilisant un autre onglet de votre navigateur :  
<http://quickstart.internetofthings.ibmcloud.com/iotsensor>

L'écran suivant s'affiche et présente la température :

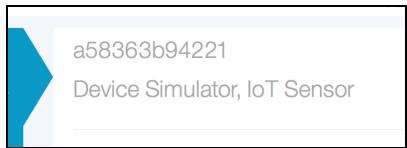


En utilisant les flèches latérales, on obtient l'humidité. En utilisant les flèches en bas de l'écran, on obtient une hausse ou une baisse de la température. Il y a une particularité sur IoT, c'est l'identifiant de votre appareil (qui doit être unique dans votre espace de gestion). Cet identifiant est mentionné en haut à droite (nombre entrecoupé par : ).

Cliquer sur ce nombre en haut à gauche – nous sommes redirigés automatiquement sur le site IoT Foundation qui nous présente l'évolution de la température automatiquement :



Le numéro de l'appareil apparaît en haut débarrassé des « : »



Dans la suite de l'exercice, on fera un copier-coller de ce numéro.

## Exercice 2.c – Création d'un flux Node-RED simple

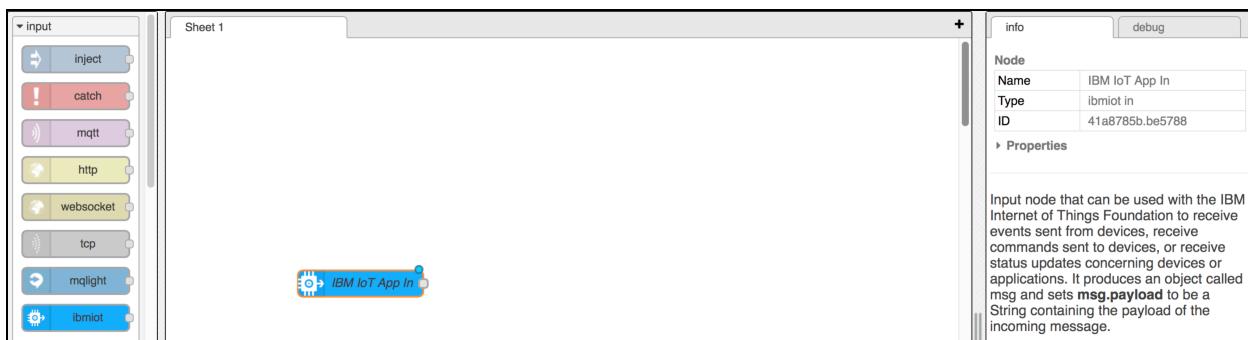
Nous allons maintenant créer un premier flux simple de collecte de données. L'écran de l'application est divisé en trois parties :

- la partie gauche qui contient une palette complète de dispositifs prêts à fonctionner (on utilisera le drag & drop vers la partie centrale)
- la partie centrale qui représente un tableau blanc pour recueillir les flux ainsi dessinés
- la partie droite qui sert à la visualisation et au débogage.

En haut à droite, il y a également un bouton 'Deploy' qui passera en rouge dès que vous ferez des modifications sur les flux. Pour activer ces nouveaux flux, vous devrez cliquer sur ce bouton.

L'application se comporte comme un gestionnaire de flux : les données arrivent sur la gauche, traversent différents nœuds de traitement et se terminent sur la droite.

Créons un premier flux très basique : dans la section de gauche 'Input', chercher 'ibmiot' et le déplacer vers la zone centrale afin d'obtenir cet écran :



Chercher maintenant dans la section 'Output' à gauche, prendre le nœud 'Debug' et le placer dans la zone centrale :

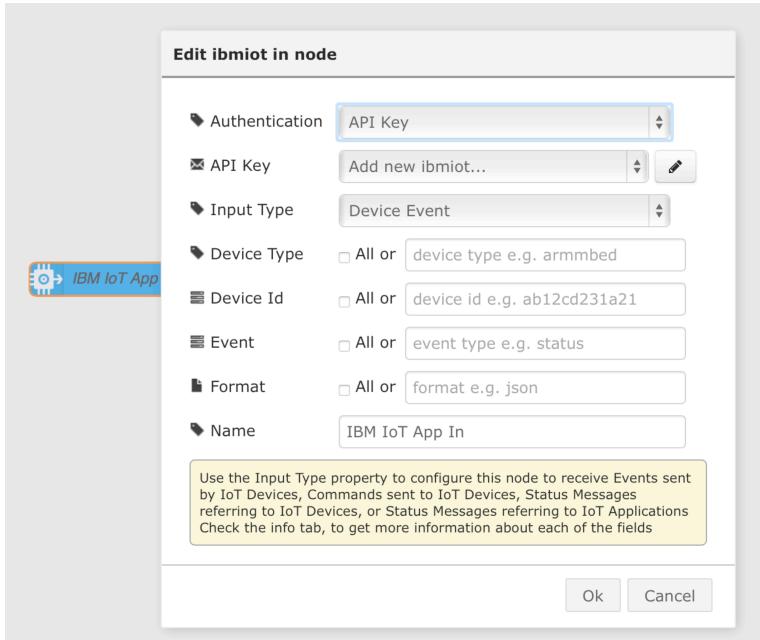


Relier les deux nœuds ensemble en tirant sur le point gris de 'ibmiot' vers le point gris de 'debug'. On obtient la figure suivante :

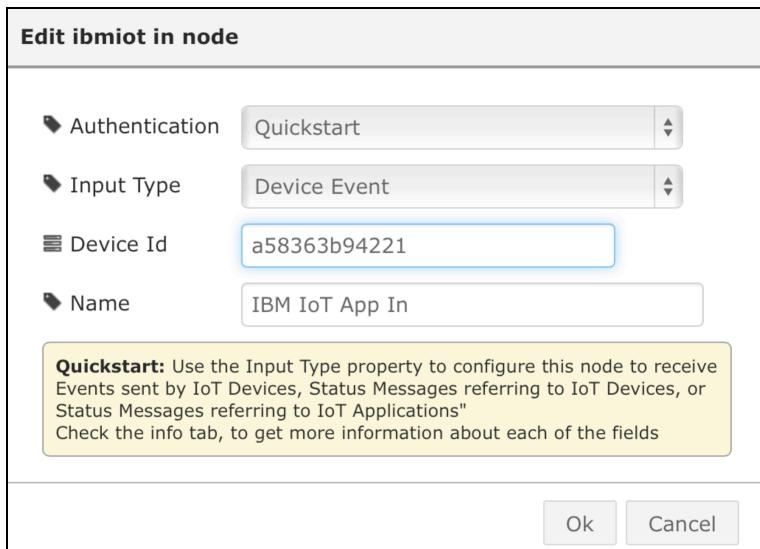


En résumé, toutes les données qui vont entrer dans le nœud bleu vont ressortir par le nœud vert qui correspond à un affichage dans la partie à droite.

En double-cliquant sur le nœud bleu, on obtient l'écran suivant :

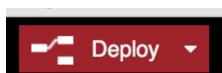


Comme on peut le voir, il existe un grand nombre de possibilités pour authentifier, sécuriser ou définir le scope de notre collecte. Pour cet exercice, nous allons simplement définir les paramètres suivants (Quickstart pour le premier et le numéro de l'appareil pour le Device Id) :

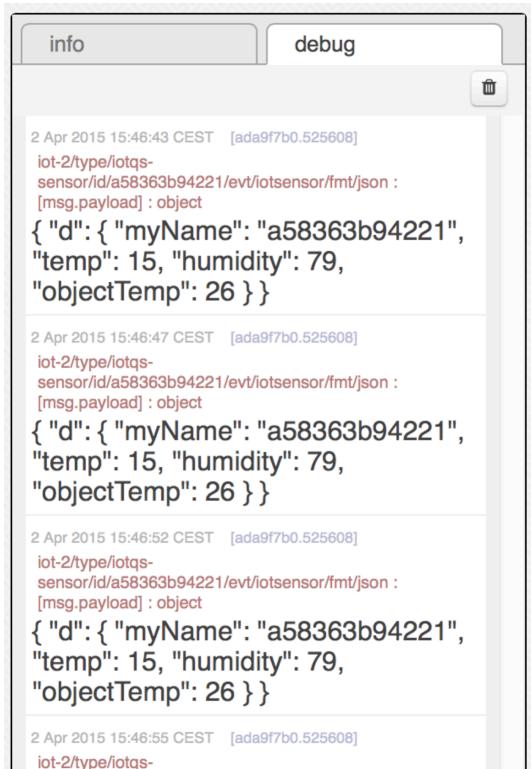


Puis cliquer sur OK.

Puis sur 'Deploy' en haut à droite:



Le flux est déployé (activé). Cliquer sur l'onglet ‘Debug’ dans la partie droite et vous devriez voir automatiquement les informations transmises sur la température et l'humidité dans un format JSON.



```
2 Apr 2015 15:46:43 CEST [ada9f7b0.525608]
iot-2/type/iotqs-
sensor/id/a58363b94221/evt/otsensor/fmt/json :
[msg.payload] : object
{ "d": { "myName": "a58363b94221",
"temp": 15, "humidity": 79,
"objectTemp": 26 } }

2 Apr 2015 15:46:47 CEST [ada9f7b0.525608]
iot-2/type/iotqs-
sensor/id/a58363b94221/evt/otsensor/fmt/json :
[msg.payload] : object
{ "d": { "myName": "a58363b94221",
"temp": 15, "humidity": 79,
"objectTemp": 26 } }

2 Apr 2015 15:46:52 CEST [ada9f7b0.525608]
iot-2/type/iotqs-
sensor/id/a58363b94221/evt/otsensor/fmt/json :
[msg.payload] : object
{ "d": { "myName": "a58363b94221",
"temp": 15, "humidity": 79,
"objectTemp": 26 } }

2 Apr 2015 15:46:55 CEST [ada9f7b0.525608]
iot-2/type/iotqs-
```

Retourner dans la page du simulateur de température et vous pourrez faire varier la température afin de vérifier que votre collecte est correcte.

Pour terminer cette partie sélectionner tous les nœuds du flux et cliquer sur la touche ‘Delete’ du clavier afin de supprimer tout le flux. La partie centrale est de nouveau vierge.

## Exercice 2.d – Création d'un flux complexe

Nous allons maintenant créer un second flux plus complexe.

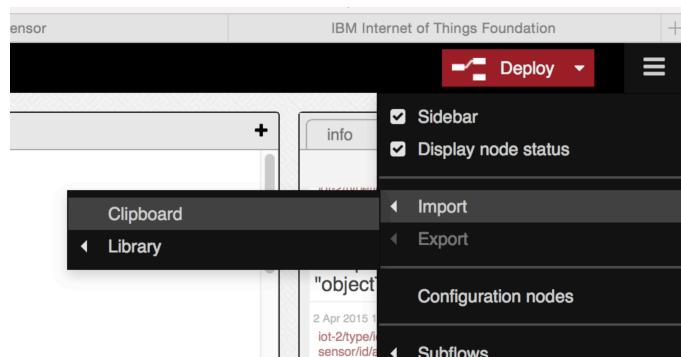
Pour réaliser cette opération, faite un copier des lignes JSON suivantes :

----- Copier en dessous de cette ligne -----

```
{"id":"41a8785b.be5788","type":"ibmiot
in","authentication":"quickstart","apiKey":"","inputType":"evt","deviceId":"a58363b94221",
applicationId:"", "deviceType": "+", "eventType": "+", "commandType": "", "format": "json", "name": "IBM IoT App
In", "service": "quickstart", "allDevices": "", "allApplications": "", "allDeviceTypes": true, "allEvents": true, "allCommands": "", "allFormats": "", "x": 127, "y": 168, "z": "3aae4e02.c551b2", "wires": [{"3
3ddabbc.cc2254", "ba6805de.4597f8"}]}, {"id": "ba6805de.4597f8", "type": "function", "name": "temp", "func": "return
{payload:msg.payload.d.temp};", "outputs": 1, "valid": true, "x": 337, "y": 181, "z": "3aae4e02.c5
51b2", "wires": [{"96ecad21.69135"}]}, {"id": "96ecad21.69135", "type": "switch", "name": "tem
p
thresh", "property": "payload", "rules": [{"t": "lte", "v": "40"}, {"t": "gt", "v": "40"}], "checkall": "true", "outputs": 2, "x": 486, "y": 182, "z": "3aae4e02.c551b2", "wires": [{"dcb0b639.234f48"}, {"c3df154
5.3c20e8"}]}, {"id": "d51c36c6.2ae3c8", "type": "debug", "name": "cpu
status", "active": true, "complete": false, "x": 797, "y": 179, "z": "3aae4e02.c551b2", "wires": []}, {"id": "33ddabbc.cc2254", "type": "debug", "name": "device
data", "active": true, "complete": false, "x": 337, "y": 92, "z": "3aae4e02.c551b2", "wires": []}, {"id": "dcb0b639.234f48", "type": "template", "name": "safe", "template": "Temperature
({{payload}}) within safe
limits", "x": 650, "y": 133, "z": "3aae4e02.c551b2", "wires": [{"d51c36c6.2ae3c8"}]}, {"id": "c3df1
545.3c20e8", "type": "template", "name": "danger", "template": "Temperature ({{payload}})
critical", "x": 649, "y": 227, "z": "3aae4e02.c551b2", "wires": [{"d51c36c6.2ae3c8"}]}
```

----- Copier au dessus de cette ligne -----

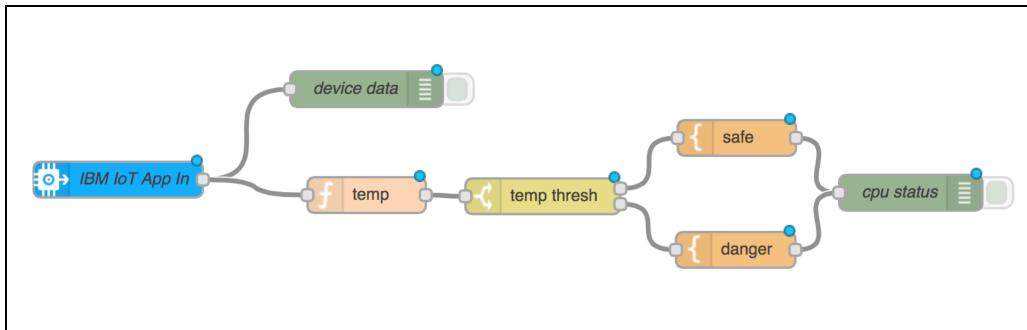
Ensuite sur l'application Node-RED, vous pourrez utiliser la fonction 'import' en utilisant le bouton en haut à droite à coté de Deploy :



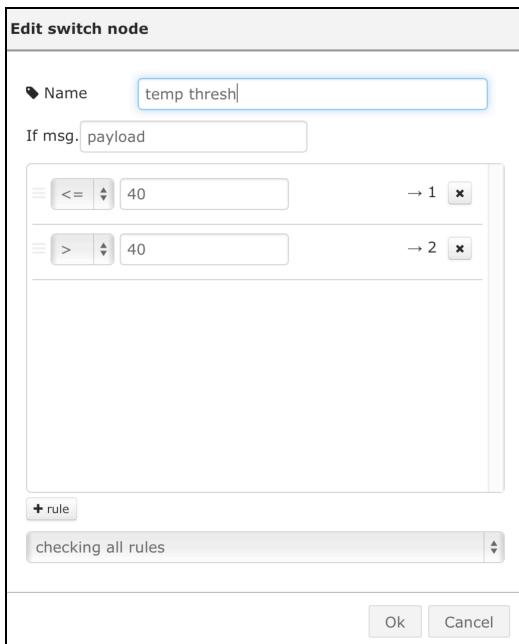
Puis coller le JSON dans la fenêtre :



Et voici le flux :

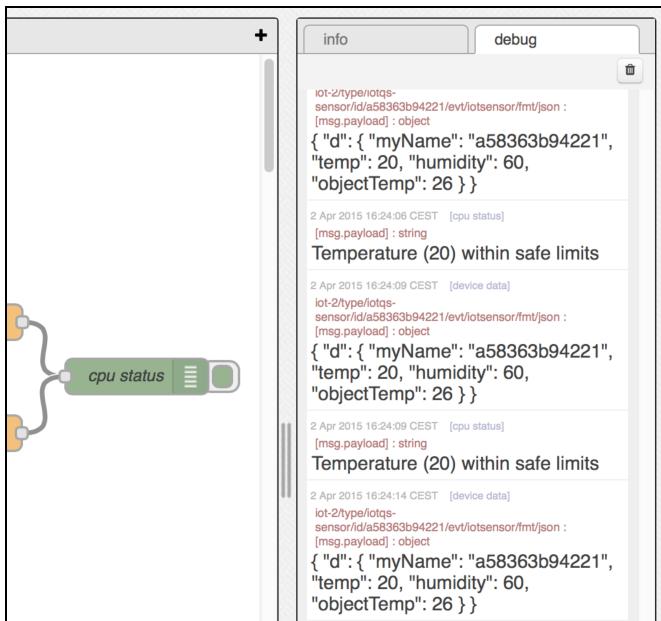


Dans ce flux, nous utilisons des fonctions, des switchs et des affichages. En cliquant sur les nœuds du flux vous pourrez mieux comprendre comment fonctionne le flux. En clair, si la température dépasse une valeur de 40°C, nous allons afficher un message de danger.



Il faudra à nouveau redéfinir le numéro de l'appareil en cliquant sur le nœud d'entrée (bleu). Et cliquer sur 'Deploy'.

Pour vérifier le fonctionnement, vous pouvez à nouveau retourner sur le simulateur et faire varier la température ou l'humidité.

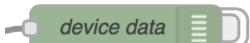


On peut voir que des messages sont mélangés.

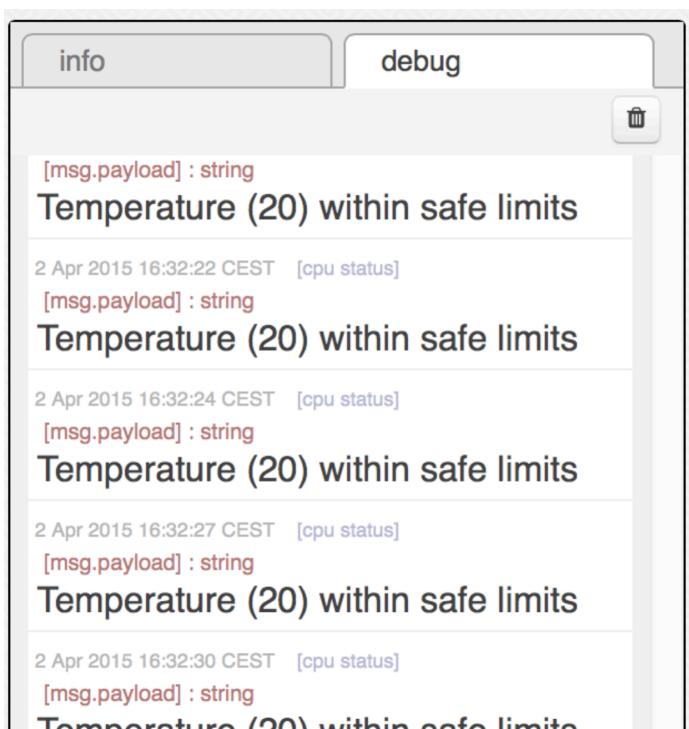
Pour supprimer les messages ‘Debug – Device Data’, il faut aller cliquer sur le bout du nœud Debug afin de le désactiver :



vers:



On obtient donc les informations suivantes sur la gauche :



Vous pouvez à nouveau jouer avec la température afin de vérifier le fonctionnement.

## Exercice 2.e – Enregistrement des données dans une base

Nous allons maintenant enregistrer automatiquement toutes les données dans une base de données No SQL sur le cloud. Ce service s'appelle Cloudant dans Bluemix.

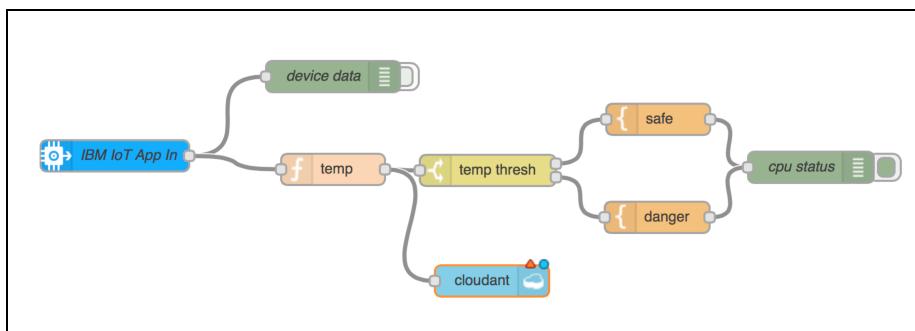
Il a déjà été ajouté à notre application (il n'y a rien à faire de ce côté là) :

The screenshot shows the IOTxx service dashboard in the Bluemix interface. At the top, it displays 'IOTxx' and 'Routes: IOTxx.mybluemix.net'. Below this, there are settings for 'INSTANCES' (set to 1) and 'MEMORY QUOTA' (set to 512 MB per Instance). A section titled 'ADD A SERVICE OR API' is visible. In the bottom left, there is a 'Cloudant NoSQL DB' entry with a hexagonal icon, labeled 'IOTxx-cloudantNoSQLDB Shared'. Buttons for 'Show Credentials' and 'Docs' are present next to this entry.

Retournons sur l'application et sur le flux complexe sur lequel nous travaillons et dans la section Storage, utilisons le nœud Cloudant (attention) :



Relier ce nouveau nœud avec le reste du flux de la façon suivante :



Puis cliquer sur le nœud Cloudant pour le paramétriser :

- Service : cela doit être le nom du service Cloudant dans Bluemix
- Database : c'est un nom en minuscule
- Name : c'est un nom en minuscule

**Edit cloudant out node**

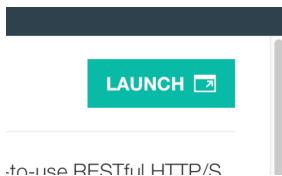
Service	IOTxx-cloudantNoSQLDB
Database	nodered
Operation	insert
<input checked="" type="checkbox"/> Only store msg.payload object?	
Name	temp
<input type="button" value="Ok"/> <input type="button" value="Cancel"/>	

Cliquez sur OK et déployez votre flux.

Afin de vérifier que nos données arrivent dans notre base NoSQL, nous devons donc retourner sur Bluemix afin d'aller sur la base en question.

Dans Bluemix, cliquer sur le service Cloudant :

Puis sur 'LAUNCH'



La console de Cloudant apparaît :

Name	Size	# of Docs	Update Seq	Actions
nodered	18.8 KB	63	6	

Cliquez sur 'nodered' pour visualiser la base :

The screenshot shows the Apache CouchDB interface. On the left, there's a sidebar with sections like 'Permissions', 'Changes', 'All Documents', 'All Design Docs', and '\_design/library'. The main area displays two JSON documents. The first document has an id of "1eca60ff05c4204c1c60049ed442b911" and a payload of "20". The second document has an id of "1eca60ff05c4204c1c60049ed443e2ed".

Les documents JSON apparaissent dans la base.  
Pour visualiser un document, cliquer sur le crayon.

A modal window is open, showing a JSON document with the following content:

```

1 {  

2   "_id": "1eca60ff05c4204c1c60049ed442b911",  

3   "_rev": "1-33c76caf9e2f7cdcbd14b3fe4d56d1d6",  

4   "payload": "20"  

5 }

```

Nous retrouvons notre température de 20°C.

Pour finir cet exercice, il est possible de définir de nouveaux nœuds (entrée, sortie ou de traitement) dans notre application Node-RED. En parcourant les sections sur la gauche, vous trouverez de nombreuses possibilités (envoyer des emails ou des tweets, recevoir des messages MQTT, écrire dans une base de données SQL ...).

## Fin de cet exercice.