

TP BlueMix



Bluemix

Les exercices nécessitent l'ouverture d'un compte blueMix et donc la création d'un ID IBM.

Les exercices nécessitent de l'espace mémoire pour être réalisés, vous devrez peut être supprimer des exercices anciens

Exercice 1

Créer une application simple avec l'interface utilisateur Bluemix

Objectifs :

Créer une application Web Java

Se familiariser avec l'interface utilisateur (tableau de bord et le catalogue de services, principalement)

Etapes à suivre :

- Dans le menu "catalogue", sélectionner le runtime Java Liberty et créer une instance.

Aide : indiquer un nom unique pour votre application



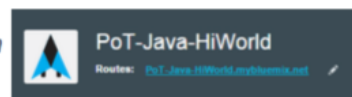
- Attendre quelques minutes....observer le résultat dans le tableau de bord.

Note : vous pouvez naviguer dans l'interface en attendant le démarrage de l'application

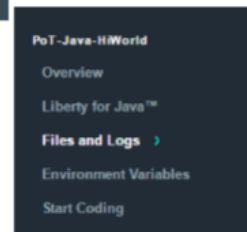


- Accéder à l'application

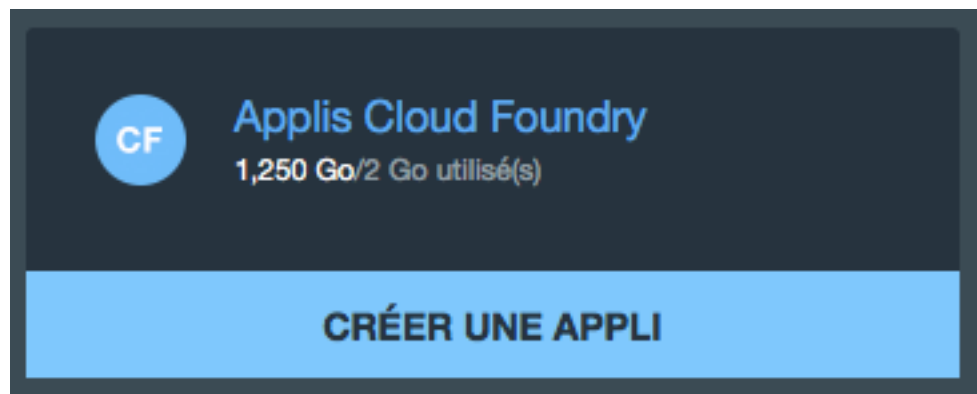
Aide : utiliser le lien indiqué dans l'application



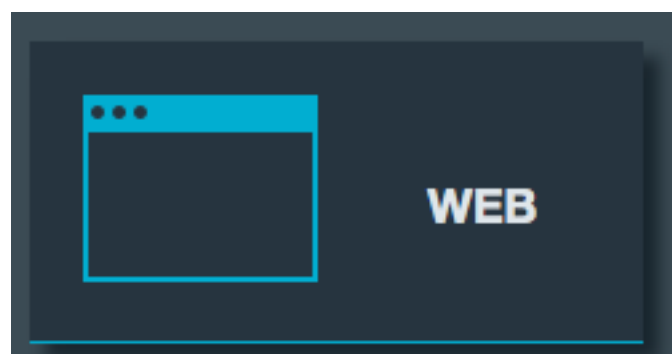
- (Optionnel) Vérifier les fichiers et journaux de l'application



1. Créer une application

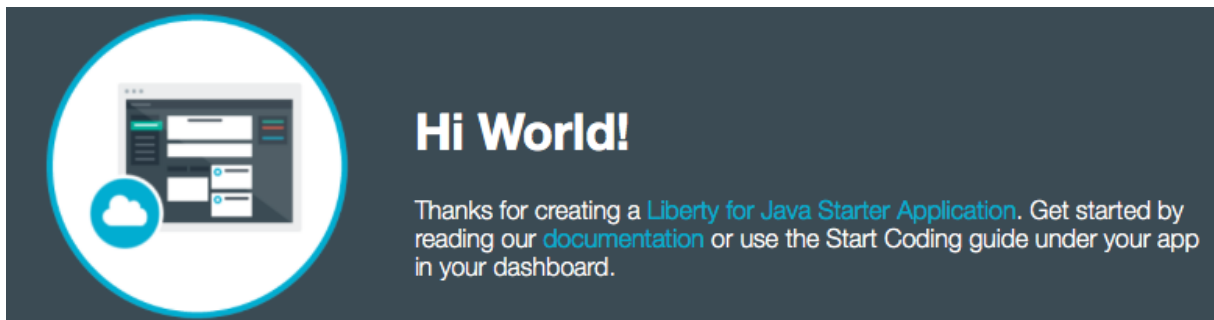


2. Choisir une application Web



3. Paramétrer votre application

4. enjoy !



A la fin de l'exercice :

Votre application Web Java sera opérationnelle
L'application sera accessible depuis un navigateur internet.

Exercice 2

Déployer une application avec la ligne de commande CloudFoundry

Code source de l'application node JS

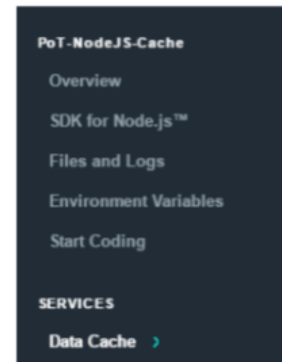
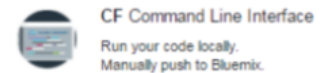
[ZIP pour le data cache](#)

Objectifs :

Déployer une application Node.js en utilisant les outils en ligne de commande
Découvrez des nouveaux services (Data Cache)

Etapas à suivre :

- Installer les outils en ligne de commande CloudFoundry
Aide : Depuis le menu "Commencer le codage/Start Coding", cliquer sur le lien pour obtenir les outils en ligne de commande
Aide 2 : Vérification de l'installation (et de sa version) avec la commande 'cf -v'
- Connexion à Bluemix
 - `cf login -a <Bluemix API endpoint>`
 - Avec <Bluemix API endpoint> défini ainsi :
 - <https://api.ng.bluemix.net> pour Bluemix public à Dallas, US
 - <https://api.eu-gb.bluemix.net> pour Bluemix public à Londres, UK
- Créer le service Data Cache
 - `cf create-service DataCache starter <CacheServiceName>`
 - Avec <CacheServiceName> = un nom unique dans l'espace Bluemix
- Editer le fichier *manifest.yml*
 - Indiquer des noms uniques pour service, host et application
- Déployer l'application vers Bluemix
 - `cf push`
- Utiliser l'application
- (Optionnel) Regarder le fichier *package.json*
- (Optionnel) Accéder au service Data Cache



A la fin de l'exercice :

Vous disposerez d'une application Node.js, qui utilisera le service Data Cache.

L'application sera accessible depuis un navigateur Internet.

Application Example: run DataCache using REST API

Grid Operations:

Key:

Value:

The get was successful for the key BlueMix.

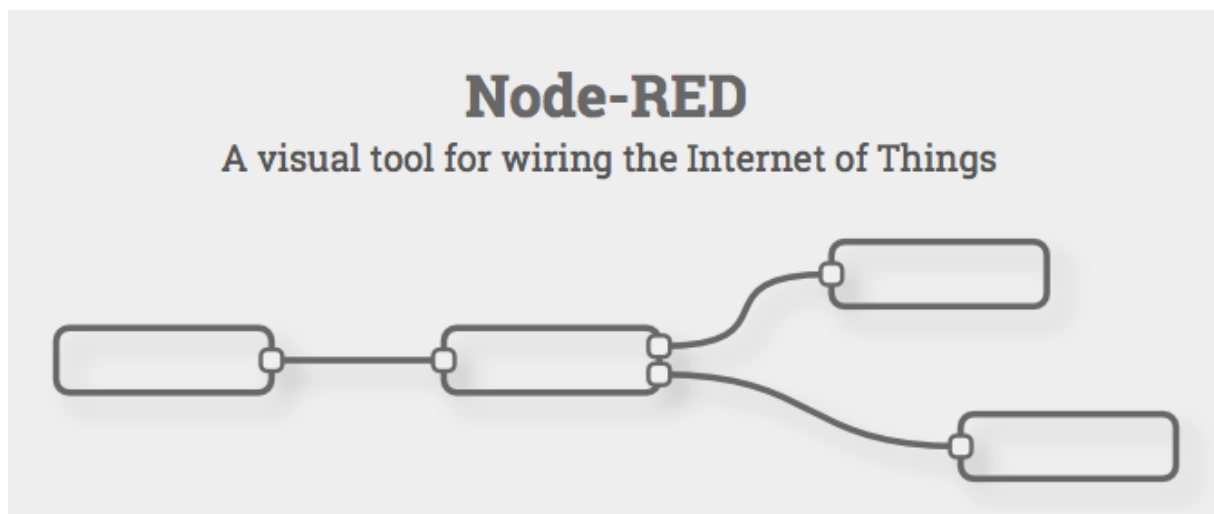
Exercice 3

Développer une application IoT avec NodeRED

Objectifs :

Créer et modifier une application en utilisant NodeRED

NodeRED est un atelier de création visuel pour relier des matériels, des services, dans l'internet des objets.



Découvrir des nouveaux services (IoT) et NodeRED (outil visuel pour utiliser les objets connectés)

Etapes à réaliser :

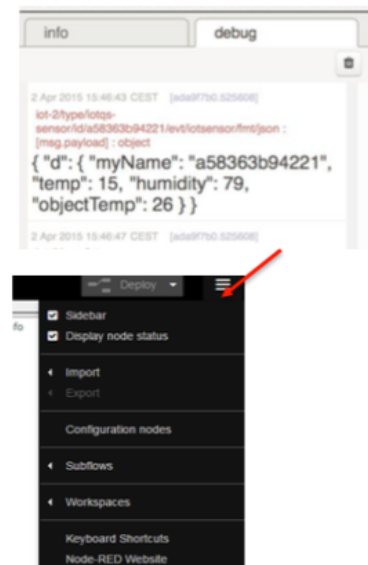
– Etapes à suivre (1/3)

- Dans le menu catalogue, choisir le “boilerplate” NodeRED et créer une instance.
- Exécuter l'application, et démarrer l'éditeur de flux/flow editor.
- Démarrer un simulateur et identifier son device ID (dans le coin haut droit).
<http://quickstart.internetofthings.ibmcloud.com/iotsensor>
- Dans l'éditeur, créer votre premier flux, (cliquer/déposer des boites disponibles dans le panneau de gauche). Choisir le noeud « IBM IoT App In ». Ajouter un noeud de « Debug ». Relier les 2 noeuds ensemble. Configurer le noeud « IBM IoT App In » :
 - Authentication : Quickstart
 - Device ID : <valeur fournie par le simulateur>
- Déployer le flux.



– Etapes à suivre (2/3)

- Cliquer sur l'onglet Debug (panneau de droite), et observer ce qu'il se passe quand on joue avec le simulateur
- Supprimer le flux. Sélectionner tous les noeuds et cliquer sur la touche Delete.
- Créer un nouveau flux, mais cette fois en important le code source fourni.
 - Cliquer sur le bouton en haut à droite
 - Sélectionner import
 - Copier/Coller le code JSON .
- Déployer le nouveau flux. Modifier la température et observer les résultats dans la console de debug.



– Etapes à suivre (3/3)

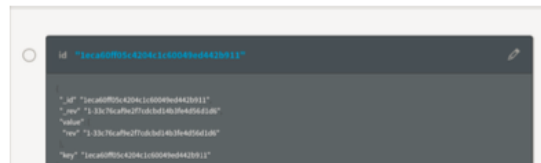
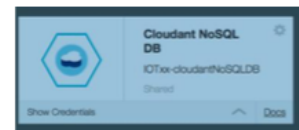
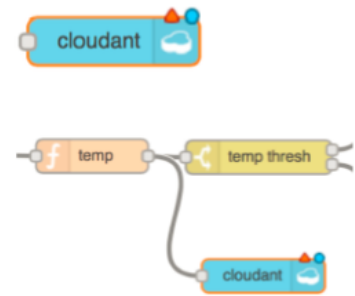
- Ajouter un nœud Cloudant (output node). Lier le nouveau nœud au nœud de fonction *temp*.

Configurer le nœud Cloudant :

- Service : nom du service Cloudant lié au runtime Node.js
- Database : nom de votre choix (en minuscule)
- Name (du node) : nom de votre choix.

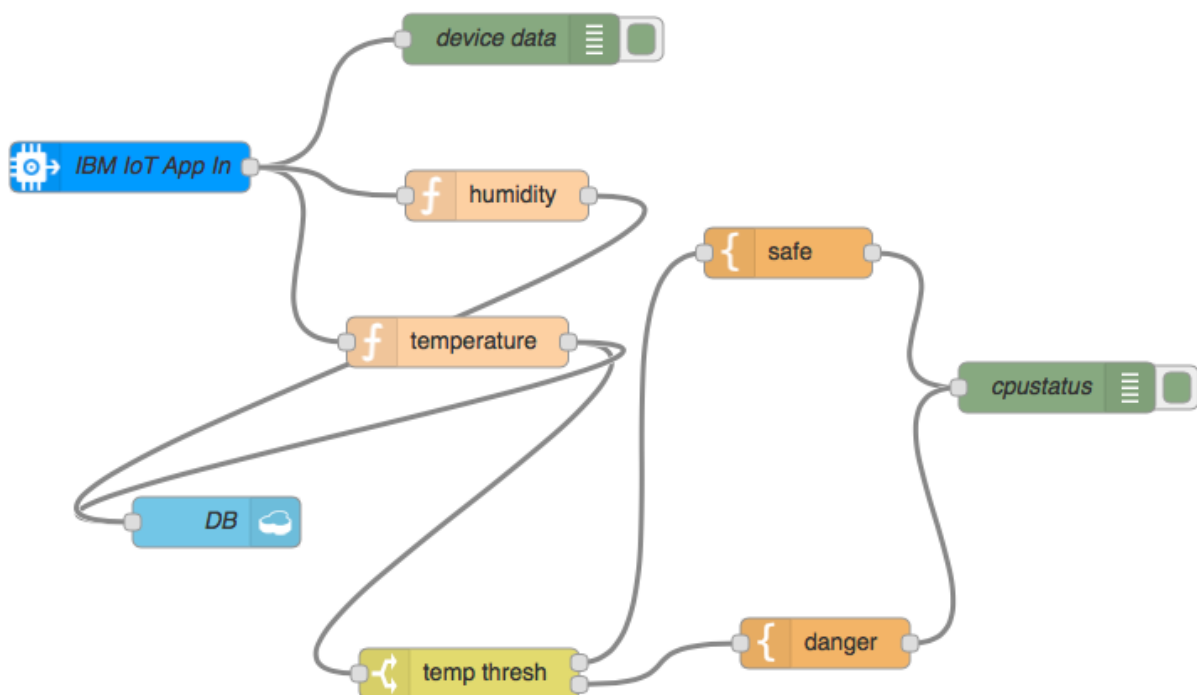
- Déployer le nouveau flux.

- Démarrer la console Cloudant, sélectionner la base de données et y naviguer.



A la fin de l'exercice :

Votre application NodeRED sera opérationnelle (utilise un runtime Node.js), accèdera à des services Cloudant et IoT Foundation Services (QuickStart)



Le mode debug de nodeRED, permet de visualiser les captures de données à intervalles réguliers :



```
"payload": { "d": { "name": "955286d330a4",
"temp": 16, "humidity": 78, "objectTemp": 25 }
}, "deviceId": "955286d330a4", "deviceType":
"iotqs-sensor", "eventType": "iotsensor",
"format": "json", "_msgid": "700ddec9.8ff22" }
```

29 octobre 2015 19:23:53 UTC+1 cpustatus
msg.temperature : number
16

29 octobre 2015 19:23:53 UTC+1 DB
msg : string [37]
Property '_msgid' renamed to 'msgid'.

29 octobre 2015 19:23:53 UTC+1 DB
msg : string [37]
Property '_msgid' renamed to 'msgid'.

29 octobre 2015 19:23:56 UTC+1 device data
iot-2/type/iotqs-sensor/id/955286d330a4/evt/iotsensor/fmt/json :
msg : object
{ "topic": "iot-2/type/iotqs-
sensor/id/955286d330a4/evt/iotsensor/fmt/json",
"payload": { "d": { "name": "955286d330a4",
"temp": 16, "humidity": 78, "objectTemp": 25 }
}, "deviceId": "955286d330a4", "deviceType":
"iotqs-sensor", "eventType": "iotsensor",
"format": "json", "_msgid": "adf42176.520be" }

L'application sera en ligne (accessible sur Internet), et sera accédée depuis un simulateur de température.

La base de données Cloudant associée se remplira de données issues du capteur de température :


```
{
  "_id": "1d02304f91884d1c3994006d02d4f2ba",
  "_rev": "1-87c28624b0206ebc76fdb7500f797f74",
  "humidity": 78,
  "msgid": "ccb1b00d.334e5"
}
```

```
{
  "_id": "1d02304f91884d1c3994006d02d4fb2f",
  "_rev": "1-da0299cce2a64bc9aa357c2baeda526e",
  "temperature": 16,
  "msgid": "ccb1b00d.334e5"
}
```

Valeurs du capteur :

