Lab 1 - Partie 4

# Azure CosmosDB

Dans les labs précédents, nous avons créé notre IoTHub, notre fonction et notre site web.

Dans cette partie nous allons créer la base de données CosmosDB (documentDB) pour stocker l’historique de nos devices.

Ce lab va se composer en 3 phases :

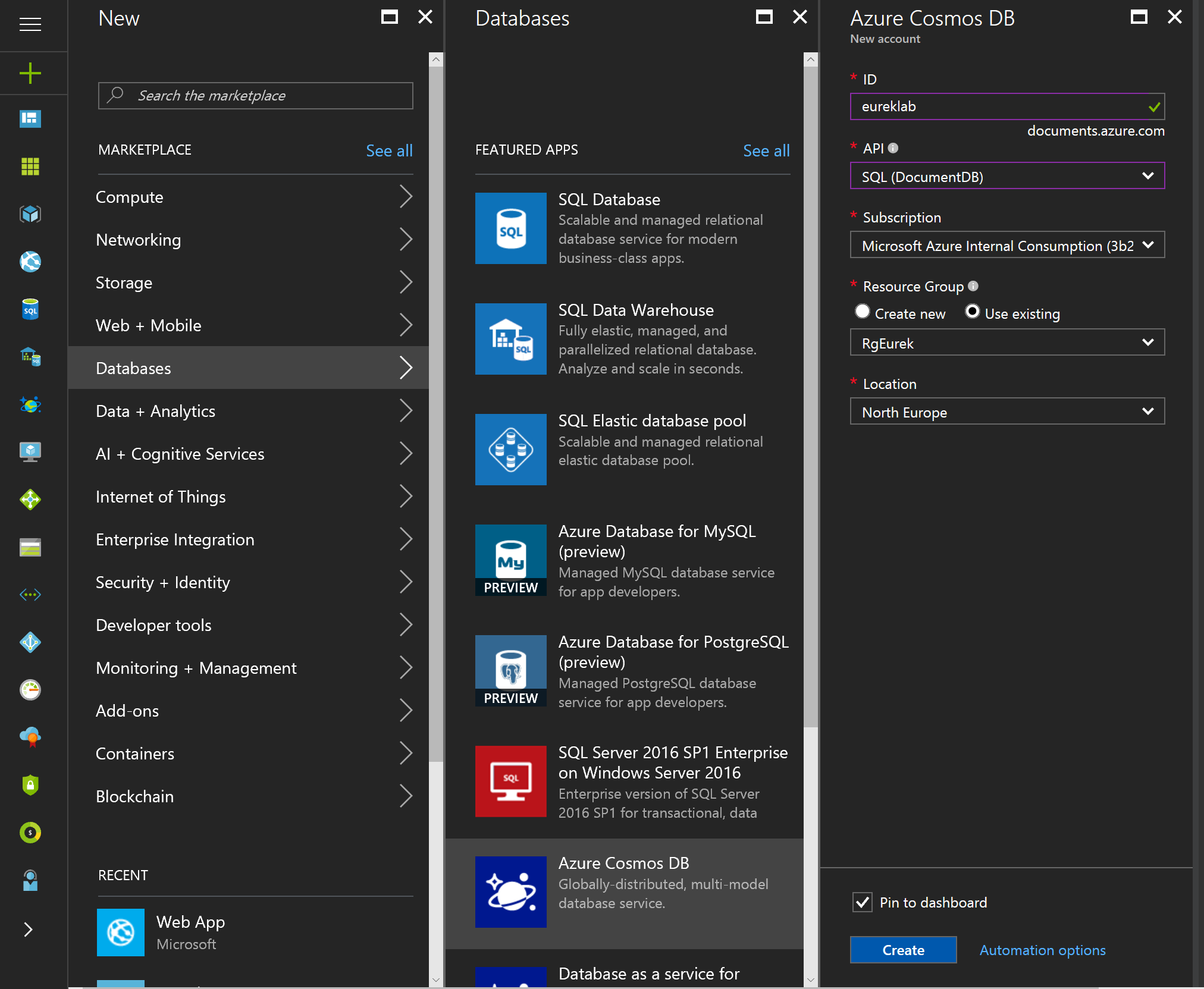
* Création de la BD CosmosDB
* Ajout d’une fonction pour récupérer les messages et stocker l’historique des états dans la base
* Tester si cela fonctionne correctement.

### Création de la BD CosmosDB

Dans le portail Azure

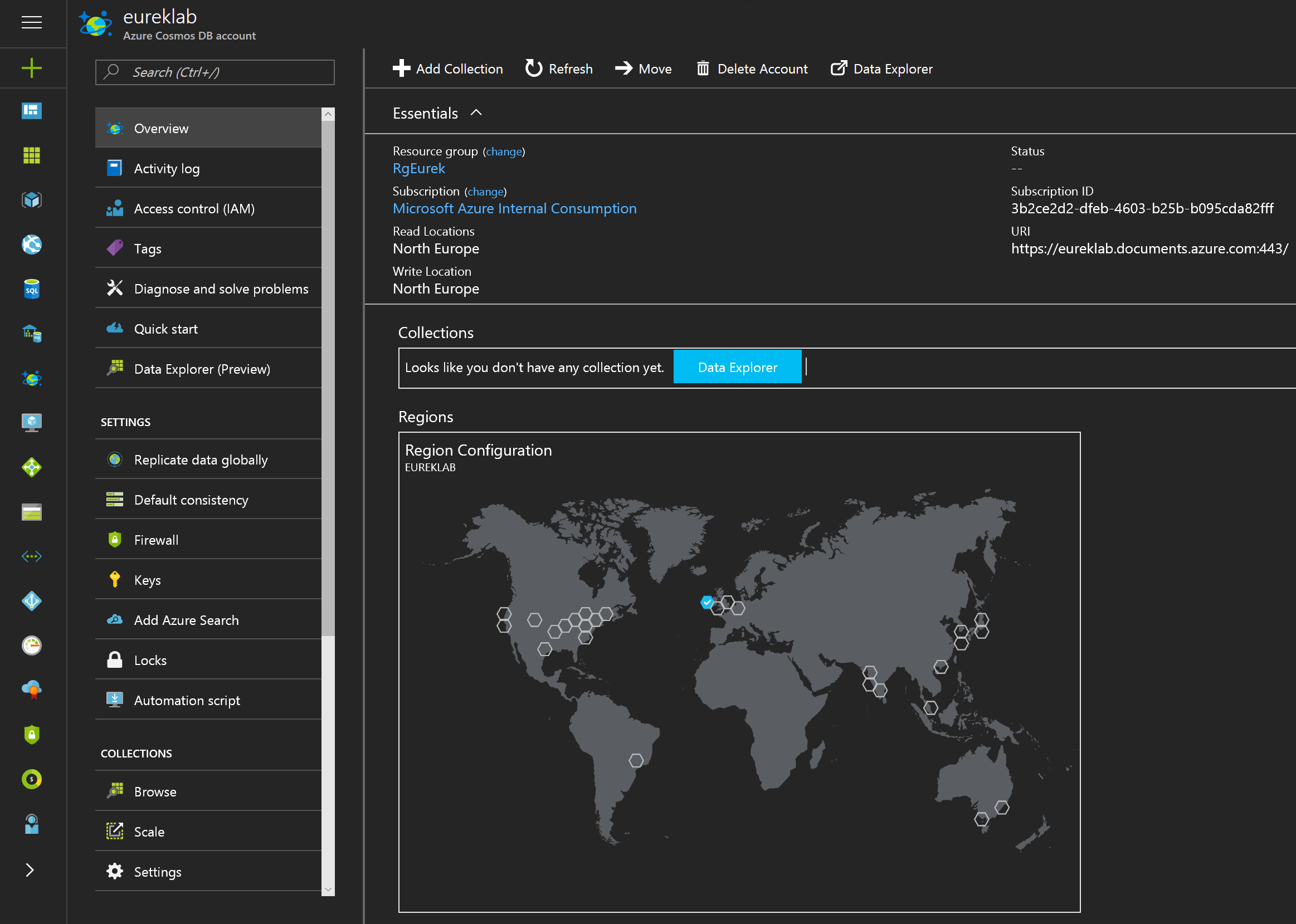
***WARNING : Tous les noms de services et key utilisés dans les Lab (Lab1 – xxx) doivent être modifié pour votre propre usage.***

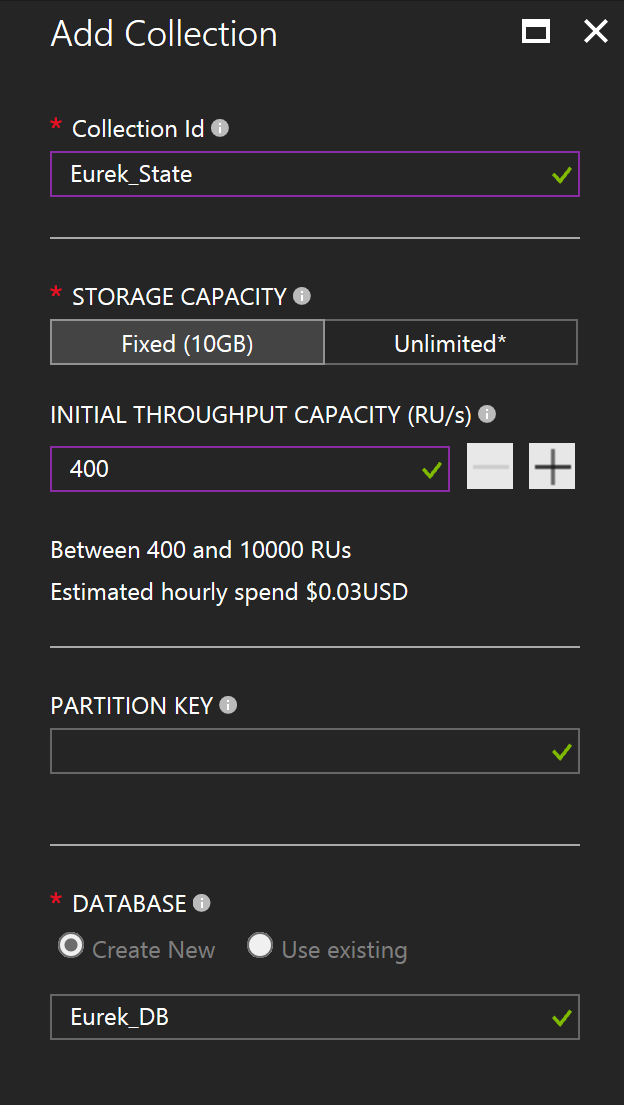
* ID : Enpoint du compte : eureklab
* API : SQL (DocumentDB)
* Resource Group : RgEurek
* Location North Europe

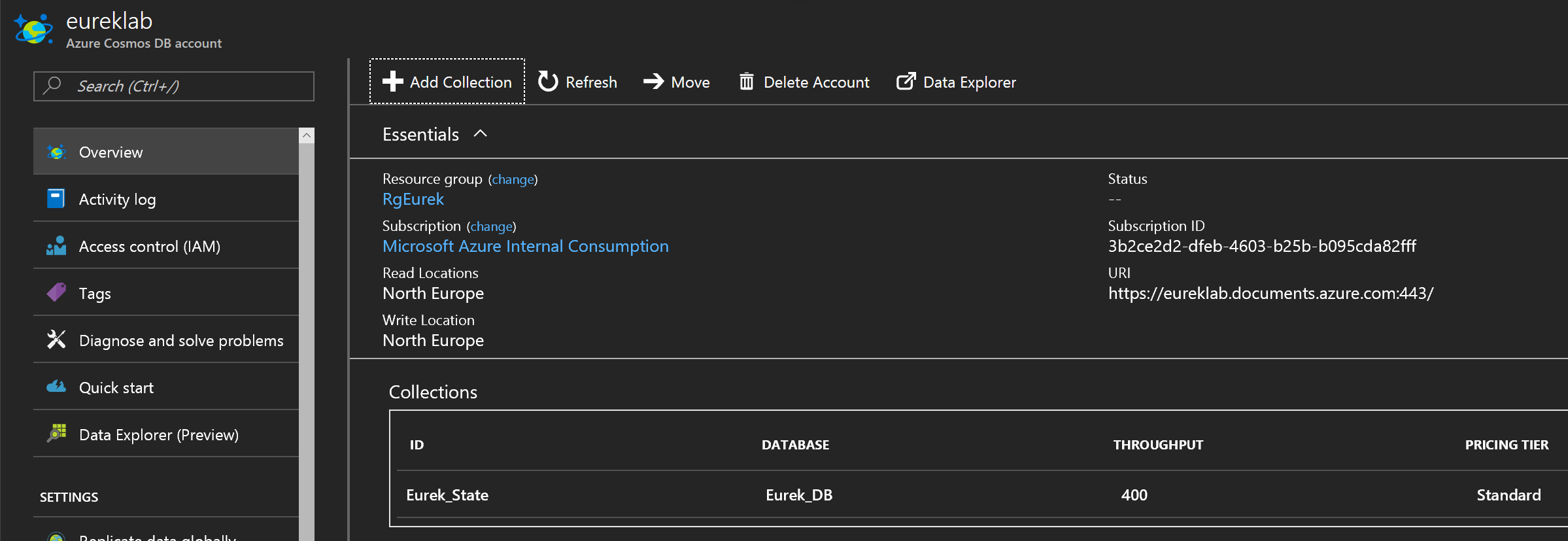


Attendre 3 mn que le service soit provisionné.

Créons une nouvelle collection

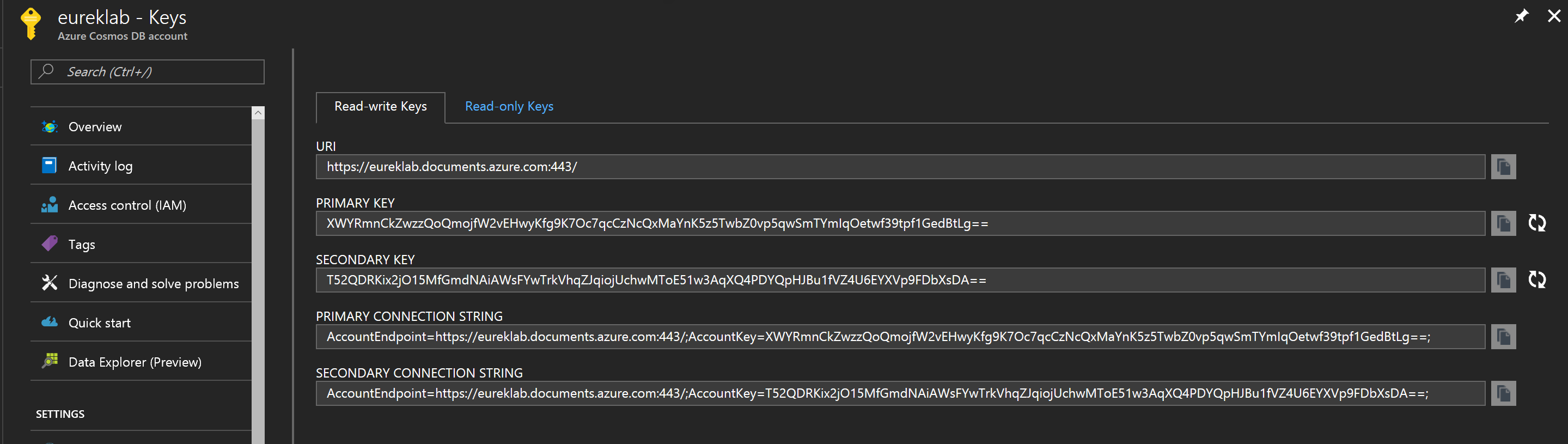






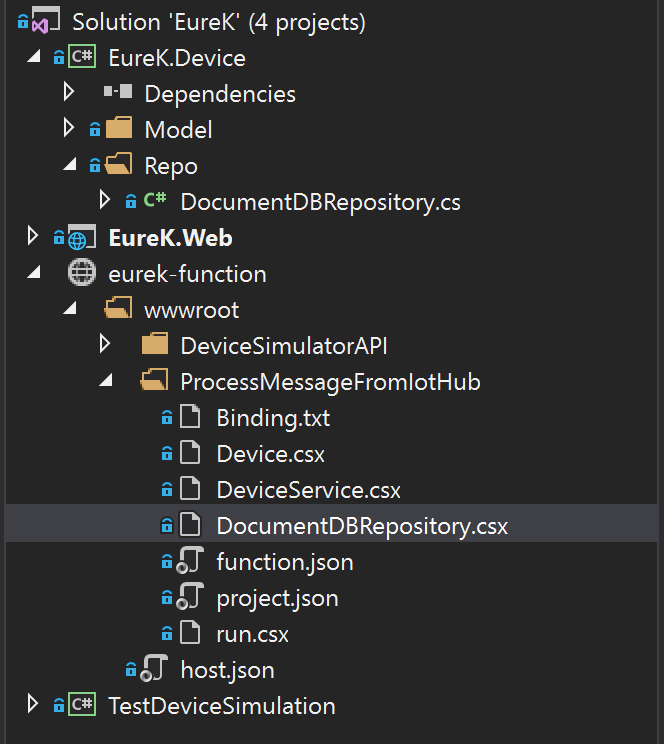
Nous allons mettons changer le code de la fonction qui permet de lire et d’écrire dans cette base de données.

Dans CosmosDB, allez dans « Keys »



Nous aurons besoin de l’URI et de l’une des clés (Primary Key ou Secondary key).

Dans Visual Studio



Remplacez le code

|  |
| --- |
| private static string enpoint = "https://<Your CosmosDB Name>.documents.azure.com:443/";  private static string authKey = "<Your CosmosDB Key>"; |

Par ce que vous avez dans votre CosmosDB :

|  |
| --- |
| private static string enpoint = "https://eureklab.documents.azure.com:443/";  private static string authKey = "XWYRmnCkZwzzQoQmojfW2vEHwyKfg9K7Oc7qcCzNcQxMaYnK5z5TwbZ0vp5qwSmTYmIqOetwf39tpf1GedBtLg=="; |

Nous allons ensuite créer notre fonction et y copier le code (comme dans le Lab 1 – Partie 2).

### Création de la fonction

Comme nous allons créer une fonction qui se déclenche quand un message arrive sur notre IoTHub, nous avons besoin d’un « endpoint » spécifique de notre IoTHub.

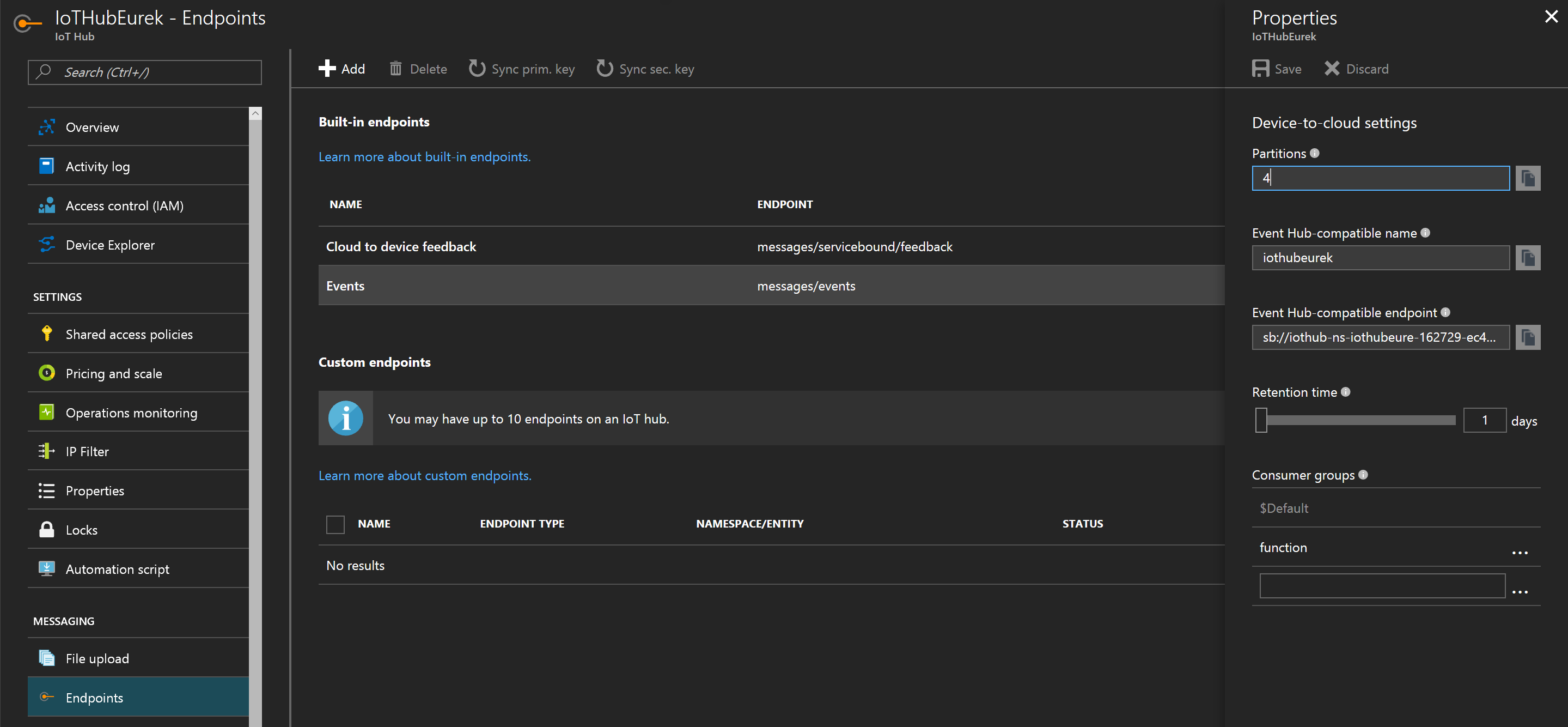
#### Recupération du Enpoint

Nous n’allons pas trouver le endpoint exactement comme la fonction l’attend. On va être obligé de recomposer ce endpoint pour la fonction en recomposant 3 parties.

Le endpoint final doit ressembler à ça :

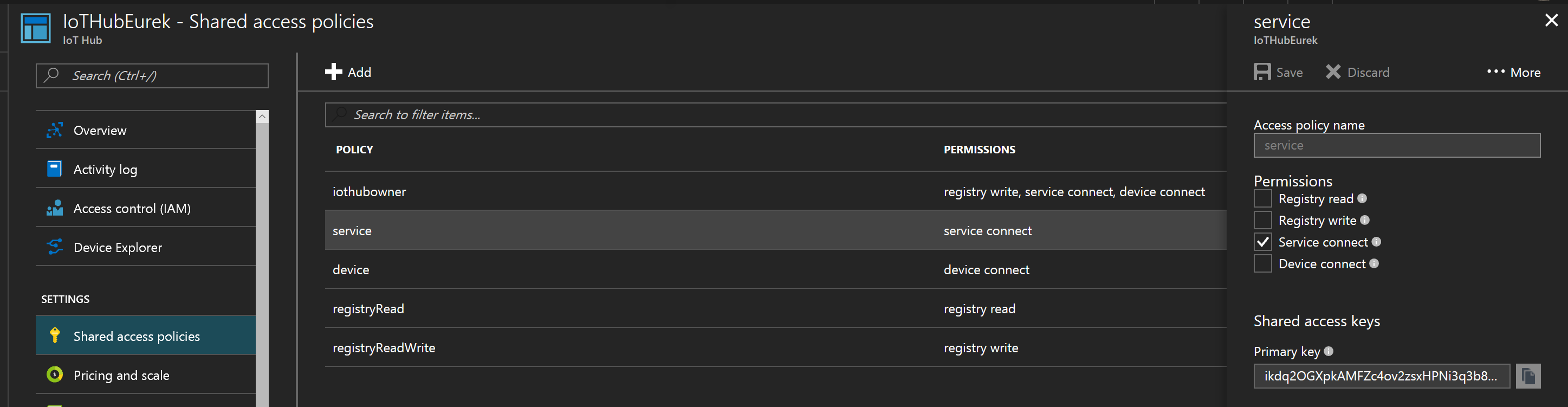
|  |
| --- |
| Endpoint=sb://<iotHub>.servicebus.windows.net;SharedAccessKeyName=service;SharedAccessKey=<Key> |

**1er partie du endpoint dans l’IoTHub**



Endpoint : **sb://iothub-ns-iothubeure-162729-ec4cf99f77.servicebus.windows.net/**

2ème partie du endpoint est la clé partagée :

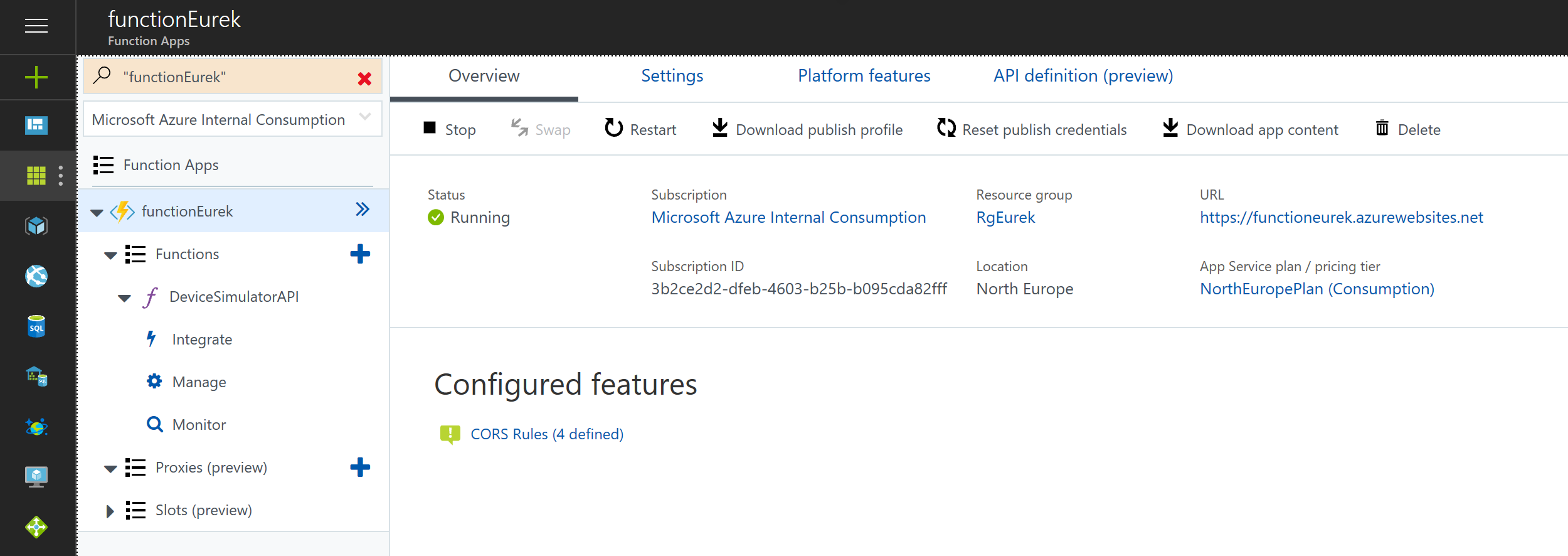


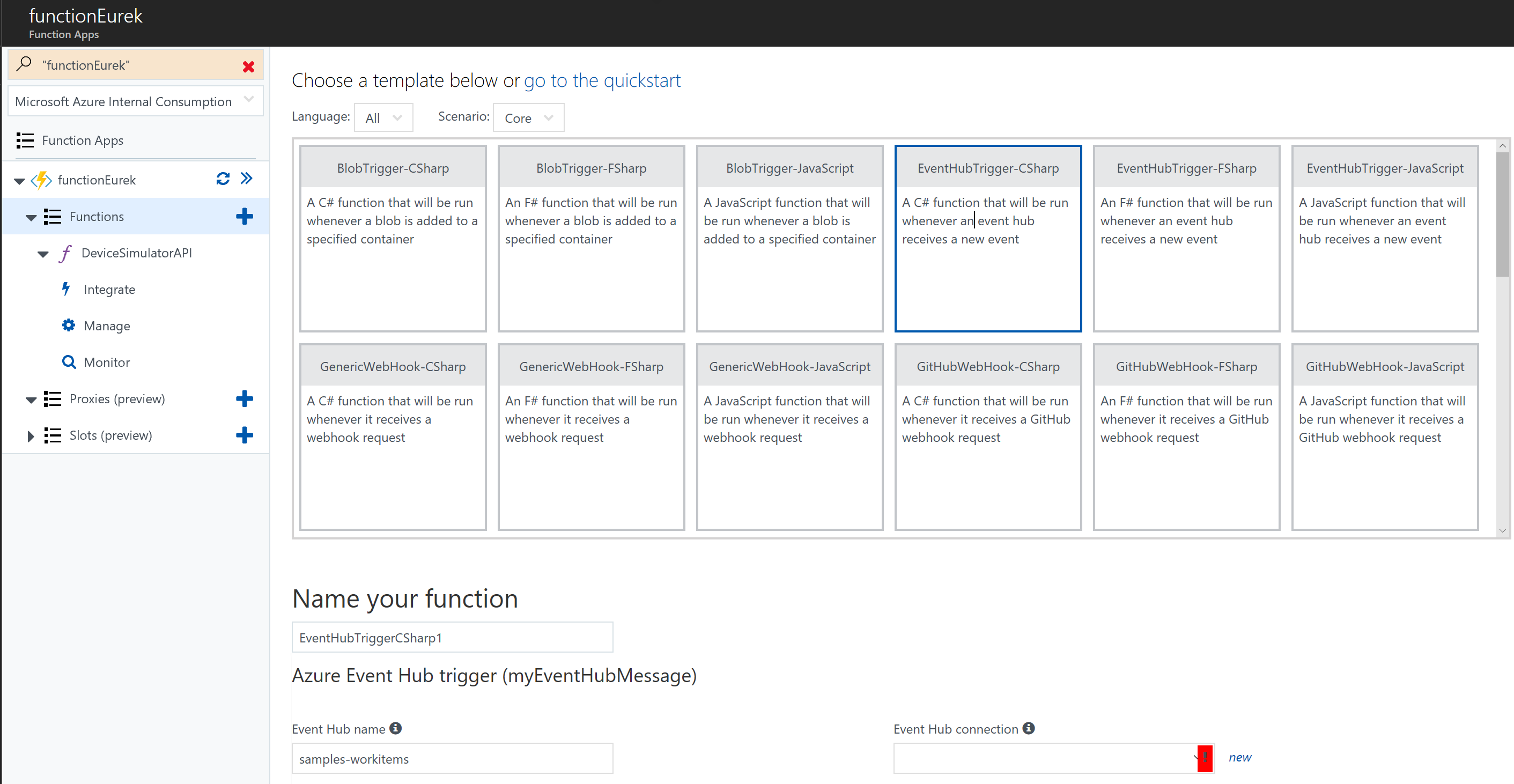
Key : ikdq2OGXpkAMFZc4ov2zsxHPNi3q3b8r+7vnvlknvrE=

Si je recompose mon endpoint :

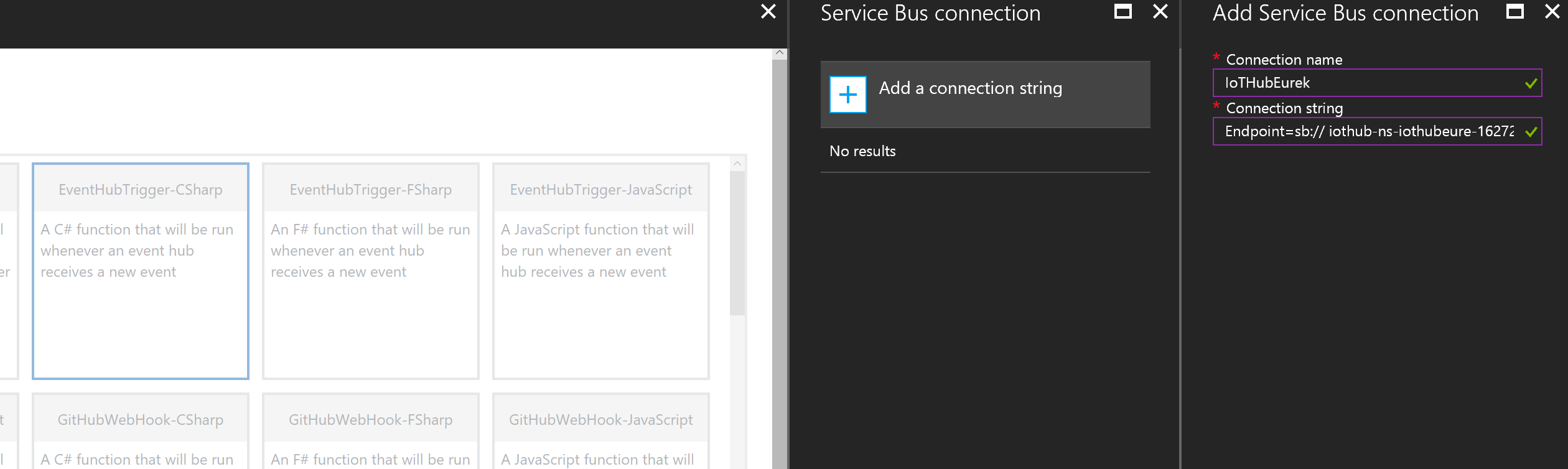
|  |
| --- |
| Endpoint=sb://**iothub-ns-iothubeure-162729-ec4cf99f77**.servicebus.windows.net;SharedAccessKeyName=service;SharedAccessKey=**ikdq2OGXpkAMFZc4ov2zsxHPNi3q3b8r+7vnvlknvrE=** |

#### Création de la fonction

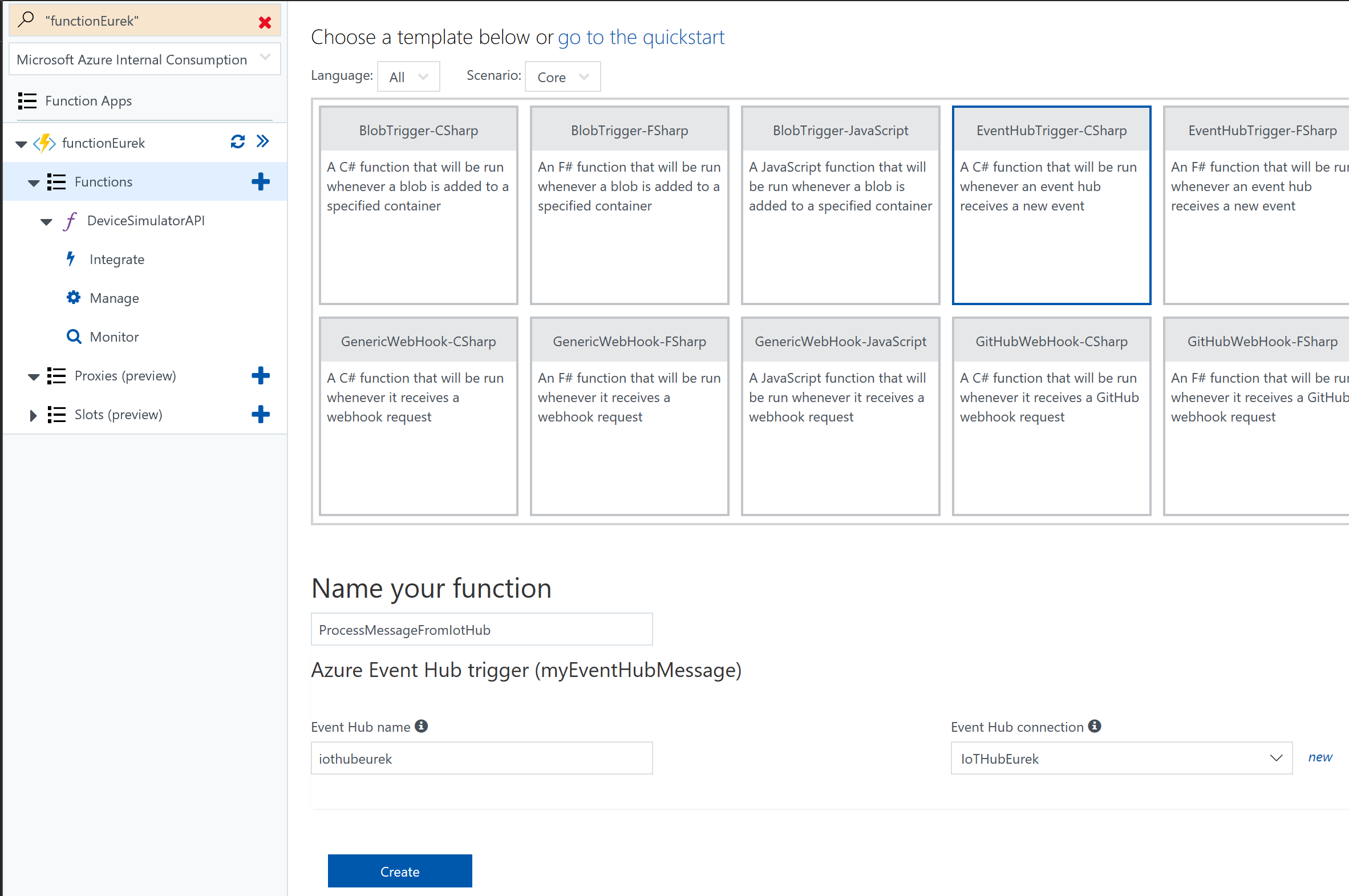




* Connection Name : IoTHubEurek
* Connection String : Endpoint=sb://iothub-ns-iothubeure-162729-ec4cf99f77.servicebus.windows.net;SharedAccessKeyName=service;SharedAccessKey=ikdq2OGXpkAMFZc4ov2zsxHPNi3q3b8r+7vnvlknvrE=



* Name your function : ProcessMessageFromIotHub
* Event Hub Name : iothubeurek

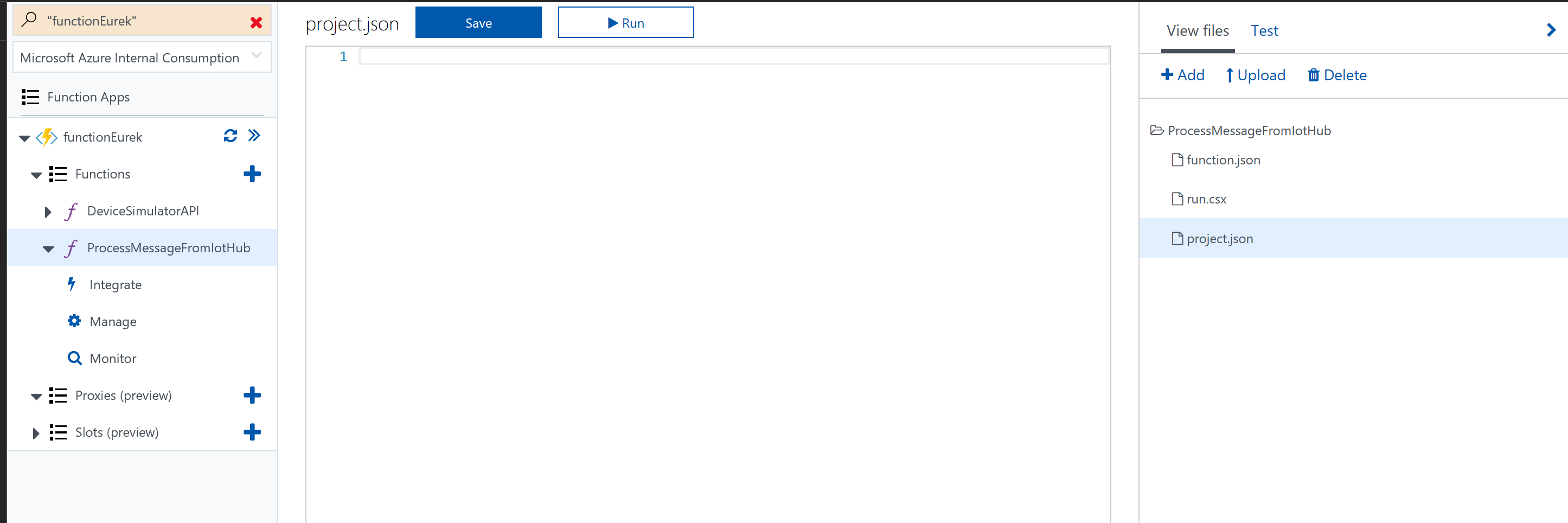


Il est fort probable que les messages envoyés dans nos tests précédents déclenchent notre fonction.

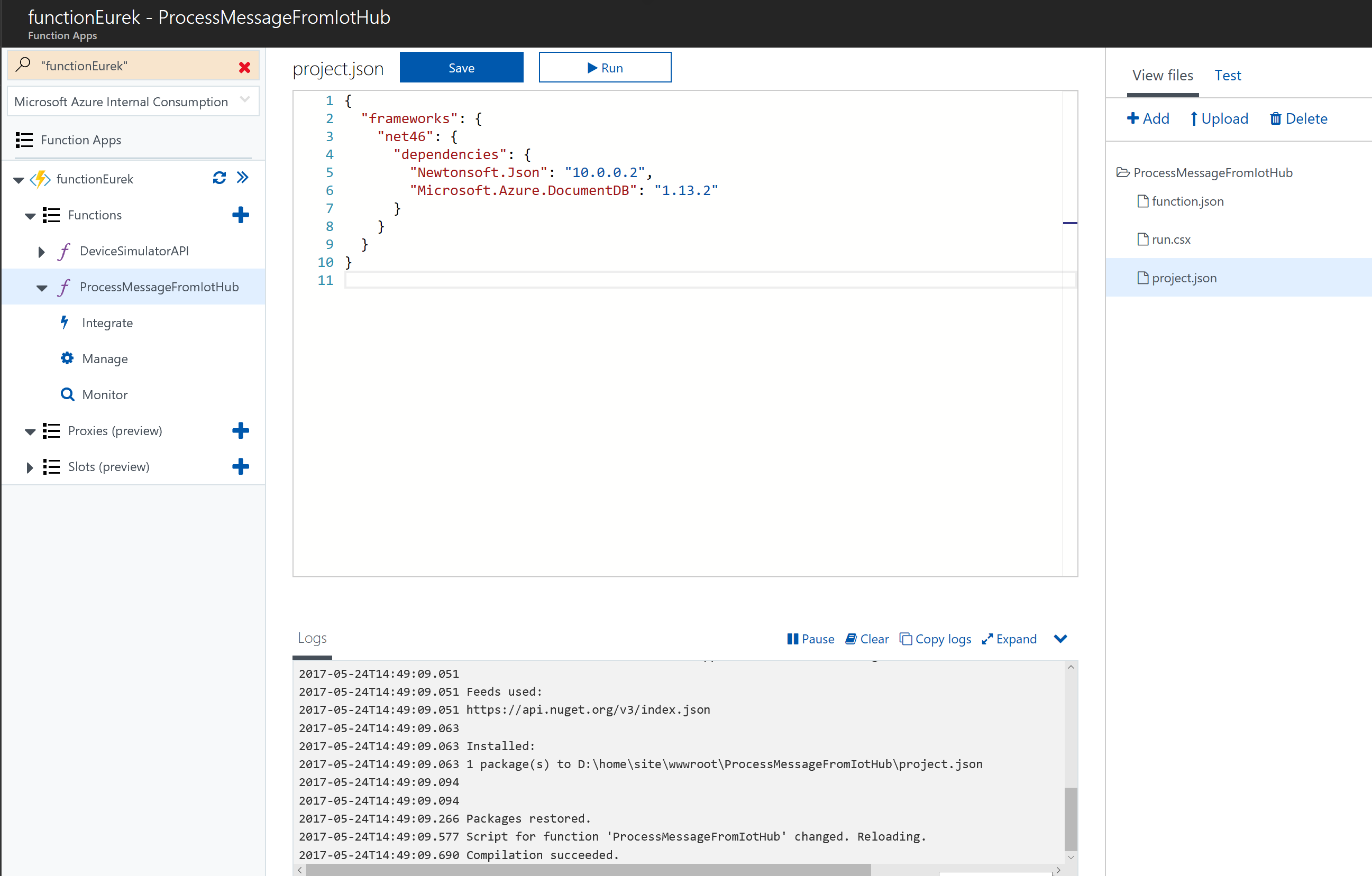


# Modifier la fonction

Commençons par créer le fichier « project.json » dans ma fonction dans Azure.



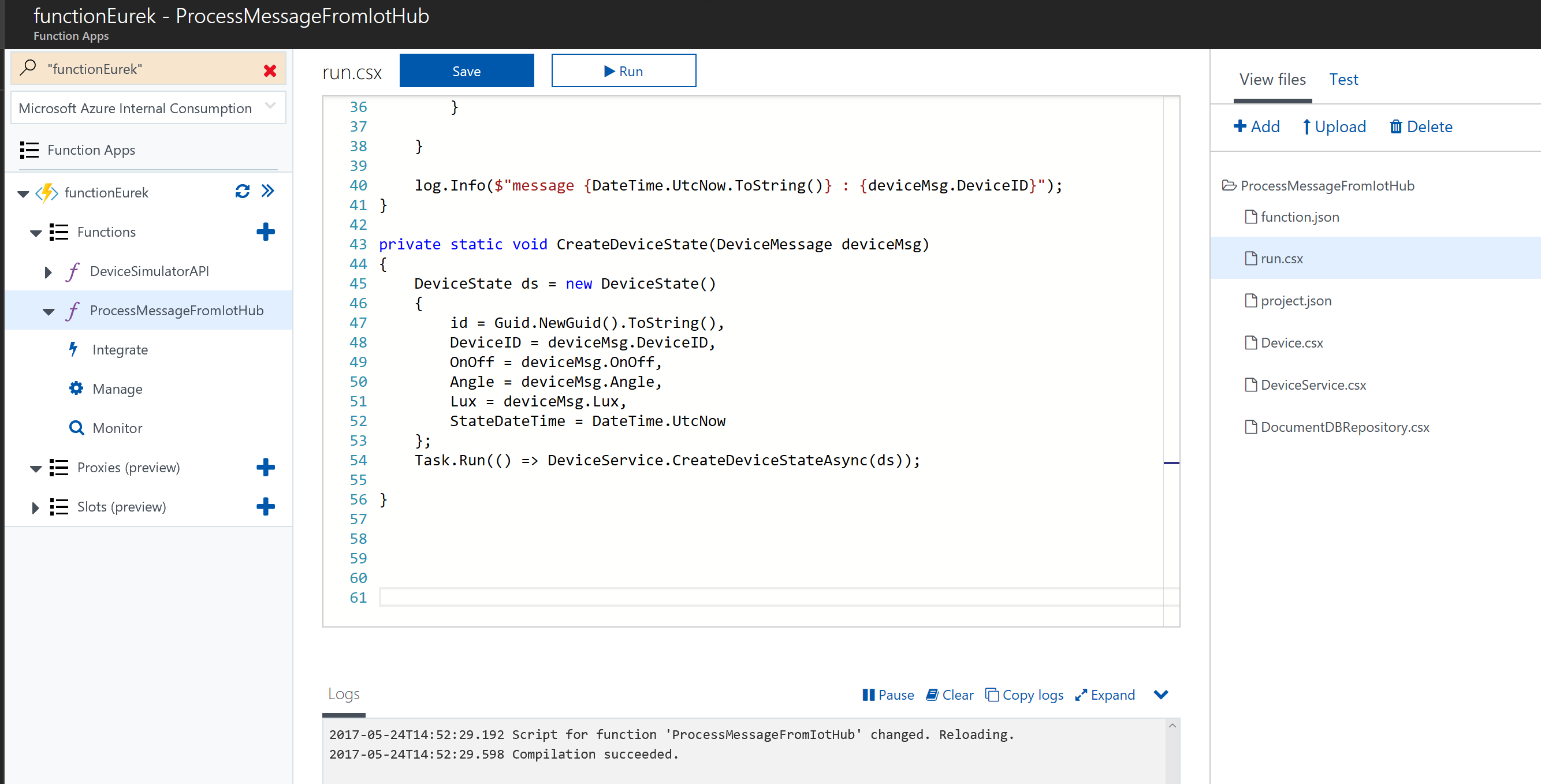
Et ensuite on va copier le contenue du fichier « project.json » du gitHub vers Azure.



Ensuite, on va faire pareil pour :

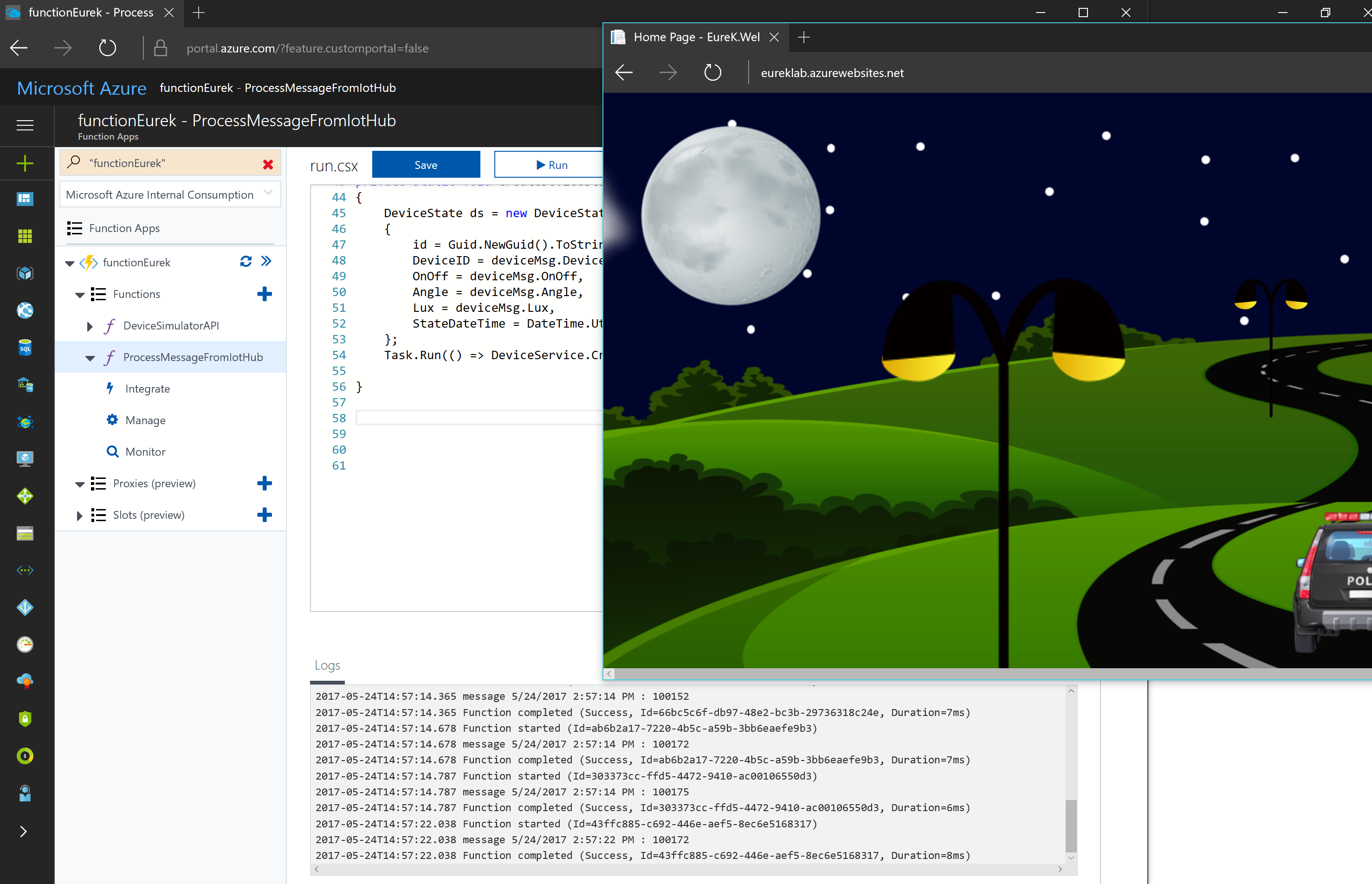
* Device.csx
* DeviceService.csx
* DocumentDBRepository.csx

Et enfin on va modifier (copier/coller) le code pour le fichier « Run.csx »



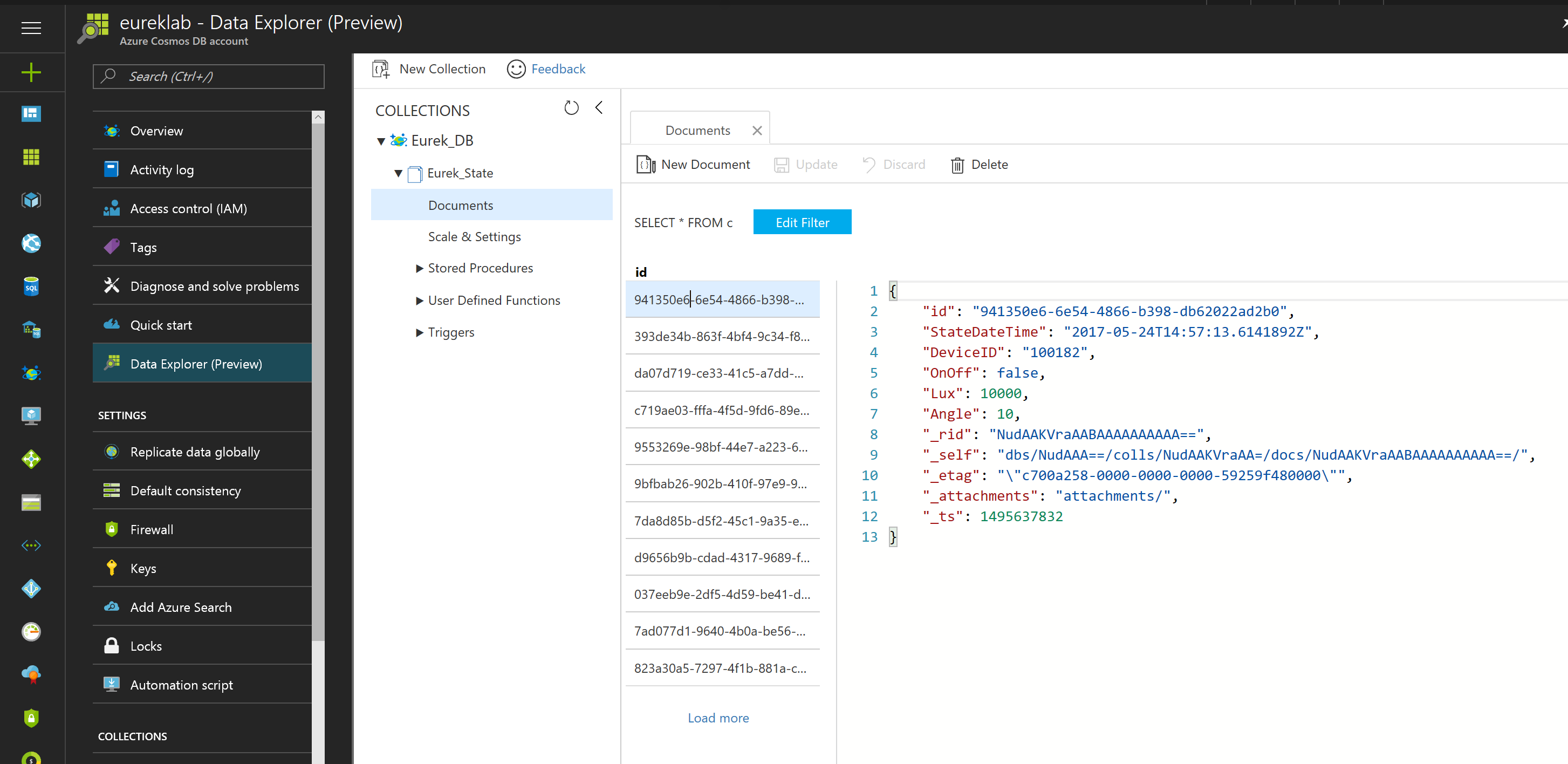
# TEST

Nous allons tester avec le site Web :



On voit bien les messages passer par notre fonction.

Allons vérifier la Base de données CosmosDB :



On y voit bien des documents qui contiennent tous l’état des devices.

# Test avec l’application console

Pour envoyer plus de mesure, vous pouvez utiliser l’application console « TestDeviceSimulation ». Vous devez uniquement modifier ces 2 paramètres :

private static string iotHubUri = "<Your IotHub Name>.azure-devices.net";

private static string connectionString = "HostName=<Your IotHub Name>.azure-devices.net;SharedAccessKeyName=registryReadWrite;SharedAccessKey=<Your SAK>";

Nous avons déjà vu comment dans le « Lab1 – Partie 2 – FunctionApp.docx » comment faire.