

Documentation de test

Fréquence de lecture

Table des matières

1. Contexte Python – IoT.....	2
2. Fonctionnalité.....	2
3. Code.....	2
4. Tests.....	2
4.1 Test avec une fréquence non valide.....	2
4.2 Test fréquence valide.....	3

1. Contexte Python – IoT

Nous devons réaliser des solutions avec l'Internet des Objets (IoT) pour simplifier la gestion des entrepôts de stockage ou faire des économies d'énergie de notre client.

Dans la partie python, nous devons lire en fonction du fichier de configuration crée par Java des données à l'aide du capteur AM107.

2. Fonctionnalité

Nous devons gérer la lecture du bus à une certaine fréquence correspondant au paramétrage du fichier de configuration

3. Code

```
def get_data(mqtt, obj, msg):
    jsonMsg = json.loads(msg.payload)
    now = datetime.datetime.now()
    date = now.strftime("%Y-%b-%d %H-%M-%S")
    for e in config["object"]:
        alert = False
        if config["object"][e]:
            if e == "activity":
                if config["seuils"][e] is not None:
                    if config["seuils"][e] < jsonMsg["object"][e] and (now.hour > 18 or now.hour < 6):
                        alert = True
            elif e == "co2" or e == "humidity" or e == "temperature":
                if config["seuils"][e][0] is not None:
                    if config["seuils"][e][1] is not None and config["seuils"][e][1] < jsonMsg["object"][e] or \
                        config["seuils"][e][0] > jsonMsg["object"][e]:
                        alert = True
            else:
                if config["seuils"][e] is not None:
                    if config["seuils"][e] < jsonMsg["object"][e]:
                        alert = True
        if alert:
            with open(config["name"] + "_alert_" + e + "_donnees.txt", "a", encoding="utf-8") as file:
                file.write(str(jsonMsg["object"][e]) + ":" + date + "\n")

            with open(config["name"] + "_" + e + "_donnees.txt", "a", encoding="utf-8") as file:
                file.write(str(jsonMsg["object"][e]) + "\n")

    sleep(config["frequency"])
```

A la fin de la fonction get_data, nous mettons un sleep en fonction de la fréquence qui suspend le code python, nous obtiendrons donc les données toutes les x temps fréquences exprimés en secondes.

4. Tests

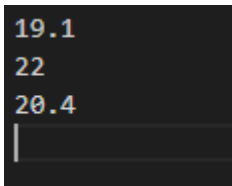
4.1 Test avec une fréquence non valide

On test avec une fréquence non valide, c'est-à-dire inférieur à 0

```
ValueError: sleep length must be non-negative
```

Normalement, nous ne pouvons pas écrire une fréquence inférieur à 0 dans le fichier de configuration car l'interface Java pourra interdire ça.

4.2 Test fréquence valide



```
19.1
22
20.4
|
```

Ce traitement est le même pour toutes les types de données