

Documentation de test

Écrire la / les valeurs dans un / des fichiers

Table des matières

1. Contexte Python – IoT.....	2
2. Fonctionnalité.....	2
3. Code.....	2
4. Tests.....	3
4.1 Sans un type de données.....	3
4.2 Avec zéro type de données.....	4
4.3 Avec tous les types de données.....	4

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 écrire la / les valeurs lues dans un / des fichiers et avoir le format UTF-8 afin que Java puisse lire ces fichiers.

3. Code

Les types de données sont choisis dans le fichier de configuration en attribuant « true » ou « false » au type.

```
object:
  activity: false
  co2: true
  humidity: true
  illumination: true
  infrared: true
  infrared_and_visible: true
  pressure: true
  temperature: true
  tvoc: true
```

Le traitement est le même pour tous les types de données

```
if config["object"]["activity"]:
    if config["seuils"]["activity"] is not None and config["seuils"]["activity"] < jsonMsg[
        if now.hour > 18 or now.hour < 6:
            print("[Activité détecté en dehors des horaires!]")
    with open(config["name"] + "_activity_donnees.txt", "a", encoding="utf-8") as file:
        file.write(str(jsonMsg["object"]["activity"]) + " PIR\n")
    file.close()
```

Nous regardons si la valeur est « true » ou « false », si la verification est « true », on écrit dans un fichier qui a pour nom le nom que nous avons choisi dans le fichier de configuration, on écrit « a » pour append, ajouter à la suite, pour pouvoir écrire plusieurs fois si la fréquence est supérieur à 1, et on encode en utf-8. Nous écrivons ensuite dans le fichier en format de chaîne de caractères la valeur qui est dans le jsonMsg cf. DocTest_2.pdf puis son ordre de grandeur et un saut de ligne.

Et quand nous voulons exécuter encore un fois le code, les anciens fichiers sont supprimés si les noms sont similaires

```

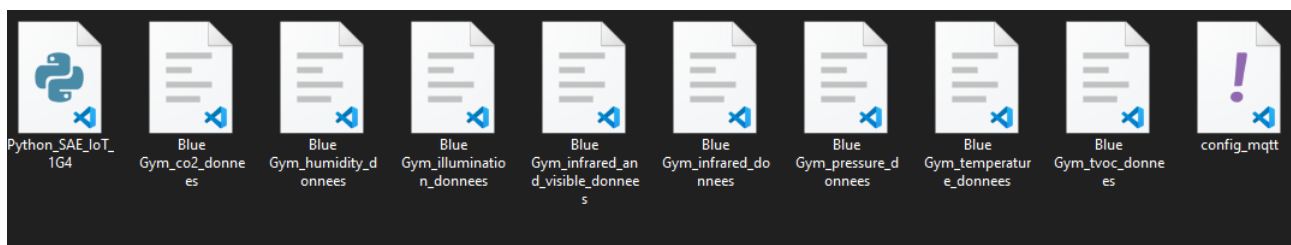
✓ try:
|     os.remove(config["name"] + "_activity_donnees.txt")
✓ except OSError:
|     pass
✓ try:
|     os.remove(config["name"] + "_co2_donnees.txt")
✓ except OSError:
|     pass
✓ try:
|     os.remove(config["name"] + "_humidity_donnees.txt")
✓ except OSError:
|     pass
✓ try:
|     os.remove(config["name"] + "_illumination_donnees.txt")
✓ except OSError:
|     pass
✓ try:
|     os.remove(config["name"] + "_infrared_donnees.txt")
✓ except OSError:
|     pass
✓ try:
|     os.remove(config["name"] + "_infrared_and_visible_donnees.txt")
✓ except OSError:
|     pass
✓ try:
|     os.remove(config["name"] + "_pressure_donnees.txt")
✓ except OSError:
|     pass

```

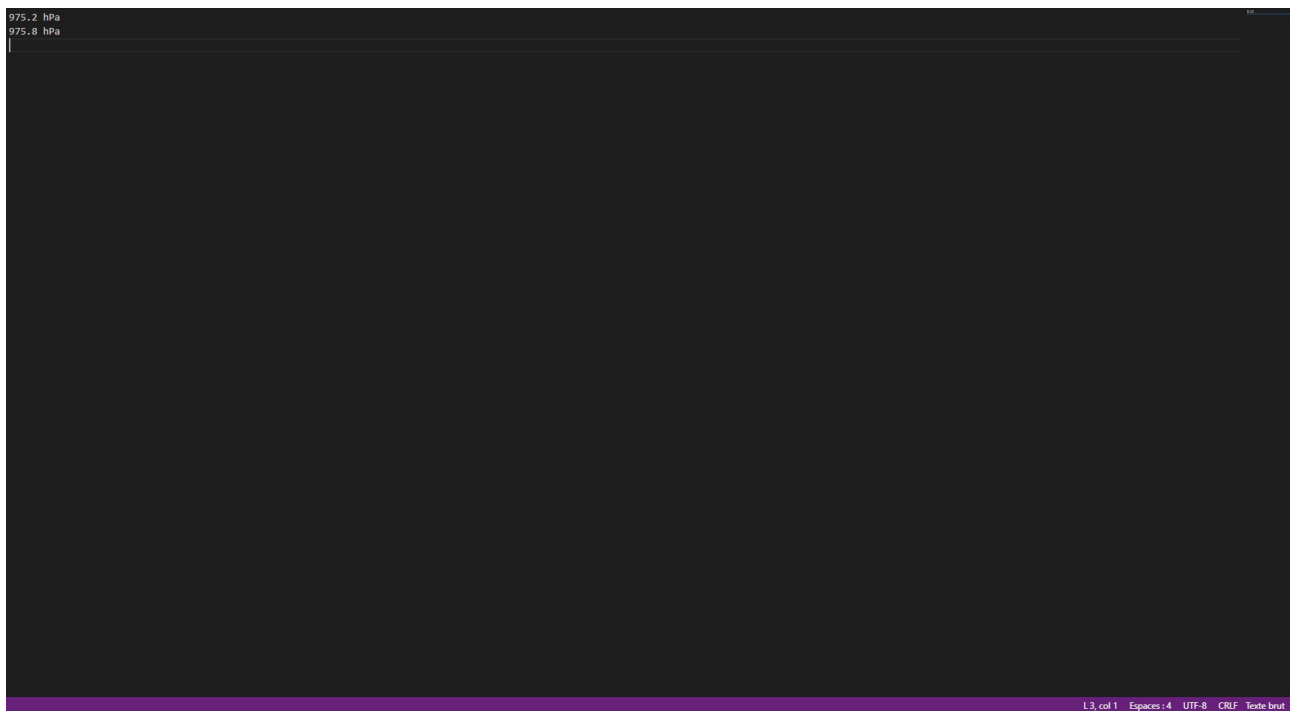
4. Tests

4.1 Sans un type de données

Nous testons en mettant la donnée activity en false



Nous pouvons voir qu'il y a toutes les données à part l'activité.



Nous pouvons voir les valeurs de la pression que le format du fichier .txt est utf-8.

4.2 Avec zéro type de données










Nous testons en mettant tous les types de données en « false » dans le fichier de configuration.

```
if((config["object"]["temperature"] == False and config["object"]["tvoc"] == False and config["object"]["pressure"] == False and config["object"]["infrared_and_visible"] == False and config["object"]["infrared"] == False)):  
    client.disconnect()
```

Si tout est en false, nous nous déconnectons.

4.3 Avec tous les types de données

Nous mettons tout en « true » dans le fichier de configuration.

	Blue Gym_activity_donnees	09/12/2022 15:39	Fichier source Text	1 Ko
	Blue Gym_co2_donnees	09/12/2022 15:39	Fichier source Text	1 Ko
	Blue Gym_humidity_donnees	09/12/2022 15:39	Fichier source Text	1 Ko
	Blue Gym_illumination_donnees	09/12/2022 15:39	Fichier source Text	1 Ko
	Blue Gym_infrared_and_visible_donnees	09/12/2022 15:39	Fichier source Text	1 Ko
	Blue Gym_infrared_donnees	09/12/2022 15:39	Fichier source Text	1 Ko
	Blue Gym_pressure_donnees	09/12/2022 15:39	Fichier source Text	1 Ko
	Blue Gym_temperature_donnees	09/12/2022 15:39	Fichier source Text	1 Ko
	Blue Gym_tvoc_donnees	09/12/2022 15:39	Fichier source Text	1 Ko

Nous avons bien tous les types de données en format .txt.