**I. Architecture :**



3- config,js

webviewer/frontend/src/common/

4- Fichiers json :

schedule.json

state.json

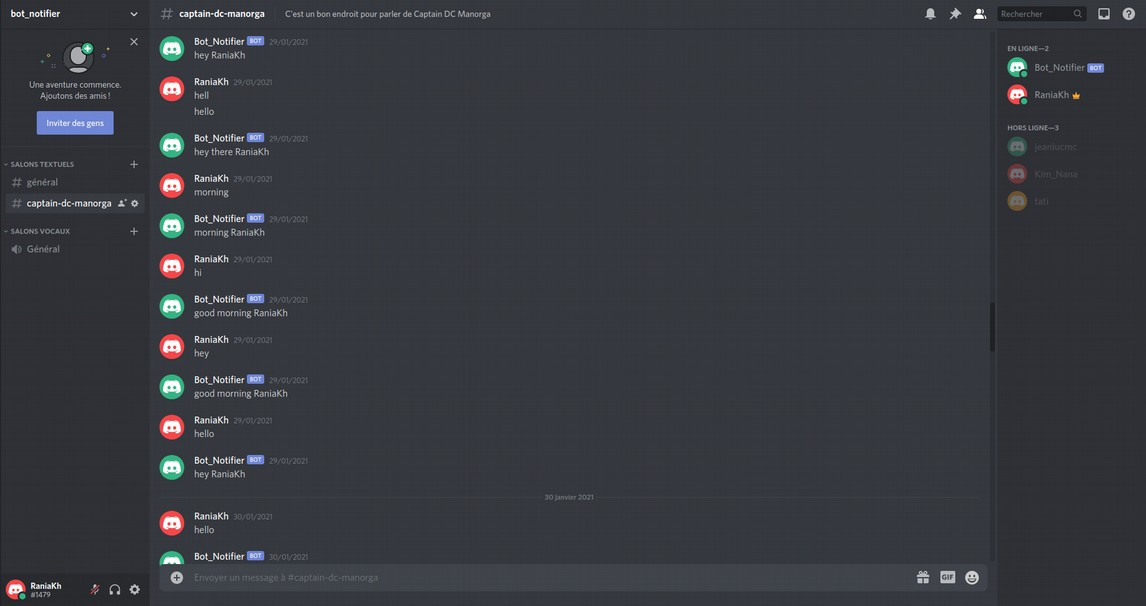
5-WebHook

(Dans notre cas

CDCHook)



**1- Discord Application :** c’est l’interface où on discute



**2- Discord Server :**

ce serveur est écrit avec l’éditeur **repl.it** qui est un IDE en ligne.

Avantage de repl.it : permet d’avoir **une adresse ip public** (serveur hébergé gratuitement) .

Cette adresse est utilisée dans **uptimerobot.com**

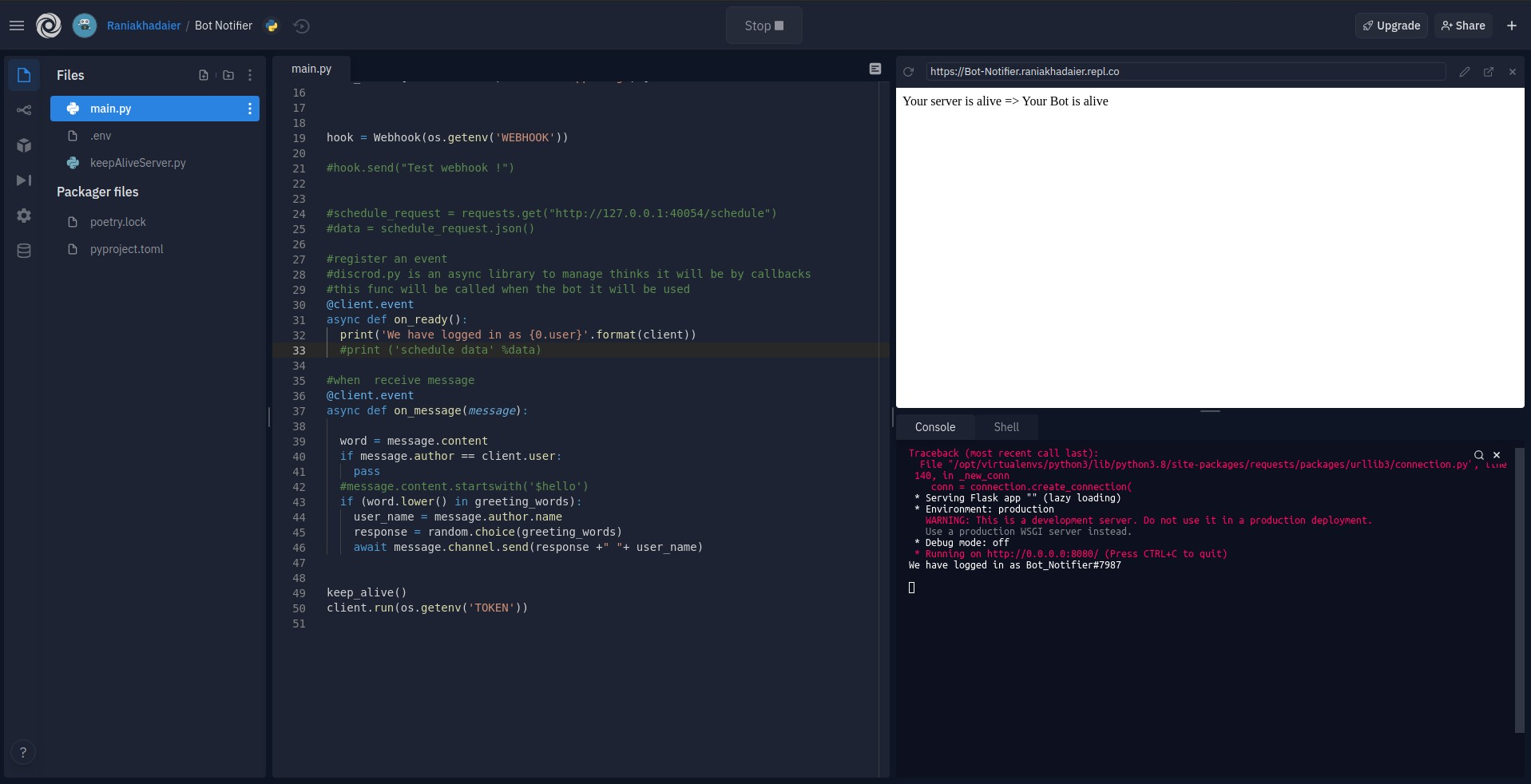
**uptimerobot.com** va nous servir à maintenir notre bot disponible 24/24 (avec un refresh chaque 5

minutes). En utilisant ces deux outils mentionnés on garantit la disponibilité du Bot Notifier.

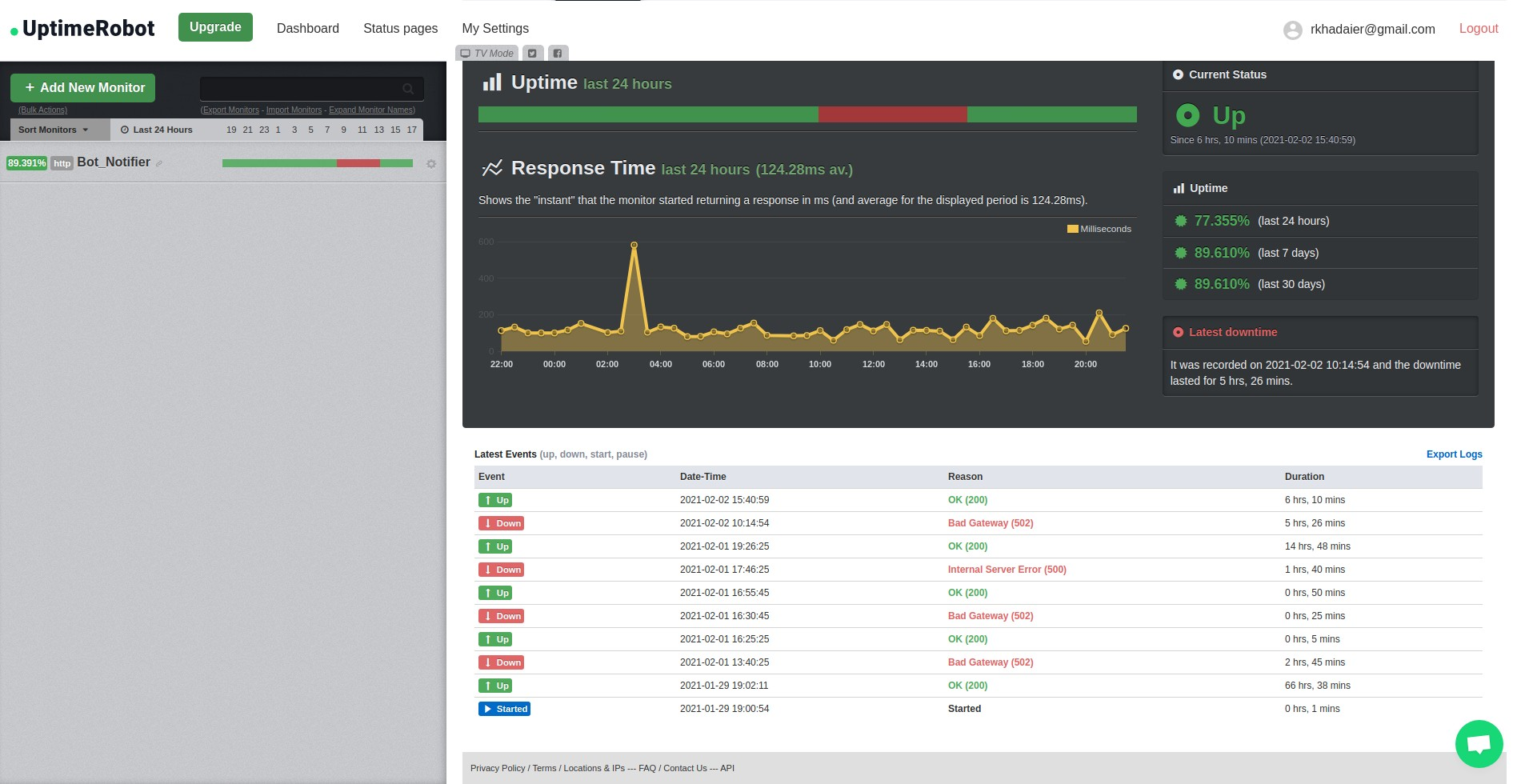
Espace de repl.it : https://repl.it/@Raniakhadaier/Bot-Notifier#main.py

UptimeRobot.com : <https://uptimerobot.com/>

**Espace repl.it :**

****

**Compte UptimeRobot :**

****

**3- Fichier config,js :**

Ce fichier serre à récupérer les seuils de températures , humidité et disque.

Ce fichier se trouve sous le répertoire : « webviewer/frontend/src/common/config.js »

**4- Fichiers JSON :**

Le Rest Server stocke les résultats de ses requêtes dans des fichiers JSON. Dans notre cas , on va utiliser deux fichiers.

**schedule.json :** afin de récupérer les dates de début/fin de l’événement.

**state.json :** afin de récupérer : le status du robot les valeurs de température , humidité , batterie et valeur d’espace vide de disque.

NB : on va tester sur CDC Manorga pour vérifier le choix sinon on passe à utiliser les topics ROS au lieu des fichiers JSON

**5- Webhook :**

qui se produit lorsque quelque chose se produit côté serveur ; une simple notification d'événement via HTTP POST.

**6- Structure des notifications :**

**\*\*\*Notification de début d’événement\*\*\***

**03/02/2021 09H00 => Date et heure de début**

**Failed: Next start AUTOPILOT NOW END 09H30 => récupération de l’état du robot si Autopilot OK sinon Failed**

**Failed: ROBOT STATUS AUTOPILOT**

**WARNING : Batterie status 0% minimum 10%**

**WARNING : Temperature 10.1°C bellow minimum 15 °C**

**WARNING : Hydrometry 51% above max value 40%**

**=> récupération des valeurs de batterie/température/humidité avec leurs seuils : si les valeurs respectent les seuils donnés OK sinon WARNING**

**\*\*\*Notification de fin de l’événement avec Status DOCKING\*\*\***

**26/01/2021 17H30 => Date et heure de fin**

**OK: Next start AUTOPILOT NOW => récupération de l’état du robot avant DOCKING si Autopilot OK sinon Failed**

**OK: ROBOT STATUS changed to DOCKING => récupération de l’état du robot si DOCKING OK sinon Failed**

**WARNING : Batterie status 0% below minimum 10%**

**WARNING : Temperature 8.1°C bellow minimum 15 °C**

**WARNING : Hydrometrie 47% above max value 40%**

**=> récupération des valeurs de batterie/température/humidité avec leurs seuils : si les valeurs respectent les seuils donnés OK sinon WARNING**

**\*\*\*Notification avec le status DOCKED\*\*\***

**26/01/2021 17H30 => Date et heure où le robot est bien positionné sur la base de recharge**

**Next start AUTOPILOT 27/01/2021 09H30 => indiquer le prochain événement**

**OK: ROBOT STATUS changed to DOCKED => récupération de l’état du robot si DOCKED OK sinon Failed**

**OK : Batterie status 100%**

**WARNING : Temperature 8.2°C bellow minimum 15 °C**

**WARNING : Hydrometrie 47% above max value 40%**

**=> récupération des valeurs de batterie/température/humidité avec leurs seuils : si les valeurs respectent les seuils donnés OK sinon WARNING**

**II.** **Exigences :**

1- Installation de **Python 3.5** comme version minimale pour le fonctionnement de Webhook discord:

https://www.python.org/downloads/release/python-3510/

https://tecadmin.net/install-python-3-5-on-ubuntu/

2- Installation de **dhooks** pour interagir avec les webhooks Discord à l'aide de python :

https://pypi.org/project/dhooks/

3-Installation de **Threading** pour faire l’appel à la fonction d’envoi des notifications d’une façon répétitive avec un timer pour ne pas rater aucun événement de schedule :

https://pypi.org/project/threaded/#description

**III. Sprint 26 Avril 2021 – 10 Mai 2021 :**

**User Stories :**

1-En tant que User je veux recevoir la notification de début d'un event : TO VERIFY=> Test sur CDC Manorga  
2-En tant que User je veux recevoir la notification de fin de l'event avec state DOCKING : TO VERIFY=> Test sur CDC Manorga  
3-En tant que User je veux recevoir la notification de fin de l'event avec state DOCKED   
4-En tant que User je veux savoir dans chaque notification les seuils de Batterie /Température /Humidité /Disque

**IV. Next Step :**

Intégration de tensorflow/dialogflow avec Bot de Discordafin d’avoir un chatbot intelligent qui peut interagir avec les membres de la chaîne Discord (A négocier)