RAPPORT SAE 23 – Partie Collecte

DESCAMPS Nathan, WAGNER Nicolas et LOSSER Julien

# Table des Matières

[Table des Matières 1](#_Toc106961217)

[1) MQTTBOX 1](#_Toc106961218)

[a) Installation 1](#_Toc106961219)

[b) Mise en place du browser & du topic 1](#_Toc106961220)

[1er. MQTT LOAD 2](#_Toc106961221)

[2e. MQTT CLIENT 3](#_Toc106961222)

[2) Python 4](#_Toc106961223)

[a) Csv 5](#_Toc106961224)

# MQTTBOX

## Installation

Installation depuis Windows Store pour être sûr de la bonne compatibilité

[MQTTBox – Applications du Microsoft Store](https://apps.microsoft.com/store/detail/mqttbox/9NBLGGH55JZG?hl=fr-fr&gl=FR)

## Mise en place du browser & du topic

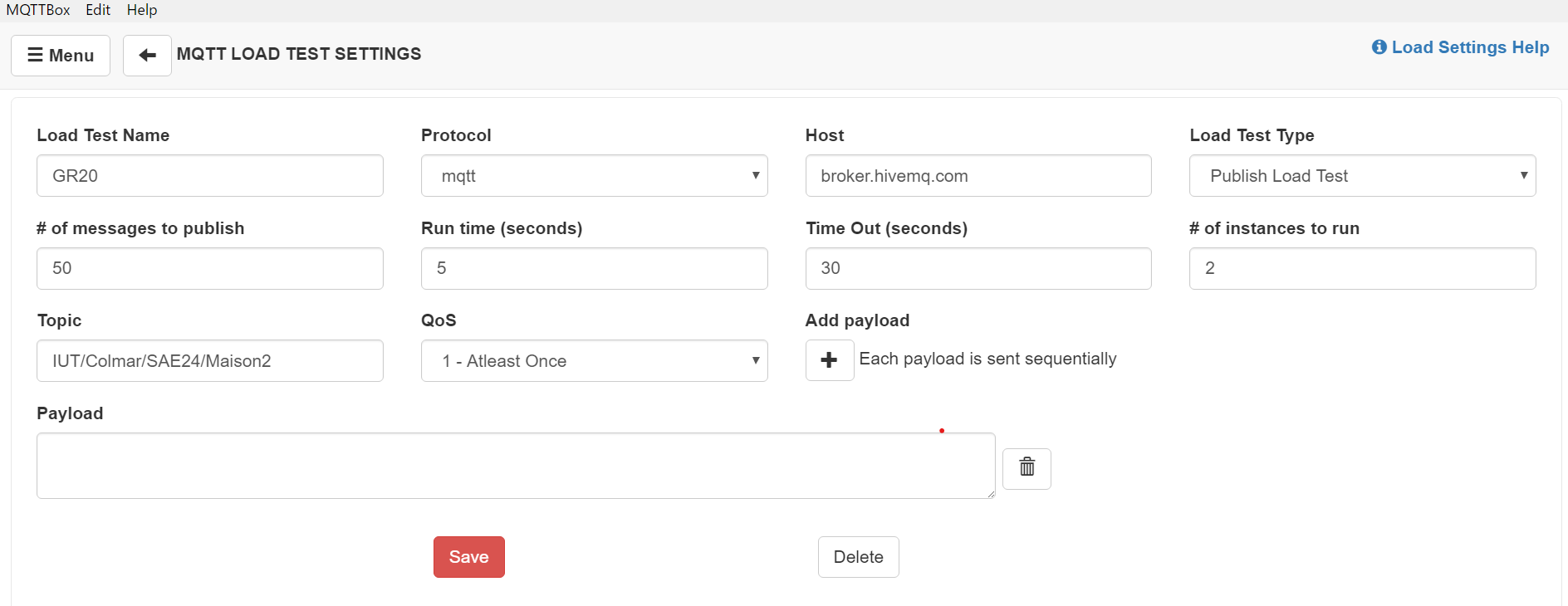
•Broker : broker.hivemq.com

•Topic : IUT/Colmar/SAE24/Maison2

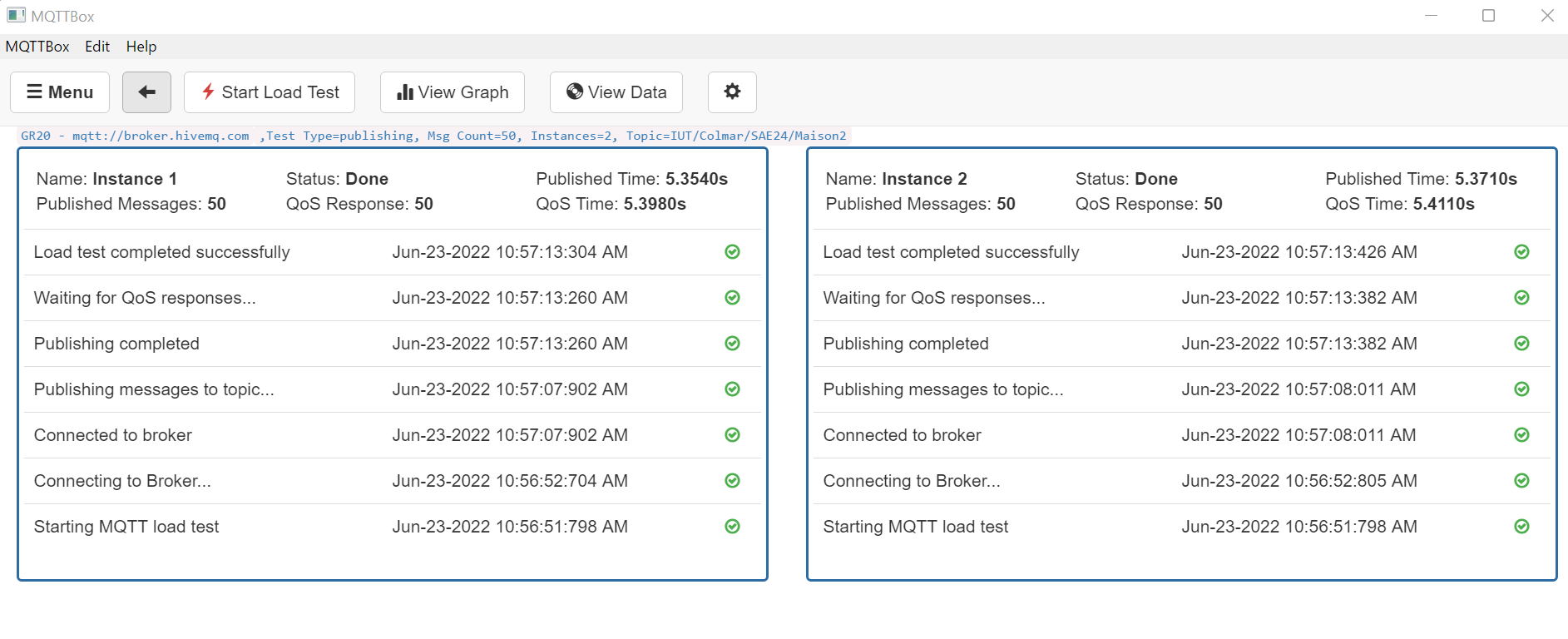
### MQTT LOAD

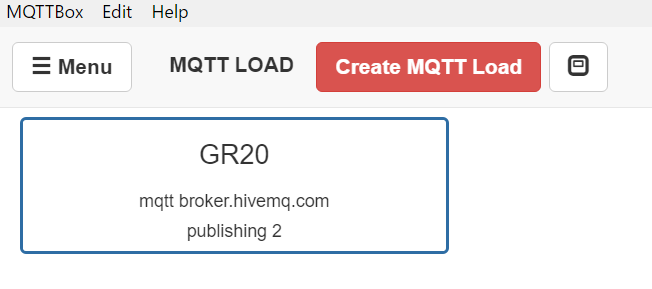
Il faut mettre le broker que vous voulez utiliser dans « Host »

Il faut sélectionner le protocol « MQTT »



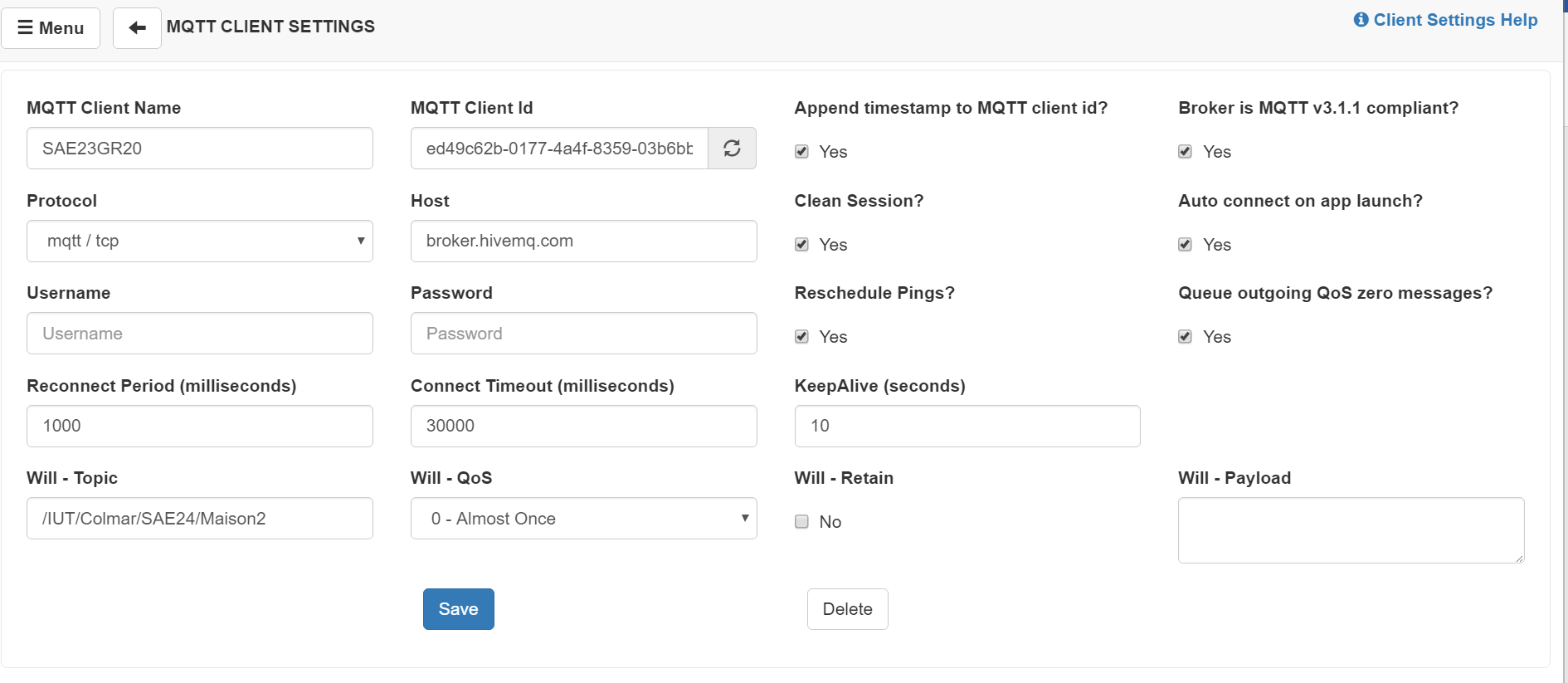
Vous pouvez constater que la connections au broker a bien été effectuer et qu’il y a eu aucune erreur lors de la connections.

Lorsqu’on sort du menu précèdent, nous avons un lien rapide vers notre configuration.

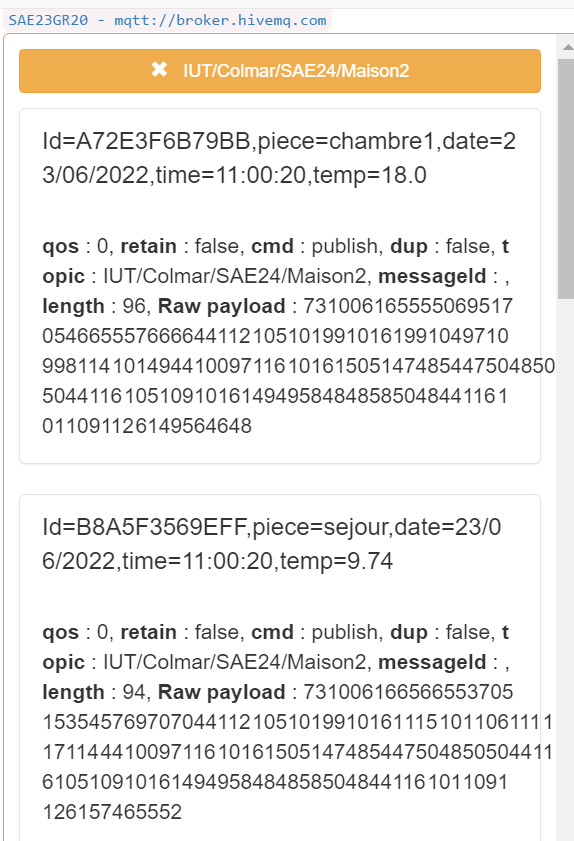


Nous pouvons voir, quelques informations comme le Nom, que nous avons donné.  
Le protocol utilisé et le broker.

### MQTT CLIENT

Comme précédemment , nous devons entrer le protocol mqtt/tcp, le broker dans l’onglet Host et le topic.

N’oubliez pas de « Save » avant de fermer la page.



Vous pouvez voir le nom que nous avons donné, le protocol mqtt avec le broker utiliser.

On est correctement connecté au topic « IUT/Colmar/SAE24/Maison2 »

Les informations publier sont afficher sous la forme id= #,piece=#,date=#,time=#,temp=#

Id corresponds à l’adresse MAC du capteur, cela est important vu qu’on dispose de 2 capteurs pour pouvoir les identifier comme l’adresse MAC est unique.

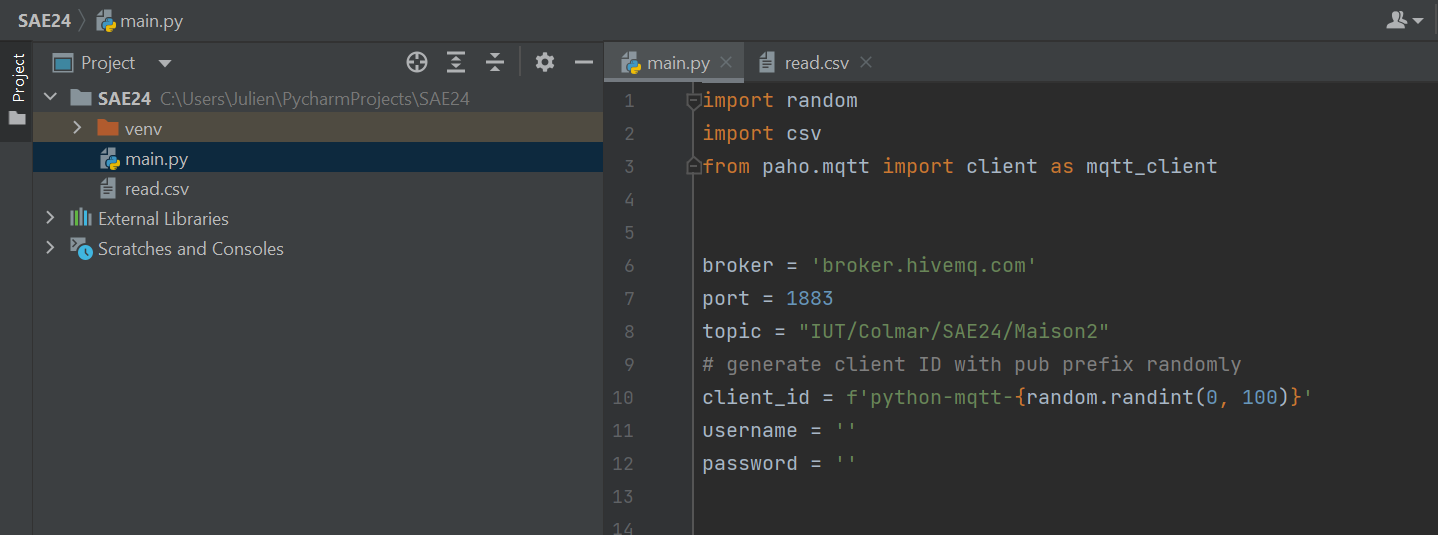
Piece correspond à la location des capteurs, car chaque capteur est dans un lieu différent, chambre1 ou séjour dans notre cas.

Date et time corresponds au moment de la prise de la capture de la température.

Temp corresponds à la température prise par le capteur.

# Python

Ci-dessus vous pouvez voir notre script python qui nous permet de récupère les données du MQTT et de créer un fichier csv avec ces données.  
Tout d’abord vérifier que vous avez python 3.6 et au-dessus. Pouvez utiliser la commande :   
python3 –version  
il faut installer dans le terminal la librairie « paho.mqtt » avec la commande  
pip3 install paho-mqtt  
Tout d’abord on identifier notre broker, puis le port 1883 et le topic.  
Nous n'avons pas besoin d’identifier l’user et le mot de passe, car notre broker/topic est en public et nous n'avons pas besoin de s’identifier pour y accéder.



import random  
import csv  
from paho.mqtt import client as mqtt\_client  
  
  
broker = 'broker.hivemq.com'  
port = 1883  
topic = "IUT/Colmar/SAE24/Maison2"  
# generate client ID with pub prefix randomly  
client\_id = f'python-mqtt-{random.randint(0, 100)}'  
username = ''  
password = ''  
  
  
def connect\_mqtt() -> mqtt\_client:  
 def on\_connect(client, userdata, flags, rc):  
 if rc == 0:  
 print("Connected to MQTT Broker!")  
 else:  
 print("Failed to connect, return code %d\n", rc)  
  
 client = mqtt\_client.Client(client\_id)  
 #client.username\_pw\_set(username, password)  
 client.on\_connect = on\_connect  
 client.connect(broker, port)  
 return client  
  
def traiter\_donnee(mac,piece,date,time,temp):  
 if mac == "B8A5F3569EFF":  
 chaine = f"{mac};{piece};{date};{time};{temp};"  
 file = open("capteur1.csv", "a")  
 file.write(chaine + "\n")  
 file.close()  
  
 if mac == "A72E3F6B79BB":  
 chaine = f"{mac};{piece};{date};{time};{temp};"  
 file = open("capteur2.csv", "a")  
 file.write(chaine + "\n")  
 file.close()  
  
 timestamp = date + " , " + time  
 global id\_donnee  
 id donnée += 1  
 global mycursor  
 # sql = "INSERT INTO data (id,data,timestamp,capteur\_id) VALUES (%s, %s, %s, %s)"  
 if id == 'A72E3F6B79BB':  
 val = (id\_donnee,temp,timestamp,2)  
 if id == "B8A5F3569EFF":  
 val = (id\_donnee, temp, timestamp,1)  
 # mycursor.execute(sql, val)  
  
 # mydb.commit()  
  
def subscribe(client: mqtt\_client):  
 def on\_message(client, userdata, msg):  
 print(f"Received `{msg.payload.decode()}` from `{msg.topic}` topic")  
 donne = msg.payload.decode()  
  
 data = donne.split(',')  
 print(data)  
 mac = data[0][3:len(data[0])]  
 piece = data[1][6:len(data[1])]  
 date = data[2][5:len(data[2])]  
 time = data[3][5:len(data[3])]  
 temp = data[4][5:len(data[4])]  
  
 datafinal = []  
 datafinal.append(mac)  
 datafinal.append(piece)  
 datafinal.append(date)  
 datafinal.append(time)  
 datafinal.append(temp)  
  
 with open('read.csv','a+', newline='') as f:  
 write = csv.writer(f, delimiter=';', quotechar='"', quoting=csv.QUOTE\_MINIMAL)  
 write.writerow(datafinal)  
  
 client.subscribe(topic)  
 client.on\_message = on\_message  
  
  
def run():  
 client = connect\_mqtt()  
 subscribe(client)  
 client.loop\_forever()  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 run()

## Csv

On voit bien que le script python créer le fichier csv avec l’affichage des données demander.

Du type : **Id=12A6B8AF6CD3, piece=sejour,date=15/06/2022,heure=12:13:14,temp=26,35**

