

IUT DE COLMAR

SAE 24

PROJET INTÉGRATIF

Partie Collecte

RT11

Martin BAUMGAERTNER

RT12

Mehdi REHM

RT11

Sâji DOGHMANE

24 juin 2022

Table des matières

1	Introduction	2
2	Récupération de données	2
2.1	Configuration du script	2
2.2	Exécution du script	3
3	Sauvegarde des valeurs dans la base de données	3
3.1	Explications	3
3.2	Configuration du script	3
4	Vérification de la récupération des données	5

Table des codes

1	Configuration des IDs de connexion	2
2	Installation des paquets nécessaire au script MQTT	2
3	Connexion à la base de données	3
4	Création des tables	4

Table des figures

1	Récupération des données	3
2	Adresse IP de ma machine windows	4
3	Récupération des tables sur DataGrip	5
4	Les capteurs sur DataGrip	5
5	Les capteurs sur MySQL	6
6	Les valeurs sur DataGrip	6
7	Les valeurs sur MySQL	7

1 Introduction

Au courant de l'année nous avons pu voir différents mode de collecte de données, notamment la récupération via MQTT. D'abord, qu'est-ce que MQTT ? MQTT, pour "Message Queuing Telemetry Transport", est un protocole open source de messagerie qui assure des communications non permanentes entre des appareils par le transport de leurs messages.

Le but de cette partie étant de récupérer des données. Nous devons réceptionner des valeurs de température sur une pièce. Puis être capable de les afficher selon les critères définis, et les intégrer dans une base de données qui nous servira plus tard pour la partie Web.

2 Récupération de données

2.1 Configuration du script

Pour pouvoir récupérer les données depuis le MQTT, j'ai donc dû adapter le script python qui nous a été donné dans le diaporama et j'ai dû l'adapter pour qu'il récupère les bonnes données.

J'ai modifié les lignes suivantes, en y ajoutant les bonnes valeurs de connexion :

```
broker = 'test.mosquitto.org'
topic = "IUT/Colmar/SAE24/Maison1"
```

Code 1 – Configuration des IDs de connexion

Par la suite j'ai dû installer un paquet qui était prérequis pour que le script puisse s'exécuter correctement à savoir :

```
pip3 install paho-mqtt python-etcd
```

Code 2 – Installation des paquets nécessaire au script MQTT

2.2 Exécution du script

Au moment de l'exécution du programme, j'obtiens bien les valeurs que nous voulions recevoir comme nous pouvons le constater ci dessous :

```
martin@SRV-UBUNTU:~/SAE-24/SAE-24/Collecte$ python3 mqtt.py
Connected to MQTT Broker!
Received 'Id=B8A5F3569EFF,piece=sejour,date=21/06/2022,time=13:45:41,temp=23.46' from 'IUT/Colmar/SAE24/Maison1' topic
Received 'Id=B8A5F3569EFF,piece=sejour,date=21/06/2022,time=13:45:46,temp=10.92' from 'IUT/Colmar/SAE24/Maison1' topic
Received 'Id=B8A5F3569EFF,piece=sejour,date=21/06/2022,time=13:45:51,temp=16.41' from 'IUT/Colmar/SAE24/Maison1' topic
Received 'Id=B8A5F3569EFF,piece=sejour,date=21/06/2022,time=13:45:56,temp=5.22' from 'IUT/Colmar/SAE24/Maison1' topic
Received 'Id=B8A5F3569EFF,piece=sejour,date=21/06/2022,time=13:46:01,temp=28.23' from 'IUT/Colmar/SAE24/Maison1' topic
Received 'Id=B8A5F3569EFF,piece=sejour,date=21/06/2022,time=13:46:06,temp=17.01' from 'IUT/Colmar/SAE24/Maison1' topic
Received 'Id=B8A5F3569EFF,piece=sejour,date=21/06/2022,time=13:46:11,temp=17.59' from 'IUT/Colmar/SAE24/Maison1' topic
Received 'Id=B8A5F3569EFF,piece=sejour,date=21/06/2022,time=13:46:16,temp=10.97' from 'IUT/Colmar/SAE24/Maison1' topic
Received 'Id=B8A5F3569EFF,piece=sejour,date=21/06/2022,time=13:46:21,temp=22.23' from 'IUT/Colmar/SAE24/Maison1' topic
Received 'Id=B8A5F3569EFF,piece=sejour,date=21/06/2022,time=13:46:26,temp=26.55' from 'IUT/Colmar/SAE24/Maison1' topic
Received 'Id=B8A5F3569EFF,piece=sejour,date=21/06/2022,time=13:46:31,temp=20.13' from 'IUT/Colmar/SAE24/Maison1' topic
Received 'Id=B8A5F3569EFF,piece=sejour,date=21/06/2022,time=13:46:36,temp=8.38' from 'IUT/Colmar/SAE24/Maison1' topic
Received 'Id=B8A5F3569EFF,piece=sejour,date=21/06/2022,time=13:46:41,temp=13.44' from 'IUT/Colmar/SAE24/Maison1' topic
```

FIGURE 1 – Récupération des données

3 Sauvegarde des valeurs dans la base de données

3.1 Explications

L'objectif finale de cet exercice est d'ensuite pouvoir sauvegarder les données dans une base de données pour ensuite permettre l'affichage de ces données dans une page web. Pour ce faire, j'ai donc du écrire un script qui permet premièrement la réception des données MQTT, puis qui les stock dans une base de données.

3.2 Configuration du script

Le script est composé de plusieurs parties. Il commence par se connecter à la base de données :

```
try:
    db=_mysql.connect("10.37.129.3","martin","martin", "temp")
except OperationalError:
    db=_mysql.connect("10.37.129.3","martin","martin")
    db.query("CREATE DATABASE martin")
    db.query("USE martin")
```

Code 3 – Connexion à la base de données

L'adresse IP renseigné correspond à celle de ma machine Windows où est installé ma base de données :

```
C:\Users\martinbaumgaertner>ipconfig

Configuration IP de Windows

Carte Ethernet Ethernet :

    Suffixe DNS propre à la connexion. . . : localdomain
    Adresse IPv6. . . . . : fdb2:2c26:f4e4:1:9840:9be8:e5f:e37f
    Adresse IPv6 temporaire. . . . . : fdb2:2c26:f4e4:1:5968:247d:f271:8282
    Adresse IPv6 de liaison locale. . . . : fe80::9840:9be8:e5f:e37f%4
    Adresse IPv4. . . . . : 10.37.129.3
    Masque de sous-réseau. . . . . : 255.255.255.0
    Passerelle par défaut. . . . . :
```

FIGURE 2 – Adresse IP de ma machine windows

Puis, on défini les tables que l'on créer et les valeurs que l'on ajoute etc. :

```
db.ping(True)
db.query("""
CREATE TABLE IF NOT EXISTS martin.sensors (
    id INT NOT NULL AUTO_INCREMENT,
    macaddr VARCHAR(12) NOT NULL,
    piece VARCHAR(50) NOT NULL,
    emplacement VARCHAR(50),
    nom VARCHAR(50),
    UNIQUE (macaddr),
    PRIMARY KEY (id))
""")

db.query("""
CREATE TABLE IF NOT EXISTS martin.sensors_data (
    id INT NOT NULL AUTO_INCREMENT,
    sensor_id INT NOT NULL,
    CONSTRAINT sensorFK
        FOREIGN KEY (sensor_id)
        REFERENCES martin.sensors(id),
    datetime DATETIME NOT NULL,
    temp FLOAT NOT NULL,
    PRIMARY KEY (id))
""")
```

Code 4 – Création des tables

4 Vérification de la récupération des données

Pour finir, on définit donc de récupérer les valeurs avec une boucle infini. Nous pouvons donc vérifier que les données sont bien reçues. Et en effet, toutes les données récupérées depuis le broker sont bien enregistrées dans la base de données SQL. Nous pouvons vérifier cela avec le logiciel DATAGRIP, en nous connectant à la base de données :

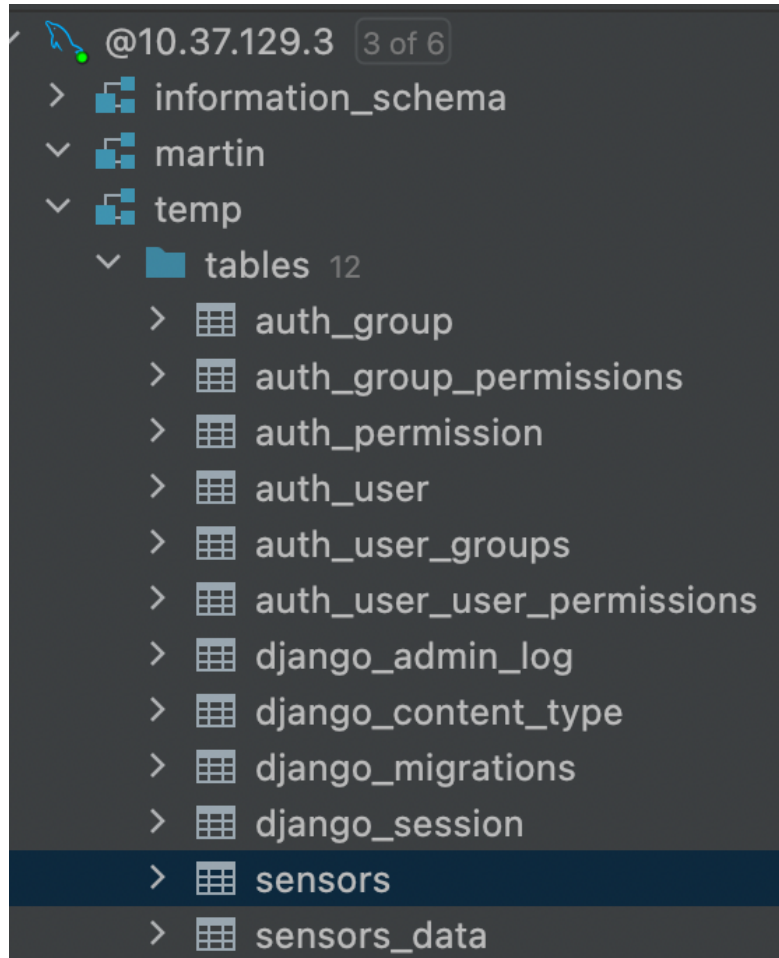


FIGURE 3 – Récupération des tables sur DataGrip

	id	macaddr	piece	emplacement	nom
1	1	B8A5F3569EFF	Cuisine	Chez moi	Capteur 1
2	4	A72E3F6B79BB	Chambre	Chez toi	Capteur 2

FIGURE 4 – Les capteurs sur DataGrip

id	macaddr	piece	emplacement	nom
1	B8A5F3569EFF	Cuisine	Chez moi	Capteur 1
4	A72E3F6B79BB	Chambre	Chez toi	Capteur 2

FIGURE 5 – Les capteurs sur MySQL

	id	sensor_id	datetime	temp
1	2	1	2022-06-22 20:34:44	2.32
2	3	1	2022-06-22 20:37:29	5.87
3	20	1	2022-06-23 09:40:59	6.36
4	85	1	2022-06-23 15:06:46	18.12
5	86	4	2022-06-23 15:06:46	21.19
6	87	1	2022-06-23 15:06:51	28.79
7	88	4	2022-06-23 15:06:51	4.77
8	89	1	2022-06-23 15:07:31	25.23
9	90	4	2022-06-23 15:07:31	11.85
10	91	1	2022-06-23 15:07:36	21.2
11	92	4	2022-06-23 15:07:36	8.99
12	93	1	2022-06-23 15:07:41	5.1
13	94	4	2022-06-23 15:07:41	12.03
14	95	1	2022-06-23 15:07:46	4.73
15	96	4	2022-06-23 15:07:46	0.24
16	97	1	2022-06-23 15:07:51	16.14
17	98	4	2022-06-23 15:07:51	27.12
18	99	1	2022-06-23 15:07:56	10.39
19	100	4	2022-06-23 15:07:56	27.64
20	101	1	2022-06-23 15:08:01	16.15
21	102	4	2022-06-23 15:08:01	22.98
22	103	1	2022-06-23 15:33:33	25.08
23	104	4	2022-06-23 15:33:33	12.42
24	105	1	2022-06-23 15:33:48	2.64
25	106	4	2022-06-23 15:33:48	1.12
26	107	1	2022-06-23 15:35:43	10.7
27	108	4	2022-06-23 15:35:43	29.06

FIGURE 6 – Les valeurs sur DataGrip

id	sensor_id	datetime	temp
2	1	2022-06-22 20:34:44	2.32
3	1	2022-06-22 20:37:29	5.87
20	1	2022-06-23 09:40:59	6.36
85	1	2022-06-23 15:06:46	18.12
86	4	2022-06-23 15:06:46	21.19
87	1	2022-06-23 15:06:51	28.79
88	4	2022-06-23 15:06:51	4.77
89	1	2022-06-23 15:07:31	25.23
90	4	2022-06-23 15:07:31	11.85
91	1	2022-06-23 15:07:36	21.2
92	4	2022-06-23 15:07:36	8.99
93	1	2022-06-23 15:07:41	5.1
94	4	2022-06-23 15:07:41	12.03
95	1	2022-06-23 15:07:46	4.73
96	4	2022-06-23 15:07:46	0.24
97	1	2022-06-23 15:07:51	16.14
98	4	2022-06-23 15:07:51	27.12
99	1	2022-06-23 15:07:56	10.39
100	4	2022-06-23 15:07:56	27.64
101	1	2022-06-23 15:08:01	16.15
102	4	2022-06-23 15:08:01	22.98
103	1	2022-06-23 15:33:33	25.08
104	4	2022-06-23 15:33:33	12.42
105	1	2022-06-23 15:33:48	2.64
106	4	2022-06-23 15:33:48	1.12
107	1	2022-06-23 15:35:43	10.7
108	4	2022-06-23 15:35:43	29.06

27 rows in set (0.00 sec)

FIGURE 7 – Les valeurs sur MySQL

Nous récupérons donc comme voulu toutes les valeurs que nous souhaitons récupérer. Toutes les valeurs sont stockées dans la base de données MySQL et permettront dans la dernière partie de pouvoir afficher sur une page Web développée en DJANGO les valeurs de températures.