

IUT DE COLMAR

SAE 24

PROJET INTÉGRATIF

Partie Web - Base de données

RT11

Martin BAUMGAERTNER

RT12

Mehdi REHM

RT11

Sâji DOGHMANE

24 juin 2022

Table des matières

| | | |
|----------|---|----------|
| 1 | Introduction | 2 |
| 2 | Mise en place du projet Django | 2 |
| 2.1 | Connexion du projet Django à la base de données | 2 |
| 2.2 | Récupération des tables | 3 |
| 3 | Développement | 4 |
| 3.1 | Création du fichier <code>models.py</code> à partir de la base de données | 4 |
| 3.2 | Affichage des 2 capteurs sur une page d'accueil | 4 |
| 3.3 | Affichage des données en fonction d'un capteur | 5 |

Table des codes

| | | |
|---|--|---|
| 1 | Configuration de <code>settings.py</code> | 2 |
| 2 | Création de l'utilisateur | 3 |
| 3 | Création du fichier <code>models.py</code> | 4 |
| 4 | Création de la vue <code>all</code> | 4 |
| 5 | Création de la vue <code>sensor</code> | 5 |

Table des figures

| | | |
|---|--|---|
| 1 | Les tables créées par Django | 3 |
| 2 | Page d'accueil | 4 |

1 Introduction

Le but de cette dernière partie est de pouvoir afficher sur une page web que nous développons en Django les valeurs de températures avec plusieurs types de complications. Par exemple, nous devons être capable de lister les données de températures en fonction d'un capteur, ou de proposer des modifications de champs.

2 Mise en place du projet Django

2.1 Connexion du projet Django à la base de données

Pour pouvoir connecter notre projet à la base de données que nous avons remplie dans la partie précédente avec le script de récupération des données, nous devons modifier le fichier `settings.py` de notre projet Django.

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'temp',
        'USER': 'martin',
        'HOST': '10.37.129.3',
        'PORT': '3306',
        'PASSWORD': 'martin',
    }
}
```

Code 1 – Configuration de `settings.py`

Nous mettons l'adresse `10.37.129.3`, qui correspond à l'adresse de ma machine windows où est hébergée notre base de données. Puis, le nom de notre base de données est `temp`. Enfin, `martin` correspond au nom utilisateur. avec le mot de passe `martin`. J'ai au préalable créé cet utilisateur et autorisé la connexion depuis la machine où est mon projet Django.

L'adresse IP 10.37.129.4 correspond à l'adresse de ma machine où je fais mon projet Django. Pour créer cet utilisateur, lui donner tous les droits et l'autoriser à se connecter il faut utiliser les commandes suivantes :

```
CREATE USER 'martin'@'10.37.129.4' IDENTIFIED BY 'martin';
GRANT ALL PRIVILEGES ON *.* TO 'martin'@'10.37.129.4';
FLUSH PRIVILEGES;
```

Code 2 – Création de l'utilisateur

2.2 Récupération des tables

Nous pouvons donc désormais nous connecter à la base de données depuis le projet Django, et faire un `python3 manage.py makemigrations` et `python3 manage.py migrate` pour que les données soient migrées. Toutes les tables sont sauvegardées et nous observons que les tables que Django crée lors de la migrations du projet sont bien présentes :

```
mysql> show tables;
+-----+
| Tables_in_temp |
+-----+
| auth_group      |
| auth_group_permissions |
| auth_permission |
| auth_user       |
| auth_user_groups |
| auth_user_user_permissions |
| django_admin_log |
| django_content_type |
| django_migrations |
| django_session  |
| sensors         |
| sensors_data    |
+-----+
12 rows in set (0.05 sec)
```

FIGURE 1 – Les tables créées par Django

3 Développement

3.1 Création du fichier `models.py` à partir de la base de données

Ensuite, nous allons créer le fichier `models.py` qui va permettre de créer les modèles nécessaires pour la suite du projet. Pour ce faire, nous allons utiliser la commande suivante :

```
python3 manage.py inspectdb > models.py
```

Code 3 – Création du fichier `models.py`

Suite à cela, notre fichier est bien créé et nous pouvons commencer le développement.

3.2 Affichage des 2 capteurs sur une page d'accueil

Nous nous retrouvons donc avec les 2 capteurs, je veux les afficher sur une page d'accueil. Pour ensuite permettre à l'utilisateur de naviguer à travers les données des capteurs en fonction du capteur qu'il sélectionne. Premièrement je fais une vue `all` pour récupérer les informations des deux capteurs :

```
def index(request):  
    sensors = Sensors.objects.all()  
    sensorsdata = SensorsData.objects.all()  
    return render(request, 'index.html', {'sensors': sensors})
```

Code 4 – Création de la vue `all`

Avec un peu de développement HTML et CSS nous obtenons ce style de page, qui nous permet d'avoir les deux capteurs sur une seule page :



| Nom du capteur : | Lieu : | Pièce : | Adresse MAC : |
|------------------|----------|---------|---------------|
| Capteur 1 | Chez moi | Cuisine | B8A5F3569EFF |
| Capteur 2 | Chez toi | Chambre | A72E3F6B796B |

FIGURE 2 – Page d'accueil

3.3 Affichage des données en fonction d'un capteur

Pour pouvoir afficher les valeurs d'un capteur en fonction de celui que l'on choisit nous mettre en place un filtre qui sélectionne les données du capteur que l'on veut afficher. Pour ce faire, je vais créer cette vue :

```
def temp(request, id):  
    temp = SensorsData.objects.filter(sensor__id=id)  
    return render(request, 'data.html', {'temp': temp})
```

Code 5 – Création de la vue `sensor`