RAPPORT SAE 23 – Partie Web

WAGNER Nicolas, LOSSER Julien et DESCAMPS Nathan

# Table des Matières

[Table des Matières 1](#_Toc106977715)

[1) Introduction 1](#_Toc106977716)

[2) Création Django et CRUDs 2](#_Toc106977717)

[a) Form.py 2](#_Toc106977718)

[b) Urls.py 3](#_Toc106977719)

[c) Views.py 3](#_Toc106977720)

[3) Mise en place databases mysql 4](#_Toc106977721)

[a) Installation des paquets 4](#_Toc106977722)

[b) Mise en place de la database 4](#_Toc106977723)

[c) Exportation de la database vers django 5](#_Toc106977724)

[d) Résultats dans le prompt MySQL 5](#_Toc106977725)

[4) Aspect graphique 6](#_Toc106977726)

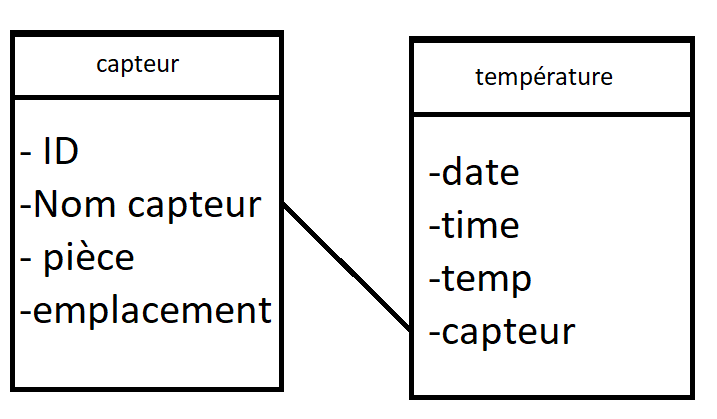
[a) HTML 6](#_Toc106977727)

[b) CSS 7](#_Toc106977728)

# Introduction

Dans ce projet, nous devons mettre en place une base de données avec MySQL sous Windows possédant 2 tables (ID, nom du capteur, pièce et emplacement du capteur). Nous devons ensuite migrer notre projet sous Ubuntu. Nous finirons par afficher les données avec un filtre.

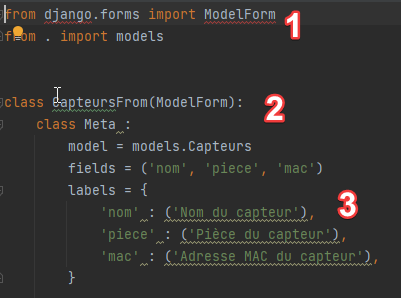
Tel que :



# Création Django et CRUDs

Nous avons créé à l’aide de pycharm des Cruds avec le paquet Django, nous avons suivi le cours vu dans les ressources web dynamique précédentes notamment la ressource R209.

## Form.py

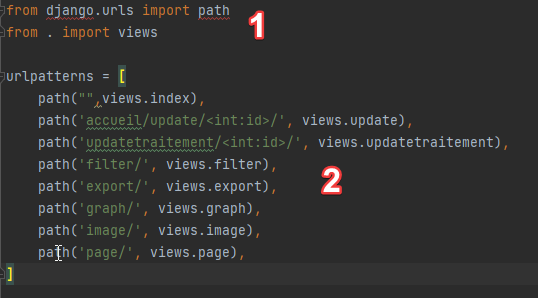


1 : On importe de ModelForm la django.forms ce qui nous permet d’appliquer une première structure a notre form.

2 : On crée la classe CapteursFrom dans laquelle on va définir les différents champs et l’affichage que l’on va leurs attribué. Les champs et les labels.

3 : On crée 3 labels nom, pièce et mac qui renverront le nom du capteur, la pièce où il se trouve et l’adresse mac de celui-ci.

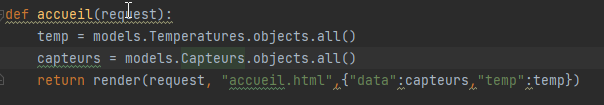
## Urls.py



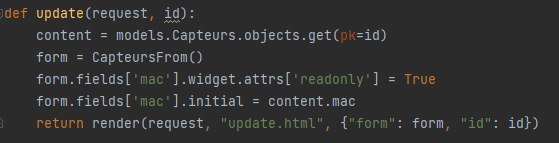
1 : Dans la première partie on peux importer path de django.urls pour crée les différents chemins, on y importe également notre views.

2 : On crée nos chemins courants à partir de nos fonctions dans le views, soit l’update, l’index, le traitement, etc… On notera un système d’ID qui permet de repérer les valeurs dans les pages correspondantes.

## Views.py



Ici on a la fonction accueil qui retourne l’accueil, avec les valeurs des températures et des capteurs qui sont redirigé vers accueil.html.



Ici on a la fonction d’update avec la form qui es définis sur la form du capteur et donc qui renvoie l’id et la form définis plus tôt dans le rapport sur « CapteursFrom ».

# Mise en place databases mysql

## Installation des paquets

Sur le terminal pycharm on installe les paquet MySQL client :

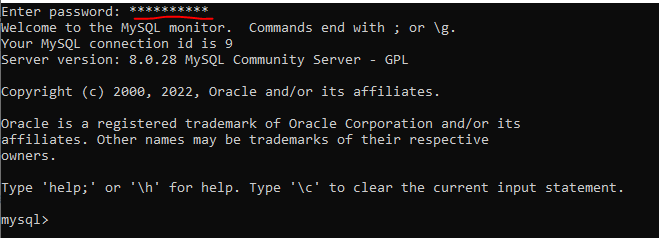


Cela nous permet de lié la base de données réalisé avec MySQL command prompt sur windows, là où l’on crée notre databases Mysql.

Au préalable pour récupéré l’application MySQL command prompt on doit allé sur internet et installé Mysql

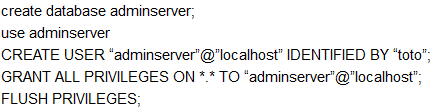


Pour cela on suis l’installation et on choisi l’option « full installation », une fois cela fait un a accès a notre prompt Mysql :



On rentre le mot de passe initialisé plus tôt sous le nom de « frangipane »

## Mise en place de la database



Ici sur MySQL prompt  On crée la base de données avec les commandes précédentes et on se donne les privilèges sur un utilisateur crée précédemment.

## Exportation de la database vers django

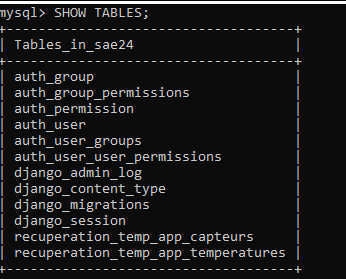


Ici on a lié la base de données databases sae24 crée précédemment sur MySQL avec le nom d’user ‘root’, le mot de passe initialisé sur ‘frangipane’ et le localhost sur le localhost crée précédemment dans MySQL :

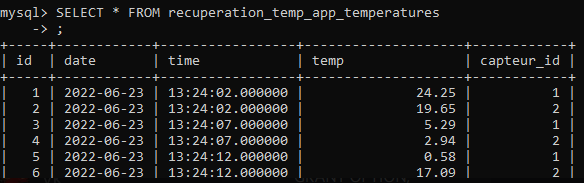


Et on y entre le port 3306.

## Résultats dans le prompt MySQL

Dans le prompt MySQL on voit qu’après avoir fait un show TABLES on a la liste des tables qui apparait : 

Et avec la commande SELECT \* FROM recuperation\_temp\_app\_temperatures on récupère les données listées dans un tableau en ASCII des différentes données récupéré par le broker et Mqtt.

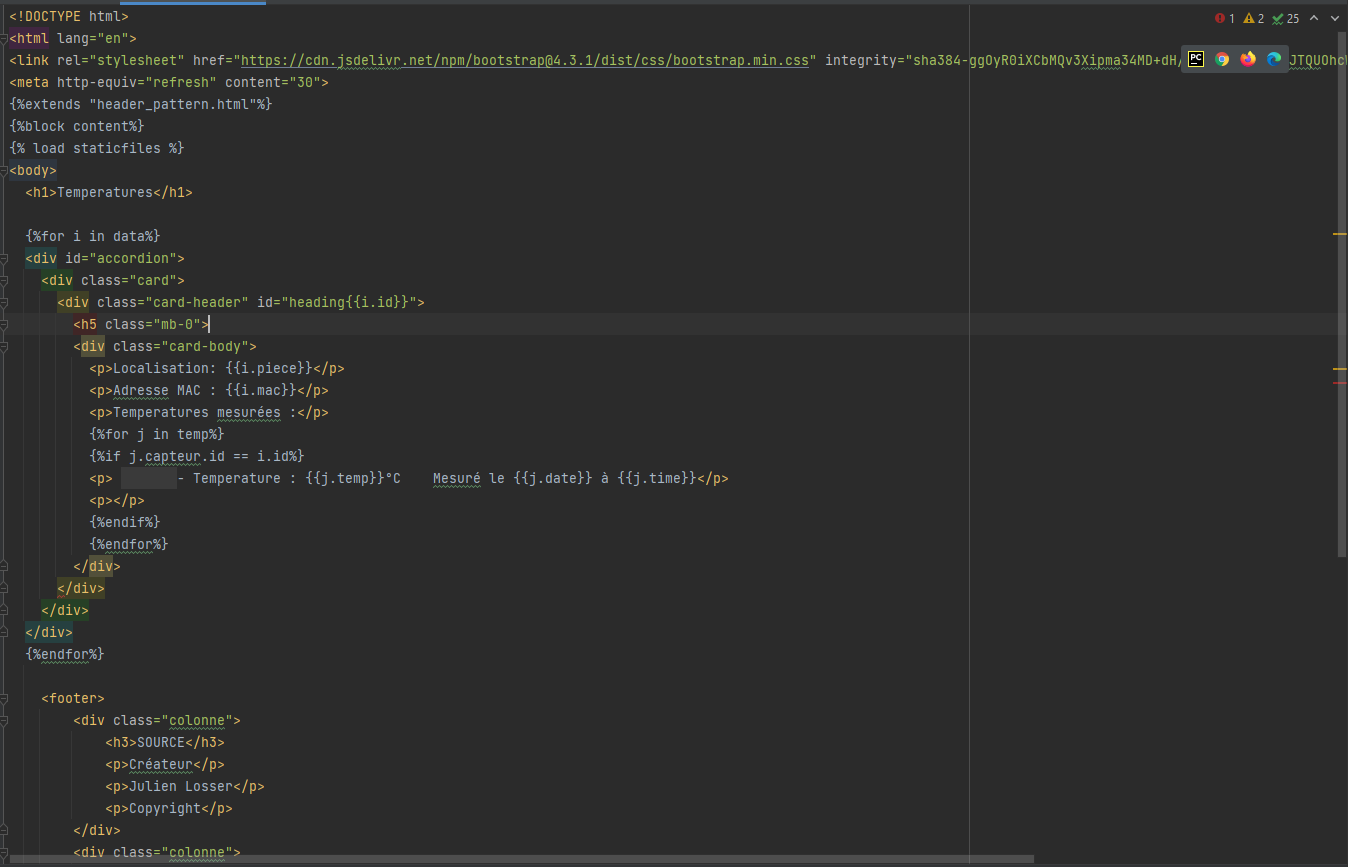


# Aspect graphique

Lors de la partie graphique nous avons utilisé pycharm pour la création d’un site qui permet de visualisé les valeurs et de voir les différentes températures sur les deux capteurs.

## HTML

Ici un exemple de notre fichier accueil.html qui es la page principale de notre projet où sont affiché les valeurs remarquables :



## CSS

Ici on a créé un CSS qui nous permet de naviguer dans les tables température et capteurs.

1. body {  
    background: #222;  
    font-family: Raleway, Helvetica, sans-serif;  
    position: relative;  
   }  
     
   /\*Navigation \*/  
     
   /\* Position \*/  
   .side-nav {  
    position: fixed;  
    height: 100vh;  
    left: 0;  
    top: 0;  
   }  
     
   /\* Pour masquer la sous-navigation \*/  
   .wrapper {  
    background: #333;  
    height: 100vh;  
    width: 75px;  
   }  
     
   .nav-bloc {  
    padding: 20px 0;  
    display: flex;  
    justify-content: center;  
    cursor: pointer;  
    border-bottom: 1px solid #f2f2f21e;  
   }  
   .nav-bloc:hover {  
    background: rgb(106, 146, 196);  
   }  
   .nav-bloc:hover .sub-nav {  
    transform: translateX(75px);  
   }  
   .nav-bloc img {  
    width: 45px;  
    transition: 0.2s ease-in-out ;  
   }  
     
   .sub-nav {  
    padding: 0px;  
    width: 200px;  
    height: 100vh;  
    position: absolute;  
    top: 0;  
    left: 0;  
    background: #333;  
    z-index: -1;  
    color: white;