



30/12/2024

: LOGEMENT-ECO-RESPONSABLE
IOT 2024/2025



Amine Oumert
SORBONNE UNIVERSITE

Rapport : LOGEMENT-ECO-RESPONSABLE

1. Schéma de la base de données

Le schéma de la base de données a été validé par le prof.

2. Code SQL pour la base de données

Questions 2 et 3 :

Le code est disponible dans le script SQL : logementEcoResponsable.sql.

Ce script crée une base de données structurée pour gérer les éléments suivants :

- Logements
- Pièces
- Capteurs/Actionneurs
- Mesures

Question 4 :

Lignes 88 à 98 : Ajout de tous les ordres SQL permettant de créer un logement avec 4 pièces.

Question 5 :

Lignes 100 à 107 : Ajout de tous les ordres SQL permettant de créer au moins 4 types de capteurs/actionneurs.

Question 6 :

Lignes 110 à 114 : Ajout de tous les ordres SQL permettant de créer au moins 2 capteurs/actionneurs.

Question 7 :

Lignes 117 à 124 : Ajout de tous les ordres SQL permettant de créer au moins 2 mesures par capteur/actionneur.

Question 8 :

Lignes 126 à 132 : Ajout de tous les ordres SQL permettant de créer au moins 2 mesures par capteur/actionneur.

1.2 Remplissage de la base de données

Le script Remplissage.py connecte une base SQLite pour insérer des données fictives, notamment :

- Logements
- Pièces
- Capteurs
- Mesures
- Factures

Le script utilise des requêtes SQL dynamiques, ainsi que des structures Python comme des dictionnaires et des boucles. Une gestion d'erreurs est incluse pour assurer la stabilité des interactions avec la base de données.

2. Serveur RESTful

2.1 Exercice 1 : Remplissage de la base de données

Le code est disponible dans le script : serveur.py.

Source : <https://fastapi.tiangolo.com/tutorial/sql-databases/>

Requêtes GET :

1. Consulter les logements

- **Route** : /logement
- **Ligne** : 239
- **Description** : Récupère tous les logements dans la base de données.

2. Consulter les pièces

- **Route** : /piece
- **Ligne** : 245
- **Description** : Récupère toutes les pièces.

3. Consulter les types de capteurs/actionneurs

- **Route** : /type-capteur-actionneur
- **Ligne** : 268
- **Description** : Récupère tous les types de capteurs/actionneurs.

4. Consulter les capteurs/actionneurs

- **Route** : /capteur-actionneur
- **Ligne** : 286
- **Description** : Récupère tous les capteurs/actionneurs.

5. Consulter les mesures

- **Route** : /mesure
- **Ligne** : 312
- **Description** : Récupère toutes les mesures.

6. Consulter les factures

- **Route** : /factures
- **Ligne** : 325
- **Description** : Récupère toutes les factures.

Requêtes POST :

1. Créer un logement

- **Route** : /logement
- **Ligne** : 230
- **Description** : Ajoute un logement dans la base de données.

2. Créer une pièce

- **Route** : /piece
- **Ligne** : 245
- **Description** : Ajoute une pièce pour un logement spécifique.

3. Créer un type de capteur/actionneur

- **Route** : /type-capteur-actionneur
- **Ligne** : 261
- **Description** : Ajoute un type de capteur/actionneur dans la base de données.

4. Créer un capteur/actionneur

- **Route** : /capteur-actionneur
- **Ligne** : 274
- **Description** : Ajoute un capteur/actionneur dans une pièce donnée.

5. Créer une mesure

- **Route** : /mesure
- **Ligne** : 291
- **Description** : Ajoute une mesure pour un capteur/actionneur spécifique.

6. Créer une facture

- **Route** : /factures
- **Ligne** : 318
- **Description** : Ajoute une facture pour un logement spécifique.

Exercice 3 : Météo

Certains problèmes ont été corrigés avec ChatGPT.

1. Consulter la météo

- **Route** : /meteo-5jours
- **Ligne** : 49

- **Description** : Récupère toutes les données météorologiques depuis OpenWeatherMap.

3. Intégration

Exercice 4 : Intégration

Le code est disponible dans le script : capteur.py.

Ce script génère des valeurs aléatoires proches des données météorologiques obtenues. Cette approche a été choisie en raison d'absence de capteur de température et d'ESP.

4. Interface Web

Technologies : HTML/CSS/JavaScript

Toute l'interface web a été générée par ChatGPT. De nombreux problèmes ont été corrigés avec son aide.

Répertoire :

- Templates
- static