Pense-bête : Utiliser SQLite et Requests en Python

1. Importer les bibliothèques nécessaires :

import sqlite3 import requests ■■ Pas besoin d'importer obligatoirement sqlite3, tout dépend de ce que tu veux utiliser (par exemple mysql.connector pour MySQL).

2. Se connecter à la base de données :

connection = sqlite3.connect("chemin_vers_ton_fichier.db") cursor = connection.cursor()

3. Créer une table si elle n'existe pas déjà :

cursor.execute(""" CREATE TABLE IF NOT EXISTS measure (id INTEGER PRIMARY KEY AUTOINCREMENT, date TIMESTAMP DEFAULT CURRENT_TIMESTAMP, description VARCHAR(255), value FLOAT) """)

4. Lire une liste d'URL depuis un fichier :

with open("list_url.txt", "r") as file: urls = file.readlines() for url in urls: url = url.strip() # Supprime les espaces et retours à la ligne with open(...) permet d'ouvrir un fichier. - "r" → lecture (read) - "w" → écriture (write) - "a" → ajout (append) - "r+" → lecture/écriture

5. Effectuer une requête HTTP pour chaque URL :

if url: # Vérifie que l'URL n'est pas vide response = requests.get(url, timeout=5) status_code = response.status_code print(f"{url} returned status code: {status_code}") Le timeout=5 empêche le programme de rester bloqué si le serveur ne répond pas.

6. Préparer les données à insérer :

description = f"Status for {url}" value = float(status_code)

7. Insérer les données dans la base :

cursor.execute("INSERT INTO measure (description, value) VALUES (?, ?)", (description, value))

8. Sauvegarder et fermer la base de données :

connection.commit() # Enregistre définitivement les modifications connection.close() # Libère les ressources

■ En résumé :

- sqlite3.connect() → ouvre ou crée une base de données. - cursor.execute() → exécute une requête SQL. - requests.get() → envoie une requête HTTP. - connection.commit() → sauvegarde

les changements. - connection.close() → ferme la connexion.