

Rapport de Projet : Scraping et Visualisation des Donnees

1. Introduction

Dans le cadre de ce projet, j'ai developpe une solution complete permettant de scraper des donnees d'un site web, de les stocker dans une base de donnees et de les afficher via une interface web. Ce projet m'a permis de consolider mes competences en web scraping, bases de donnees, developpement d'API, et deploiement avec Docker.

2. Presentation du Projet

L'objectif etait de creer une application capable de :

- Scraper des donnees depuis un site web specifique.

- Stocker ces donnees dans une base PostgreSQL.

- Creer une API pour acceder aux donnees.

- Construire une interface web pour afficher les resultats.

- Tout conteneuriser avec Docker pour faciliter le deploiement.

3. Choix Technologiques

- Scraping : BeautifulSoup et Requests

- Base de donnees : PostgreSQL

- API : FastAPI

- Interface Web : Dash (Plotly)

- Conteneurisation : Docker & Docker-Compose

4. Difficultes Rencontrees et Solutions Apportees

Erreur 1 : ModuleNotFoundError: No module named 'Database'

Probleme : Le module Database n'etait pas trouve dans le container Scraper.

Solution : Modification du Dockerfile pour copier correctement les fichiers.

Erreur 2 : connection to server at 'database' failed: Connection refused

Probleme : L'API essayait de se connecter a PostgreSQL alors qu'il n'etait pas encore pret.

Solution : Ajout d'un delai d'attente avant d'executer l'API.

Erreur 3 : No such file or directory: '/app/Scraper/requirements.txt'

Probleme : Le chemin d'accès aux fichiers dans le container n'était pas correct.

Solution : Verification et modification du contexte de build dans Dockerfile.

Erreur 4 : database 'books_db' already exists

Probleme : Le fichier schema.sql tentait de recréer la base existante.

Solution : Suppression de la ligne CREATE DATABASE books_db; dans schema.sql.

5. Conclusion et Perspectives

Ce projet m'a permis de mettre en oeuvre plusieurs competences techniques et de mieux comprendre l'interconnexion

entre le scraping, les bases de donnees, les API, et le deploiement avec Docker.

Ameliorations futures :

Scraping en temps reel avec Celery.

Ajout d'Elasticsearch pour une recherche plus efficace.

Utilisation de Redis pour optimiser la mise en cache.

En surmontant les difficultes rencontrees, j'ai acquis une meilleure comprehension des defis lies aux architectures

distribuees et a la conteneurisation.