

AXE D'amélioration futur du code

Stockage dans une base de donnée:

L'idée ici est de remplacer le stockage des données dans des fichiers CSV par une base de données. Les avantages incluent une meilleure gestion des données, la possibilité de faire des requêtes plus complexes, et la facilité d'ajouter, mettre à jour ou supprimer des enregistrements.

Comment procéder:

- Choisir une base de données (SQLite, MySQL, PostgreSQL, etc.) en fonction des besoins.
 - Utiliser une bibliothèque Python pour interagir avec la base de données (par exemple, SQLAlchemy pour MySQL ou SQLite).
-
- Modifier le code pour insérer les données extraites dans la base de données plutôt que dans des fichiers CSV.

Test d'intégration:

Ajouter des tests d'intégration pour assurer que les différentes parties du programme fonctionnent correctement ensemble. Cela garantit que les mises à jour futures ne cassent pas les fonctionnalités existantes. On peut retrouver Pytest qui est un framework qui fournit l'ensemble des outils nécessaires pour effectuer les tests. On créera un fichier test pour chaque fichier source (exemple : `test_get_all_categories.py` ce fichier contiendra le code qui correspond à une fonction qui permettra de voir si le résultat attendu correspond au résultat donné,) Pytest signalera automatiquement les échecs d'assertion et fournira des informations détaillées sur les erreurs.

Support multithreading:

Le multithreading est une technique qui permettra au programme d'accomplir plusieurs tâches simultanément. Cela sera utile car les requêtes réseau peuvent prendre du temps, et pendant ce temps, le programme pourrait faire d'autres choses au lieu d'attendre.

Pour l'implémenter, il faut utiliser la bibliothèque `concurrent.futures` de Python. Ainsi après avoir extrait toutes les catégories, à l'aide d'une fonction le reste du code s'exécutera pour chaque catégorie en même temps.