**Documentation du Script d’Agrégation**

#### **Dépendances**

Le script d’agrégation utilise plusieurs bibliothèques *Python (langage)* pour l'extraction, le traitement et le stockage des données :

* requests : Pour effectuer des appels HTTP vers l'API de géocodage d'OpenWeatherMap.
* pandas : Pour la manipulation et le traitement des données.
* sqlite3 : Pour interagir avec la base de données SQLite.
* os : Pour les opérations de gestion des fichiers et des répertoires.
* fastapi : Pour créer l'API REST.
* uvicorn : Pour exécuter le serveur FastAPI.

#### Commandes

Voici les commandes nécessaires pour exécuter le script d’agrégation et l’api :

Installation des dépendances :

* Utilisez pip pour installer les bibliothèques nécessaires : *pip install requests pandas fastapi uvicorn*

Exécution du script :

* Exécutez le script principal pour extraire, traiter et stocker les données :

*python Request\_data.py*

Démarrage du serveur FastAPI :

* Démarrez le serveur FastAPI pour accéder à l’API : *uvicorn Api\_data:app --reload*

#### Enchaînements Logiques de l’Algorithme

Le script suit une série d’étapes pour s'assurer que les données sont correctement extraites, enrichies et stockées :

Définition des informations de base :

* Initialisation des constantes telles que l'API\_KEY, le BASE\_URL pour l'API de géocodage, et le dictionnaire state\_info contenant les informations sur les états.

Fonction pour obtenir les coordonnées géographiques :

* La fonction get\_lat\_lon utilise l’API de géocodage pour obtenir les coordonnées des villes principales de chaque état.

Lecture du fichier CSV :

* Utilisation de pandas pour lire le fichier CSV contenant les données d’émissions.

Extraction des coordonnées géographiques :

* Boucle sur chaque état pour obtenir et stocker les coordonnées géographiques dans un dictionnaire.

Ajout des coordonnées au DataFrame :

* Enrichissement du DataFrame avec les coordonnées géographiques obtenues.

Renommage des colonnes du DataFrame :

* Homogénéisation des noms de colonnes pour correspondre aux noms utilisés dans la base de données SQLite.

Sauvegarde du DataFrame enrichi :

* Écriture du DataFrame enrichi dans un nouveau fichier CSV.

Création et mise à jour de la base de données SQLite :

* Création des tables state et emissions dans la base de données SQLite.
* Insertion des données dans les tables.

#### Nettoyage et Homogénéisation des formats de données

#### **Gestion des erreurs de lecture du CSV** :

* + Le script vérifie si le fichier CSV existe, s'il est vide ou s'il contient des erreurs de formatage avant de procéder.

Homogénéisation des noms de colonnes :

* Renommage des colonnes du DataFrame pour correspondre aux noms utilisés dans la base de données.

Vérification et ajout des coordonnées :

* Utilisation d’une fonction map pour ajouter les coordonnées géographiques au DataFrame.

Création des tables et insertion des données :

* Création des tables state et emissions dans SQLite et insertion des données.