Rapport de Développement Back-End : API de Gestion de Pandémies

Contexte du projet

Dans le cadre d'un projet de groupe, nous avons conçu et développé une application permettant de suivre les situations pandémiques par pays et par maladie. Cette application vise à offrir une visualisation claire et à jour de données pandémiques, exploitables par des outils de datavisualisation comme **Power BI**.

Je me suis occupé du **développement back-end** de l'application.

Technologies utilisées

- **Langage** : Python 3.11
- **Framework** : FastAPI
- **ORM** : SQLAlchemy
- **Base de données** : PostgreSQL
- **Outils de documentation** : Swagger (OpenAPI v3.1.0, intégré par FastAPI)
- **Gestion des schémas** : Pydantic
- **Outil de requête et test API** : Swagger UI / Postman
- **Connexion à Power BI** (prévue)

Fonctionnalités principales de l'API

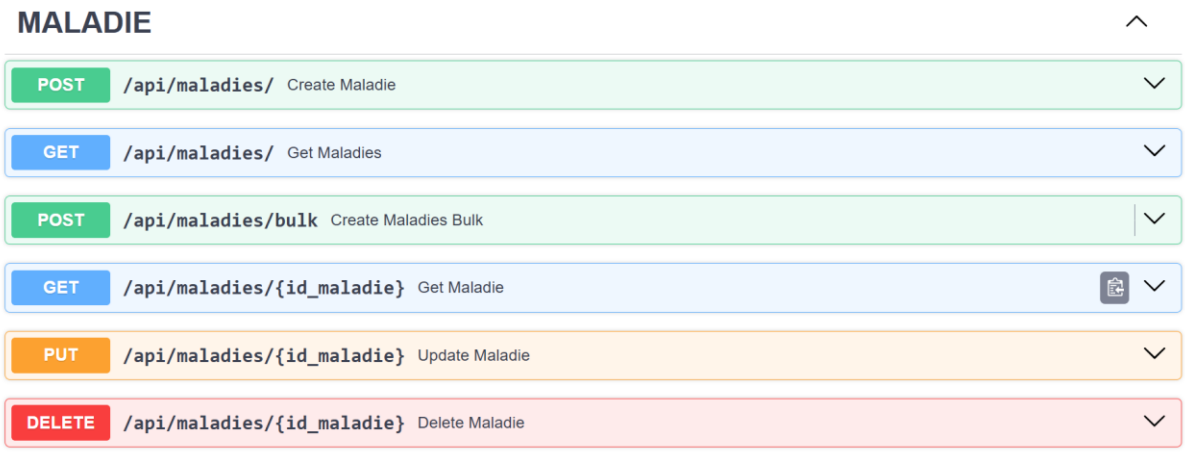
L'API propose des fonctionnalités CRUD (Create, Read, Update, Delete) sur trois entités principales :

1. Pays

- Endpoint : /api/pays/
- Attributs : id_pays, nom_pays, region_oms
- Fonctionnalités : Ajouter, consulter, modifier, supprimer, ou insérer en masse plusieurs pays.

2. Maladie

- Endpoint : /api/maladies/
- Attributs : id_maladie, nom_maladie, type_maladie
- Fonctionnalités : Gestion complète CRUD + insertion groupée.



3. Situation Pandémique

- Endpoint : /api/situations/
- Attributs :
    - id_situation
    - pays (clé étrangère)
    - maladie (clé étrangère)
    - date_observation
    - cas_confirmes
    - deces
    - guerisons
    - cas_actifs
- Fonctionnalités :
    - Suivi des cas par pays et par maladie.
    - Insertion et mise à jour groupée.
    - Filtrage des situations par pays, maladie ou date.

Sécurité & Validation

- Tous les champs sont validés avec **Pydantic**, garantissant la cohérence des données reçues et retournées.
- Les erreurs sont gérées de façon explicite avec des messages clairs (e.g. 404 Not Found, 400 Bad Request).
- L'API peut facilement intégrer un mécanisme d'authentification si nécessaire (via FastAPI Security).

Intégration avec Power BI

L'API est conçue pour s'intégrer avec Power BI via les endpoints GET :

- GET /api/pays/
- GET /api/maladies/
- GET /api/situations/

Ces routes retournent les données sous format JSON structuré, facilement interprétables par les outils de datavisualisation.

Une version optimisée des endpoints est également prévue (type /api/situations/export) pour exporter toutes les données sous un format plat, simplifiant l'import dans Power BI.

Résultats obtenus

- Une API fonctionnelle, testée, et conforme aux normes REST.
- Une base de données relationnelle PostgreSQL bien structurée, avec intégrité référentielle.
- Une documentation automatique générée par FastAPI pour faciliter les tests par l'équipe front-end et les analystes Power BI.
- Une architecture propre, facilement extensible et maintenable.

Conclusion

Ce projet m'a permis de renforcer mes compétences en développement back-end, notamment avec FastAPI, SQLAlchemy, et PostgreSQL. J'ai appris à structurer une API REST robuste, à valider les données efficacement et à collaborer avec des outils de datavisualisation.

Je suis prêt à déployer ou améliorer cette API pour des usages en production ou en lien avec d'autres outils métiers.