

Comment utiliser le projet

Avec docker

La première méthode pour lancer le projet est avec l'utilisation de Docker

1. Installer Docker
2. Utiliser la commande `docker-compose up -d` dans un terminal pointant sur la racine du projet
3. Le frontend du projet sera disponible sur <http://localhost:5173>

Pour arrêter le projet, utilisez la commande : `docker-compose down`

Cela permettra de fermer tous les conteneurs Docker ouverts.

Pour relancer le projet, il suffit de refaire la commande : `docker-compose up -d` à la racine du projet.

Sans Docker

La deuxième méthode pour lancer le projet est de l'installer directement sur l'ordinateur sans passer par Docker.

Installation du Backend

1. Installer Python en allant sur le site officiel : <https://www.python.org/downloads>
2. Ouvrir un terminal de commande dans le dossier **Backend**
3. Créer un environnement virtuel avec la commande : `python -m venv venv`
4. Activer l'environnement virtuel :
 - Sur Windows : `*venv\Scripts\activate*`
 - Sur Linux/Mac : `*source venv/bin/activate*`
5. Installer les dépendances du projet avec la commande : `pip install -r requirements.txt`

Installation du Frontend

1. Installer Node.js (et npm) depuis le site officiel : <https://nodejs.org>
2. Ouvrir un terminal dans le dossier **mspr502-front**.
3. Installer les dépendances du projet avec la commande : `*npm install*`

Lancer le projet

1. Ouvrir un terminal dans le dossier **Backend**
2. Lancer le serveur backend avec la commande : `python app.py`
3. Ouvrir un terminal dans le dossier **mspr502-front**
4. Lancer le projet avec la commande : `npm run dev`
5. Le frontend sera disponible à l'adresse : <http://localhost:5173>

Changer de model IA

Le nouveau modèle doit être placé dans le dossier Backend/models et doit être nommé **model.keras**.

Seuls les modèles au format .keras sont pris en charge pour l'instant.

Les fichiers **scaler_features.pkl** et **scaler_targets.pkl** doivent également se trouver dans le dossier models.

Si vous avez installé le projet avec Docker, vous devrez reconstruire l'image Docker pour prendre en compte le nouveau modèle avec la commande *docker-compose build* et relancer les conteneurs avec *docker-compose up -d*, les deux commande doivent être fait à la racine du projet.

Débug

Si le projet rencontre des problèmes avec le serveur, vous pouvez généralement résoudre le souci de deux manières :

1. Vérifier que le backend est bien lancé.
2. Vérifier que les noms des fichiers dans le dossier models correspondent bien à ceux indiqués précédemment.

Commandes utiles pour le débogage

Si vous utilisez Docker :

- Voir les conteneurs en cours d'exécution : *docker ps*
- Voir tous les conteneurs (y compris arrêtés) : *docker ps -a*
- Voir les logs d'un conteneur (exemple pour le backend) :
docker logs <nom_du_conteneur>
- Arrêter tous les conteneurs : *docker-compose down*
- Relancer les conteneurs après modification : *docker-compose up -d*

Si vous utilisez le projet sans Docker :

- Vérifier que le backend tourne en ouvrant un terminal dans le dossier Backend et en lançant : *python app.py*
- Tester que l'API répond : Ouvrir un navigateur ou utiliser curl : curl
<http://localhost:5000/health>
- Si le backend ne démarre pas, vérifiez les messages d'erreur dans le terminal pour identifier le problème.