

Objectifs :

L'objectif de ce projet est de mettre en place, et peupler une base de données MongoDB à partir d'un jeu de données de l'open data (disponible ici : <https://www.kaggle.com/zynicide/wine-reviews?select=winemag-data-130k-v2.json>), et d'implémenter une API permettant de requêter cette base de données.

- 1- Mettre en place une base de données MongoDB et la peupler.
- 2- Construction d'une API Flask permettant d'interroger.
- 3- Conteneurisation de la base MongoDB, du script de peuplement, et de l'API via Docker.
- 4- Déploiement via un fichier docker-compose.yml permettant de lancer les différents services (bdd, script de peuplement de la bdd, API) en même temps.

Livrables :

- a. Le code source de l'API ainsi que les dépendances au format standard requirements.txt
- b. Un Dockerfile pour l'API conteneurisée,
- c. Les Dockerfiles pour chaque conteneur de test,
- d. Dans un dossier, l'ensemble des fichiers utilisés pour créer les tests,
- e. Un docker-compose.yml pour lancer l'ensemble des tests.
- f. Les fichiers de déploiement Kubernetes.

Requirements

- ⇒ Packages: pandas, Flask, flask_httpauth, Requests, pymongo
- ⇒ Docker
- ⇒ Fichiers: docker-compose.yml et mongo-init
- ⇒ Dossiers ./image_mongo_flaskapi ./image_pymongo permettant de créer les images docker de l'API et du script de peuplement de la base.

1- Création des images docker de l'API et du script de peuplement de la base.

- API

Dans le dossier ./image_mongo_flaskapi , exécuter la commande suivante :

```
docker image build . -t sdereynal/mongo_flaskapi:latest
```

- Script de peuplement :

Dans le dossier ./image_pymongo , exécuter la commande suivante :

```
docker image build . -t sdereynal/image_pymongo:latest
```

2- Déploiement des différents services : bdd, script de peuplement de la bdd, API

Dans le dossier courant (dans lequel se situe les fichiers : docker-compose.yml et mongo-init), exécuter la commande suivante :

```
docker-compose up
```

3- Requêtage de la base de données

L' API permet d'interroger la base de données MongoDB. Voici comment l'utiliser :

Endpoints :

- /mongodb/wine/wine_test/status : retourne 1 si l'API fonctionne.
- Exemple : `curl -i -X GET http://0.0.0.0:5000/mongodb/wine/wine_test/status`
- /mongodb/wine/wine_test/databases : retourne le nom des bases de données présentes dans MongoDB
- /mongodb/wine/wine_test/distinct ?key=<key> : renvoie les valeurs uniques de la clé sélectionnée (par exemple le nom de tous les testeurs de vin ayant rédigé un avis, ou le nom de tous les pays présentant au moins un vin noté dans la base).
- /mongodb/wine/wine_test/filter?crit=<crit> : renvoie les documents satisfaisants la/les conditions du/des critères retenu(s).