

# HapiVueMongo & Microservices

La police de Boston a besoin d'un nouvel outil d'exploration et recensement des crimes. Votre travail est de le créer !

Pour ce faire vous avez à votre disposition un export de la base de données de la police sous format CSV.

Le chef de la police vous demande explicitement de :

- Fournir des outils de visualisation des données (crimes par date, par type, par localisation, ...)
- Forcer l'identification de chaque utilisateur, et leur attribuer des droits différents selon leurs grades. Agent (consultation), Détective (consultation, ajout, et modification), Chef de la police (consultation, ajout, modification et suppression)
- Chaque nouveau compte devra être validé par le Chef de la police.

## Technos

- Un front avec VueJS / Vuex / Nuxt
- Une API en HapiJS
- Une base de données Mongo & MySQL
- Lumen (pour vos microservices)

# Microservices

Vous devez réaliser 3 microservices à l'aide de Lumen.

- L'Enregistrement des utilisateurs sur une base MySQL
- L'export des utilisateurs sous le format CSV
- La recherche

Exemple : <http://127.0.0.1:8080/exportToCsv/> et <http://127.0.0.1:5832/searchResult> ...

L'idée n'est pas de réaliser toute une architecture microservices, mais simplement de développer 3 microservices qui eux seront consommés par votre projet. Vous devez utiliser Docker pour faire tourner vos microservices.

# #Fonctionnalités

- Une page de connexion / inscription
- Une page d'administration pour la validation des nouveaux profils (réservée au Chef de la police, vous devez garantir que chacun a les bons droits), vous pouvez utiliser un template CSS pour l'interface admin si vous le souhaitez, (SB admin) par exemple.
- Une page de visualisation des données :
  - Visualisation des détails
  - Fonctionnalité de filtres (par armes, ...)
  - Barre de recherche (microservice)
- Une page de saisie des crimes et délits
- Les utilisateurs doivent être enregistrés dans une base de données MySQL (microservice)
- Export des utilisateurs sous le format CSV (microservice)
- Pour l'API
- Elle devra être RESTFUL et créée à partir d'Hapi JS.

## Liens utiles

How to use mongodb with your lumen API

GOTO 2014 • Microservices • Martin Fowler

Architecture microservices

Autre

Évitez au maximum les opérations synchrones avec node (idéalement il ne devrait pas y en avoir du tout !).

Vous devez faire attention au temps de chargement (pagination).

Vous ne devez pas altérer les données qui vous sont fournies de base (malus de 10).

Bonus de 2 points si vous réalisez des tests unitaires pour vos microservices.