

Mongodb + node.js + express + Jade/pug

Dans ce TD, vous allez importer une base json dans un nœud MongoDB.

Vous allez utiliser la console **mongo** pour vérifier l'import et faire quelques requêtes d'exploration. Vous allez ensuite réaliser un mini serveur express pour exploiter cette base avec des pages écrites en templates jade/pug.

1. Récupérez la **base.json** sur le site Moodle et créez une instance de **mongod** qui contient cette base (**mongoimport**). Installez une interface http de type REST pour pouvoir consulter des informations sur vos traitements sur la base.
2. Utilisez une console **mongo** pour vérifier la création de la base. Vérifiez également avec l'interface http.
3. En utilisant la console **mongo**, lancez les commandes pour répondre aux questions suivantes :
 - a. Combien de documents dans la collection ?
 - b. Combien de restaurants à Manhattan ?
 - c. Combien de pizzerias à Manhattan ?
 - d. Combien de restaurants Français dans le Bronx ?
 - e. Combien de pizzerias à Manhattan qui ont déjà été évalués en catégorie "A" ?
4. Vous allez utiliser un serveur node.js en utilisant express pour mettre à disposition les résultats des recherches sur la base. Vous vous brancherez à l'aide d'un navigateur sur votre serveur node.js (qui pourra faire des requêtes sur la base MongoDB). Un chemin d'accès spécifique sera utilisé pour tester les fonctionnalités (voir plus loin).

Vérifiez bien que vous avez accès au package mongodb (pas d'utilisation de mongoose). Vous utiliserez jade/pug comme système de templates.

Avant d'installer le serveur, vérifiez que vous arrivez à vous connecter à la base en utilisant un client mongodb (**mongoClient**). Faites une requête simple qui affiche le nom des restaurants français sur la console.

- a. Réalisez un serveur **express** (avec gestion des erreurs – 404 etc).
- b. Réalisez une page **index.html** qui affiche le nombre de restaurants dans la base.
- c. Gérez les erreurs de connexion en affichant une page qui explique le problème si le client n'arrive pas à se connecter à la base. (Remarque : utilisez systématiquement Jade/pug pour les pages web).

5. Vous allez réaliser la page web suivante : l'adresse est /noms/specialité où spécialité est remplacée par la spécialité du restaurant.

Par exemple, **/noms/Pizza** : dans ce cas ; la page affiche le nombre des pizzerias, le nom des pizzerias et leur adresse. Dans le cas d'un nom composé (ou spécialité), le nom apparaît dans le chemin avec un – pour remplacer les espaces. Encore une fois, c'est Jade/Pug qui génère la page (each, for while etc).

7. Même chose avec /noms/quartier/Pizza : par exemple /noms/Bronx/Pizza

6. Trouvez un mécanisme pour trier le résultat par quartiers et par nom de restaurant (par exemple les Domino's Pizza du Bronx sont ensemble). Avez-vous le même nombre de restaurants (triés ou pas triés) pour la même requête ? Remarques ?

- **Broadway Joe'S Pizza : 5985 Broadway 10463 (BRONX)**
- **Bronx Lala Pizza : 87 East 161 Street 10451 (BRONX)**
- **Bronx Pizza : 3414 Jerome Avenue 10467 (BRONX)**
- **Brook Avenue Deli & Pizza : 348 Brook Avenue 10454 (BRONX)**
- **C & M Pizza : 2519 Webster Avenue 10458 (BRONX)**
- **Capri Li Pizza : 149 Dreiser Loop 10475 (BRONX)**
- **Capri Pizza Restaurant : 2401 Westchester Avenue 10461 (BRONX)**
- **Carbo'S Pizzeria : 2456 Eastchester Road 10469 (BRONX)**

7. Réalisez une page web qui permet d'entrer des coordonnées (x,y) et qui affiche l'adresse du restaurant le plus proche (son type, son nom etc..). Adaptez votre serveur et réalisez cette page web.