

Séance pratique MongoDB

TD 3IF, Février 2023, Előd Egyed-Zsigmond

Prérequis :

- MongoDB server Community, v6.x (téléchargez la version .zip) téléchargeable depuis : <https://www.mongodb.com/try/download/community> .
- MongoDB Shell (outil de connexion à la base de données en ligne de commande (téléchargez la version .zip)) depuis : <https://www.mongodb.com/try/download/shell> .
- MongoDB Database Tools pour charger des données (téléchargez la version .zip), téléchargeable depuis : <https://www.mongodb.com/try/download/database-tools> .
- Un éditeur de code pour écrire du code non auto correcteur (type Notepad++, SublimeText, VS Code, emacs, Notepad, Eclipse, ...) mais pas MS Word, Libre Office, Google Doc qui modifient le texte que vous écrivez sans vous le dire)
- Le client MongoDB : Studio3T Free : <https://studio3t.com/download-studio3t-free>

Préparation

A. Installation et lancement du serveur MongoDB

1. Installez le serveur MongoDB (notez bien le répertoire d'installation que nous notons *MongoDBServerHome*), téléchargé depuis le lien ci-dessus. Pour ce cours, décocher la case d'installation du serveur comme service. On va le lancer à la main. Documentation ici : <https://docs.mongodb.com/manual/administration/install-community/>
2. Créez un répertoire permettant de stocker les données de MongoDB (typiquement *C:\data\db*) que nous notons *MongoDBDataDir*
3. Lancez le serveur MongoDB à partir de la ligne de commande.
Documentation détaillée : <https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/#run-mongodb-from-cmd>
 - a. Sous Windows, lancez une fenêtre de ligne de commande (Bouton Windows + R, puis cmd puis Entrée)
 - b. Positionnez-vous dans la ligne de commande, dans le répertoire *MongoDBServerHome\bin* (remplacez avec le nom de répertoire d'installation du serveur MongoDB noté à l'étape 1) en utilisant la commande `cd`.
 - c. Tapez la ligne :
`mongod --dbpath "MongoDBDataDir"`
 - d. Des messages apparaissent en format json. La dernière devrait se terminer par :
...`"ctx":"listener","msg":"Waiting for connections","attr":{"port":27017,"ssl":"off"}}`
 - e. Le serveur MongoDB tourne en local et attend les connexions sur le port 27017

B. Connexion au serveur MongoDB et création de la base de données utilisée en TP.

1. Lancez une nouvelle fenêtre d'invité de commandes (Bouton Windows + R, puis cmd puis Entrée), positionnez-vous dans cette fenêtre de ligne de commande dans le répertoire *MongoDBServerHome\bin* (remplacez avec le nom de répertoire d'installation du serveur MongoDB noté à l'étape 1).
2. Lancez le client mongodb avec l'instruction : `mongosh`, après quelques messages, un curseur « > » doit apparaître. Ça y est vous êtes connectés.
3. Une fois connecté, créer la base tpMongo avec l'instruction
`use tpMongo`

C. Remplissage de la base de données

4. Décompresser le fichier zip contenant les outils MongoDB Database Tools.
5. Déplacez les fichiers, notamment : mongoimport.exe (sous Windows ou son équivalent pour d'autres OS) dans le répertoire `MongoDBServerHome\bin` de votre installation du serveur (à côté de mongod.exe).
6. Copiez `velov_geo.json` se trouvant dans l'archive du TP dans le répertoire bin de votre installation mongodb (`MongoDBServerHome\bin`). Ce fichier texte contient les documents décrivant les stations Vélo'V
(Ces données étaient disponibles sur le site <https://data.grandlyon.com/> plus précisément : <https://data.grandlyon.com/jeux-de-donnees/stations-velo-v-metropole-lyon-disponibilites-temps-reel/telechargements> le 11 janvier 2021)
7. Lancez une nouvelle fenêtre de ligne de commande et placez-vous dans le répertoire `MongoDBServerHome\bin`
8. Dans ce répertoire se trouve maintenant le fichier exécutable mongoimport (mongoimport.exe sous Windows) ainsi que `velov_geo.json`.
9. Lancez l'instruction (pas depuis le client mongo!! Mais bien depuis l'invité de commande du système d'exploitation) :

```
mongoimport --db tpMongo --collection velov_geo --file  
"velov_geo.json"
```

10. Cet instruction crée la base tpMongo si elle n'existe pas, la collection `velov_geo` et y insère les documents contenus dans le fichier `velov_geo.json`.
Vous pouvez exécuter mongoimport depuis n'importe où à condition que le chemin vers l'exécutable mongoimport soit enregistré dans la variable d'environnement PATH. Le paramètre « --file » doit être suivi par le chemin complet vers le fichier json entre guillemets.
11. Vous pouvez lister les données importées dans le client mongoDB lancé dans la partie B. en utilisant l'instruction :

```
db.velov_geo.find().pretty()
```

vous devriez voir des documents json correspondants aux informations des stations Vélo'V affichés ainsi que la chaîne « Type "it" for more » à la fin.

12. Votre base maintenant est créée et remplie
- Partie optionnelle : insertion de documents un par un

Créer le document suivant : { "type": "Feature",

```
"properties" : {  
  "lat" : 45.7873840849558,  
  "available_bike_stands" : NumberInt(7),  
  "gid" : NumberInt(1065),  
  "last_update" : "2017/02/02 14:49:23",  
  "number" : NumberInt(9013),  
  "address2" : "",  
  "nmarrond" : NumberInt(13),  
  "pole" : "pôle d'activité (Cegid, Infogrammes)",  
  "bike_stands" : NumberInt(20),  
  "banking" : NumberInt(0),  
  "address" : "Quai Paul Sédallon",  
  "lng" : 4.81437377940006,  
  "availabilitycode" : NumberInt(1),  
  "availability" : "Vert",  
}
```

```

    "name" : "Sédaillan / Naviguation",
    "commune" : "Lyon 9 ème",
    "bonus" : "Non",
    "status" : "OPEN",
    "available_bikes" : NumberInt(13),
    "last_update_fme" : "2017/02/02 14:54:00"
  },
  "geometry" : {
    "type" : "Point",
    "coordinates" : [
      4.814373779400059,
      45.787384084955846
    ]
  }
}

```

Insérer le document dans la collection `velov_geo` avec l'instruction :

```
db.velov_geo.save(...);
```

Créer les fiches json de deux autres stations Vélo’V (allez sur la page : <https://velov.grandlyon.com/fr/mapping>) et insérez les dans la base.

Interrogation simple

Exprimez des requêtes simples pour les recherches suivantes :

1. Liste de toutes les stations dont la commune est VILLEURBANNE
2. Lister toutes les stations qui ne sont pas à VILLEURBANNE
3. Liste des stations qui n'ont pas de seconde adresse (address2 vide)
4. Nombre de stations dont la seconde adresse a une valeur
5. Lister les stations avec plus de 2 vélos disponibles (available_bikes)
6. Lister toutes les communes distinctes
7. Lister toutes les communes distinctes, trier les résultats par ordre alphabétique
8. Lister les stations du 9^{ème} arrondissement (Lyon 9^{ème}) par ordre croissant de vélos disponibles
9. Projeter les résultats sur le nom, l'adresse et le nombre de vélos disponibles
10. Afficher les communes et le nombre de stations Vélo'V par commune
11. Triez par commune
12. Triez par nb de stations en ordre décroissante
13. Nombre de stations Vélo'V de Villeurbanne groupés par nombre de vélos disponibles triés selon ce nombre
14. Rajouter au résultat précédent, les noms des stations pour chaque nombre de vélos disponible
15. Calculer le nombre moyen de vélos disponibles par commune

Requêtes géographiques

Préciser que les coordonnées sont des données géographiques en créant un index 2dsphere :

```
use tpMongo;
db.velov_geo.createIndex( { "geometry.coordinates" : "2dsphere" } )
```

Vérifiez que l'index a bien été créé avec :

```
db.velov_geo.getIndexes() ;
```

16. Chercher les stations Vélo'V qui sont à moins de 500m du point : \$geometry: { type: "Point", coordinates: [4.863132722360224, 45.77022676914935] } (utilisez : \$near)
17. Listez les 5 stations les plus proches des coordonnées type : "Point", coordinates: [4.863132722360224, 45.77022676914935] } (utilisez : geoNear)

Requêtes textuelles

Préciser que le champ pole est de type texte et créez un index sur ce champ

```
db.velov_geo.createIndex( { "properties.pole": "text" } , { default_language: "french" })
```

18. Cherchez les stations dont le pole répond à la requête « quartiers »