

TP MongoDB

Analyse de données Twitter

Avant-propos : pour ce TP, la documentation MongoDB (<https://docs.mongodb.com/manual/>) et les supports de cours et TD (sous Moodle) sont vos meilleurs amis !

Présentation : Ce TP reprend le cas d'étude Twitter vu en TD ten appliquant la modélisation de données établie en TD. Après installation et mise en route de l'environnement MongoDB, une série de manipulations (insertions, modifications, suppressions et interrogations) seront effectuées sur un exemple de données.

1 Installation et mise en route de mongoDB

1.1 Installation

La première étape consiste à installer MongoDB Community Edition (https://www.mongodb.com/try/download/community?tck=docs_server) à partir d'une archive tgz (adaptée à votre configuration matérielle).

Les étapes d'installation de MongoDB ne sont pas détaillées ici mais dans la documentation (<https://docs.mongodb.com/manual/administration/install-community/>). Elles dépendent de votre configuration matérielle. Il suffit a priori de décompresser l'archive dans un dossier pour lequel on dispose des des droits d'accès suffisants.

L'environnement MongoDB repose sur une architecture Client-Serveur, il faut donc d'abord lancer un serveur MongoDB puis un client MongoDB dans deux fenêtre de type *Terminal* différentes.

1.2 Mise en route du serveur mongoDB

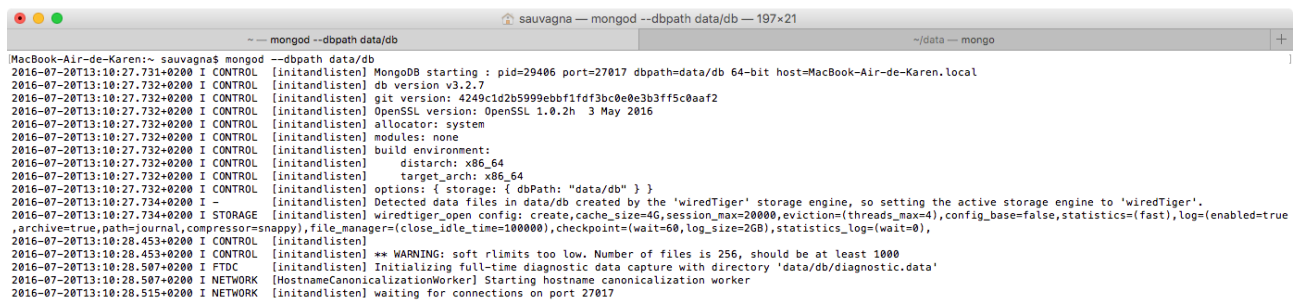
La première étape consiste à lancer le serveur (`mongod`) à partir d'une fenêtre *Terminal*.

```
>mongod
```

Si au moment de l'installation vous n'avez pas choisi le chemin par défaut pour le stockage de vos données, il faut spécifier le chemin choisi.

```
>mongod --dbpath votreChemin
```

Vous devez obtenir quelque chose du genre :



```
MacBook-Air-de-Karen:~ sauvagna$ mongod --dbpath data/db
2016-07-20T13:10:27.731+0200 I CONTROL [initandlisten] MongoDB starting : pid=29466 port=27017 dbpath=data/db 64-bit host=MacBook-Air-de-Karen.local
2016-07-20T13:10:27.732+0200 I CONTROL [initandlisten] db version v3.2.7
2016-07-20T13:10:27.732+0200 I CONTROL [initandlisten] git version: 4249c1d2b5999ebbf1fd3bc0e0e3b3ff5c0aaf2
2016-07-20T13:10:27.732+0200 I CONTROL [initandlisten] OpenSSL version: OpenSSL 1.0.2h  3 May 2016
2016-07-20T13:10:27.732+0200 I CONTROL [initandlisten] allocator: system
2016-07-20T13:10:27.732+0200 I CONTROL [initandlisten] modules: none
2016-07-20T13:10:27.732+0200 I CONTROL [initandlisten] build environment:
2016-07-20T13:10:27.732+0200 I CONTROL [initandlisten]   distarch: x86_64
2016-07-20T13:10:27.732+0200 I CONTROL [initandlisten]   target_arch: x86_64
2016-07-20T13:10:27.732+0200 I CONTROL [initandlisten] options: { storage: { dbPath: "data/db" } }
2016-07-20T13:10:27.734+0200 I - [initandlisten] Detected data files in data/db created by the 'wiredTiger' storage engine, so setting the active storage engine to 'wiredTiger'.
2016-07-20T13:10:27.734+0200 I STORAGE [initandlisten] wiredtiger_open config: create,cache_size=4G,session_max=20000,eviction=(threads_max=4),config_base=false,statistics=(fast),log=(enabled=true
,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000),checkpoint=(wait=60,log_size=2GB),statistics_log=(wait=0),
2016-07-20T13:10:28.453+0200 I CONTROL [initandlisten]
2016-07-20T13:10:28.453+0200 I CONTROL [initandlisten] ** WARNING: soft limits too low. Number of files is 256, should be at least 1000
2016-07-20T13:10:28.507+0200 I FTDC [initandlisten] Initializing full-time diagnostic data capture with directory 'data/db/diagnostic.data'
2016-07-20T13:10:28.507+0200 I NETWORK [HostnameCanonicalizationWorker] Starting hostname canonicalization worker
2016-07-20T13:10:28.515+0200 I NETWORK [initandlisten] waiting for connections on port 27017
```

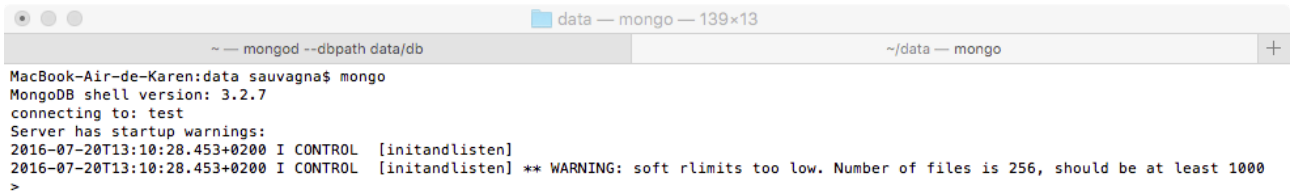
Le serveur est lancé, la dernière ligne doit indiquer que le serveur MongoDB attend les connexions de clients MongoDB sur le port 27017. Il est nécessaire de ne pas fermer cette fenêtre *Terminal*.

1.3 Invite de commande

Dans une nouvelle fenêtre *Terminal*, vous pouvez maintenant lancer le client MongoDB pour obtenir l'invite de commande :

```
>mongo
```

La connexion au serveur est établie sur la base `test` (qui est la base par défaut) et le prompt (`>`) attend les requêtes.



```
MacBook-Air-de-Karen:data sauvagna$ mongo
MongoDB shell version: 3.2.7
connecting to: test
Server has startup warnings:
2016-07-20T13:10:28.453+0200 I CONTROL [initandlisten]
2016-07-20T13:10:28.453+0200 I CONTROL [initandlisten] ** WARNING: soft rlimits too low. Number of files is 256, should be at least 1000
>
```

Quelques commandes de gestion des bases de données sur votre serveur :

- `show dbs` permet de voir toutes les bases de données existantes,
- `use mabasededonnees` permet de se connecter à la base 'mabasededonnees' (et de la créer si elle n'existe pas),
- `db` permet de savoir sur quelle base on se trouve.

2 Premières manipulations

Pour ces premières manipulations, nous reprenons l'exemple vu en TD en suivant la modélisation définie en TD, basé sur Twitter.

2.1 Insertion des données

Pour rappel, au sein d'une base de données, les données sont organisées au sein de collections (ce qui correspond à la notion de table en relationnel). En MongoDB, lorsqu'une collection n'existe pas, elle est créée dès la première insertion de données.

Il y a 2 manières d'insérer des données :

1. Insertion manuelle des enregistrements (ce que nous allons faire dans cette section)
2. Importation de données à partir d'un fichier au format Json (ce que vous devrez faire dans le projet).

Consignes – Insérez les données suivantes dans la base de données `test` suivant la **modélisation** définie en TD à laquelle on rajoutera la date de création de chaque tweet :

- 5 utilisateurs : toto, titi, tutu, tata, lolo
- titi, tutu, tata suivent toto / tutu, tata suivent titi / toto, titi, tutu suivent tata / tutu et lolo n'ont pas de followers
- toto a posté 4 tweets
 - "Ceci est mon premier tweet #youpi #BacASable", créé le 10/02/2016 à 10:50:42
 - "@titi tu fais quoi ce soir ?", créé le 10/02/2016 à 18:50:42
 - "Allez voir ceci <http://bit.ly/29O9yYc> #youpi #DanseAvecLesStars", créé le 10/05/2016 à 09:12:42
 - "Vivement les vacaaaaaances ! #jeVeuxDuSoleil", créé le 04/07/2016 à 10:17:25

- titi a posté 2 tweets :
 - "Tests sur le dernier Iphone <http://bit.ly/29TunHh> #appleJeTAime #iPhone6", créé le 11/01/2016 à 09:12:42
 - "@toto on va boire un coup ? #HappyHour", créé le 10/02/2016 à 18:54:13
- tutu a posté 2 tweets :
 - "Ceci est un test #BacASable", créé le 12/04/2016 à 18:59:03
 - "Je n'ai vraiment rien a dire #BacASable", créé le 12/04/2016 à 18:54:13
- tata et lolo n'ont pas posté de tweet

Attention : la gestion des dates est particulière dans MongoDB, il convient de lire attentivement la documentation pour insérer ce type de donnée.

Pour vérifier que l'insertion s'est bien passée, exécuter la commande `db.users.find()`, qui doit vous renvoyer tout le contenu de la collection `users`.

Maintenant que quelques données sont présentes dans la base, nous allons pouvoir effectuer quelques manipulations et interrogations.

3 Mises à jour

Consignes – Écrivez les requêtes permettant de répondre aux besoins suivants :

- Q1. Changer le nom de l'utilisateur "lolo" par "lulu".
- Q2. Ajouter le tweet "youpi un nouveau tweet" créé par l'utilisateur "titi" le 13/02/2016 à 17:44:43.
- Q3. Ajouter "tata" comme follower de "lulu".

4 Interrogation

Consignes – Écrivez les requêtes permettant de répondre aux besoins suivants :

- Q4. Lister les utilisateurs (pour vérifier une dernière fois que les informations insérées et mises à jour sont correctes).
- Q5. Obtenir les tweets postés par l'utilisateur "toto".
- Q6. Obtenir les utilisateurs qui ont utilisé le hashtag « bacasable ».
- Q7. Obtenir les utilisateurs qui sont suivis par les utilisateurs 2 ou 4.
- Q8. Obtenir les utilisateurs qui sont suivis à la fois par les utilisateurs 2 et 4.
- Q9. Obtenir les utilisateurs dont le premier tweet date d'avril 2016.
- Q10. Obtenir les tweets contenant une url (présence de la chaîne de caractères 'http').
- Q11. Obtenir le plus grand nombre de followers.
- Q12. Obtenir le deuxième tweet de l'utilisateur "titi".
Indice : allez voir du côté de l'opérateur de projection `slice` dans la documentation.
- Q13. Obtenir le nombre de tweets par utilisateur.
- Q14. Obtenir le nombre de hashtags pour chaque tweet.
- Q15. Obtenir le texte (uniquement) des tweets contenant le mot 'Ceci'.