

TP MongoDB

Avant-propos : pour ce TP, et l'utilisation de MongoDB en général, la documentation MongoDB (<https://docs.mongodb.com/manual/>) est très utile... car MongoDB n'est pas ce qu'on pourrait appeler un logiciel intuitif !

1 Installation et mise en route de mongoDB

1.1 Installation

La première étape consiste à installer MongoDB Community Edition (<https://www.mongodb.com/download-center#community>). Les étapes d'installation de MongoDB sont détaillées dans la documentation (<https://docs.mongodb.com/manual/administration/install-community/>) et dépendent de la configuration matérielle.

MongoDB étant un logiciel client-serveur, il y a deux parties à exécuter, la partie serveur et ensuite la partie client.

1.2 Mise en route du serveur mongoDB


La première étape consiste à lancer le serveur (mongod) à partir d'un terminal.

```
>mongod
```

Si au moment de l'installation aucun chemin par défaut n'a été spécifié pour le stockage des données, il faut spécifier le chemin choisi.

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

Il s'affiche alors 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=29406 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: 4249c1d2b5999ebb1fd3bc0e8e3b3ff5c0aa2
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 rlimits 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, et il attend les connexions sur le port 27017.

1.3 Invite de commande

Il est alors possible de lancer l'application cliente à partir d'un autre terminal pour obtenir l'invite de commande :

```
>mongo
```

Il s'affiche alors quelque chose du genre :



```
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
>
```

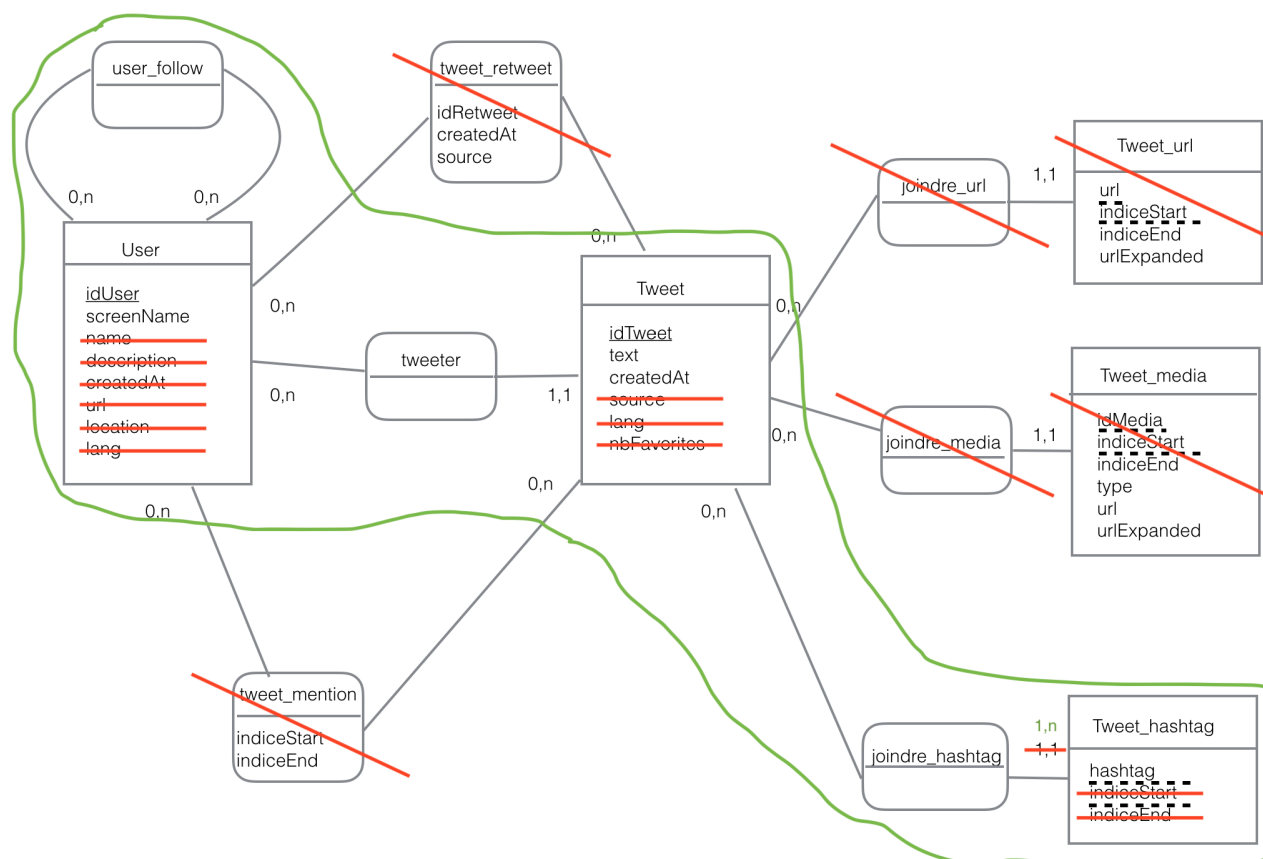
La connexion au serveur est établie sur la base **test** (qui est la base par défaut) et le prompt (>) attend les requêtes.

Quelques commandes de gestion des bases de données sur le serveur sont utiles à retenir :

- `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, en le simplifiant, le cas d'étude Twitter introduit précédemment.



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. Utilisation d'un fichier de données au format Json (utile si les données sont issues d'autres applications).

Q1. : Insérer les enregistrements suivants dans la collection `users` sur la base de données `test`:

```
1 {
2   _id: 1,
3   screenName: "toto",
```

```

4  nbFollowers: 3,
5  tweets: [{
6      text: "Ceci est mon premier tweet #youpi #BacASable",
7      createdAt: "2016-02-10T10:50:42",
8      hashtags: ["youpi", "bacasable"]
9  },
10     {
11        text: "@titi tu fais quoi ce soir ? ",
12        createdAt: "2016-02-10T18:50:42"
13    },
14     {
15        text: "Allez voir ceci http://bit.ly/29O9yYc #youpi #DanseAvecLesStars",
16        createdAt: "2016-05-10T09:12:42",
17        hashtags: ["youpi", "danseaveclesstars"]
18    },
19     {
20        text: "Vivement les vacaaaaaances ! #jeVeuxDuSoleil",
21        createdAt: "2016-07-04T10:17:25",
22        hashtags: ["jeVeuxDuSoleil"]
23    }
24 ],
25 followers: [2, 3, 4]
26 }
27 {
28     _id: 2,
29     screenName: "titi",
30     nbFollowers: 2,
31     tweets: [{
32         text: "Tests sur le dernier Iphone http://bit.ly/29TunHh #appleJeTAime #iPhone6",
33         createdAt: "2016-01-11T09:12:42",
34         hashtags: ["applejetaime", "iphone6"]
35     },
36     {
37         text: "@toto on va boire un coup ? #HappyHour",
38         createdAt: "2016-02-10T18:54:13",
39         hashtags: ["HappyHour"]
40     }
41 ],
42 followers: [3, 4]
43 }
44 {
45     _id: 3,
46     screenName: "tutu",
47     nbFollowers: 0,
48     tweets: [{
49         text: "Ceci est un test #BacASable",
50         createdAt: "2016-04-12T18:59:03",
51         hashtags: ["bacasable"]
52     },
53     {
54         text: "Je n'ai vraiment rien a dire #BacASable",
55         createdAt: "2016-04-12T18:54:13",
56         hashtags: ["bacasable"]
57     }
58 ]
59 }
60 {
61     _id: 4,
62     screenName: "tata",
63     nbFollowers: 3,
64     followers: [1, 2, 3]
65 }

```

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

2.2 Requêtes simples

Écrire les requêtes MongoDB suivantes :

- Q2.1.** Changez le nombre de followers de l'utilisateur toto à 5.
- Q2.2.** Ajoutez le tweet text:"youpi un nouveau tweet" à l'utilisateur d'_id 2.
- Q2.3.** Donnez la requête permettant d'obtenir le deuxième tweet de l'utilisateur "titi". Aide: Voir du côté de l'opérateur de projection `slice` (<https://docs.mongodb.com/manual/reference/operator/projection/slice/>).
- Q2.4.** Donnez la requête permettant d'obtenir les tweets contenant une url (présence de la chaîne de caractères 'http')
- Q2.5.** Donnez la requête permettant d'obtenir le plus grand nombre de followers.
- Q2.6** Donnez la requête permettant d'obtenir les utilisateurs qui sont suivis par les utilisateurs 2 ou 4.
- Q2.7** Donnez la requête permettant d'obtenir les utilisateurs qui sont suivis à la fois par les utilisateurs 2 et 4.
- Q2.8.** Donnez la requête permettant d'obtenir les utilisateurs dont le premier tweet date d'avril 2016.
- Q2.9.** Donnez la requête permettant d'obtenir les utilisateurs qui ont utilisé le hashtag « bacasable ».
- Q2.10.** Donnez la requête permettant d'obtenir le nombre de tweets par utilisateur.
- Q2.11.** Donnez la requête permettant d'obtenir le nombre de hashtags pour chaque tweet.
- Q2.12.** Supprimez toutes les données de la collection `users`.