

MongoDB

1. Premiers pas

1.1. Mac OS

<https://docs.mongodb.com/manual/tutorial/install-mongodb-on-os-x/>

1.1.1. Avec Homebrew

1.1.1.1. Installation

Dans un terminal :

```
$ brew update
```

```
$ brew install mongodb
```

1.1.1.2. Démarrage

Dans un terminal, démarrer le serveur :

```
$ mongod
```

Dans un second terminal (laisser le premier actif), démarrer le shell Mongo :

```
$ mongo
```

1.1.2. Avec les fichiers d'installation

Télécharger les sources correspondant à votre PC, **onglet Community**:

<https://www.mongodb.com/download-center#community>

Dans la suite, vous devrez probablement modifier les commandes surlignées en jaune afin d'ajouter les chemins complets par exemple.

Se placer dans le bon répertoire:

```
$ cd path_to_downloaded_file
```

Décompresser (peut se faire avec l'explorateur de fichiers):

```
$ tar zxvf mongodb-xxxx.tgz
```

Placer le dossier décompressé (sans l'extension.tgz) dans le HOME :

```
$ mv mongodb-xxxx ~/
```

Créer le répertoire de stockage:

```
$ sudo mkdir -p /data/db
```

```
$ sudo chmod 777 /data/db
```

1.1.2.1. Démarrage

Se placer dans le répertoire du fichier binaire:

```
$ cd ~/mongodb-linux-x86_64-ubuntu1604-3.6.1/bin
```

Lancement du serveur MongoDB:

```
$ ./mongod --dbpath /data/db
```

Le terminal reste "figé", c'est normal !

Dans un **2ème terminal** (ne pas fermer le 1er), se placer dans le répertoire du fichier binaire et lancer le MongoDB shell:

```
$ cd ~/mongodb-linux-x86_64-ubuntu1604-3.6.1/bin
```

```
$ ./mongo
```

1.2. Debian / Ubuntu

<https://docs.mongodb.com/manual/administration/install-on-linux/>

1.3. Documentation

<https://docs.mongodb.com/>

Comparaison avec la syntaxe SQL :

<https://docs.mongodb.com/manual/reference/sql-comparison/>

Guides de démarrage :

<https://docs.mongodb.com/manual/tutorial/getting-started/>

<http://blog.xebia.fr/2010/12/15/mongodb-en-pratique/>

<https://scotch.io/tutorials/an-introduction-to-mongodb>

<https://www.codeproject.com/Articles/1037052/Introduction-to-MongoDB>

1.4. Création d'une collection à partir d'un fichier JSON

Par exemple, création de la **collection emails** dans la **base de données test**. Dans un **3ème terminal**, se placer dans le répertoire des fichiers binaires et importer le fichier :

```
$ cd ~/mongodb-linux-x86_64-ubuntu1604-3.6.1/bin
```

```
$ ./mongoimport --db test --collection emails --file enron.json
```

2. TP

2.1. Emails

Décompresser le fichier `enron.json.bz2` (peut être fait en double-cliquant sur le fichier dans l'explorateur):

```
$ bunzip2 enron.json.bz2
```

1. Importer `enron.json` dans une collection nommée "mails"
2. Nombre total d'emails ?
3. Nombre total d'emails envoyés avec des adresses du domaine `enron.com`
4. Lister les emails transférés (*subject* commence par "FW:")
5. Combien de temps a pris la dernière requête ?
6. Créer un index avec le bon champ pour faire en sorte que la requête précédente soit plus rapide
7. Combien de temps a pris cette nouvelle requête ?
8. Trouver la date, l'expéditeur (*sender*) et l'objet (*subject*) de tous les messages envoyés à rosalee.fleming@enron.com
9. Supprimer lizard_ar@yahoo.com dans tous les emails dont il est le destinataire
10. Ajouter rod.bradley@enron.com comme destinataire de tous les emails envoyés par rosalee.fleming@enron.com

2.2. Codes postaux

Décompresser le fichier `zips.json.bz2` et importer `zips.json` dans une collection "zips"

1. Lister les 10 zones (une zone correspond à un code postal = 1 "zip code") les plus peuplées en Californie. Même chose pour la Louisiane
2. Lister les 10 zones suivantes (classées entre 11 et 20)
3. Ajouter un champ "country" avec la valeur "USA" à tous les documents
4. Lister toutes les zones avec plus de 100 000 habitants situées à l'ouest du méridien -74
5. Quelle est la zone la plus proche des coordonnées -73.996705, 40.74838 ?
Il faut créer un index :

```
> db.zips.createIndex( { loc : "2d" } ) ;
```
6. Lister les villes qui sont distantes de moins de 5 km de -73.996705, 40.74838
7. Lister les villes qui ont plus de 500 000 habitants
8. La requête suivante ne retourne aucun résultat :

```
> db.zips.find({city:{regex:"^N"}})
```


Quel est le problème ?
Pourquoi MongoDB ne retourne pas d'erreurs lors de son exécution ?
Quelle est la requête correcte ?