

Objectifs

prendre en main une base de données NoSQL orientée documents. l'exemple de MongoDB

1 Installation du serveur MongoDB

Afin de pouvoir faire tourner une version de MongoDB en local, nous allons le télécharger sous sa forme tarball. Mais sinon il existe une version “packagée” pour chacun des systèmes. `apt-get` permet de l'installer et mettre à jour tout ce qui est nécessaire, mais il faut pour cela être `root` ;-)

1. Aller sur <https://www.mongodb.com/download-center>
2. Dans l'onglet `server`, mettre `system` : Ubuntu et package : `tar` puis télécharger la version `tgz`
3. Décompresser cette archive : `tar zxvf mongodb-linux-aarch64-ubuntu1804-4.2.0.tgz`
4. Créer une variable système `MONGO` contenant le répertoire d'installation : `export MONGO=le_bon_chemin`
5. Mettre à jour la variable `PATH` pour faciliter l'accès aux exécutables `export PATH=$PATH:$MONGO/bin`

(Idéalement, si vous souhaitez conserver cette configuration, ces variables devraient être déclarées dans le `.bashrc`)

2 Bien démarrer

1. Créez un répertoire dans votre arborescence (test par exemple) : `mkdir test`
2. Lancez le serveur avec ce répertoire comme hébergement de bases : `mongod -dbpath /test`
3. Lancez dans une autre fenêtre shell le client : `mongo`
4. Le client mongo est avant tout une console javascript. Testez d'abord un peu de code Javascript !

```
2+3
a=5
2*a
function f(x){return 2*x+5;}
f(3)
function fact(n) { if (n == 0) { return 1; } else { return n * fact(n-1); } }
for(i=1;i<20;i++){print(i+"\t"+ fact(i));}
```

5. Au départ il existe 3 bases : `admin`, `config`, `local`
`show dbs`
6. Au lancement du client, si rien n'est précisé, on se trouve alors dans une base nommée `test`
`db`
7. Le changement de base se fait avec la commande `use`

```
use toto
db
use test
```

8. Pour quitter le client mongo
`quit()`

3 premiers pas

1. Re-connectez vous à la base test

```
mongo test
```

2. Créer une nouvelle collection nommée auteur

```
db.createCollection("auteur")
```

3. Ajouter quelques données dedans

```
db.auteur.insert({nom:"DURAND",prenom:"philippe",ville:"ici"})
```

4. On peut aussi utiliser des objets javascript pour effectuer l'insertion

```
var obj={}
obj.nom="DURAND"
obj.prenom="pierre"
obj.ville="labas"
obj
db.auteur.insert(obj)
```

5. Ou même toute une liste de documents d'un coup

```
var liste =
[
{nom:"BEDIER", prenom:"Vincent", login:"bedierv"},
{nom:"BLANQUART", prenom:"Hugo", login:"blanquah"},
{nom:"BRESSY", prenom:"Nicolas", login:"bressyn"},
{nom:"CANONNE", prenom:"Theo", login:"canonnet"},
{nom:"DELGRANGE", prenom:"Pierre", login:"delgranp"},
{nom:"DUBOIS", prenom:"Guillaume", login:"duboisg"},
{nom:"FRETEL", prenom:"Liza", login:"fretell"},
{nom:"HEUGUE", prenom:"Guilhem", login:"heugueg"},
{nom:"KARAOUZENE", prenom:"Nagi", login:"karaouzn"},
{nom:"LAHOUSSE", prenom:"Ludovic", login:"lahoussl"},
{nom:"LANDSCHOOT", prenom:"Tony", login:"landscht"},
{nom:"LETORET", prenom:"Jimmy", login:"letoretj"},
{nom:"MONCHEAUX", prenom:"Bastien", login:"moncheab"},
{nom:"PAKULA", prenom:"Nicolas", login:"pakulan"},
{nom:"PASI", prenom:"Gaetan", login:"pasig"},
{nom:"PICAVET", prenom:"Alexandre", login:"picaveta"},
{nom:"PLAISANT", prenom:"Florentin", login:"plaisanf"},
{nom:"PROIX", prenom:"Danae", login:"proixd"},
{nom:"PROVOST", prenom:"Florian", login:"provostf"},
{nom:"SHIPMAN", prenom:"Thomas", login:"shipmant"},
{nom:"TELLIER", prenom:"Benjamin", login:"tellierb"},
{nom:"THEBAUD", prenom:"Emeric", login:"thebaude"},
]
db.auteur.insert(liste)
```

Vous remarquerez que les 2 premiers enregistrements (qui ont une ville) n'ont pas le même schema que les suivant (qui ont un login). Un SGBD comme Mongo se moque complètement des schémas.

6. Pour voir les collections créées : show collections

7. Pour lister un document de la collection : db.auteur.findOne()

8. Pour lister toute la collection db.auteur.find()

9. et pour l'afficher formatée : db.auteur.find().pretty()

10. Pour rechercher un document : db.auteur.find(prenom:"pierre")

11. Pour supprimer les documents contenant un prenom de valeur "pierre"

```
db.auteur.remove({prenom:"pierre"})
```

12. Pour quitter : quit()

13. Et bien sûr lire le manuel ! : <https://docs.mongodb.com/manual/>

Vous voilà prêts à être lancés dans le grand bain !

4 Exercice

1. Récupérez à partir de moodle l'archive employes.bson

2. Importez cette archive dans votre base : mongorestore --db test employes.bson

3. Afficher toutes les collections de la base
4. Lister le contenu de la collection `employes`
5. Compter le nombre de documents de la collection `employes`
6. Afficher deux documents sous la forme pretty
7. Afficher la liste des employés dont le prénom est `David`
8. Afficher la liste des employés dont le prénom commence ou se termine par D (utiliser une expression régulière)
9. Afficher les nom et prénom de chaque employé ayant une ancienneté > 10
10. Afficher la liste des employés dont soit le prénom est `David`, soit l'ancienneté est > 10
11. Afficher les nom et adresse complète des employés ayant un attribut `rue` dans l'objet `adresse`
12. Incrémenter de 200 la prime des employés ayant déjà un champ `prime`
13. Afficher les trois premières personnes ayant la plus grande valeur d'ancienneté
14. Regrouper les personnes dont la ville de résidence est Toulouse (afficher nom, prénom et ancienneté uniquement)
15. Mettre à jour l'adresse de Dominique Mani: nouvelle adresse (numero : 20, ville : 'Marseille', codepostal : '13015'). Attention, il ne doit plus y avoir d'attribut `rue` dans `adresse`
16. Calculer et afficher la moyenne des anciennetés de toute la collection
17. Calculer et afficher la somme de l'ancienneté par prénom d'employé