

Pronote : installer et découvrir MongoDB

Installation

Lien : <https://docs.mongodb.com/manual/tutorial/install-mongodb-on-debian/>

- se connecter en SSH au serveur Debian sur lequel est installer NodeJS
- importer la clé publique

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv  
0C49F3730359A14518585931BC711F9BA15703C6
```

- créer le fichier /etc/apt/sources.list.d/mongodb-org-3.4.list <code shell> echo "deb <http://repo.mongodb.org/apt/debian> jessie/mongodb-org/3.4 main" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.4.list </code>
- Mettre à jour la liste des paquets <code shell> sudo apt-get update </code>
- installer MongoDB <code shell> sudo apt-get install -y mongodb-org </code>
 - lancer MOngoDB qui sera en écoute sur le port 27017 <code shell> sudo service mongodb start </code>

Première prise en main

Lien : <https://openclassrooms.com/courses/guide-de-demarrage-pour-utiliser-mongodb>

- Le format JSON (JavaScript Object Notation) est utilisé pour l'insertion et la restitution des documents.
 - MongoDB stocke les documents dans un format binaire appelé BSON.
 - * Lancer l'interpréteur de commandes :

```
<code shell> mongo
```

```
|
```

```
<
```

```
/code>
```

Créer une base de données

- il suffit de se positionner sur une base qui est alors créée si elle n'existe pas : <code>

```
| use moyennesswitched to db moyennes
```

```
|<
```

```
/code>
```

- Une base de données contient des collections dans lesquelles on ajoute des documents de

même nature. Dans un SGBDR il s'agirait d'une table.

création d'une collection

<code>

```
db.annee.insert({"annee":2016,"classe":"B1 SIO","periode":"Semestre 2","nbEleves":35,"moyenneClasse":11,19})
```

<

/code>

- lire le contenu d'une table ; MongoDB ajoute une propriété "*id*" qui est un identifiant unique *ObjectId* pour le document si celui-ci n'est pas spécifié à l'insertion. <code> > db.annee.find() { "id" : ObjectId("5913641110464938586b0faf"), "annee" : 2016, "classe" : "B1 SIO", "periode" : "Semestre 2", "nbEleves" : 35, "moyenneClasse" : 11,19 } </code>
- MongoDB est schemaless, ce qui signifie que les documents n'ont pas tous à respecter le même format.

Faire une recherche

- ajout de plusieurs enregistrements de classe puis recherche sur une classe précise

```
>db.annee.insert({"annee":2016,"classe":"B1 SIO","periode":"Semestre 1","nbEleves":35,"moyenneClasse":11,76})
>db.annee.insert({"annee":2016,"classe":"B1 SIO","periode":"Année (Deux semestres)","nbEleves":35,"moyenneClasse":11,48})
> db.annee.find({"moyenneClasse":11,76})
```

- compter le nombre de résultats <code>

```
db.annee.find({"classe":"B1 SIO"}).count()
```

<

/code>

- Les opérandes

\$gt	plus grand que
\$lt	plus petit que
\$gte	plus grand ou égal à
\$lte	plus petit ou égal à
\$or	ou
\$and	et
\$in, \$all, \$exist, \$type et \$regex ...	

<code> db.annee.find({moyenneClasse: {\$gte:11, \$lt:11,50}}) </code>

Choisir les informations à afficher

```
db.annee.find({moyenneClasse: {$gte:"11", $lt:"11,50"}}, {"_id":0, "periode":1})
```

- Mettre la valeur 0 à *id* pour que ce champ ne soit pas affiché ===== Trier avec la méthode *sort()* ===== <code> db.annee.find({"moyenneClasse": {\$gte:"11", \$lt:"11,50"}}, {"id":0, "periode":1}).sort({"moyenneClasse":1})</code>
- mettre -1 pour un tri décroissant

Exemple

- créer un document année et ajouter successivement les classes

```
db.annee.insert({annee: 2016})
// ajouter une propriété
db.annee.update({annee:2016}, {$set : {classe:"B1 SIO"}})
// supprimer une propriété
db.annee.update({annee:2016}, {$unset : {classe:1}})
// ajouter un tableau classe
db.annee.update({annee:2016}, {$set : {classe:[]}})
// ajouter un élément au tableau
db.annee.update({annee:2016}, {$push : {classe: "B1 SIO"}})
db.annee.update({annee:2016}, {$push : {classe: "B2 SIO"}})
db.annee.update({annee:2016}, {$push : {classe: {libelle:"B2 SIO"}}})

//supprimer un document
db.annee.remove({})
```

- gérer les tableaux de moyennes du lycée Suzanne Valadon <code>
 db.classes.insert({'établissement':'0870019y', 'année':'2017'})
 db.classes.update({'établissement':'0870019y', 'année':'2017'}, {\$push : {classe: {libelle:"B1SIO"} }})
 db.classes.update({'établissement':'0870019y', 'année':'2017'}, {\$push : {classe: {libelle:"B2SIO"} }})
 </code>

MongoDB avec Node.JS

Lien : <http://mongodb.github.io/node-mongodb-native/2.2/quick-start/quick-start/>

Retour projet 2017

- PROJET 2017

From:

/ - APs et stages du BTS SIO du lycée Suzanne Valadon



Permanent link:

/doku.php/hackathon/pronote/mongodb

Last update: **2017/06/07 10:09**