

MongoDB

Michaël MATHIEU

10 février 2023

1 Introduction

MongoDB est le leader des bases de données NoSQL. Notre vision est qu'une base de données NoSQL est utilisée pour restituer rapidement et simplement des données, mais n'est pas le *master* des données. En effet, une base de données relationnelle est plus adaptée pour garantir notamment l'absence de redondance, et surtout l'intégrité référentielle.

2 L'écosystème MongoDB

Si pour une base de données MongoDB en production, il suffit d'installer le serveur (*community edition* dans notre cas) et d'y ajouter les données¹, pour notre environnement de développeur, nous avons besoin d'installer plusieurs outils afin que l'écosystème MongoDB nous permette de travailler.

2.1 Serveur

1. télécharger le serveur community edition au **format zip** (<https://www.mongodb.com/try/download/community>)
2. décompresser le zip (par exemple, `C:\mongodb`)
3. créer un répertoire `data` (`C:\mongodb\data`)
4. ajouter le répertoire `bin` à la variable d'environnement système `Path` (`C:\mongodb\bin`)
5. démarrer le serveur (depuis le répertoire de MongoDB : `mongod --dbpath data`)

2.2 Outils de base de données et shell

1. télécharger
 - (a) les outils de base de données (<https://www.mongodb.com/try/download/database-tools>)
 - (b) le shell (<https://www.mongodb.com/try/download/shell>)
2. décompresser les zip
3. placer le contenu du répertoire `bin` des zip dans le répertoire `bin` de MongoDB

2.3 GUI – Compass

1. télécharger l'interface graphique permettant de voir ce qu'il y a dans la base de données et modifier des données (<https://www.mongodb.com/try/download/compass>)
2. décompresser le zip
3. lancer `MongoDBCompass.exe`
4. cliquer sur le bouton *Connect*

1. soit programmatiquement, soit en important un dump de données (souvent le cas lorsque l'on télécharge une base de données opendata comme par exemple <https://fr.openfoodfacts.org/data>)

3 Importer un dump de données

1. télécharger le zip contenant le dump (par exemple celui d'*OpenFoodFacts*)
2. décompresser le zip qui contient le répertoire `dump`
3. importer le dump (depuis le répertoire où se trouve le répertoire `dump : mongorestore`)
4. vérifier avec *Compass* que la base de données a été importée
5. le répertoire `dump` peut être supprimé

4 Créer un dump de données

1. depuis le dossier où l'on souhaite créer le dump de données : `mongodump --db nomDeMaBaseDeDonnes` (si on souhaite extraire *toutes* les bases de données que l'on a (par exemple pour une sauvegarde), il suffit simplement de lancer `mongodump` sans préciser le nom d'une base de données
2. un répertoire `dump` a été créé
3. compresser le répertoire pour sa distribution

5 Clé primaire

- le champ `_id` joue le rôle de la *clé primaire*
- chaque *document* d'une table contient un champ `_id`, si vous ne l'indiquez pas, lors de l'insertion d'un document, MongoDB en générera un
- le champ `_id` peut être composé de plusieurs champs contenu dans un `Object` par exemple
- △ dans une clé composée de plusieurs champs, l'**ordre** dans lequel les champs sont définis a son importance : `{"a": 1, "b": 1} != {"b": 1, "a": 1}`

6 Utiliser MongoDB avec C#

6.1 Bibliothèques à installer

Tools → NuGet Package Manager → Manage NuGet Packages for Solution...

Toutes les bibliothèques nécessaires sont fournies dans le paquet *MongoDB.Driver*.

6.2 Les types MongoDB

- `BsonDocument` : correspond à un objet
- `BsonValue` : correspond à un attribut
- `BsonArray` : correspond à une liste d'objets

6.3 Exemples

Inutile de réinventer la roue : un tutoriel très intéressant a déjà été écrit et vous permettra d'aller plus loin : <https://zetcode.com/csharp/mongodb/>

7 Activer la réplication

Pour illustrer ce mécanisme², nous avons besoin d'au minimum 3 instances de MongoDB.

△pour que la réplication fonctionne correctement, il faut un nombre **impair** d'instances³!

Soit sur des machines différentes (chaque instance utilisant une adresse IP différente⁴) :

Hôte	Port	Rôle	Dossier	ReplicaSet
192.168.0.10	27017	Primary	data	rep1
192.168.0.20	27017	Secondary	data	rep1
192.168.0.30	27017	Secondary	data	rep1

Soit sur la même machine (chaque instance utilisant un port différent) :

Hôte	Port	Rôle	Dossier	ReplicaSet
localhost	27010	Primary	data1	rep1
localhost	27020	Secondary	data2	rep1
localhost	27030	Secondary	data3	rep1

7.1 Configurer le nœud primaire

1. Ouvrir la console
2. Démarrer le nœud : `mongod --dbpath data1 --logpath "data1/mongod.log" --bind_ip localhost --port 27010 --journal --replSet rep1`
3. Ouvrir une nouvelle console (cmd1) et se connecter au nœud : `mongosh --host localhost --port 27010`
 - (a) `> rsconf={_id:"rep1", members:[{_id:0, host:"localhost:27010"}]}`
 - (b) `> rs.initiate(rsconf)`
 - (c) `> enter`
 - (d) la console doit afficher désormais : `rep1 [direct: primary] test>`
 - (e) garder la console ouverte

7.2 Configurer les nœuds secondaires

1. Ouvrir une nouvelle console
2. Démarrer le nœud : `mongod --dbpath "data2" --logpath "data2/mongod.log" --bind_ip localhost --port 27020 --journal --replSet rep1`
3. Ouvrir une nouvelle console
4. Démarrer le nœud : `mongod --dbpath "data3" --logpath "data3/mongod.log" --bind_ip localhost --port 27030 --journal --replSet rep1`

7.3 Ajouter les nœuds secondaires

Depuis cmd1 :

1. `> rs.add("localhost:27020")`
2. `> rs.add("localhost:27030")`
3. `> rs.status()` doit afficher l'instance 27010 comme PRIMARY et les instances 27020 et 27030 comme SECONDARY

2. <https://www.mongodb.com/docs/manual/replication/>

3. <https://www.mongodb.com/docs/manual/tutorial/deploy-replica-set-for-testing/#overview>

4. il faut remplacer les adresses 192.168.0.x par l'adresse IP de vos machines

7.4 Vérifier que la réplication fonctionne

1. Ajouter un élément avec *Compass* sur l'instance 27010
2. Eteindre l'instance 27010
3. Se connecter sur le clusters
`mongodb://localhost:27010,localhost:27020,localhost:27030/?replicaSet=rep1`
4. Vérifier que la donnée ajoutée précédemment est disponible