



Introduction NoSQL

TP

Laurent Broudoux (@lbroudoux)
Janvier 2014

Exercice 1

Objectifs : Appréhender l'utilisation d'une base « document » : MongoDB

Conditions : Installation de MongoDB sur le poste. Installation déroulée au début du TP. Projet sous Eclipse avec dépendance externe qui sera installée. Les librairies junit-4.11.jar et harmcrest-core-1.3.jar pourront éventuellement être utilisées pour compléter le projet par des tests unitaires.

Préparation : Installation

Installation de MongoDB sous Windows :

- Récupérer la dernière version (version 2.4.8) sous <http://www.mongodb.org/downloads> (bien choisir 32 ou 64 bits) sous forme d'archive Zip,
- Extraire le tout sous un répertoire unique (ex : [d:\](#) ou [e:\mongodb](#))
- Créer un répertoire \data sous ce répertoire unique et un autre répertoire \db sous \data
- Lancer la commande suivante (hypothèse : vous avez installé sous [d:\](#)) depuis un Command Prompt DOS :

```
> d:\mongodb\bin\mongod.exe --dbpath d:\mongodb\data
```

Installation de MongoDB sous Unix / Linux : si vous êtes assez geek pour avoir un Unix / linux, je ne me fais pas de soucis pour vous ;-)

Présentation Cas d'usage

Nous allons utiliser MongoDB pour stocker les données relatives à une plateforme de blogging. Les entités métiers à gérer seront donc les suivantes :

- Article (title, body, note, author)
- Optionnelle : User (username, displayname, password, birthdate)

Partie 1 : Console et client

- Lancer le client mongo.exe depuis l'invite de commande DOS en utilisant la commande suivante :

> d:\mongodb\bin\mongo.exe

- Avec le client, appréhender l'environnement Mongo en déroulant les exemples de <http://docs.mongodb.org/manual/tutorial/getting-started/>
- Télécharger le client graphique opensource Umongo depuis <http://edgytech.com/umongo/> (version 1.6.1). Dézipper l'archive dans un répertoire de votre choix. Executer simplement le fichier umongo.exe ou écrivez un fichier .cmd vous permettant de trouver JAVA si le système n'a pas de défaut
- Une fois lancé, connectez-vous à la base locale (utilisation du défaut). Celle-ci fonctionne sur le port 27017. Vous pouvez notamment essayer d'entrer l'URL suivante dans votre navigateur : <http://localhost:27017>
- Appréhender le client Umongo en reexécutant les requêtes réalisés dans le tutorial avec le client console.

Note : prenez le temps de bien noter la nature et la structure du champs « _id » présent sur tous les documents

Partie 2 : Programmation Java

- Commencer par créer un nouveau projet Eclipse (Projet Java simple). Ce projet devra être enrichi avec le driver MongoDB pour Java disponible en téléchargement ici : <http://central.maven.org/maven2/org/mongodb/mongo-java-driver/2.11.3/mongo-java-driver-2.11.3.jar>. Ce jar devra être enregistré comme librairie du projet Eclipse
- Créer une simple classe Article qui correspondra à l'objet devant être persisté dans MongoDB. Pour pouvoir identifier de façon unique et non ambiguë un article, ne pas oublier de lui ajouter un attribut id de type String
- A l'instar du TP sur AOP, créer maintenant une interface en utilisant

le design pattern Repository (voir <http://martinfowler.com/eaCatalog/repository.html>). Elle portera des méthodes permettant de converser avec le système de stockage

- Définir une méthode Article save(Article)
 - Définir une méthode findByAuthor(String)
 - Bonus : définir toutes les autres méthodes que vous pensez utiles : findByTitleLike(String), count(), countByAuthor(String), etc ...
- Il est maintenant temps de créer une implémentation de cette interface Repository ; implémentation qui utilisera MongoDB et masquera toutes les mécaniques de mapping entre notre objet Article et les objets techniques propres à MongoDB

Aide : la classe de base à utiliser est la classe com.mongodb.MongoClient. Comme vous avez déroulé le tutorial, vous connaissez maintenant les concepts qui permettent de gérer la structure de MongoDB. Chaque concept possède son pendant en classe Java ... La javadoc est disponible ici : <http://api.mongodb.org/java/2.11.3/>

Aide : pour la gestion des id, on pourra exécuter simplement les commandes
article.setId(object.getObjectId("_id").toString()) et
new ObjectId(article.getId())

- Afin que les choses soient les plus propres possibles, créer maintenant en utilisant le design pattern Factory (voir <http://www.oodeesign.com/factory-pattern.html>), une instance de cette implémentation MongoDB du repository

Aide : afin de conserver les choses les plus simples possibles, on pourra partir du fait que c'est la factory qui fournit le nom de la base ainsi que les informations de connexion à l'implémentation (vous pouvez utiliser par exemple une base nommée « ensim »)

- Créer finalement une classe de Test Junit ou une classe main permettant de tester. Cette classe aura les responsabilités suivantes :
- Initialiser un ensemble d'objets Article qui serviront de test (faites en sorte que les notes positionnés sur les articles soient définies de façon aléatoire – entre 0 et 5 par exemple)
 - Appeler la factory et utiliser l'implémentation retournée
 - Tester les méthodes de votre repository
- Créer pour terminer une classe de Test Junit ou une classe main au

sein de laquelle nous allons définir un job Map/Reduce

- Ce job devra retourner la note moyenne attribuée aux articles de notre base
- Pour MongoDB, les fonctions Map et Reduce sont à exprimer en Javascript (voir <http://docs.mongodb.org/manual/tutorial/map-reduce-examples/>). Ces fonctions doivent être passées en argument de la méthode `DBCollection.mapReduce(1, 2, 3, 4, 5)` où le 3ème et 5ème arguments seront positionnés à null

Aide : afin d'élaborer plus facilement vos fonctions, regarder du côté de Umongo...

Exercice 2

Objectifs : Appréhender l'utilisation d'une base «graph » : Neo4J

Conditions : Installation de MongoDB sur le poste. Installation déroulée au début du TP.

Préparation : Installation

Installation de Neo4j sous Windows :

- Récupérer la dernière version (2.0.0) à l'adresse suivante : <http://www.neo4j.org/download>. Extraire le contenu de l'archive dans un répertoire unique (ex : [e:\neo4j](#))
- Dans une invite de commande DOS, se positionner sous le répertoire d'installation et lancer simplement la commande suivante (le JDK 1.7 doit être installé)

> bin/neo4j start

Client Web

- Lancer votre navigateur et aller sur <http://localhost:7474>. L'installation Neo4j contient un client web qui va permettre d'expérimenter Cypher.
- Quelques commandes pour commencer :

```
CREATE (p:Person {name:"Arthur"}) RETURN "Hello", p.name
```

```
CREATE (p:Person {name:"Fanette"}) RETURN "Hello", p.name
```

```
MATCH (a:Person),(b:Person) WHERE a.name = 'Arthur' AND b.name =  
'Fanette' CREATE (a)-[r:Soeur]->(b) RETURN r
```

```
MATCH (a:Person),(b:Person) WHERE a.name = 'Arthur' AND b.name =  
'Fanette' CREATE (b)-[r:Frere]->(a) RETURN r
```

```
MATCH (n) OPTIONAL MATCH (n)-[r]-() DELETE n,r
```

- Quelques commandes pour continuer de découvrir :
<http://docs.neo4j.org/chunked/milestone/cypher-query-lang.html>