

**LO54 - MongoDB**

BRAVO Yvan / KENDE Aloïs

*Printemps 2017*

*Projet de gestion de l'offre de formation d'une école privée à travers un développement J2EE.*

I – Contexte du projet

II – Tutoriel d’utilisation de MongoDB

* Installation du client & serveur MongoDB via exécutable
* Lancement (et création) de la base (mongod.exe)
* Création d’une base de données
* Création d’une collection
* Installation du driver JDBC Mongo (pom.xml)
* Ajouter un élément à une collection depuis Java
* Visualiser le contenu de la base de données avec Robo 3T (optionnel)

III – Rapport d'expérience

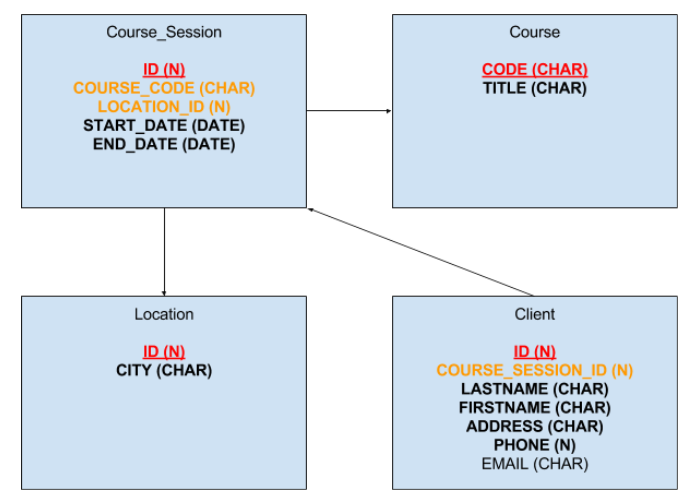
* Type de base orienté document
* No-SQL n’est pas la réponse à tout

IV – Conclusion

1 - Contexte du projet

Le projet consiste à réaliser une application permettant la gestion des inscriptions à des cours en utilisant les technologies web, annotations et diverses conventions JEE. Le système se base sur une architecture SOA et stock ainsi les informations dans une base de données relationnelle.

Notre sujet complémentaire est MongoDB, un système de gestion de base de données NoSQL. Dans notre projet, la dimension MongoDB se cantonnera à ajouter chaque enregistrement d’étudiant dans la base de données MongoDB en parallèle de la base de données relationnelle classique (JavaDB (?)).



*Schéma de l’architecture de la base de données*

2 – Tutoriel d’utilisation mongoDB

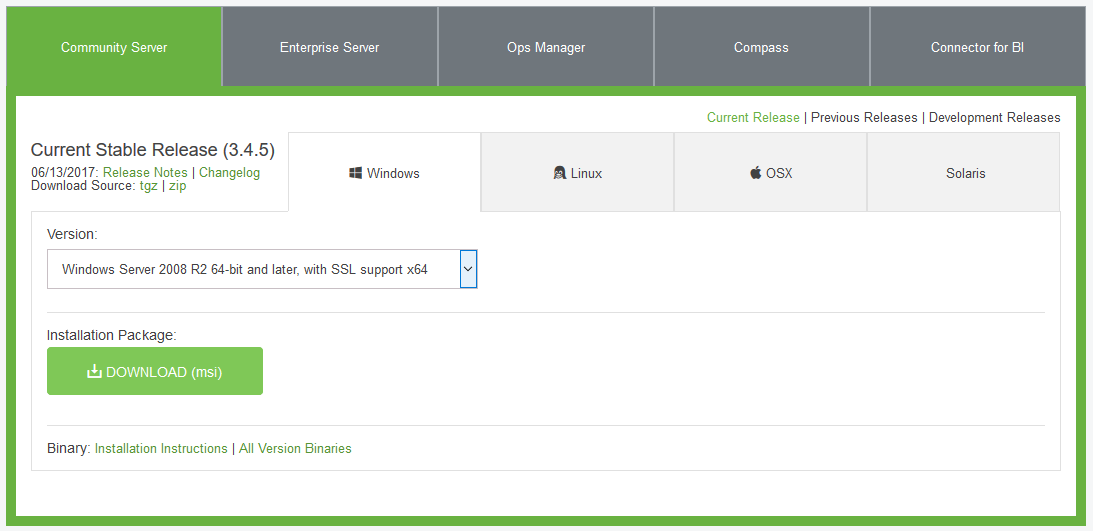
Plusieurs éléments sont nécessaires à mettre en place pour faire fonctionner la base MongoDB ; nous allons en faire le tour.

D’un côté il faut installer la base de données, sur un serveur ou en local (dans notre cas, ce sera en local). Ensuite on aura besoin d’avoir le driver JDBC pour MongoDB pour aller manipuler la base avec notre code. Enfin, un dernier élément facultatif à installer est Robo 3T, anciennement Robomongo, un outil avec une interface graphique simple qui permet de visualiser et manipuler la base de données MongoDB.

* Installation du client & serveur MongoDB

On peut télécharger la dernière version stable directement sur le site officiel de MongoDB :

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



Pour Windows 7 ou Windows 10, vous pouvez prendre sans soucis la version proposée par défaut en arrivant sur la page (Windows Server 2008 R2 64-bit & later, with SSL support x64).

Lancez ensuite le fichier téléchargé. Il vous faut d’abord accepter les termes du contrat de licence. Choisissez ensuite installation Custom si vous souhaitez choisir le répertoire d’installation de la base MongoDB.

* Lancement (et création) de la base

Selon l’OS utilisé et les droits du compte, il est possible que vous deviez créer manuellement les dossiers /data/db à l’endroit où vous souhaitez que la base soit créée. Par défaut, si vous souhaitez faire au plus simple, créez ce dossier /data/db à la racine du disque où vous venez d’installer votre serveur MongoDB.

Il vous suffit ensuite de démarrer mongod.exe présent dans le répertoire d’installation où vous avez installé la base (par défaut :

C:\Program Files\MongoDB\Server\3.4\bin\mongod.exe

Si vous avez souhaité créer /data/db ailleurs qu’à la racine, créez un fichier mongod.bat et inscrivez ceci dedans :

@echo off

start "C:\Program Files\MongoDB\Server\3.4\bin\mongod.exe --dbpath C:\mongodb\data”

exit **A TESTER**

Remplacez la première adresse par là où est installé le serveur MongoDB, et remplacez la seconde adresse par l’adresse où vous avez créé les dossiers /data/db.

Au lieu de lancer le serveur avec mongod.exe, vous lancerez alors le serveur avec votre fichier mongod.bat

* Création d’une base de données

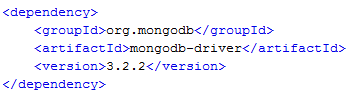
Cette étape existe ?

* Création d’une collection

Cette étape existe ?

* Installation du driver JDBC Mongo (pom.xml)

Pour installer le driver JDBC pour MongoDB, il suffit de copier-coller la dépendance dans Maven, c’est-à-dire dans le fichier pom.xml du projet. La dépendance sera ensuite importée au prochain « build Maven » du projet.



*La dépendance à ajouter dans le pom.xml (à jour du 18/06/2017)*

* Ajouter un élément à une collection depuis Java

Pour ajouter un élément à une collection depuis Java il faut déjà instancier un MongoClient en lui donnant l’adresse de la base (localhost si le serveur est sur votre PC), et le port de connexion (27017 par défaut).



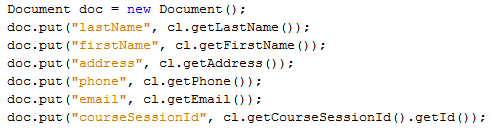
Ensuite il suffit de créer un élément MongoDatabase qui récupère les informations d’une base de données issue du MongoClient avec en argument bien sûr le nom de la base :

C:\Users\Norgannon\AppData\Local\Microsoft\Windows\INetCache\Content.Word\MongoDatabase.png

La classe MongoCollection permet de manipuler une collection de données.

C:\Users\Norgannon\AppData\Local\Microsoft\Windows\INetCache\Content.Word\MongoCollection.png

Pour ajouter un élément à la collection, on pourra simplement instancier un nouveau Document et lui ajouter toutes les informations que l’on souhaite :



Via la méthode insertOne([document]) de MongoCollection, on pourra ajouter ce nouvel élément à la collection :

C:\Users\Norgannon\AppData\Local\Microsoft\Windows\INetCache\Content.Word\InsertOne.png

Si vous n’avez plus besoin de la connexion, il est possible de la fermer en utilisant la méthode close de MongoClient :

C:\Users\Norgannon\AppData\Local\Microsoft\Windows\INetCache\Content.Word\close.png

Mis bout à bout, l’exemple donne ceci :



* Visualiser le contenu de la base de données avec Robo 3T (optionnel)

Documentation additionnelle pour l’installation de MongoDB :

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