

## TME - Développement et réplication avec MongoDB



### I°) Environnement de travail

Télécharger Maven 3 à l'adresse : <https://maven.apache.org/download.cgi>

Dans le répertoire \${HOME}/.m2/, créer un fichier settings.xml et ajouter les informations de proxy de l'université. La documentation de la configuration d'un proxy maven est ici :

<https://maven.apache.org/guides/mini/guide-proxies.html>

Si vous utilisez IntelliJ IDEA, pensez à ajouter l'auto import sur maven (une popup apparaît en bas de l'IDE)

Télécharger le projet Java à l'adresse suivante :

<https://github.com/olivier-pitton/dant/raw/master/mongodb/tp/tp.zip>

Dézippez le fichier téléchargé.

Dans Eclipse, faites Clic droit (dans *Package Explorer*) -> *Import project* et sélectionnez le projet.

Dans IntelliJ, importez le projet en tant que Maven Project. S'il y a eu un soucis d'import (pom.xml et fichiers Java non reconnus ...), faites Clic droit sur le project, *Add Framework Support, Maven*.

**Attention importez votre projet en tant que Maven project !**

Une fois le projet importé, vous pourrez démarrer le TP.

Le projet est structuré de la manière suivante :

1. *src* : Contient les sources du projet. Chaque classe développée devra se trouver dans le package adéquat.
2. *test* : Contient un simple test pour valider votre travail

### II°) Développement de l'API

## **2.1°) Création de l'objet métier**

Mettez à jour la classe Account pour qu'elle respecte :

- Ajoutez une variable d'instance de type ObjectId
- Annotez votre classe et variables d'instance avec ce que l'on a vu en cours
- Ajoutez les getters / setters sur les champs email / created
- Ajoutez les méthodes equals / hashCode. Le champ à tester est "email"
- Ajoutez une méthode toString qui renvoie bien l'email

## **2.2°) Implémentation standard avec MongoDB**

Mettez à jour les méthodes de la classe AccountServiceImpl pour qu'elles effectuent bien leur travail. Vous pouvez récupérer une instance de Datastore de Morphia grâce à la classe MongoUtil.

Lancez le test unitaire AccountServiceImpl pour valider votre travail.

## **2.3°) Mise en place d'un mécanisme de réplication**

Lancez deux serveurs sur votre machine de telle manière que le serveur sur le port 27017 soit le maitre et le serveur sur le port 27018 soit un esclave.

Relancez le test unitaire et regardez grâce au shell Mongo que la réplication a bien eu lieu

Faites valider votre travail par l'encadrant.

## **2.4°) Optimisation de l'architecture (optionnel)**

Notre système comprend 10 fois plus de lectures que d'écritures et ne doit pas perdre de données, en aucun cas.

Comment optimiser notre modèle en prenant cela en considération ?

Faites valider votre travail par l'encadrant.