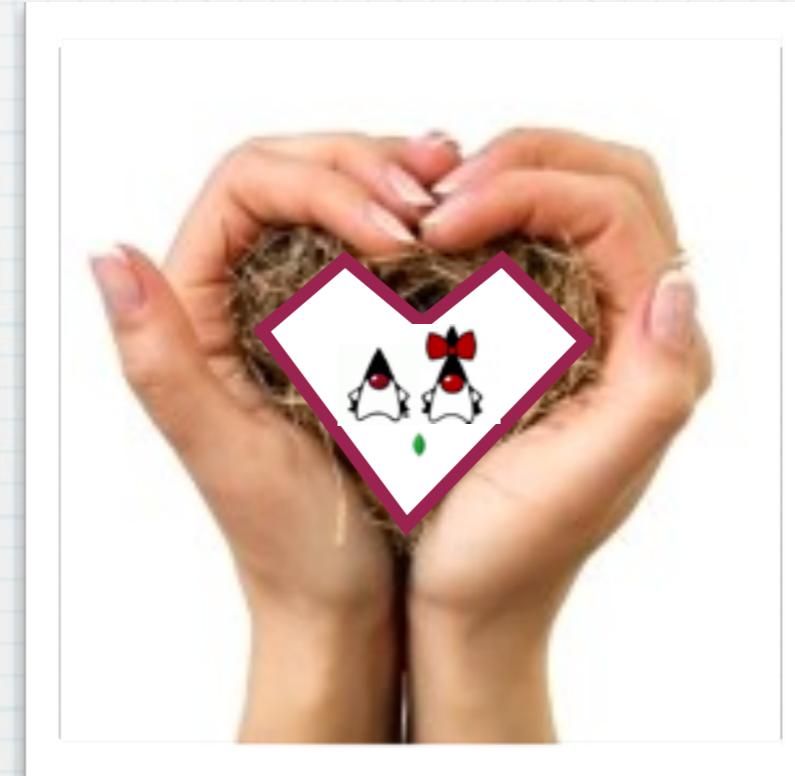


Hands-On mongoDB



Xebia



Workshop Guide

Planning



{ "Les outils" : 1 }



{ "Driver Java" : 2 }



{ "Replica Set" : 3 }

Part 1 : Les Outils

- * Ex 1 : Installation
- * Ex 2 : Quick Shell
- * Ex 3 : MongoRestore
- * Ex 4 : MongoDump
- * Ex 5 : MongoExport (Bonus)
- * Ex 6 : Mongolimport (Bonus)





HELP

<http://www.mongodb.org/display/DOCS/Quickstart+OS+X>

<http://www.mongodb.org/display/DOCS/Import+Export+Tools>



Ex 1 : Installation

- * Décompresser le fichier téléchargé :
 - * `mongodb-osx-x86_64-2.0.1.tgz`
- * Créer répertoire pour stocker la data
 - * Ex : `/Users/karesti/`
- * Exécuter :
 - * `./mongod --dbpath /Users/karesti/HANDS-ON-DATA --rest`
 - * `http://localhost:28017/_commands`



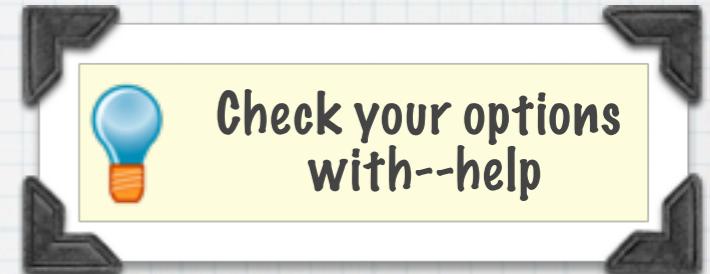
Ex 2 : Quick Shell

- * Démarrer le shell
 - * `./mongo`
- * `> db, >db.help()`
- * `> use Duchess`
- * `> db, > show dbs`
- * Check REST -> <http://localhost:28017>



Ex 3 : MongoRestore

- * Décompresser le fichier hands-on-mongodb/dumpMongo/WhatToDo.zip
- * Utiliser MongoRestore pour importer la base de données
 - * `./mongorestore /{your_path}/WhatToDo`
- * Vérifier quelques données -> `use WhatToDo`
 - * `db.events.count() > 13.000`, `db.events.findOne()`
 - * `db.events.find({"name": /Concert/})`.-> Events avec le mot concert dans le champ name
 - * `db.events.find({"name": /Concert/}).limit(5)`.-> Limiter à 5
 - * `db.events.find({"name": /concert/}).count() -> 32`
 - * `db.events.find({"name": /concert/, "venue.town": "Paris"}, {"name": 1}) ->`





Ex 4 : MongoDump

- * Réaliser un backup

- * db.copyDatabase("WhatToDo", "WhatToDoBackUp")

- * Utiliser **.mongodump** pour faire un dump de la base «WhatToDoBackUp»

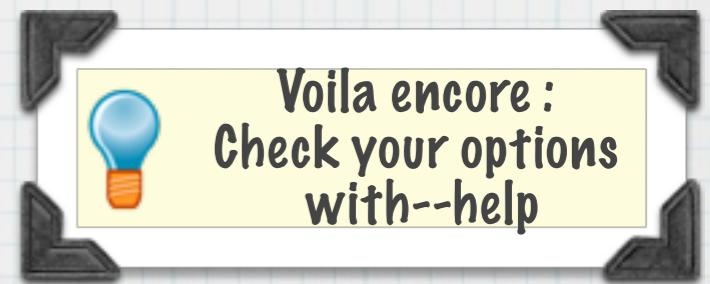
- * Check File System !





Ex 5 : MongoExport

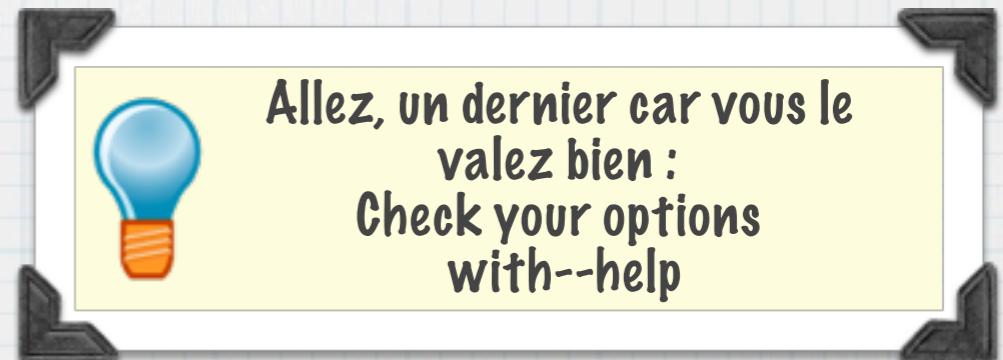
- * Exporter dans un fichier uniquement le champ «**venue**» des events avec le mot concert dans leur nom de la base **WhatToDo**
 - * `{"name": /concert/}`
- * Utiliser les paramètres --query, --db





Ex 6 : Mongolimport

- * Importer le fichier que vous avez exporté dans la base «**JustSomeVenues**» dans la collection «**venues**»
- * Vérifier les données :
 - * `use JustSomeVenues`
 - * `db.venues.count()`



Part 2 : Java Driver

- * Ex 1 : Création d'une connexion par défaut à la base de données locale
- * Ex 2 : Create, Find and Remove By ID
- * Ex 3 : Find By Criterias
- * Ex 4 : Update One, Update More
- * Ex 5 : Create Index





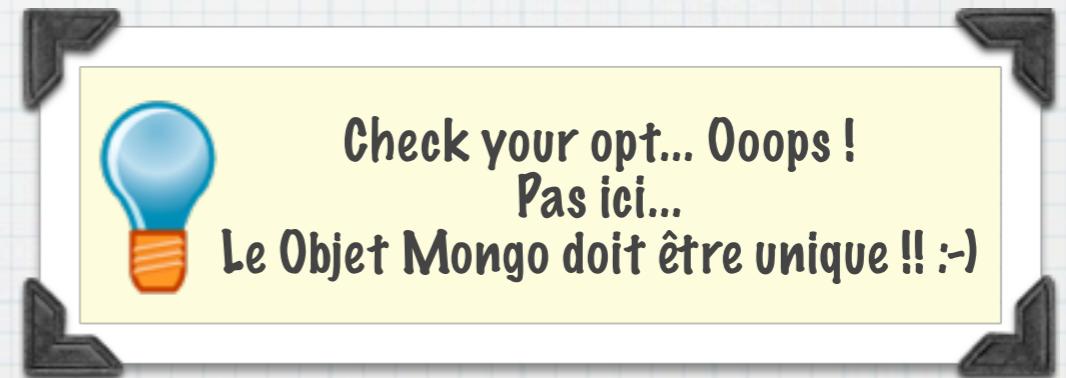
Part 2 : Help

<http://www.mongodb.org/display/DOCS/Java+Tutorial>



Ex 1 : Connexion DB

- * //TODO: Part 2 - EX1 : Create connection
- * Création d'une instance de Mongo





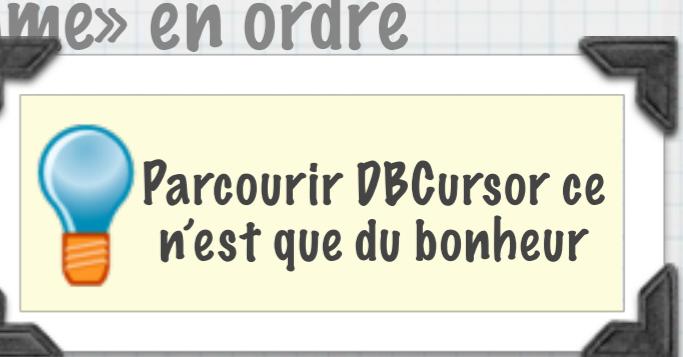
EX 2 : Create, Find and Remove By ID

- * // TODO: Part 2 - EX2 : Create, Find and Remove By ID
- * Annotations Morphia sur les beans :
@Entity, @Embedded, @Id
- * QueryBuilder.start(...)
- * Utiliser l'objet **mapper** de Morphia pour réaliser la transformation DBObject - Event



Ex 3 : Find By Criterias

- * // TODO: Part 2 - EX3 - Search
should_search_with_multiple_criteres
- * Find All Events Avec «concert» dans leur «name».
- * Max 10 résultats
- * Pattern patternGiven = Pattern.compile
("." + string + ".");
- * Trier les résultats par le champ le champ «name» en ordre
ascendant
- * Utiliser la classe Pattern de java.





EX 4 : Update 1, Update More

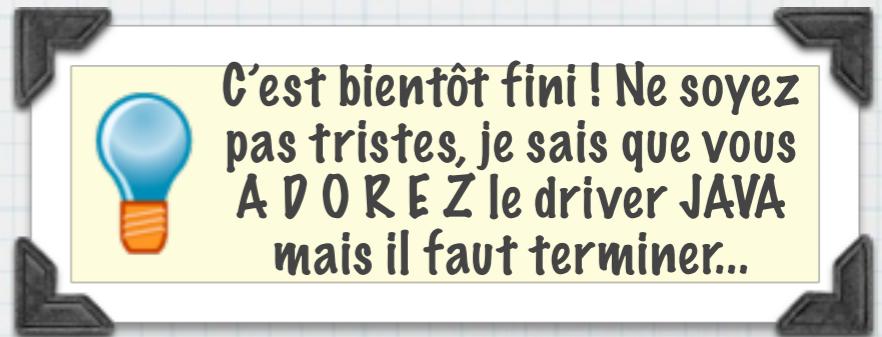
- * // TODO: Part 2 - EX4 : Update One, Update More - should_be_able_to_add_another_session
 - * Query par ID, update query «`$push`»
- * // TODO: Part 2 - EX4 : Update One, Update More - should_be_able_to_do_multiple_updates
 - * La méthode «`updateNameWithSessionDate`» qui est franchement mal nommée, doit chercher tous les événements avec une session à la date donnée et modifier son nom avec le nom passé.
- * Query avec Session , update query «`$set`»

 Cette fonctionnalité n'a aucun sens ...



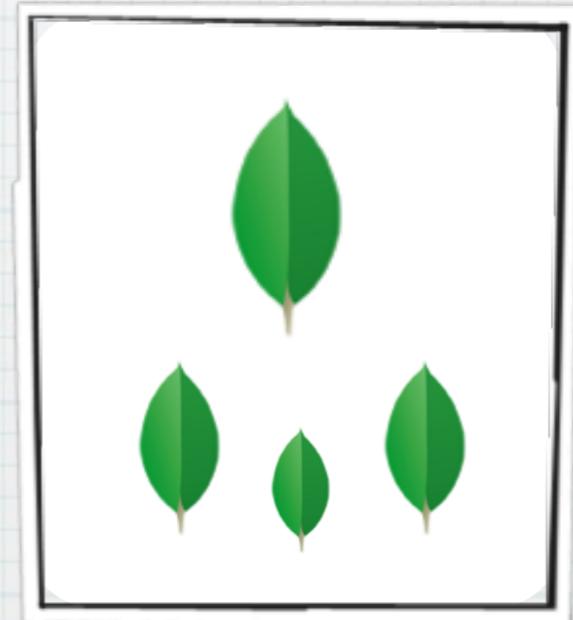
Ex 5 : Index On Name

- * // TODO: Part 2 - EX5 : Create index on name
 - * Vérifie dans la console -> db.events.getIndexes()
 - * C'est possible aussi avec une annotation Morphia



Part 3 : Replica Set

- * Ex 1 : Init
- * Ex 2 : Ajouter des membres et Arbitre
- * Ex 3 : Kill The Master
- * Ex 4 : Adapte Code Java

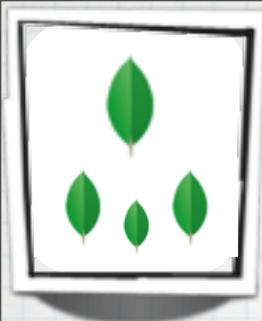




Part 3 : Help

<http://www.mongodb.org/display/DOCS/Replica+Sets>

[http://www.mongodb.org/display/DOCS/Replica+Set
+Semantics](http://www.mongodb.org/display/DOCS/Replica+Set+Semantics)



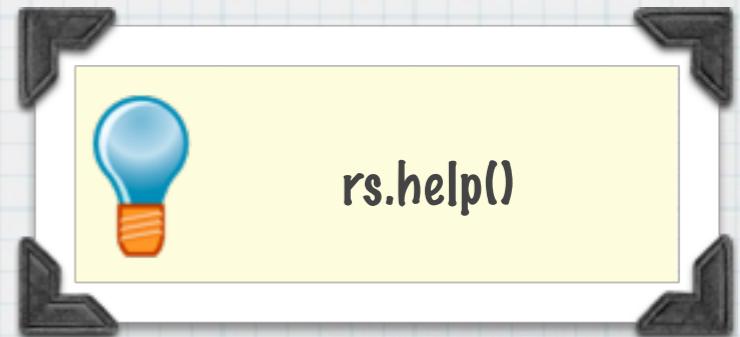
Ex1 : Init

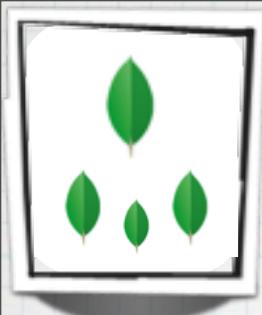
- * Kill Current Running Server
 - * use admin
 - * db.shutdownServer()
- * Création de 3 dossiers
 - * data1, data2, data3
- * Démarrer trois instances de mongod chacun avec dpath et un port différent (27017, 27018, 27019)
- * replSet -> une valeur, le même pour tous ex: mySet



Ex 2 : Quick Shell

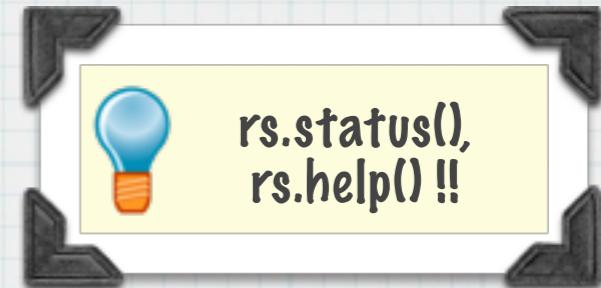
- * se connecter ./mongo
- * rs.initiate() -- attendre
- * rs.status()
- * Ajouter member
 - * rs.add("name.host:port")
- * tester rs.stepDown()

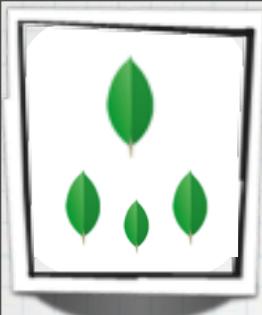




Ex 3 : Kill The Master

- * Démarrer une 4ème sur un autre port (27020)
- * rs.addArb(lle dernier server)
- * arrêter les primary à chaque fois
- * regarder ce qui se passe dans le status





Ex 4 : Java Connexion

- * Adapte le code java pour se connecter au replica set
- * utiliser mongodump avec le replica set
- * faire passer les tests

