

TP MongoDB

Alexandre GINGEMBRE et Mathieu MOUROT



Installation

- Linux et macOS :
- <https://www.mongodb.com/download-center?jmp=nav>
- Commande : mongo

Installation

- Windows :
 - <https://www.mongodb.com/download-center?jmp=nav>
 - Créez un dossier C:\data
 - Créez un dossier C:\data\db
 - Modifiez le Path pour ajouter mongo
 - Cliquez-droit sur ordinateur > Modifier les paramètres > Nouveau Ajouter le dossier de mongo : C:\Program Files\MongoDB\Server\3.4\bin
 - Lancer le démon mongo pour permettre à windows d'accéder au serveur :
 - mongod
 - Autre fenêtre de commande :
 - mongo

TP Hello World

- Utilisez la base de helloWorld :
 > use helloWorld

TP Hello World

- Insertion d'une ligne d'une collection :

```
> db.personne.insert({
  "nom": "toto",
  "prenom": "titi",
  "adresse": {
    "numero": 32,
    "rue": "Michel Ney ",
    "codePostale": "54300",
    "ville": "Lunéville"
  },
  "couleurYeux": "bleu",
  "couleurCheveux": "chatain",
  "voiture": [{
    "marque": "Renault",
    "modele": "clio"
  }, {
    "marque": "Dacia",
    "couleur": "verte"
  }]
})
```

TP Hello World

- Effectuer une recherche sur les données insérées :

```
> db.personne.find()
```

TP Hello World

- Effectuer une recherche avec des prédicats :

```
> db.personne.find({"nom":"toto"})
```

TP Hello World

- Effectuer une recherche sur un tableau :

```
> db.personne.find( { "voiture.marque":  
"Dacia" } )
```


TP Hello World

- Effectuer une recherche avec des opérateurs :

```
> db.personne.find({"adresse.numero": { $gte :  
10} })
```

TP Hello World

- Effectuer des recherches avec des opérateurs booléens :

- ET

```
> db.personne.find( { "couleurYeux": "bleu",  
"adresse.numero": 32 } )
```

- OU

```
> db.personne.find( { "couleursYeux": "bleu"},  
{ "adresse.numero": 32 } )
```

TP Hello World

- Trier des résultats :

```
> db.personne.find().sort( { "nom": 1,  
"adresse.numero": 1 } )
```

TP Hello World

- Mettre à jour une donnée :

```
> db.personne.update( { "prenom" : "titi" }, { $set:  
  { "adresse.numero": 13 } })
```

TP Hello World

- Supprimer une donnée :

```
> db.personne.remove( { "nom": "toto" } )
```

TP Hello World

- Quitter proprement :

> exit

Documentation

- Documentation <https://docs.mongodb.com>
- Documentation des commandes et pour en découvrir de nouvelles : <https://docs.mongodb.com/getting-started/shell/>
- Le GitHub de MongoDB : <https://github.com/mongodb/mongo>

TP Java - MongoDB

- Initialisation :

```
> wget https://raw.githubusercontent.com/mongodb/docs-assets/primer-dataset/primer-dataset.json
```

Ou récupérer le fichier sur la DropBox

TP Java - MongoDB

- Importer les données dans mongo

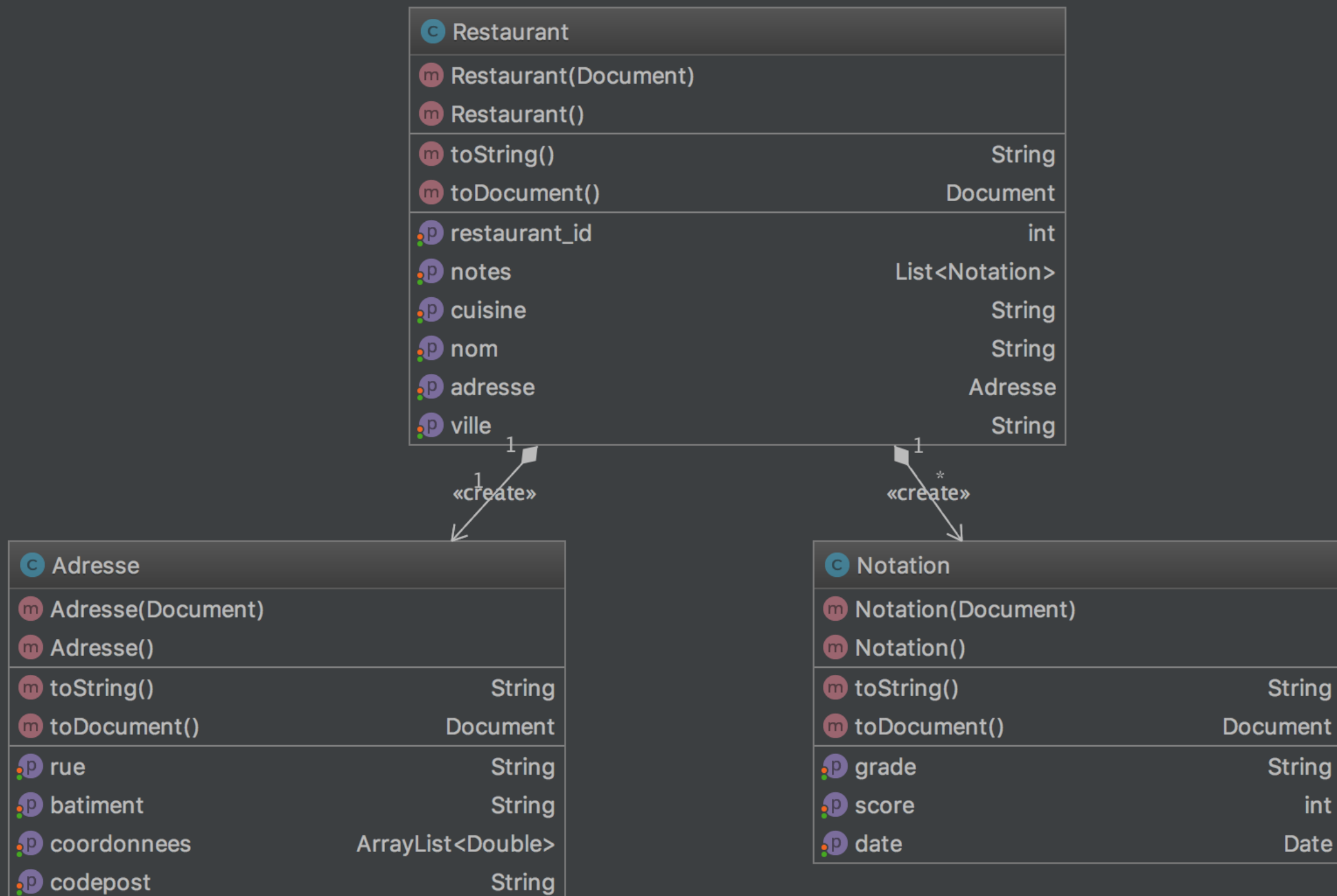
> mongoimport --db test --collection restaurants
--drop --file \$YOUR_DIRECTORY\$/primer-
dataset.json

Analyse d'un document

```
{ "_id" : ObjectId("5810715c22edb6fe9773440c"), "address" : { "building" : "129-08", "coord" : [ -73.839297, 40.78147 ], "street" : "20 Avenue", "zipcode" : "11356" }, "borough" : "Queens", "cuisine" : "Delicatessen", "grades" : [ { "date" : ISODate("2014-08-16T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2013-08-27T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2012-09-20T00:00:00Z"), "grade" : "A", "score" : 7 }, { "date" : ISODate("2011-09-29T00:00:00Z"), "grade" : "A", "score" : 10 } ], "name" : "Sal'S Deli", "restaurant_id" : "40361618" }
```

- Extrait d'un `db.restaurants.find()`
- Constat 1 : Un document est entre accolade
- Constat 2 : On a plusieurs types de données (Double, String, Array)
- Constat 3 : Un document peut contenir d'autres documents

TP Java - MongoDB



Implémentation et récupération des objets

- Objectif :
 - Manipuler les documents
- TODO :
 - Les constructeurs avec comme paramètre un document

Tests : Initialisation et connexion

```
public class HelloWorldPlusPlus {  
    static MongoClient mc ;  
    static MongoDBDatabase mdb ;  
    static MongoCollection collection;  
  
    @BeforeClass  
    public static void init(){  
        mc = new MongoClient( host: "localhost", port: 27017);  
        mdb = mc.getDatabase( databaseName: "test");  
        collection = mdb.getCollection( s: "restaurants");  
    }  
  
    @Test  
    public void testConnection(){  
        DB db = mc.getDB( dbName: "test");  
        Set<String> collections = db.getCollectionNames();  
        assertTrue(collections.contains("restaurants"));  
    }  
}
```

Parcours du dataset

- Objectifs :
 - Effectuer un parcours sur les données stockées dans MongoDB avec java
 - Tester le code de vos constructeurs pour les différentes classes implémentés précédemment
- TODO :
 - Un test sans condition pour récupérer les données

Effectuer une recherche simple

- Objectifs :
 - Effectuer une recherche dans Mongo
 - Continuer à manipuler l'objet FindIterable
 - Manipuler les BasicDBObject
- TODO :
 - Faire un test avec une recherche du restaurant où :
restaurant_id = 40361618

Effectuer une recherche un peu plus complexe

- Objectifs :
 - Se familiariser avec le parcours d'un document imbriqué
 - Utiliser une fonction de MongoDB
- TODO:
 - Faire un test qui nous retourne les restaurants ayant eu une note de 4 ou 8
- Indice : Utilisez les valeurs suivantes : 4 et 8

La personne qui renseigne la base n'est pas très douée

- Objectifs :
 - Se rappeler de ne jamais faire confiance à l'utilisateur
 - Utiliser une fonction
- TODO :
 - Une fonction qui recherche les restaurants de sushi
- Indice :
 - Mario Pizza est un restaurant qui vend des pizzas
 - Les expressions régulières, c'est la vie

Connaitre les différentes valeurs d'un champs

- Objectifs :
 - Voir en quoi Mongo est proche de SQL
 - Manipuler un autre Iterable : DistinctIterable.
- TODO:
 - Afficher (une seule fois) chaque ville

Faire une insertion d'un document

- Objectifs :
 - Insérer un document dans la collection
 - Maîtriser le marshalling d'objets
- TODO :
 - Insérer un nouveau restaurant

MAJ d'un objet

- Objectif :
 - Modifier un document existant dans la base de données
- TODO :
 - Modifier le document précédemment inséré dans la base

Suppression d'un document

- Objectif :
 - Finaliser le CRUD avec MongoDB et Java
- TODO :
 - Supprimer le document que vous avez créé puis modifié

MongoDB Rest

- Lancer le démon MongoDB avec le paramètre `--rest` et s'amuser avec Postman
- <http://127.0.0.1:28017/databaseName/collectionName/>
- <https://docs.mongodb.com/ecosystem/tools/http-interfaces/>

Feedback

Merci pour votre attention et votre participation
avez vous des questions ?

Sources

- Documentation MongoDB
- Forums