# Introduction to MongoDB

## Overview

* MongoDB is an open source [**document-oriented database**](http://en.wikipedia.org/wiki/Document-oriented_database) system developed and supported by [**10gen**](http://en.wikipedia.org/wiki/10gen). It is part of the [**NoSQL**](http://en.wikipedia.org/wiki/NoSQL) family of database systems. Instead of storing data in tables as is done in a "classical" [**relational database**](http://en.wikipedia.org/wiki/Relational_database), MongoDB stores structured data as [JSON](http://en.wikipedia.org/wiki/JSON)-like documents with dynamic schemas (MongoDB calls the format [**BSON**](http://en.wikipedia.org/wiki/BSON)), making the integration of data in certain types of applications easier and faster.

## MongoDB Data Model

* A MongoDB deployment hosts a number of databases. A [**database**](http://docs.mongodb.org/manual/reference/glossary/#term-database) holds a set of collections. A [**collection**](http://docs.mongodb.org/manual/reference/glossary/#term-collection) (equal table in SQL) holds a set of documents. A [**document**](http://docs.mongodb.org/manual/reference/glossary/#term-document) (equal record in SQL) is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection’s documents may hold different types of data.

## Main features

### Flexibility

* + MongoDB stores data in JSON documents (which we serialize to [BSON](http://bsonspec.org/)). JSON provides a rich data model that seamlessly maps to native programming language types, and the dynamic schema makes it easier to evolve your data model than with a system with enforced schemas such as a RDBMS.

### Power

* + MongoDB provides a lot of the features of a traditional RDBMS such as secondary indexes, dynamic queries, sorting, rich updates, upserts (update if document exists, insert if it doesn’t), and easy aggregation. This gives you the breadth of functionality that you are used to from an RDBMS, with the flexibility and scaling capability that the non-relational model allows.

### Speed/Scaling

* + By keeping related data together in documents, queries can be much faster than in a relational database where related data is separated into multiple tables and then needs to be joined later. MongoDB also makes it easy to scale out your database. Autosharding allows you to scale your cluster linearly by adding more machines. It is possible to increase capacity without any downtime, which is very important on the web when load can increase suddenly and bringing down the website for extended maintenance can cost your business large amounts of revenue.

### Ease of use

* + MongoDB works hard to be very easy to install, configure, maintain, and use. To this end, MongoDB provides few configuration options, and instead tries to automatically do the “right thing” whenever possible. This means that MongoDB works right out of the box, and you can dive right into developing your application, instead of spending a lot of time fine-tuning obscure database configurations.

## Operations

* MongoDB is a server process that runs on Linux, Windows and OS X. It can be run both as a 32-bits or 64-bits application. MongoDB is limited to a total data size of about 2GB for all databases in 32-bit mode.
* The MongoDB process listens on port 27017 by default.
* Clients connect to the MongoDB process, optionally authenticate themselves if security is turned on, and perform a sequence of actions, such as inserts, queries and updates.
* MongoDB stores its data in files (default location is /data/db/), and uses memory mapped files for data management for efficiency.

# Tutorial

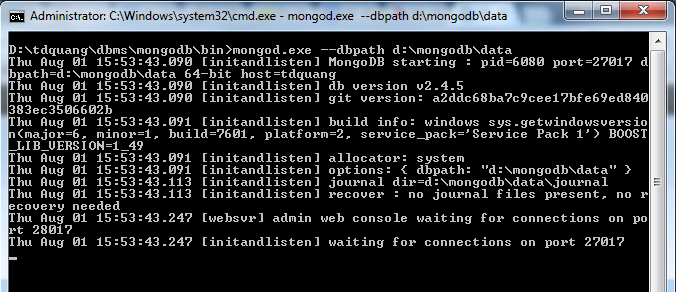
## Install MongoDB on Windows

* Download and extract the latest production release of MongoDB from the [MongoDB downloads page](http://www.mongodb.org/downloads).
* Set up the environment variable for MongoDB.
* To start MongoDB, execute from the Command Prompt:

mongod.exe

* MongoDB requires a [**data folder**](http://docs.mongodb.org/manual/reference/glossary/#term-dbpath) to store its files. The default location for the MongoDB data directory is **C:\data\db**. Please create this folder or specify an alternate path for \data\db with the [**dbpath**](http://docs.mongodb.org/manual/reference/configuration-options/#dbpath) setting:

mongod.exe --dbpath d:\mongodb\data



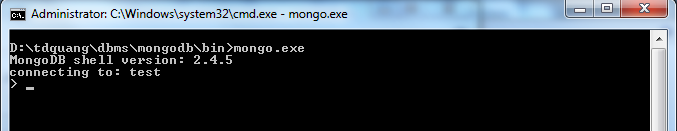
## Getting Started with MongoDB

* Start MongoDB, execute from the Command Prompt:

mongod.exe

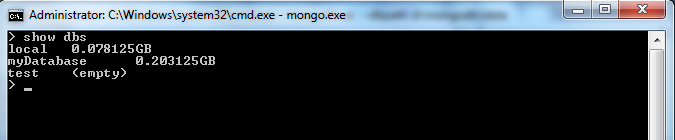
* Start [**mongo**](http://docs.mongodb.org/manual/reference/program/mongo/#bin.mongo) by issuing the [**mongo**](http://docs.mongodb.org/manual/reference/program/mongo/#bin.mongo) command, as follows:

mongo.exe



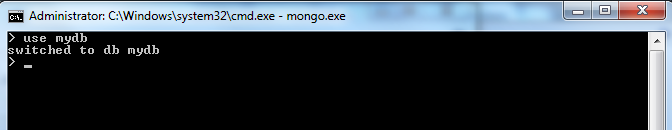
* By default, [**mongo**](http://docs.mongodb.org/manual/reference/program/mongo/#bin.mongo) looks for a database server listening on port **27017** on the **localhost** interface. To connect to a server on a different port or interface, use the --port and --host options.
* Display the list of databases, with the following operation:

show dbs



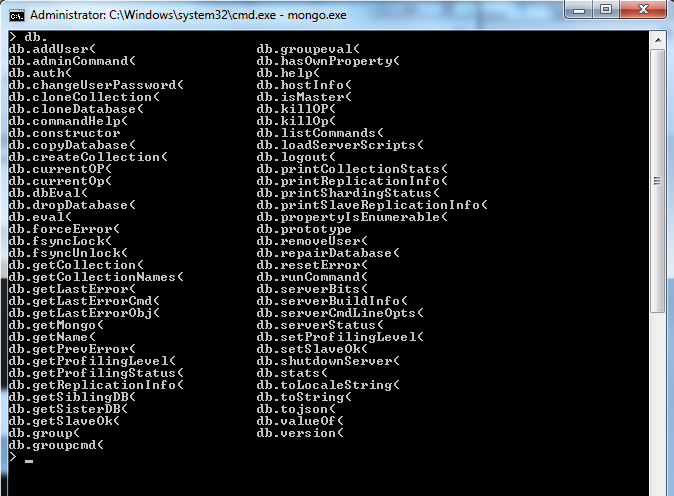
* Switch to a new database named **mydb**, with the following operation:

use mydb



* Issue the following operation at the [**mongo**](http://docs.mongodb.org/manual/reference/program/mongo/#bin.mongo) to report the name of the current database:

db



## Create a Collection and Insert Documents

* Switch to a new database named **mydb**, with the following operation (if the database has not been created, mongoDB will create database):

use mydb

* Create two documents named **j** and **k** by using the following sequence of JavaScript operations:

j = { name : "mongo" }

k = { x : 3 }

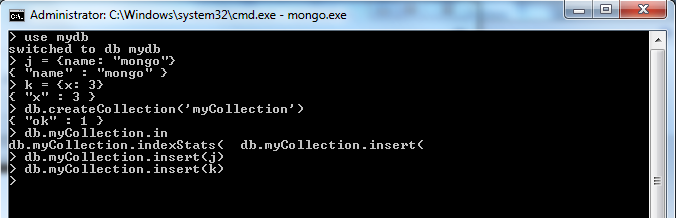
* Create a collection:

db.createCollection(“myCollection”)

* Insert the **j** and **k** documents into the **myCollection** collection with the following sequence of operations (if the collection has not been created, mongoDB will create collection):

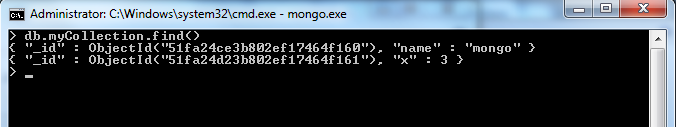
db.myCollection.insert(j)

db.myCollection.insert(k)



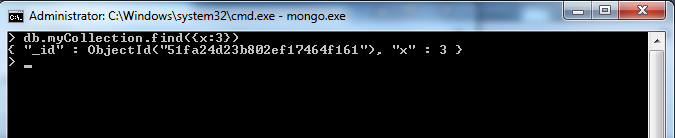
* Confirm that the documents exist in the myCollection collection by issuing a query on the collection using the [**find()**](http://docs.mongodb.org/manual/reference/method/db.collection.find/#db.collection.find) method:

db.myCollection.find()



* To query for all documents where the x field has a value of 3 by passing the { x : 3 } query **document as a parameter** to the [**find()**](http://docs.mongodb.org/manual/reference/method/db.collection.find/#db.collection.find) method:

db.myCollection.find({ x : 3 })

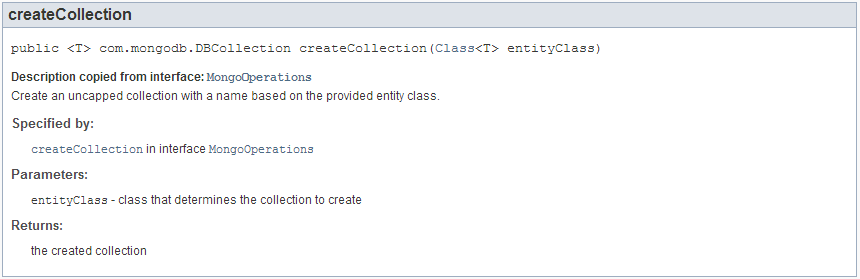


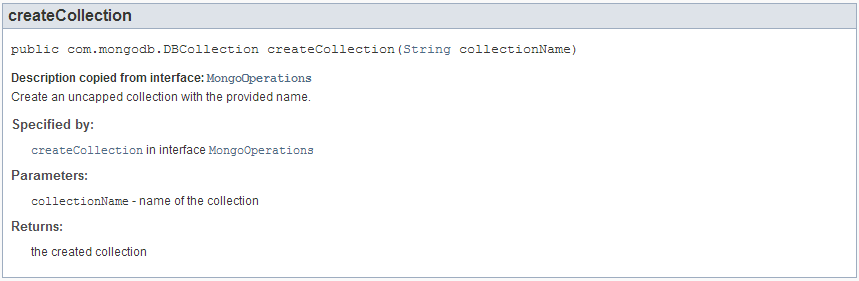
## MongoDB Spring Integration

### MongoTempla class

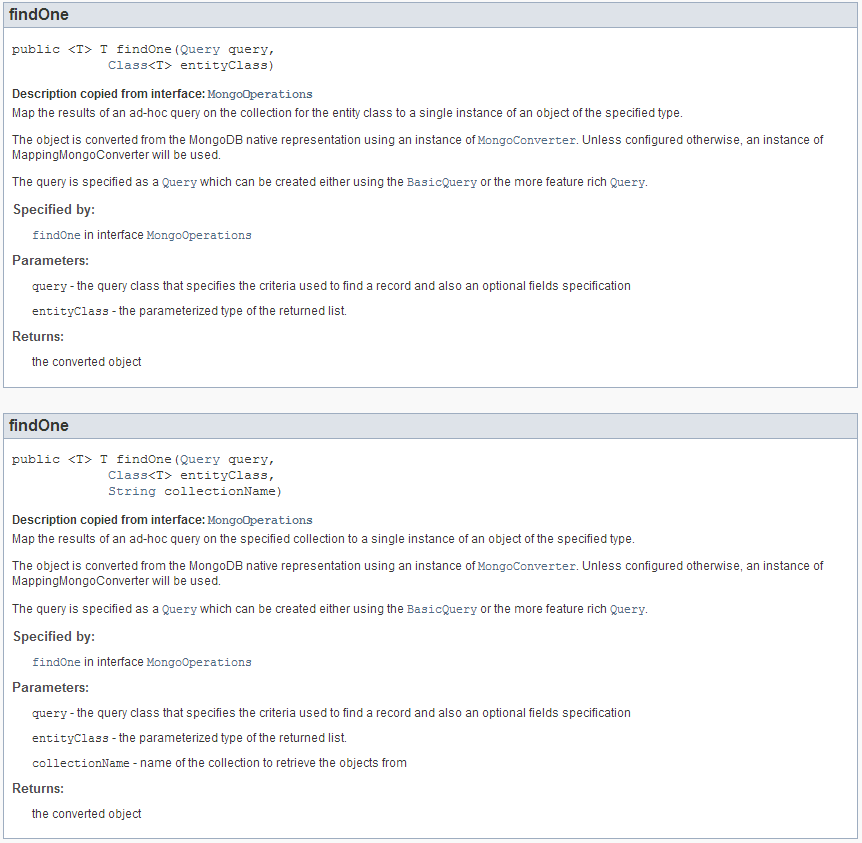
* + Spring supports class ***org.springframework.data.mongodb.core.MongoTemplate*** to working with MongoDB.
  + Main APIs of class ***org.springframework.data.mongodb.core.MongoTemplate***:

#### ***createCollection***

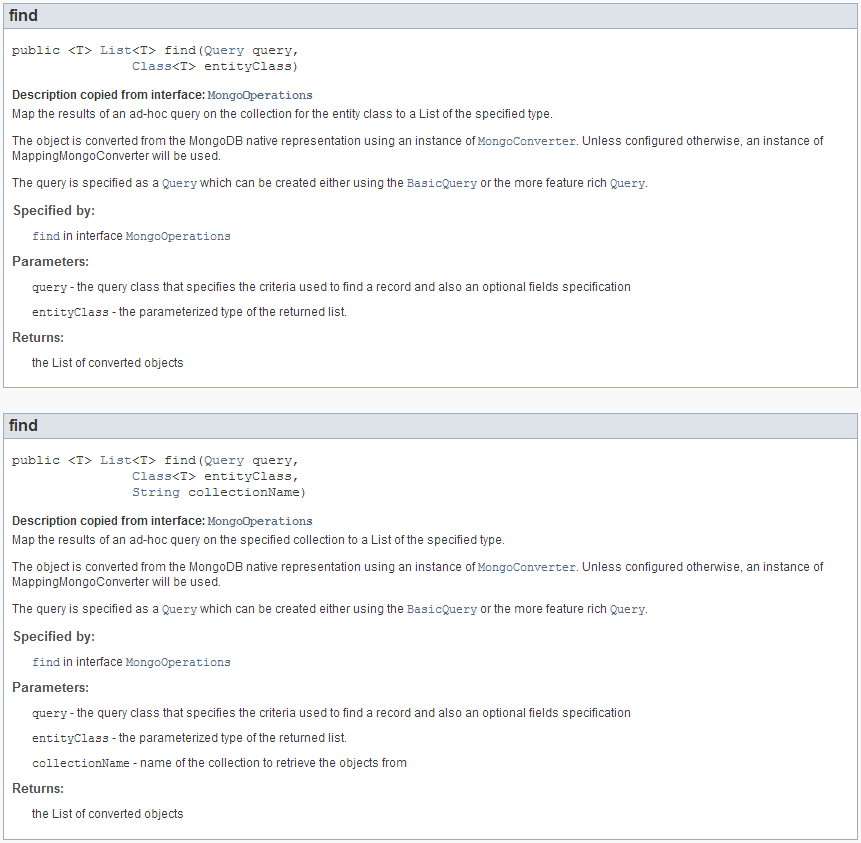




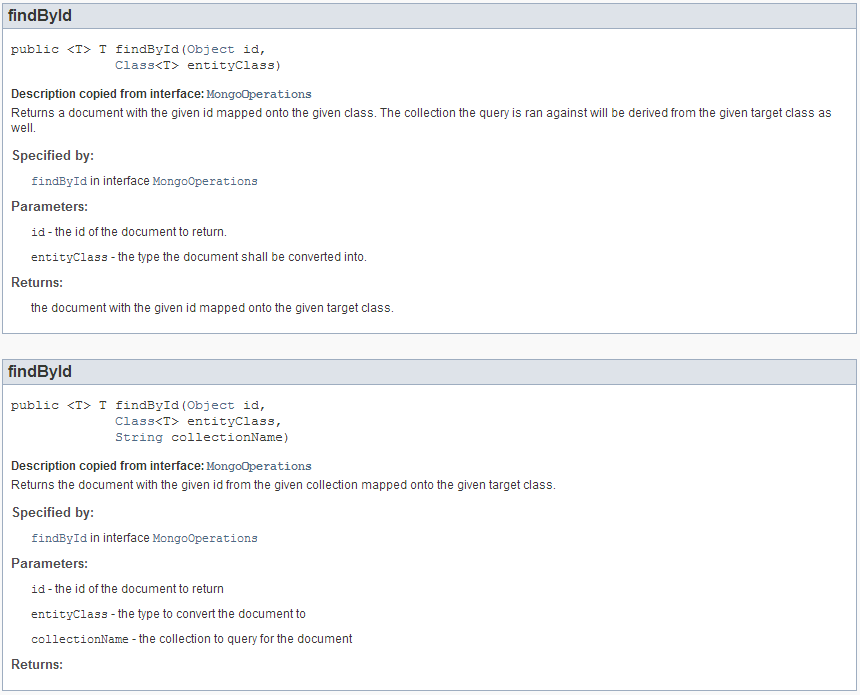
#### findOne



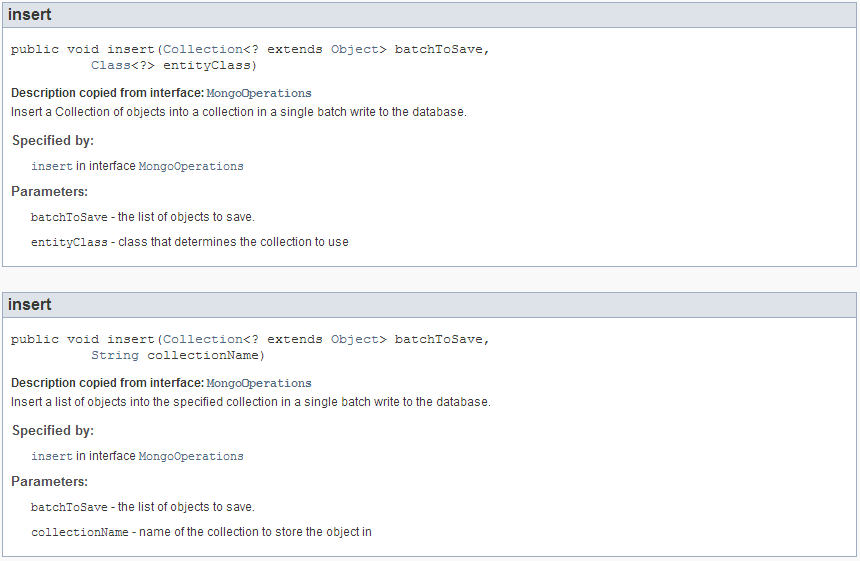
#### find



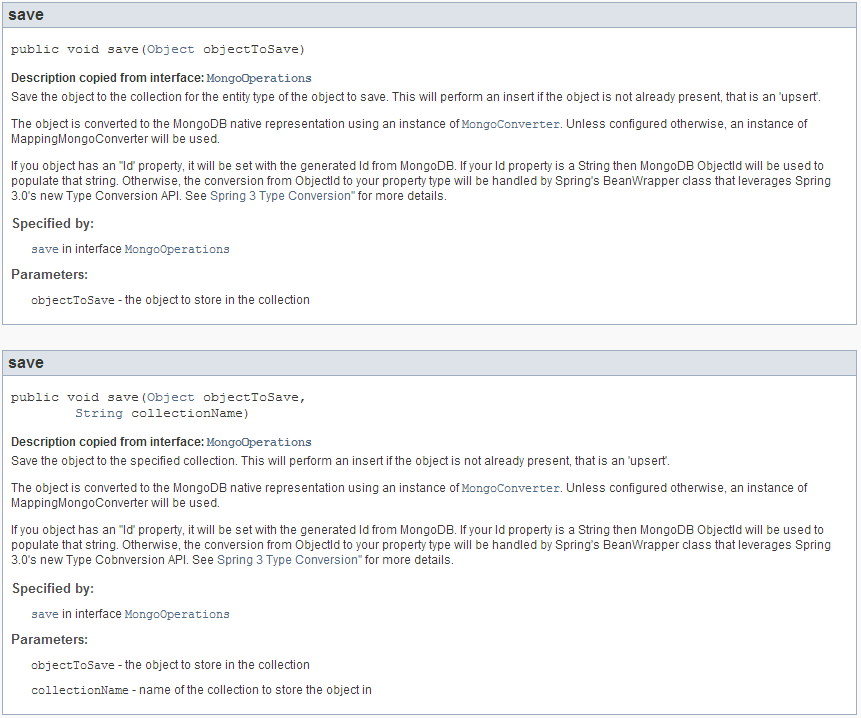
#### findById



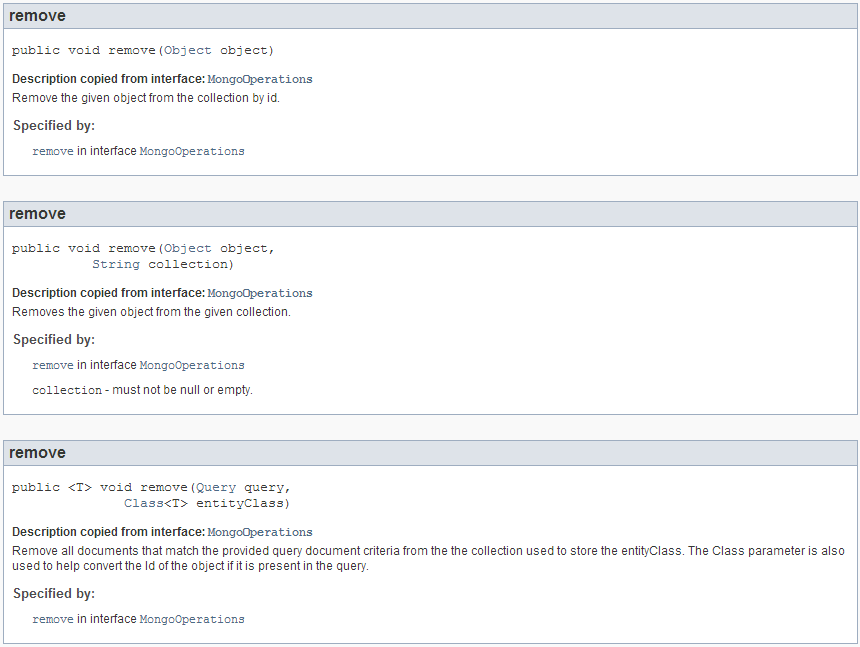
#### insert



#### save

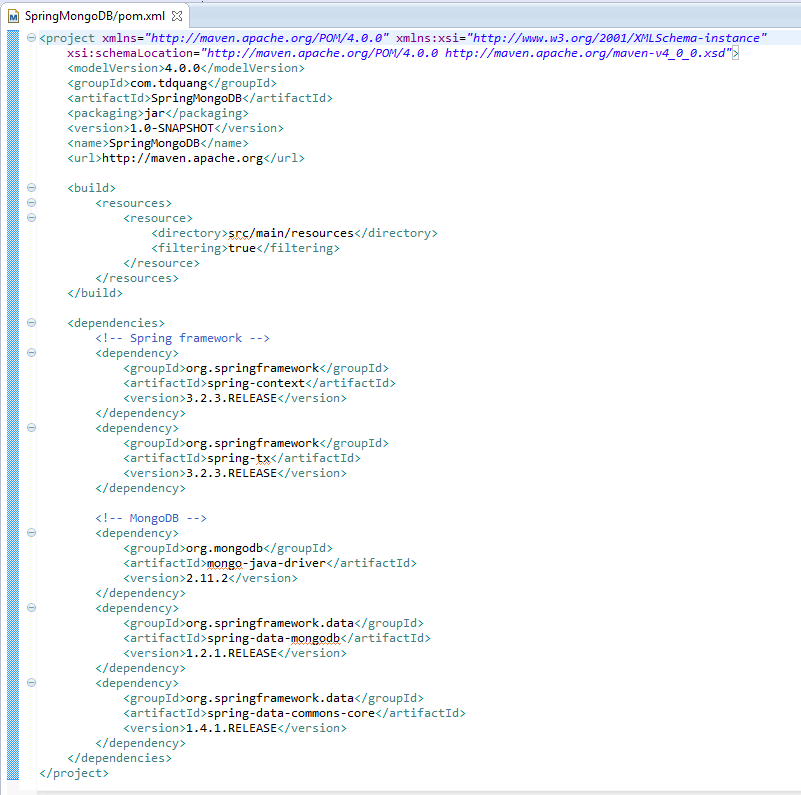


#### remove



### Example 1

#### Libraries



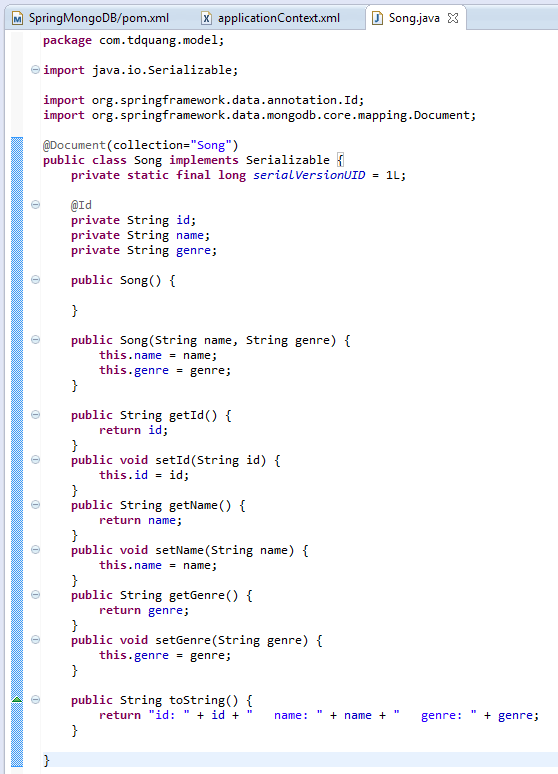
#### Config

* + Use tag <mongo:mongo host=*"127.0.0.1"* port=*"27017"* /> to connect to server mongoDB:
    - host: host of MongoDB server
    - port: port of MongoDB server
  + Use tag <mongo:db-factory dbname=*"songDB"* /> to connect to database:
    - dbname: name of database
    - username: username to authenticating when connect to database
    - password: password to authenticating when connect to database
  + Spring supports *org.springframework.data.mongodb.core.MongoTemplate* to working with mongoDB



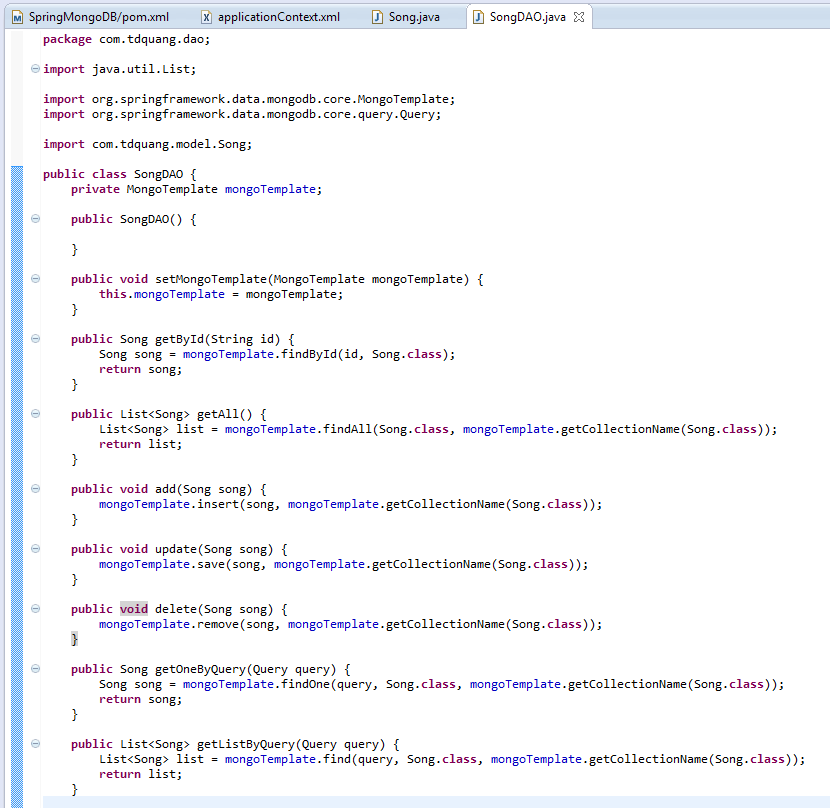
#### Pojo class

* + To mapping **java class** with **collection** of database we use **@Document** annotation
  + To using Id **auto generated** we use **@Id** annotation

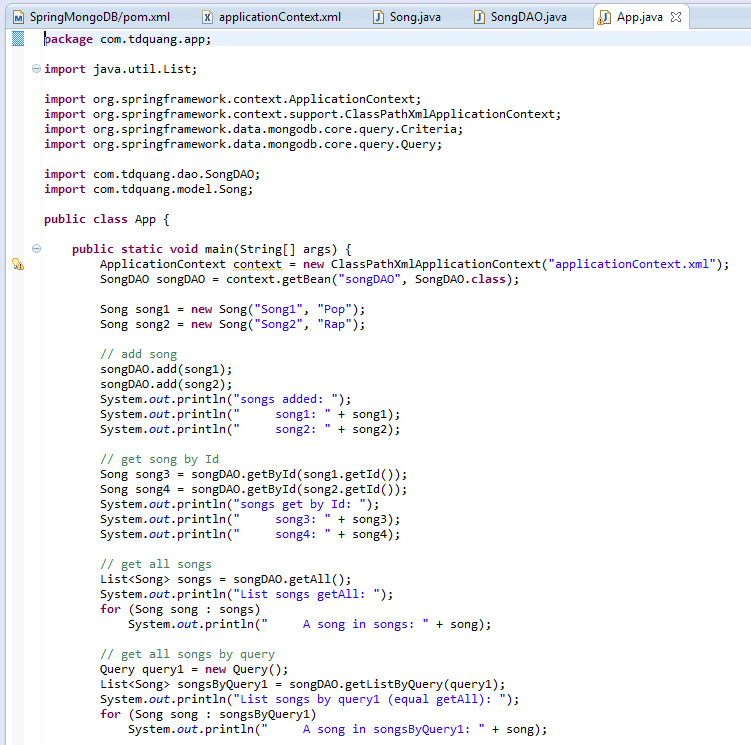


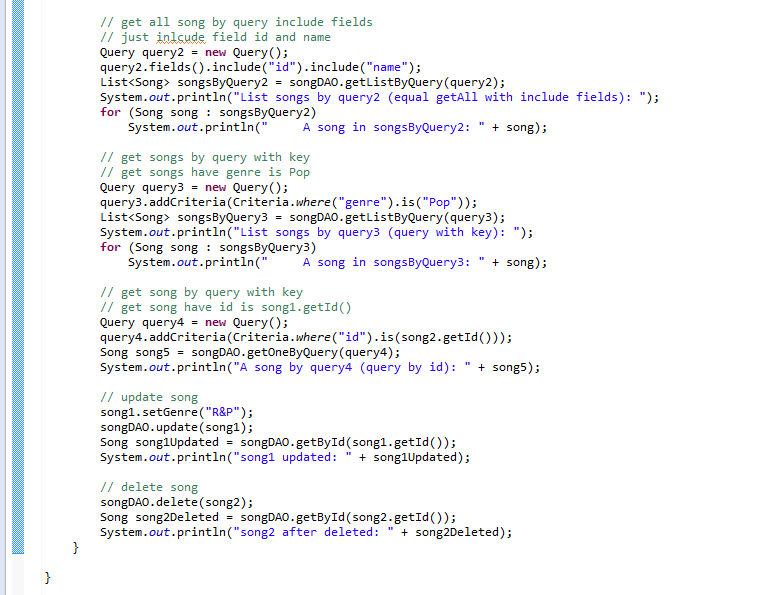
#### DAO class

* + Class *MongoTemplate* supports many APIs to working with mongoDB
  + To add, update and delete we use APIs are insert, save and delete of *MongoTemplate*
  + To query database, we use APIs are find or findOne of *MongoTemplate*

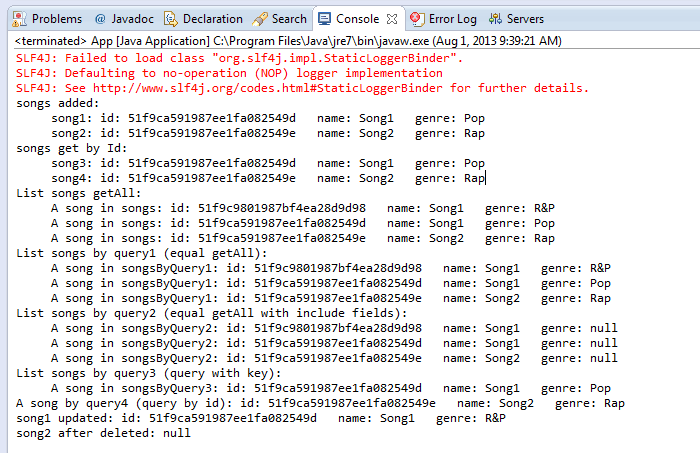


#### App class





#### Result



# References

* <http://docs.mongodb.org/manual/>
* <http://docs.mongodb.org/manual/tutorial/>
* <http://static.springsource.org/spring-data/data-mongodb/docs/current/reference/html/>
* <http://www.mkyong.com/tutorials/java-mongodb-tutorials/>