MongoDB

Introduction

MongoDB (from "humongous") is an open source document-oriented database system developed and supported by 10gen. It is part of the NoSQL family of database systems. Instead of storing data in tables as is done in a "classical" relational database, MongoDB stores structured data as JSON-like documents with dynamic schemas (MongoDB calls the format BSON), making the integration of data in certain types of applications easier and faster. MongoDB is the most popular NoSQL database management system.

Features

The following is a brief summary of some of the main features:

Ad hoc queries

MongoDB supports search by field, range queries, regular expression searches. Queries can return specific fields of documents and also include user-defined JavaScript functions.

Indexing

Any field in a MongoDB document can be indexed (indices in MongoDB are conceptually similar to those in RDBMSes). Secondary indices are also available.

Replication

MongoDB supports master-slave replication. A master can perform reads and writes. A slave copies data from the master and can only be used for reads or backup (not writes). The slaves have the ability to select a new master if the current one goes down.

Load balancing

MongoDB scales horizontally using sharding.[9] The developer chooses a shard key, which determines how the data in a collection will be distributed. The data is split into ranges (based on the shard key) and distributed across multiple shards. (A shard is a master with one or more slaves.)

MongoDB can run over multiple servers, balancing the load and/or duplicating data to keep the system up and running in case of hardware failure. Automatic configuration is easy to deploy and new machines can be added to a running database.

File storage

MongoDB could be used as a file system, taking advantage of load balancing and data replication features over multiple machines for storing files.

This function, called GridFS,[10] is included with MongoDB drivers and available with no difficulty for development languages (see "Language Support" for a list of supported languages). MongoDB exposes functions for file manipulation and content to developers. GridFS is used, for example, in plugins for NGINX.[11] and lighttpd[12]

In a multi-machine MongoDB system, files can be distributed and copied multiple times between machines transparently, thus effectively creating a load balanced and fault tolerant system.

Aggregation

MapReduce can be used for batch processing of data and aggregation operations. The aggregation framework enables users to obtain the kind of results for which the SQL GROUP BY clause is used.

Server-side JavaScript execution

JavaScript can be used in queries, aggregation functions (such as MapReduce), are sent directly to the database to be executed.

Capped collections

MongoDB supports fixed-size collections called capped collections. This type of collection maintains insertion order and, once the specified size has been reached, behaves like a circular queue.

Prerequisite

1. MongoDB 1.8.1
2. MongoDB-Java-Driver 2.5.2
3. JDK 1.6
4. Maven 3.0.3
5. Eclipse 3.6

Steps to setup MongoDB

1. Download MongoDB from the below link/site

   http://www.mongodb.org/downloads

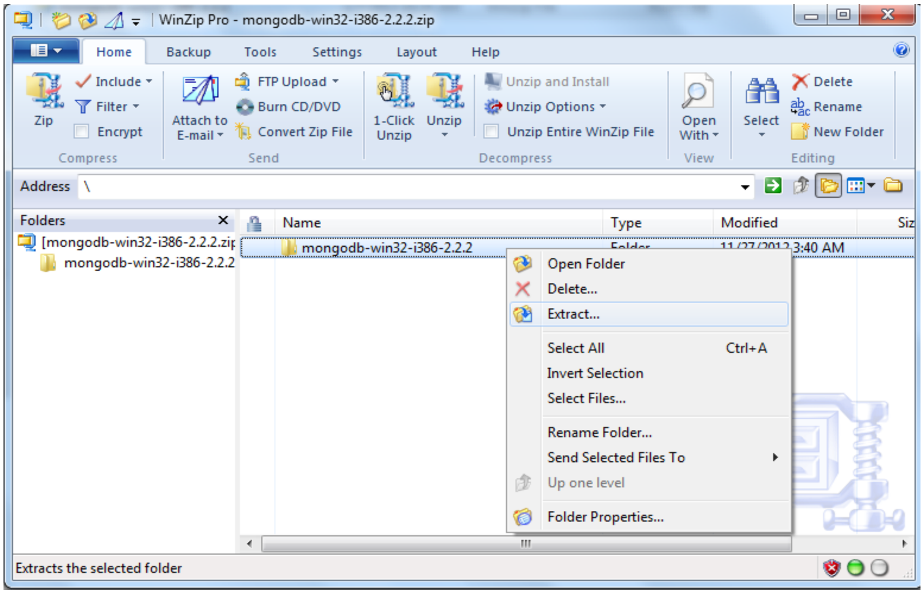
2. Note: There are three builds of MongoDB for Windows:
   - MongoDB for Windows Server 2008 R2 edition only runs on Windows Server 2008 R2, Windows 7 64-bit, and newer versions of Windows. This build takes advantage of recent enhancements to the Windows Platform and cannot operate on older versions of Windows.
   - MongoDB for Windows 64-bit runs on any 64-bit version of Windows newer than Windows XP, including Windows Server 2008 R2 and Windows 7 64-bit.

- ➢ MongoDB for Windows 32-bit runs on any 32-bit version of Windows newer than Windows XP. 32-bit versions of MongoDB are only intended for older systems and for use in testing and development systems.

3. Note: Changed in version 2.2: MongoDB does not support Windows XP. Please use a more recent version of Windows to use more recent releases of MongoDB.
   - ➢ 32-bit versions of MongoDB are suitable only for testing and evaluation purposes and only support databases smaller than 2GB.
   - ➢ You can find the architecture of your version of Windows platform using the following command in the Command Prompt

     C:\> wmic os get osarchitecture

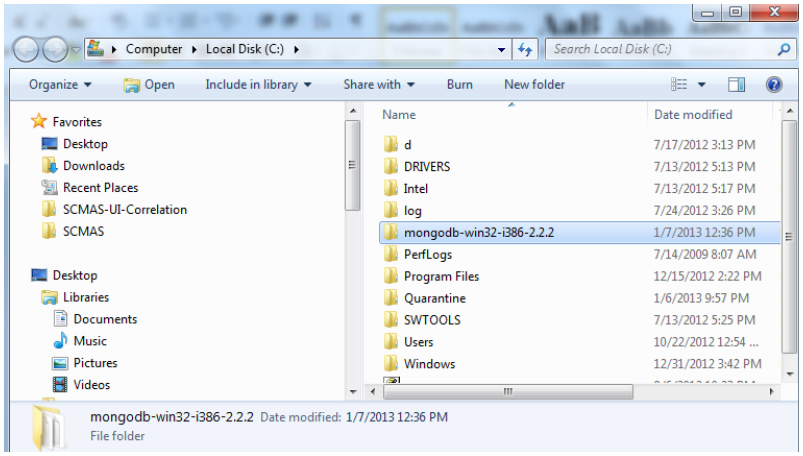
4. Extract the downloaded **mongodb-win32-i386-2.2.2.zip** to a specified drive c:\



Name	Type	Modified	Siz
mongodb-win32-i386-2.2.2	Folder	11/27/2012 3:40 AM	

Extracts the selected folder

5. Folder name will be as shown below.



Name	Date modified
d	7/17/2012 3:13 PM
DRIVERS	7/13/2012 5:13 PM
Intel	7/13/2012 5:17 PM
log	7/24/2012 3:26 PM
mongodb-win32-i386-2.2.2	1/7/2013 12:36 PM
PerfLogs	7/14/2009 8:07 AM
Program Files	12/15/2012 2:22 PM
Quarantine	1/6/2013 9:57 PM
SWTOOLS	7/13/2012 5:25 PM
Users	10/22/2012 12:54 ...
Windows	12/31/2012 3:42 PM

mongodb-win32-i386-2.2.2 Date modified: 1/7/2013 12:36 PM
File folder

Configuring/ Setting the Environment

1. Start the Command Prompt by selecting the Start Menu, then All Programs, then Accessories, then right click Command Prompt, and select Run as Administrator from the popup menu. In the Command Prompt.

2. Rename or move the extracted folder as given below

   ```
   C:\>ren mongodb-win32-i386-2.2.2 mongodb
   C:\>move mongodb-win32-i386-2.2.2 mongodb
   ```

   Note: MongoDB is self-contained and does not have any other system dependencies. You can run MongoDB from any folder you choose. You may install MongoDB in any directory (e.g. D:\karthik\mongodb)

3. MongoDB requires a data folder to store its files. The default location for the MongoDB data directory is C:\data\db. Create this folder using the Command Prompt. Issue the following command sequence:

   ```
   C:\>md data
   C:\>md data\db
   ```

   Note: You may specify an alternate path for \data\db with the dbpath setting for mongod.exe, as in the following example:

   ```
   C:\mongodb\bin\mongod.exe --dbpath d:\test\mongodb\data
   ```

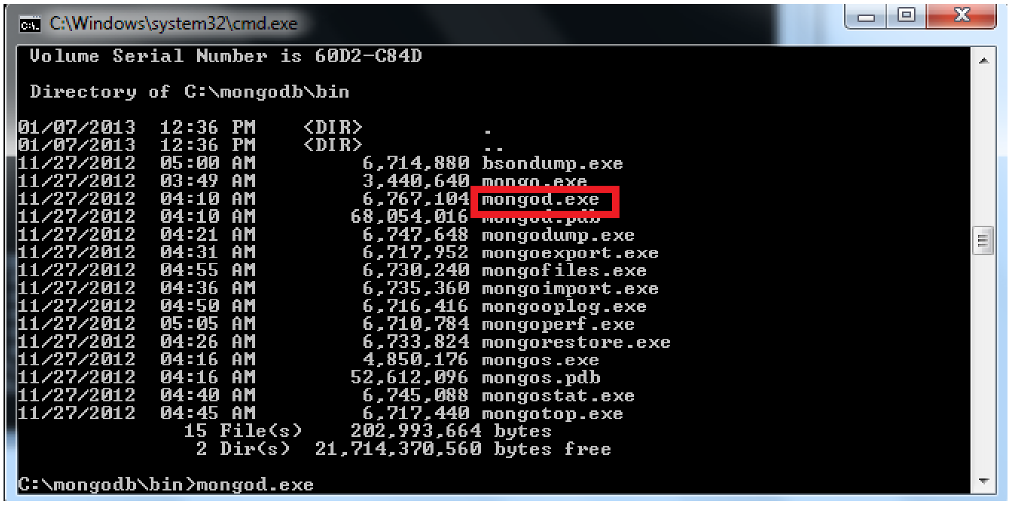
   Note: If your path includes spaces, enclose the entire path in double quotations, for example:

   ```
   C:\mongodb\bin\mongod.exe --dbpath "d:\test\mongo db data"
   ```

Start MongoDB

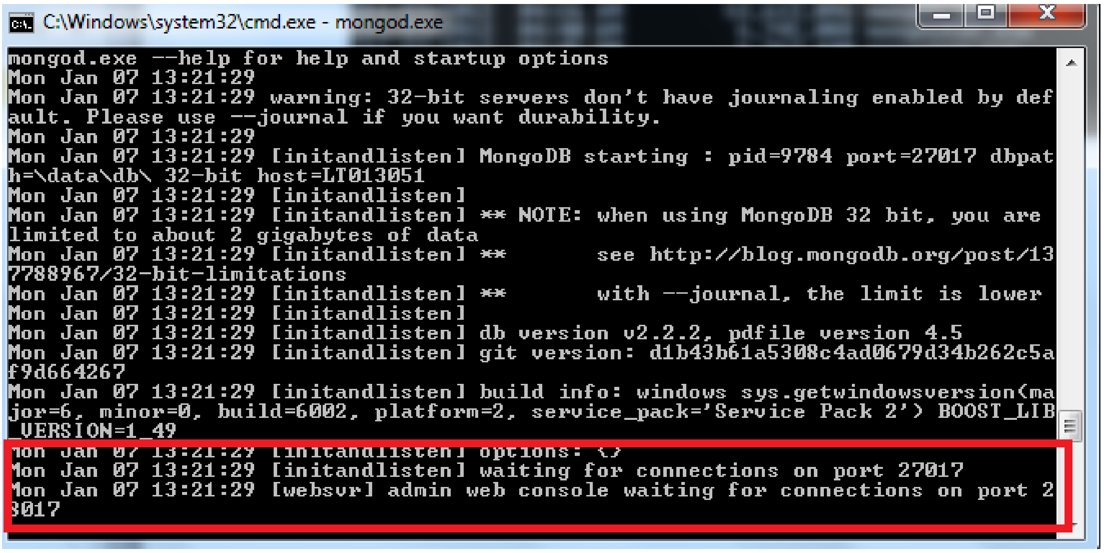
1. To start MongoDB, execute from the Command Prompt:

   ```
   C:\mongodb\bin\mongod.exe
   ```



```
 Volume Serial Number is 60D2-C84D

 Directory of C:\mongodb\bin

01/07/2013  12:36 PM    <DIR>          .
01/07/2013  12:36 PM    <DIR>          ..
11/27/2012  05:00 AM         6,714,880 bsondump.exe
11/27/2012  03:49 AM         3,440,640 mongo.exe
11/27/2012  04:10 AM         6,767,104 mongod.exe
11/27/2012  04:10 AM        68,054,016 mongod.pdb
11/27/2012  04:21 AM         6,747,648 mongodump.exe
11/27/2012  04:31 AM         6,717,952 mongoexport.exe
11/27/2012  04:55 AM         6,730,240 mongofiles.exe
11/27/2012  04:36 AM         6,735,360 mongoimport.exe
11/27/2012  04:50 AM         6,716,416 mongooplog.exe
11/27/2012  05:05 AM         6,710,784 mongoperf.exe
11/27/2012  04:26 AM         6,733,824 mongorestore.exe
11/27/2012  04:16 AM         4,850,176 mongos.exe
11/27/2012  04:16 AM        52,612,096 mongos.pdb
11/27/2012  04:40 AM         6,745,088 mongostat.exe
11/27/2012  04:45 AM         6,717,440 mongotop.exe
              15 File(s)    202,993,664 bytes
               2 Dir(s)  21,714,370,560 bytes free

C:\mongodb\bin>mongod.exe
```

This will start the main MongoDB database process. The waiting for connections message in the console output indicates that the mongod.exe process is running successfully.



```
mongod.exe --help for help and startup options
Mon Jan 07 13:21:29
Mon Jan 07 13:21:29 warning: 32-bit servers don't have journaling enabled by def
ault. Please use --journal if you want durability.
Mon Jan 07 13:21:29
Mon Jan 07 13:21:29 [initandlisten] MongoDB starting : pid=9784 port=27017 dbpat
h=\data\db\ 32-bit host=LT013051
Mon Jan 07 13:21:29 [initandlisten]
Mon Jan 07 13:21:29 [initandlisten] ** NOTE: when using MongoDB 32 bit, you are
limited to about 2 gigabytes of data
Mon Jan 07 13:21:29 [initandlisten] **       see http://blog.mongodb.org/post/13
7788967/32-bit-limitations
Mon Jan 07 13:21:29 [initandlisten] **       with --journal, the limit is lower
Mon Jan 07 13:21:29 [initandlisten]
Mon Jan 07 13:21:29 [initandlisten] db version v2.2.2, pdfile version 4.5
Mon Jan 07 13:21:29 [initandlisten] git version: d1b43b61a5308c4ad0679d34b262c5a
f9d664267
Mon Jan 07 13:21:29 [initandlisten] build info: windows sys.getwindowsversion(ma
jor=6, minor=0, build=6002, platform=2, service_pack='Service Pack 2') BOOST_LIB
_VERSION=1_49
Mon Jan 07 13:21:29 [initandlisten] options: {}
Mon Jan 07 13:21:29 [initandlisten] waiting for connections on port 27017
Mon Jan 07 13:21:29 [websvr] admin web console waiting for connections on port 2
8017
```

2. Depending on the security level of your system, Windows will issue a Security Alert dialog box about blocking "some features" of C:\\mongodb\bin\mongod.exe from communicating on networks. All users should select PrivateNetworks, such as my home or work network and click Allow access.

   For additional information on security and MongoDB, please read the Security and Authentication wiki page.

3. Warning : Do not allow mongod.exe to be accessible to public networks without running in "Secure Mode" (i.e. auth.) MongoDB is designed to be run in "trusted environments" and the database does not enable authentication or "Secure Mode" by default.

4. Connect to MongoDB using the mongo.exe shell. Open another Command Prompt and issue the following command:

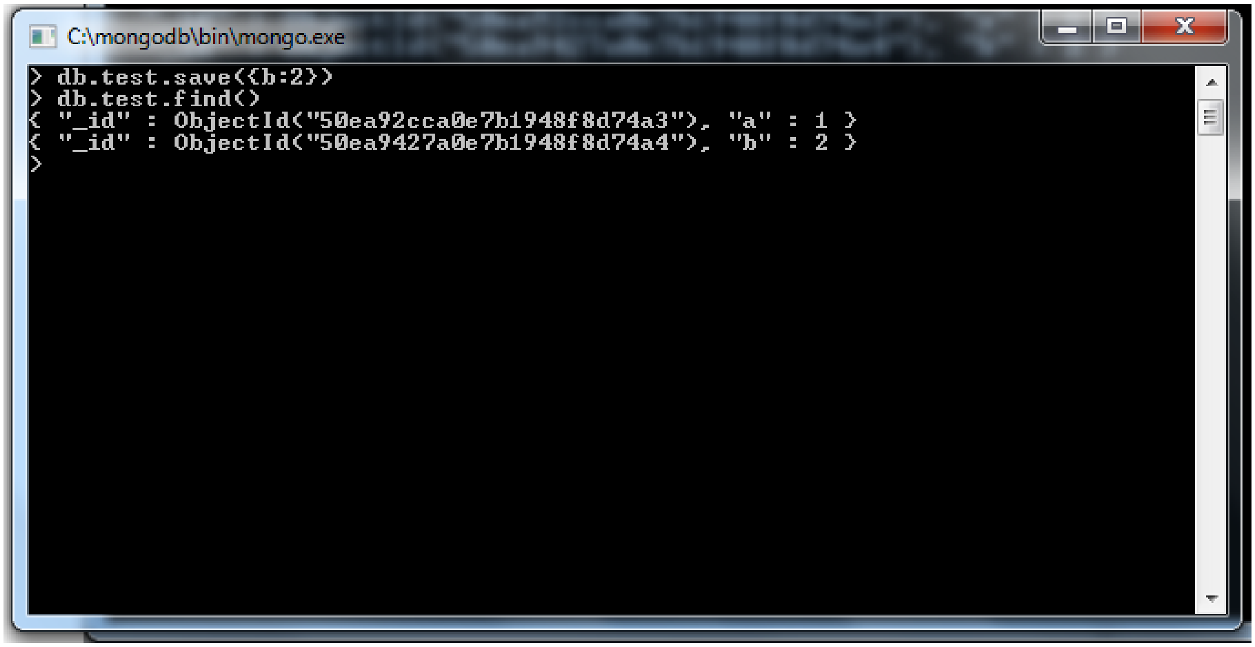
   C:\mongodb\bin\mongo.exe

5. Executing the command start C:\mongodb\bin\mongo.exe will automatically start the mongo.exe shell in a separate Command Prompt window.
6. The mongo.exe shell will connect to mongod.exe running on the localhost interface and port 27017 by default.

Executing

1. At the mongo.exe prompt, issue the following two commands to insert a record in the test collection of the default test database and then retrieve that record:

   > db.test.save({ a: 1 })
   > db.test.save({ b: 2 })
   > db.test.find()



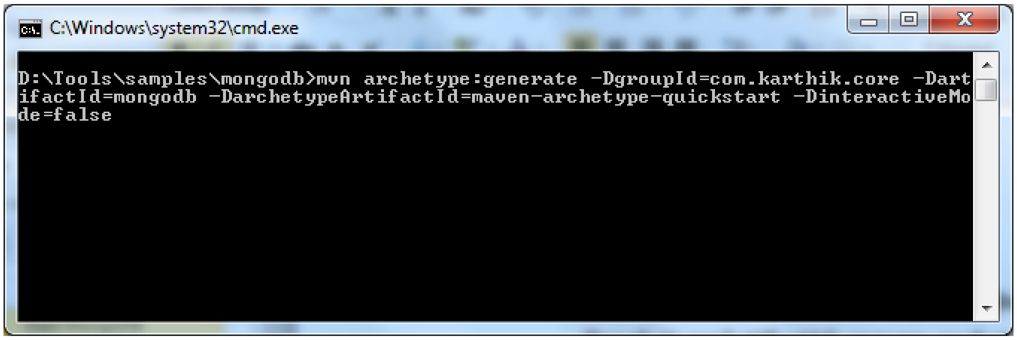
```
> db.test.save({b:2})
> db.test.find()
{ "_id" : ObjectId("50ea92cca0e7b1948f8d74a3"), "a" : 1 }
{ "_id" : ObjectId("50ea9427a0e7b1948f8d74a4"), "b" : 2 }
>
```

Connecting MongoDB via Java

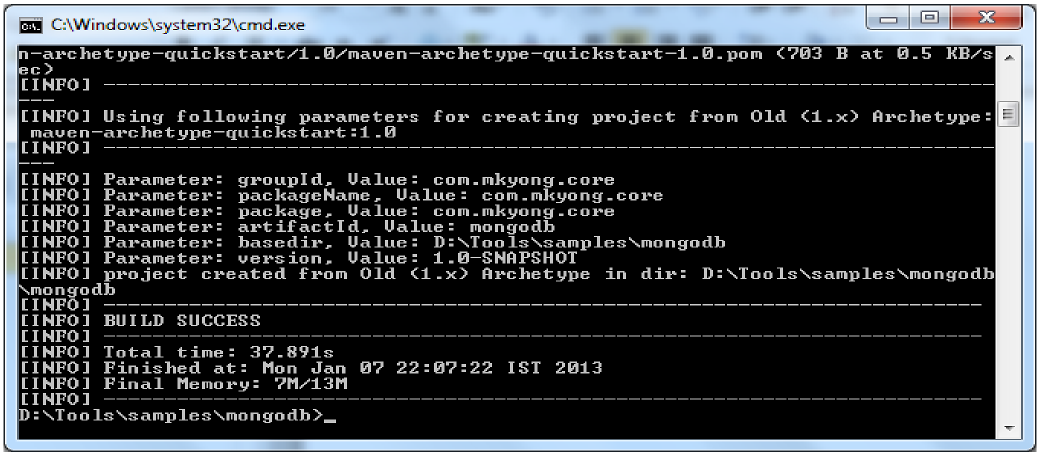
A simple Java hello world example to works with MongoDB – connect to, create database, collection and document in MongoDB, and also retrieve the saved value and display it to console.

1. Create a standard Java project using Maven.

```
mvn archetype:generate -DgroupId=com.karthik.core -DartifactId=mongodb -
DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```



```
D:\Tools\samples\mongodb>mvn archetype:generate -DgroupId=com.karthik.core -Dart
ifactId=mongodb -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMo
de=false
```



```
n-archetype-quickstart/1.0/maven-archetype-quickstart-1.0.pom (703 B at 0.5 KB/s
ec)
[INFO] -------------------------------------------------------------------------
---
[INFO] Using following parameters for creating project from Old (1.x) Archetype:
 maven-archetype-quickstart:1.0
[INFO] -------------------------------------------------------------------------
---
[INFO] Parameter: groupId, Value: com.mkyong.core
[INFO] Parameter: packageName, Value: com.mkyong.core
[INFO] Parameter: package, Value: com.mkyong.core
[INFO] Parameter: artifactId, Value: mongodb
[INFO] Parameter: basedir, Value: D:\Tools\samples\mongodb
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: D:\Tools\samples\mongodb
\mongodb
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 37.891s
[INFO] Finished at: Mon Jan 07 22:07:22 IST 2013
[INFO] Final Memory: 7M/13M
[INFO] ------------------------------------------------------------------------
D:\Tools\samples\mongodb>_
```

A new Java project named “mongodb”, and the entire project directory structure is created automatically.

2. Import the maven project into Eclipse.
3. The mongo-java driver is included in the Maven center repository. To get it, just declares the detail your pom.xml

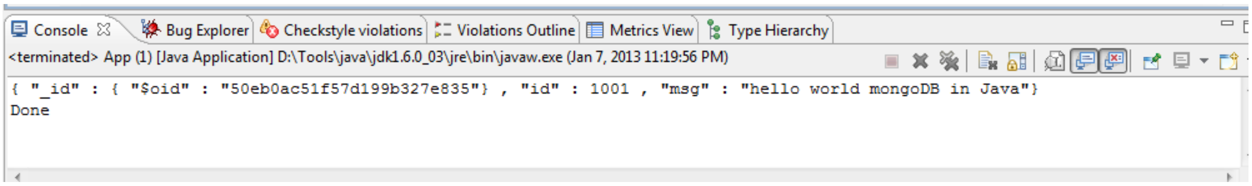
```
<project ...>
    <dependencies>
            <dependency>
                    <groupId>org.mongodb</groupId>
                    <artifactId>mongo-java-driver</artifactId>
                    <version>2.5.2</version>
            </dependency>
    </dependencies>
</project>
```

4. Create a sample java program to work with mongoDB

```
package com. karthik.core;

import java.net.UnknownHostException;

import com.mongodb.BasicDBObject;
import com.mongodb.DB;
import com.mongodb.DBCollection;
import com.mongodb.DBCursor;
import com.mongodb.Mongo;
import com.mongodb.MongoException;

/**
 * Java + MongoDB Hello world Example
 *
 */
public class App {
	public static void main(String[] args) {

		try {
			// connect to mongoDB, ip and port number
			Mongo mongo = new Mongo("localhost", 27017);

			// get database from MongoDB,
			// if database doesn't exists, mongoDB will create it
automatically
			DB db = mongo.getDB("yourdb");

			// Get collection from MongoDB, database named "yourDB"
			// if collection doesn't exists, mongoDB will create it
automatically
			DBCollection collection =
db.getCollection("yourCollection");

			// create a document to store key and value
			BasicDBObject document = new BasicDBObject();
			document.put("id", 1001);
			document.put("msg", "hello world mongoDB in Java");

			// save it into collection named "yourCollection"
			collection.insert(document);

			// search query
			BasicDBObject searchQuery = new BasicDBObject();
			searchQuery.put("id", 1001);
```

```
                // query it
                DBCursor cursor = collection.find(searchQuery);

                // loop over the cursor and display the retrieved result
                while (cursor.hasNext()) {
                    System.out.println(cursor.next());
                }

                System.out.println("Done");

            } catch (UnknownHostException e) {
                e.printStackTrace();
            } catch (MongoException e) {
                e.printStackTrace();
            }

        }
}
```

5. Execute the java. Ie, run as java application
6. Output will be as shown below



```
<terminated> App (1) [Java Application] D:\Tools\java\jdk1.6.0_03\jre\bin\javaw.exe (Jan 7, 2013 11:19:56 PM)
{ "_id" : { "$oid" : "50eb0ac51f57d199b327e835"} , "id" : 1001 , "msg" : "hello world mongoDB in Java"}
Done
```

SQL to Mongo Mapping Chart

http://docs.mongodb.org/manual/reference/sql-comparison/