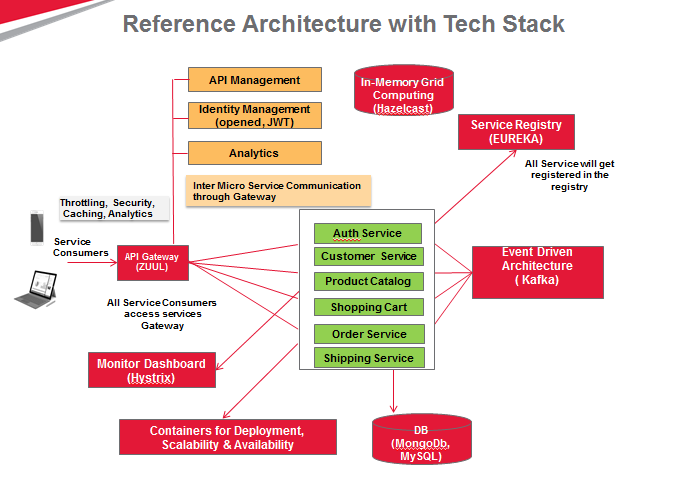
**Online ShoppingCart**



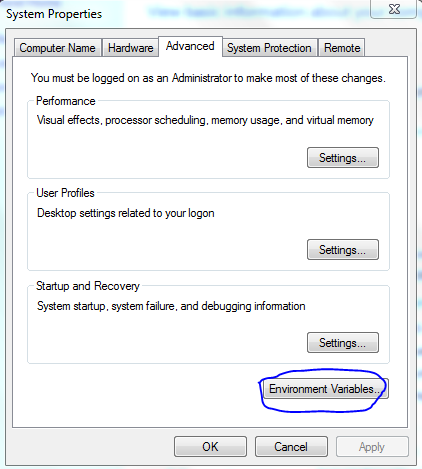
**Prerequisites**

* Java 7 or higher

**Set the environment**

Right-click on 'My Computer' and select 'Properties'.

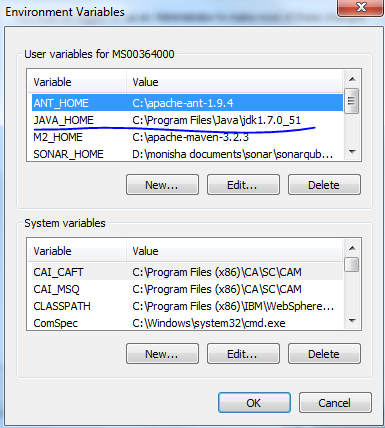
Click the 'Environment variables' button under the 'Advanced' tab.



**Setting Java Path**

Now, alter the 'Path' variable so that it also contains the path to the Java executable. Example, if the path is currently set to 'C:\WINDOWS\SYSTEM32', then change your path to read 'C:\WINDOWS\SYSTEM32;c:\Program Files\java\jdk\bin'.

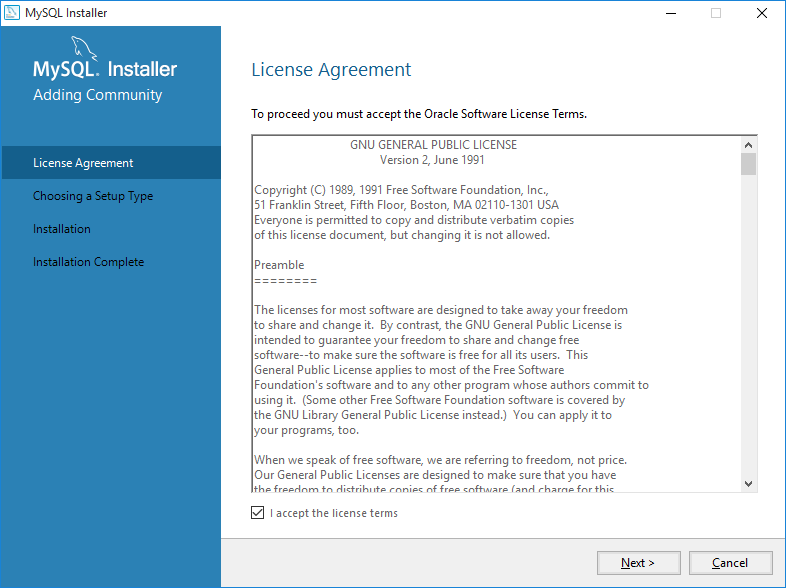
**Setting Java home**



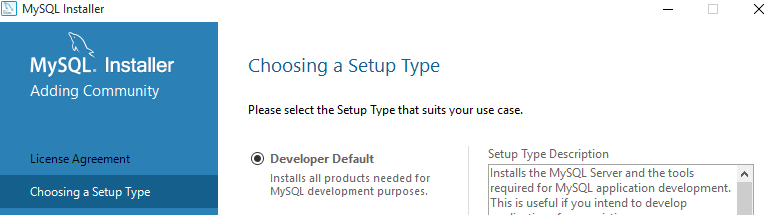
1. **Download and Install MySql**
2. **Download “mysql-installer-community-5.6.28.0.msi” from the below url**

<https://dev.mysql.com/downloads/file/?id=460602>

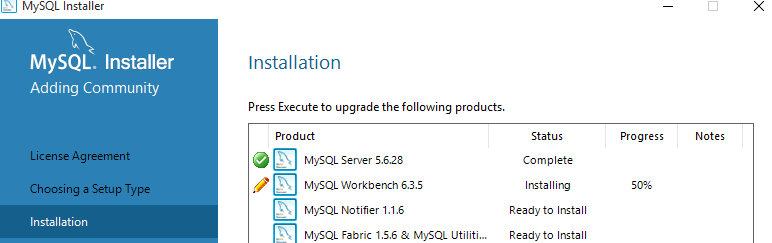
1. **Double click the “mysql-installer-community-5.6.28.0.msi”, which will start the installation process**



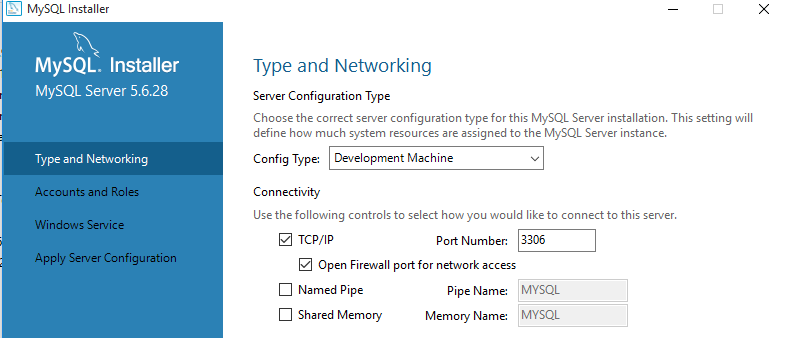
1. **Select “Default” option and click next.**



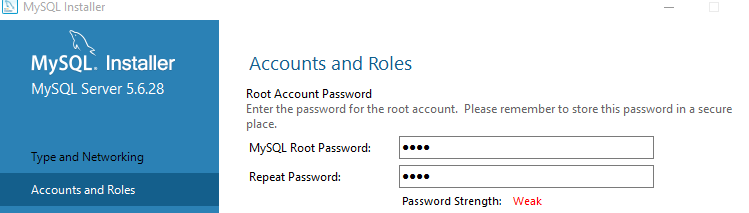
1. **Click “Next”**



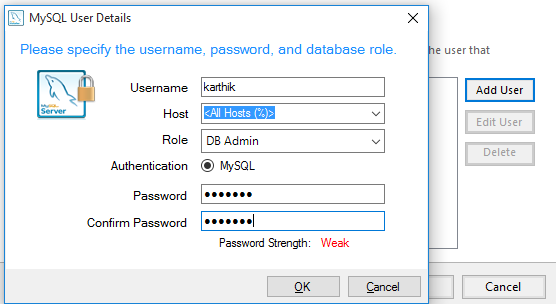
1. **Click “Next”**



1. **Enter password for “root” and also create a new user and password**



1. **If you want to create “user Accounts”, Click “Add user” and enter user id and password**



1. **Click “Execute”**
2. **Click “Finish”**

**3.Download and Install MongoDB**

**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
6. Download MongoDB from the below link/site

<http://www.mongodb.org/downloads>

1. Note: There are three builds of MongoDB for Windows:

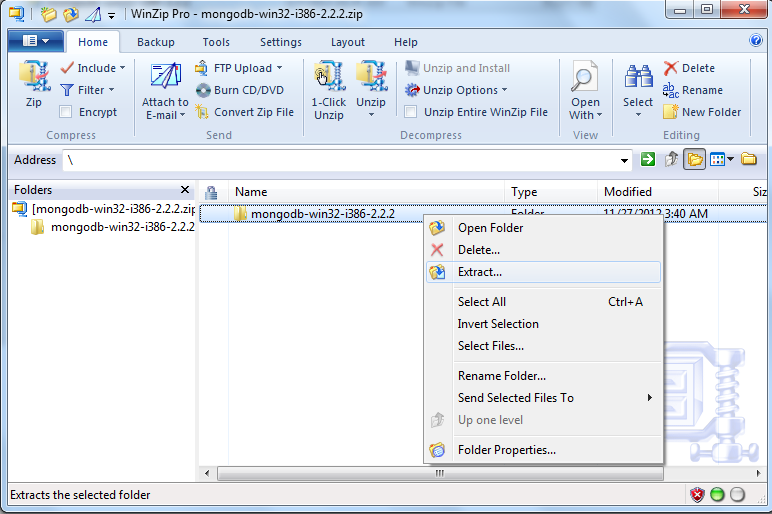
* MongoDB for Windows Server 2008 R2 edition only runs on Windows Server 2008 R2, Windows 7 64-bit, and newer versions of Windows. This build takes advantage of recent enhancements to the Windows Platform and cannot operate on older versions of Windows.
* MongoDB for Windows 64-bit runs on any 64-bit 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.

1. 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.

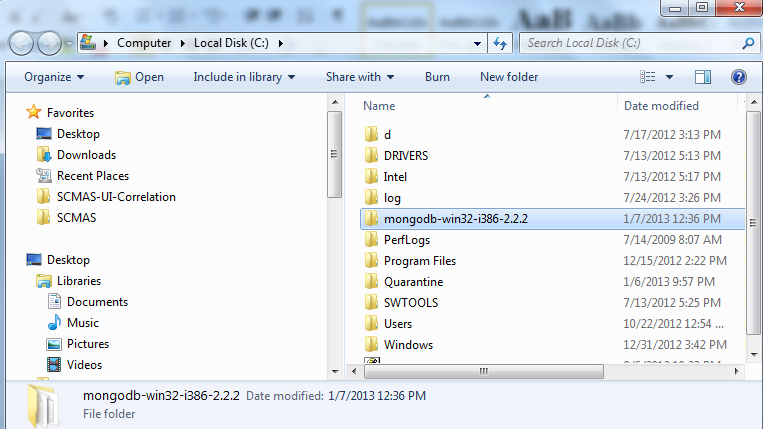
* 32-bit versions of MongoDB are suitable only for testing and evaluation purposes and only support databases smaller than 2GB.
* You can find the architecture of your version of Windows platform using the following command in the Command Prompt

**C:\> wmic os get osarchitecture**

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



1. Folder name will be as shown below.



**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)

1. 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. 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***

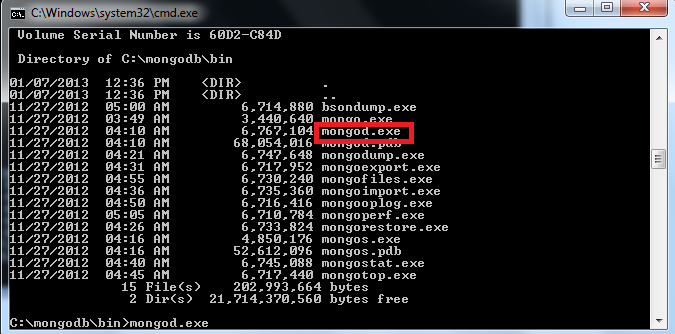
Note: If your path includes spaces, enclose the entire path in double quotations, for example:

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

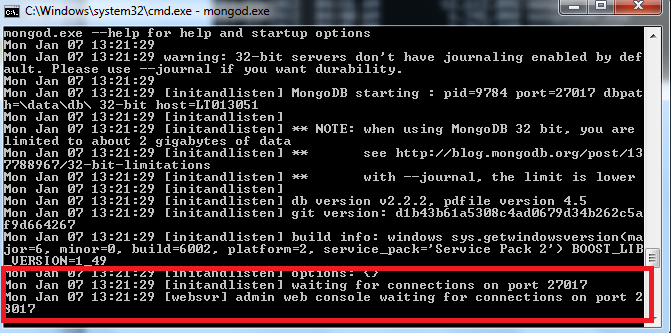
**Start MongoDB**

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

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



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.



1. 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](http://www.mongodb.org/display/DOCS/Security+and+Authentication) wiki page.

1. 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.
2. Connect to MongoDB using the mongo.exe shell. Open another Command Prompt and issue the following command:

**C:\mongodb\bin\mongo.exe**

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

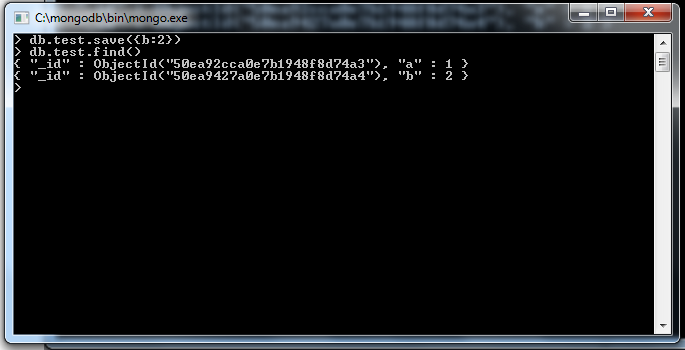
**Executing**

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()**

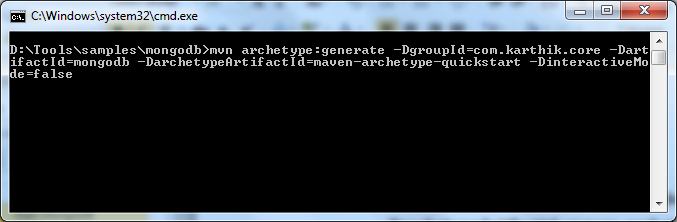
****

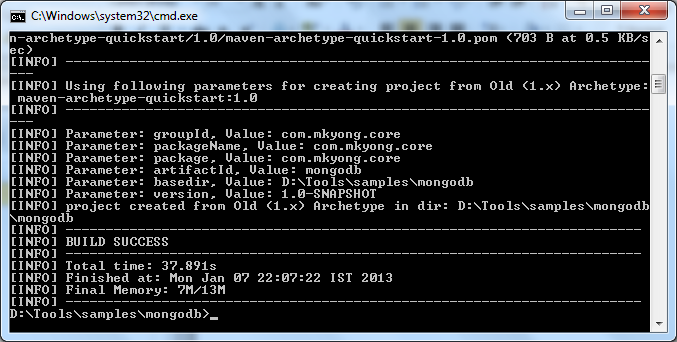
**Connecting MongoDB via Java**

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***





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

1. Import the maven project into Eclipse.
2. 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>**

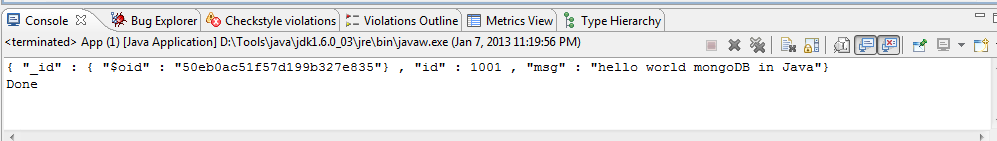
**</dependencies>**

**</project>**

1. 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();  }    }  } |

1. Execute the java. Ie, run as java application
2. Output will be as shown below



**SQL to Mongo Mapping Chart**

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

**4.Download and Install Kafka**

**Prerequisite**

* JDK 1.7+
* Kafka 2.10-0.10.1.1
* Zookeeper 3.4.9

**Installation**

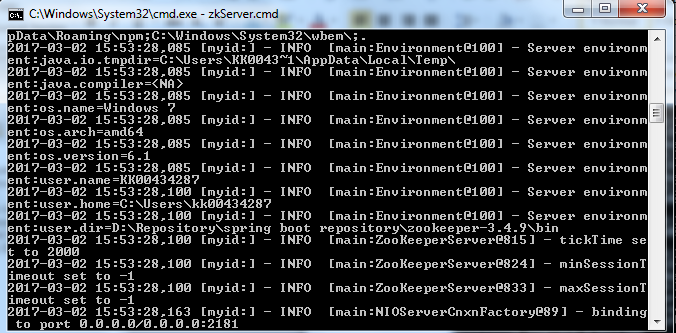
1. **Zookeeper Installation:**

In order to install kafka we need jdk and zookeeper in your system.

1. Download zookeeper from the link.

<http://zookeeper.apache.org/releases.html#download>

1. Goto your Zookeeper config directory. For me its C:\zookeeper-3.4.9\conf
2. Rename file “zoo\_sample.cfg” to “zoo.cfg”
3. Open a  command prompt window as admin and change the directory to unzipped folder
4. Run command “.\bin\zkServer”
5. This starts Zookeeper service on port 2181.



1. **Kafka Installation:**

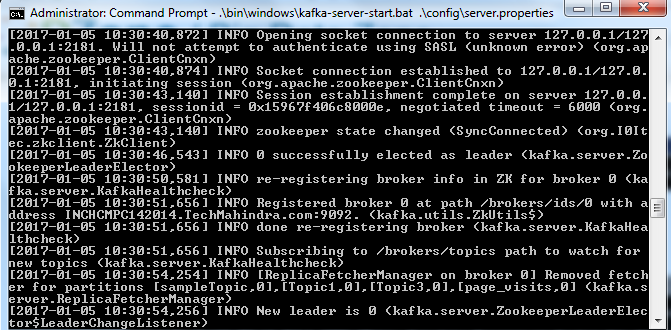
Important: Please ensure that your Zookeeper instance is up and running before starting a Kafka server.

1. Download kafka from the below link.
   * 1. [**https://kafka.apache.org/downloads.html**](https://kafka.apache.org/downloads.html)
2. Go to your Kafka directory. For me its *C:\kafka\_2.10-0.10.1.1\*
3. Open a new command prompt window as administrator and navigate to folder ”cd *C:\kafka\_2.10-0.10.1.1”*.
4. Run command

“ **.\bin\windows\kafka-server-start.bat .\config\server.properties** ”.

* + 1. 

1. This starts Kafka server.



**3. Kafka Topic Creation:**

1. Open a new command prompt as admin and go to kafka folder path *C:\kafka\_2.10-0.10.1.1.* Then run the below command and this will create the topic .

**.\bin\windows\kafka-topics --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic <topic\_name>**

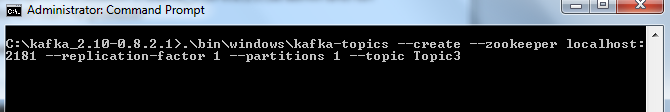
**Example:**

**.\bin\windows\kafka-topics --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic fundTransferRequestTopic**

**And**

**C:\kafka\_2.10-0.10.1.1>.\bin\windows\kafka-topics --create --zookeeper localhost**

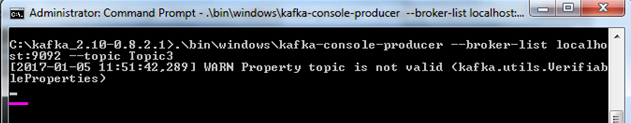
**:2181 --replication-factor 1 --partitions 1 --topic fundTransferResponseTopic**



**4. Creating the Producer and Consumer:**

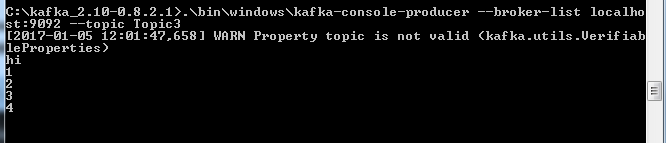
1. Open a new command prompt as administrator in the location C:\kafka\_2.10-0.8.2.1\ as admin and run the below command. This will start he producer

**.\bin\windows\kafka-console-producer --broker-list localhost:9092 --topic <topic\_name>**



Note : this below step is after creating the consumer.

You can see that the cusor is waiting for the input.if you type the data.



1. Again open a new command prompt as administrator in the same location as C:\kafka\_2.10-0.8.2.1\ as admin and run the below command.

**.\bin\windows\kafka-console-consumer --zookeeper localhost:2181 --topic <topic\_name> --from-beginning**

Note: you will see this when you enter the data at producer command prompt or it will be empty

Now if you press enter you will see the below window.

**4.Download and Install HazelCast**

**Prerequisite**

* JDK 1.7+
* Tomcat 7
* Hazelcast 3.6 or above

**1.Member Installation Steps**

1. Download Hazelcast 3.6 zip from <http://download.hazelcast.com/download.jsp?version=hazelcast-3.6&p=171028013171028013>
2. Extract the Hazelcast-3.6.zip file into working directory.
3. Following attributes value needs to be modified in Hazelcast.xml which is available under bin directory.
   * 1. Group 🡺 Cluster group name & password that would be used to enable multi node with same group name
     2. management-center 🡺 It should be enabled with mancenter web application link that provides consolidated statistic report includes all clustered members.
     3. network – multicast 🡺 It should be enabled to access all clustered members which are available in same network.

**Hazelcast.xml**

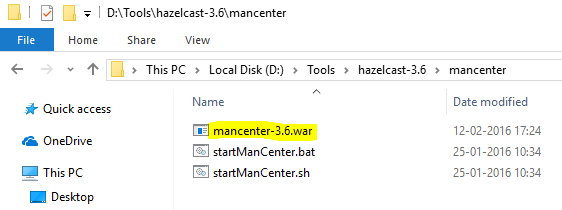
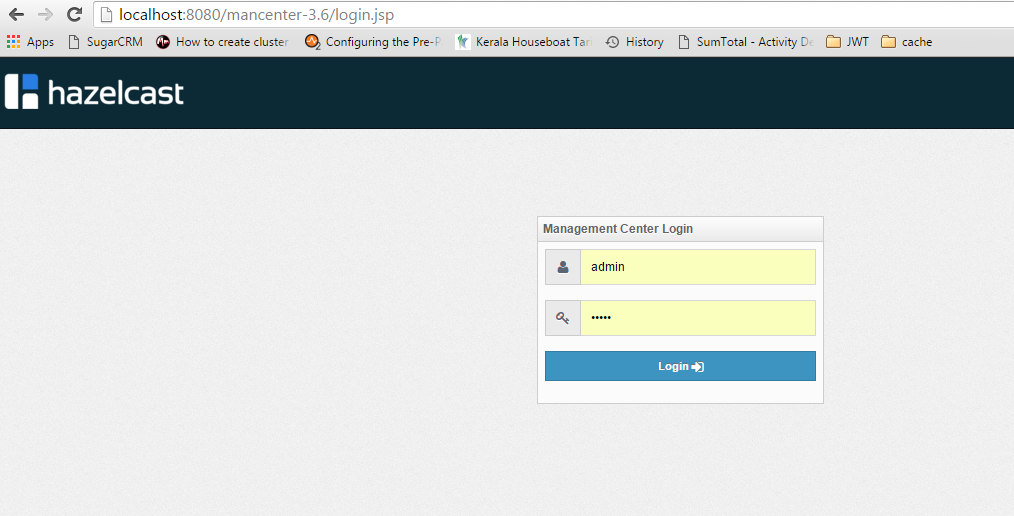
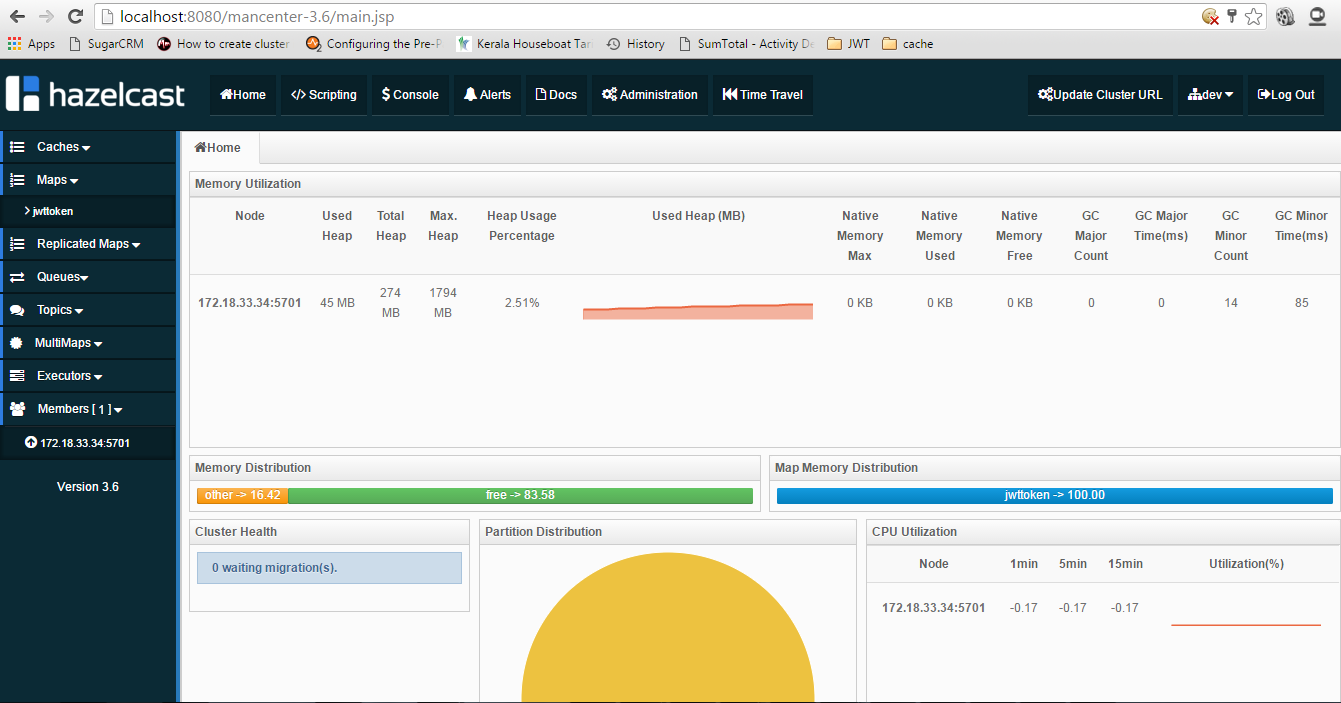
|  |
| --- |
| **<**hazelcast xsi:schemaLocation="http://www.hazelcast.com/schema/config hazelcast-config-3.6.xsd"  xmlns="http://www.hazelcast.com/schema/config"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  <group>  <name>dev</name>  <password>dev-pass</password>  </group>  <management-center enabled="true">http://localhost:8080/mancenter-3.6</management-center>  <network>  <port auto-increment="true" port-count="100">5701</port>  <outbound-ports>  <!--  Allowed port range when connecting to other nodes.  0 or \* means use system provided port.  -->  <ports>0</ports>  </outbound-ports>  <join>  <multicast enabled="true">  <multicast-group>224.2.2.3</multicast-group>  <multicast-port>54327</multicast-port>  </multicast>  <tcp-ip enabled="false">  <interface>127.0.0.1</interface>  <member-list>  <member>127.0.0.1</member>  </member-list>  ----  ----  ----- |

**2.Mancenter**

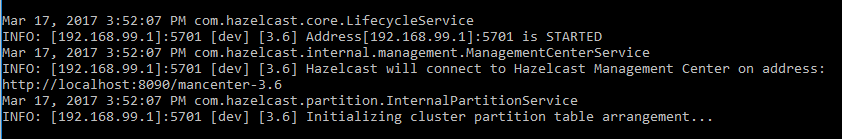
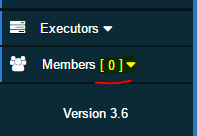
Mancenter provides dashboard that includes clustered members, caching, maps, queues, topics, partition distribution details and performance statistic report

**3.Deploy Mancenter**

**Start the Tomcat server**

1. Hazelcast provides mancenter.war file which is available under mancenter directory
   1. 
2. Deploy war file in any one application/web server
   1. In web servers like tomcat
   2. Copy mancenter.war file from mancenter directory and paste it into webapp folder of tomcat directory
3. Start the tomcat server
4. Login to mancenter Link : http://<<domain name >>/mancenter
   1. For example: <http://localhost:8080/mancenter-3.6/>
   2. 
5. Default user name & password to login: admin / admin
   1. 

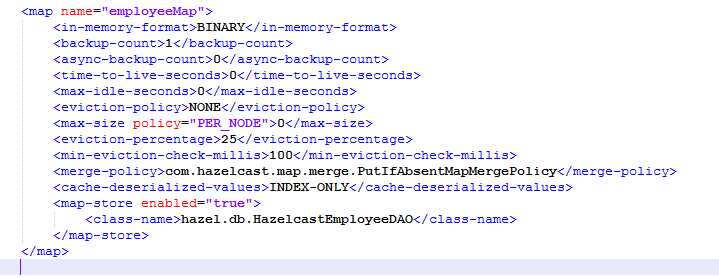
**5.Start the Hazelcast server (Mancenter)**

1. Copy the tomcat link mentioned above
   * 1. <http://localhost:8080/mancenter-3.6/>
2. Paste the link inside the hazelcast.xml available inside “\hazelcast-3.6\bin” folder.
3. Run the start.bat, which is present under hazelcast-3.6/bin directory. This will connect with the member.
   1. 
4. Before starting the haselcast, Memember will be zero as mentioned in the below screen.
   1. 
5. Once the server has been started, member value will be “1” as mentioned in the below screen.
   1. 
6. Hazelcast Installed successfully!!!

**Hints:**

We can also make changes in hazelcast.xml

Such as Map store configuration



Can make changes in the following

1. <map name=””></map> -name of the map
2. <in-memory-format>BINARY</in-memory-format>
   * 1. Number of backups. If 1 is set as the backup-count for example then all entries of the map will be copied to another JVM for fail-safety. 0 means no backup.
3. <backup-count>1</backup-count>
   1. Number of async backups. 0 means no backup.
4. <async-backup-count>0</async-backup-count>
   1. Maximum number of seconds for each entry to stay in the map. Entries that are older than <time-to-live-seconds> and not updated for <time-to-live-seconds>
   2. will get automatically evicted from the map.Any integer between 0 and Integer.MAX\_VALUE. 0 means infinite. Default is 0
5. <time-to-live-seconds>0</time-to-live-seconds>
   1. Maximum number of seconds for each entry to stay idle in the map. Entries that are
   2. idle(not touched) for more than <max-idle-seconds> will getautomatically evicted from the map. Entry is touched if get, put or containsKey is called.
   3. Any integer between 0 and Integer.MAX\_VALUE. 0 means infinite. Default is 0.
6. <max-idle-seconds>10</max-idle-seconds>
   1. Valid values are:
      1. NONE (no eviction),
      2. LRU (Least Recently Used),
      3. LFU (Least Frequently Used).
      4. NONE is the default.