

# Running Claim Service

Monday, April 18, 2016 8:36 PM

You need the following installed to be able to run the REST web service in claim\_demo\_v001.zip folder in Git as well the DB.

1. Download MongoDB for your platform/OS from here <https://www.mongodb.org/downloads#production>

Install to a location of your choice.

While there is a GUI for interacting with the DB, it will only work after you start the DB from the command line.

To start the MongoDB server, use below command in command prompt:

(depending on where you have installed it - cd d:\MongoDB\bin(mongod.exe)

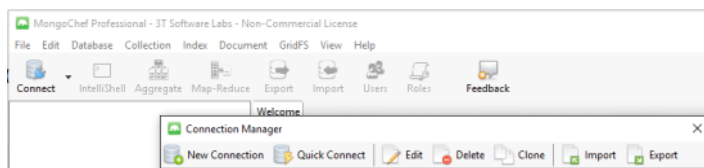
A successful start at default port 27017 will be as below. The DB also has a default database "test" where any new collections (aka tables in the SQL world will be created).

```
Command Prompt - mongod
D:\MongoDB\bin>mongod
2016-04-18T11:01:11.741-0800 I CONTROL [initandlisten] MongoDB starting : pid=3308 port=27017 dbp
2016-04-18T11:01:11.745-0800 I CONTROL [initandlisten] targetMMS: Windows Vista/Windows Server 2
2016-04-18T11:01:11.747-0800 I CONTROL [initandlisten] db version v3.2.4
2016-04-18T11:01:11.749-0800 I CONTROL [initandlisten] git version: 52ee9ffc9f95a4fad76802e18cc97
2016-04-18T11:01:11.752-0800 I CONTROL [initandlisten] allocator: tcmalloc
2016-04-18T11:01:11.754-0800 I CONTROL [initandlisten] modules: none
2016-04-18T11:01:11.757-0800 I CONTROL [initandlisten] build environment:
2016-04-18T11:01:11.758-0800 I CONTROL [initandlisten] distarch: x86_64
2016-04-18T11:01:11.765-0800 I CONTROL [initandlisten] target_arch: x86_64
2016-04-18T11:01:11.767-0800 I CONTROL [initandlisten] options: {}
2016-04-18T11:01:11.771-0800 I STORAGE [initandlisten] Detected data file in D:\data\db\ created
by the "wiredtiger" storage engine, so setting the active storage engine to "wiredtiger".
2016-04-18T11:01:11.773-0800 I STORAGE [initandlisten] wiredtiger.open config: create,cache_size=8
2016-04-18T11:01:11.775-0800 I STORAGE [initandlisten] config_base=false,wiredtiger(logs(enable=true,
archive=true,path=journal,compressor=snappy),file_manager(close_idle_time=10000),checkpoint(wait
=60,log_size=256),statistics_log(wait=60),
2016-04-18T11:01:11.338-0800 I NETWORK [mainNameCanonicalizationServer] Starting hostname canonic
2016-04-18T11:01:11.338-0800 I NETWORK [initandlisten] Initializing full-time diagnostic data capt
2016-04-18T11:01:11.340-0800 I NETWORK [initandlisten] waiting for connections on port 27017
```

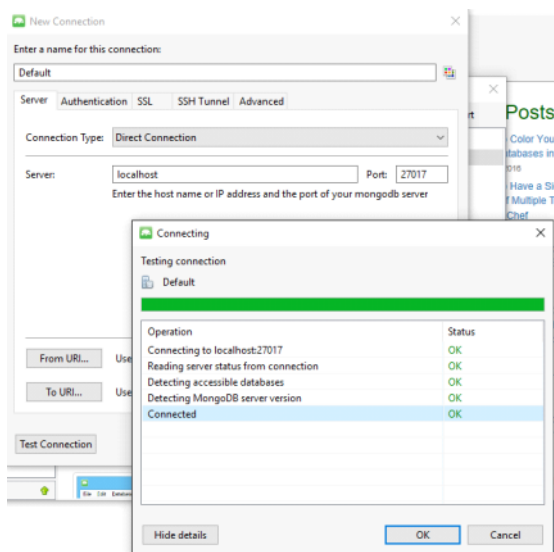
If you choose not to interact with the DB with the GUI, you can start a command window with the following command (depending on the install location).

```
D:\MongoDB\bin>mongo
MongoDB shell version: 3.2.4
connecting to: test
>
```

2. To shutdown mongoDB after you are done testing you need to shut down the DB clean using the below 2 commands.  
use admin  
db.shutdownServer()
3. Mongo Chef - Free for personal use GUI tool to use for Mongo DB (to update/view)  
Download from - <http://3t.io/mongochef/download/>  
After installing it in a location of your choice, start it up. Go to File -> New Connection

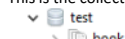


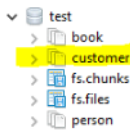
Enter a name for your connection (this can be saved and used later, no auth credentials are needed at this time). Test your connection once complete.



Import the customer.json file in via mongo chef, be default it should be imported in to test. It is in Git.

This is the collection that the rest service will go against.





Double click on customer and you should see a record that our service will use along the lines of below

Key	Value	Type
policyId	00676493	String
distance	0.039	String
latLong	35.468281 -97.532688	String
telephoneId	0910111-2222	String
city	Oklahoma City	String
state	Oklahoma	String
postalCode	73102	String
countryCode	US	String
streetName	N Denney Ave	String
streetNumber	88	String

1. You need to have Java 8 (1.8) installed on your machine.
2. Use an IDE of your choice (I use Eclipse).
3. Unzip the claim\_demo\_v001.zip to a location of your choice.
4. If you do not have Maven installed, please install it from <https://maven.apache.org/download.cgi>
5. Add it to your path

1. JDK and JAVA\_HOME. Make sure JDK is installed, and "JAVA\_HOME" variable is added as Windows environment variable.
2. Download Apache Maven. Visit Maven official website, download the Maven zip file, for example : apache-maven-3.2.2-bin.zip . ...
3. Add M2\_HOME and MAVEN\_HOME. ...
4. Add To PATH. ...
5. Verification.

6. Verify that maven is on your path by typing the command mvn in a command window. You should see something like below if your install is correct.

```
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\karmicjay>mvn
[INFO] Scanning for projects...
[INFO] -----
[INFO] BUILD FAILURE
[INFO] -----
[INFO] Total time: 0.100 s
[INFO] Finished at: 2016-04-18T21:27:58-04:00
[INFO] Final Memory: 4M/279M
[INFO] -----
```

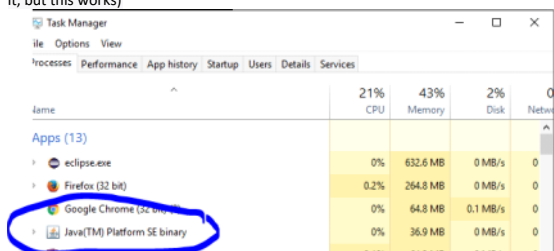
7. Import claim\_demo\_v001 to your IDE as a maven project(if you want to look at the source) if not go to the next step.
8. Build the service by typing in "mvn clean package" in a command window (without the quotes). You will see a success if there are no errors.

```
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] --- maven-jar-plugin:2.5:jar (default-jar) @ claim_demo ---
[INFO] Building jar: d:\WorkerBees\Temp\Claim demo\target\claim_demo-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:1.4.0.M2:repackage (default) @ claim_demo ---
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 18.655 s
[INFO] Finished at: 2016-04-18T21:49:12-04:00
[INFO] Final Memory: 38M/124M
[INFO] -----
```

9. Run the service with the command "mvn spring-boot:run". You should see a success with a message that Tomcat (embedded since this was created using Spring boot) is running on port 8989.

```
2016-04-18 21:50:29.538 INFO 5000 --- [main] s.b.c.e.t.TomcatEmbeddedServletContainer : Tomcat started on port(s): 8989 (http)
2016-04-18 21:50:29.563 INFO 5000 --- [main] c.c.workerbees.ClaimDemoApplication : Started ClaimDemoApplication in 17.362 seconds (JVM running for 22.254)
```

To stop the service kill the process from the task manager (not had a chance to put in the code to stop it, but this works)



10. You can then test the REST service using postman (client for testing REST webservice - available for free from the chrome store). <https://chrome.google.com/webstore/detail/postman/fhbjgbfijnidbgehcddcbncdddomop?hl=en>
11. Startup postman after the install . Point the url to <http://localhost:8989/customer/73102> "73102" is the search parameter for the GET operation. We have one record in the DB with this zip code. You should see the response in the JSON format as below.



Builder

Team Library

OFFLINE

Sign In

http://localhost:8989/cust/

No environment

GET

http://localhost:8989/customer/73102

Params

Send

Save

Authorization

Headers

Body

Pre-request Script

Tests

Generate Code

key

value

Bulk Edit

Presets

Body

Cookies

Headers (5)

Tests

Status: 200 OK

Time: 6482 ms

Pretty

Raw

Preview

JSON

```
1 {
2   "customers": [
3     {
4       "name": "John Doe",
5       "policyNbr": "008763493",
6       "distance": "0.039",
7       "latlong": "35.466501,-97.522696",
8       "telephoneNbr": "(918)111-2222",
9       "city": "Oklahoma City",
10      "state": "Oklahoma",
11      "postalCode": "73102",
12      "countryCode": "US",
13      "streetName": "N Dewey Ave",
14      "structureNum": "99"
15    }
16  ]
17 }
```