Overview.

The following document outlines how to set up the project (which is built using Maven) and then describes the testing both automated and via the browser provided with the project.

Pre-requisites.

This project relies on java 8 and maven 3 being installed on the local machine. The service was developed as a Spring Boot Web Application using Intellij, and has been tested on HSQL and MySQL databases

Installation.

The project has been zipped up with this documentation into a file called *earthquakes-jf-1.0.0.zip.* Extract the zip file into a folder (e.g. *c://Earthquakes*) which I will elsewhere refer to as the <*Root-Folder*>

Assuming java 8 and maven 3 are installed, the project can be built with the standard mavcen commands – e.g
mvn clean install

View project

It is recommended that the project be viewed using IntelliJ.It is maven project and the *pom.xml* is in the <*Root-Folder*>

Automated Tests.

Unit and Integration tests support all but the basic POJO, Exception and CrudRepository classes.

In particular each of the RestController's (*EarthquakeController, RegionController* and *StationController*) have supporting integration tests which use the Spring MVC Test Framework with an HSQL database to provide full coverage of each of the controllers' endpoints. One particular endpoint *EarthquakeController POST /earthquakes* is only available for authneticated users, there are tests for both the authenticated and non-authenticated cases – see the class *EarthquakeControllerIT* for details

From the *<Root-Folder>* enter the command :

mvn test

JaCoCo – Test Reports

The JaCoCo runtime agent has been added to the build - see *http://www.eclemma.org/jacoco/trunk/doc/maven.html*. - to provides basic test reports on coverage.
To see the code coverage, enter the command
***mvn clean jacoco:prepare-agent install jacoco:report*.**

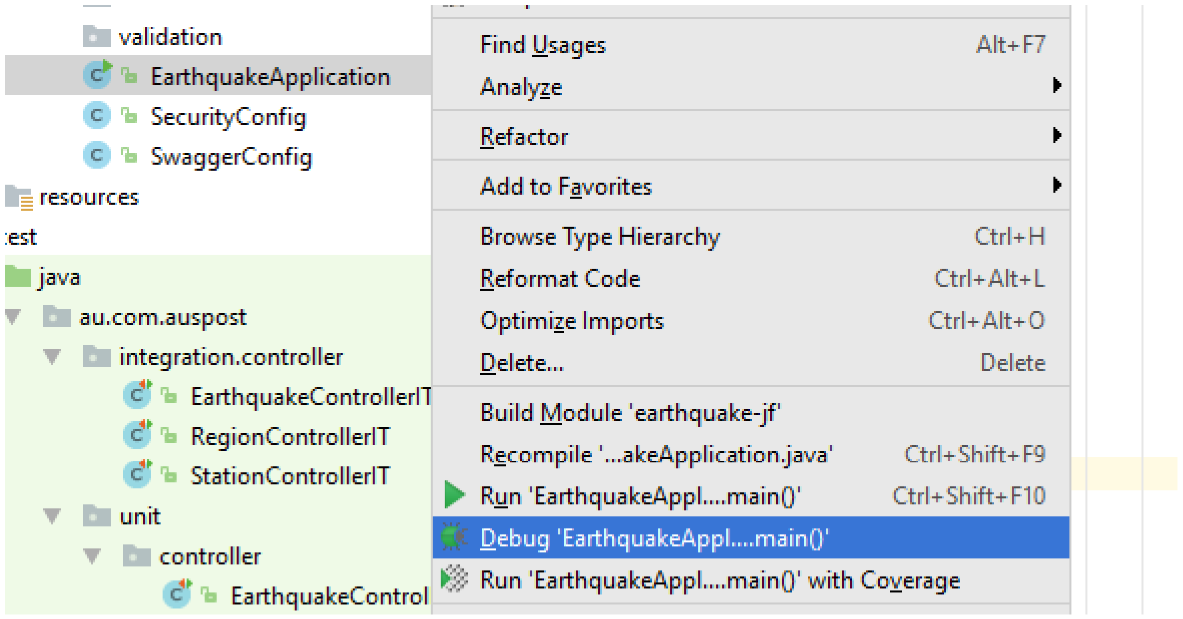
Once complete, open the file *<Root-Folder>/target/site jacoco/index.html* to view the test report.

Starting the Earthquake service

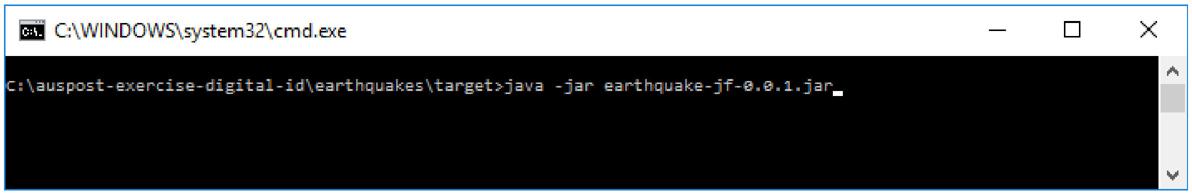
The service can be started in a number of ways.

1. ***From IntelliJ***

   With the project open, find and run (or debug) the main class *EarthquakeApplication*. The project will start in IntelliJ

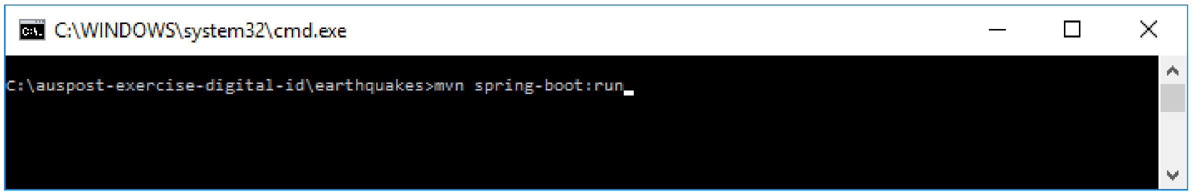


2. ***Run the* earthquake-jf-0.0.1.jar file that is created as part of the maven build**



```
C:\auspost-exercise-digital-id\earthquakes\target>java -jar earthquake-jf-0.0.1.jar
```

3. ***From the <ROOT-Folder>, enter mvn spring-boot:run***

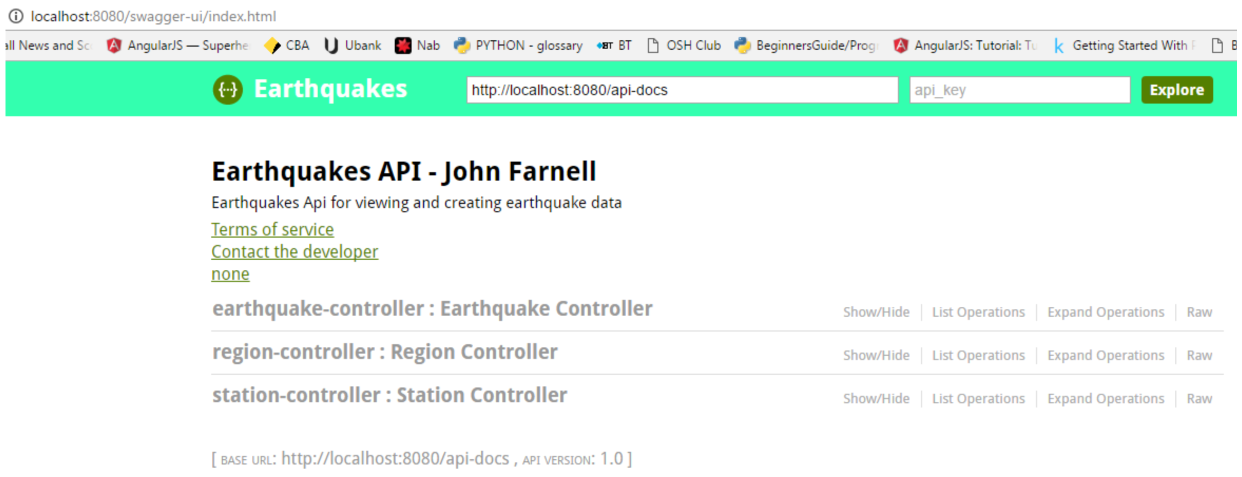


```
C:\auspost-exercise-digital-id\earthquakes>mvn spring-boot:run
```

When the service is running against DEV (see *application.properties)* which is the default setting. The Persistence layer is a HSQL database, and data is loaded up via the *import.-data.sql* – see the file for details of *User, Station, Region* and *Earthquake* data that is loaded at start-up.

***Documentation with Swagger* - (http://swagger.io)**

Swagger (http://swagger.io) has been incorporated into the API to provide REST API documentation. With the application running as described above, open the a browser and go to *http://localhost:8080/swagger-ui/index.html*. The page shown below will be displayed and the available restful endpoints can be viewed and exercised (see the *Try it Out* buttons)



Earthquakes API - John Farnell

Earthquakes Api for viewing and creating earthquake data

Terms of service
Contact the developer
none

earthquake-controller : Earthquake Controller — Show/Hide | List Operations | Expand Operations | Raw
region-controller : Region Controller — Show/Hide | List Operations | Expand Operations | Raw
station-controller : Station Controller — Show/Hide | List Operations | Expand Operations | Raw

[BASE URL: http://localhost:8080/api-docs , API VERSION: 1.0]

The *GET* requests are straightforward, the POST for *EarthController* is secured. The request below will work, but you will need to enter a username/password of *admin/password*

```
{
      "src": "NZ",
      "eqid": "EQ-IDNEW",
      "timedate": "2016-01-09 19:47:52",
      "lat": 70,
      "lon": 68,
      "magnitude": 3.56,
      "depth": 22.79,
      "region": "Nicobar Islands, India region"
}
```