

## Assignment for Athletigen

### 1. Basic information about this DEMO

Candidate: Danxun Jiao, danxun123@gmail.com

IDE: Eclipse Mars.2 Release (4.5.2)

Language: Java 8

Architecture: Spring Boot (1.4.0.M2) + Spring MVC + Hibernate + h2 (1.4.191)

Maven: apache-maven-3.3.9

Embedded Database:

H2

Table Name: STATE

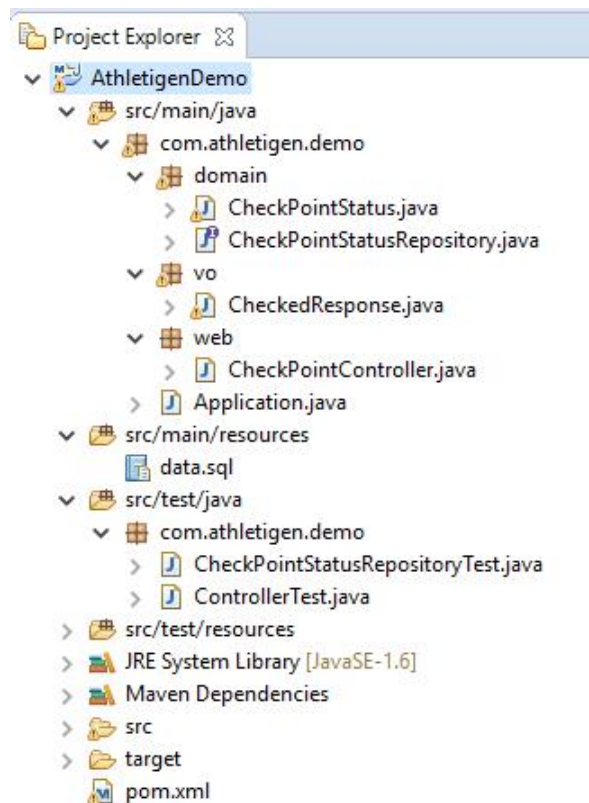
When the restful API is called, it will connect to the H2, and fetch data from this table

The default init script: INSERT INTO STATE (name,state) VALUES('isAlive','200');

### 2. How to run this DEMO:

1. Unzip to folder: AthletigenDemo
2. Or, get from <https://github.com/danxun/DemoForAthletigen0423>
3. cd AthletigenDemo
4. mvn spring-boot:run
5. Or, java -jar AthletigenDemo-0.0.1-SNAPSHOT.jar
6. Rest checkpoint: <http://127.0.0.1:8080/services/heartbeat/isAlive>
7. exit the application hit ctrl-c.

### 3. Structure of source code:



### 4. Task finished:

The application will check if the database is available, are able to accept requests.

- Application will not use JDBC to check the database status but rather will rely on

hibernate API.

Yes. [Hibernate 5.10](#)

- Application will not use native servlets but rather rely on SpringMVC

Yes. [Spring-web-4.3.0](#)

- Application will use maven build manager

Yes. [Maven 3.3.9](#)

- Application will respond with a 200 response code only if everything is ok, and 500 otherwise.

- Application will return the following JSON objects that matches the correct response Code.

Yes.

After running ( `mvn spring-boot:run`)

Access <http://127.0.0.1:8080/services/heartbeat/isAlive>

If content of data.sql is: `INSERT INTO STATE (name,state) VALUES('isAlive','200');`

will return:

```
{"developerMessage":"life is good!","code":200}
```

If content of data.sql is: `INSERT INTO STATE (name,state) VALUES('isAlive','300');`

will return:

```
{"developerMessage":"Something is wrong, check the system logs","code":500}
```

## 5. Comments

1. Two unit tests: `src/test/java`

`CheckPointStatusRepositoryTest`, it test the database connection and data available

`ControllerTest`, it test the result from restful API

2. Business logic in `CheckPointController`

Try to select \* from state using Hibernate JPA, if there is Exception, return 500

If the value of state equals 200, return 200

Else return 500

3. `CheckedResponse`

a class will be return to the load balancer, Spring convert it to json format

## 6. Some thought about this demo

Since this is a boilerplate application, I choose the spring boot solution because spring boot makes it easy to create stand-alone Spring based applications.

In order to simplify this demo, I don't use layer of Service. In a real project, this layer should be added. Controller -> service -> domain -> database.