Using the Java Starter Template

About the templates:

Two templates have been provided for use: template_database and template_basic.

- **template_database:** A spring boot application using H2 in-memory database (via Hibernate). The web pages rendered have been created using Thymeleaf.
- **template_basic:** A spring boot application with a welcome page (rendered using Thymleaf) consisting of links to tutorials to explore Spring, Hibernate and Thymeleaf.

Requirements:

- Java version 1.7. To run java version 8, change the pom.xml file. However, the Hibernate dependency in pom will have to be changed to the latest version since new Hibernate versions are compatible only with java 1.8 and above.
- Tomcat server v8.0 (in case of execution of templates without using embedded server).
- Maven

Running the web app:

- ### Running the web app via embedded server:
    1) Navigate to the folder containing pom.xml and 'src' folder.

2) Run the following command:

 mvn spring-boot:run

3) Open the browser and navigate to the following URL:

   http://localhost:8091

● Running the web app using external Tomcat server:

1) Navigate to the folder containing the source folder and pom.xml.

2) Run the following command:

   mvn package

   The war file is generated in the 'target' folder

3) Load the war file into the /webapps folder of the tomcat directory

4) Start the server

5) Navigate to the following url http://localhost:8080 (the default port used by Apache tomcat is 8080)

6) Via Manager App option, open the web app

Notes:

● To generate the war file, navigate to the folder containing the src folder and pom.xml. Run the following command:
mvn package
The war file will be generated in the target folder which can then be imported to IDE such as Eclipse.

● To change the port number used for running the embedded server, edit the 'application.properties' file.

- The Hibernate version specified in pom.xml is compatible with java version 1.7. To use the latest version of hibernate, change the java version to 1.8 in pom.xml and add the hibernate dependency, replacing the old one.
- To use custom database, create a new a file 'hibernate.config.xml' with the following content:

```xml
//This file should be in src/main/java
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
  <session-factory>
    <property
name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
    <property
name="hibernate.connection.password">TheDeadman</property>
    <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/demo?useSSL=false</property>
    <property
name="hibernate.connection.username">root</property>
    <property
name="hibernate.dialect">org.hibernate.dialect.MySQL5Dialect</property>
    <property name="hbm2ddl.auto">update</property>
  </session-factory>
</hibernate-configuration>
```

This file should be placed in src/main/java. Please refer to tutorials on the internet for implementing Hibernate using custom databases.