CST-339 Activity 1 Guide

Contents

Part 1: Tools Installation and Validation

Overview

Goal and Directions:

In this activity you will install the latest version of the Spring Tool Suite and validate your locale environment by developing a simple "Hello World" Spring Boot application. Note, these activities are written assuming the use of the Spring Tool Suite IDE. With approval from your instructor you are free to explore the use of the Microsoft Visual Studio Code IDE in this course as long as you are comfortable with finding alternative steps to complete the activities. See Appendix B for developer notes if you will be using the Microsoft Visual Studio Code IDE.

Execution

Execute this assignment according to the following guidelines:

1. Download the latest version of Spring Tool Suite at https://spring.io/tools. Take a screenshot of the Spring Tool Suite About Box.
    a. Mac OSX: Open the downloaded DMG file and copy the application to your Applications folder.
    b. Windows: Run the downloaded self-extracting JAR file. If the file does not run and the Open With Java option is not available from the context menu when right clicking on the file then you will need to install the Java Runtime Environment (JRE) version 16 from https://www.oracle.com/java/technologies/javase-downloads.html. Once the downloaded self-extracting JAR file has been run you can copy the extracted folder to a desired location in your home directory.
2. Create a new workspace named *workspaceCST-339*.
3. Create a Spring Boot Project by following steps in Appendix A and naming your Group and Package Name as *com.gcu* and your Project Name *topic1-1*.
4. Update the Java Runtime Execution version in Eclipse by right clicking on the Project, select Build Path > Configure Build Path. Select the Libraries tab. Select the JRE System Library entry and click the Edit button. Select the JavaSE-15 from the Execution Runtime dropdown list.

5. Under the *src/main/java* folder. Select the *Topic11Application.java* source file. Click the Run icon from the Spring Tool Suite tool bar (if prompted for an application type to run select Spring Boot App). Validate there are no errors in the console. To debug your application, you can select the Debug icon from the Spring Tool Suite tool bar.
6. Print a *Hello World* message in the first line of the *main()* method in the *Topic11Application.java* source file. Save the changes. Ensure the application auto-restarts and the message is printed (before the Spring splash screen) in the Console window. Take a screenshot.

```
@SpringBootApplication
public class Topic11Application
{

    public static void main(String[] args)
    {
        System.out.println("Hello from my Spring Boot application");
        SpringApplication.run(Topic11Application.class, args);
    }

}
```

```
Hello from my Spring Boot application

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.3.0.RELEASE)
```

7. Open a browser and go to *localhost:8080*. A default Spring Boot *Whitelabel Error Page* should be displayed. Take a screenshot.
8. Create a new static HTML file named *index.html* in the *src/main/resources/static* directory. Put a *Hello World* HTML content in the body and set the title of document.
9. Restart the application. Open a browser and go to *localhost:8080*. The content of the *index.html* page should be displayed. Take a screenshot.

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="UTF-8">
5          <title>Hello World</title>
6      </head>
7
8      <body>
9          <h1>Hello World from my Spring Boot Application!</h1>
10     </body>
11 </html>
```

10. Stop the application by clicking the Stop icon in the Console window.

Deliverables:

The following needs to be submitted as this part of the Activity:

a. Screenshot of the Spring Tool Suite About Box.
b. Screenshot of console output when running the *Topic11Application* class.
c. Screenshot of the *Whitelabel Error Page*.
d. Screenshot of the Hello World *index.html* page.

Part 2: Learning Maven

Overview

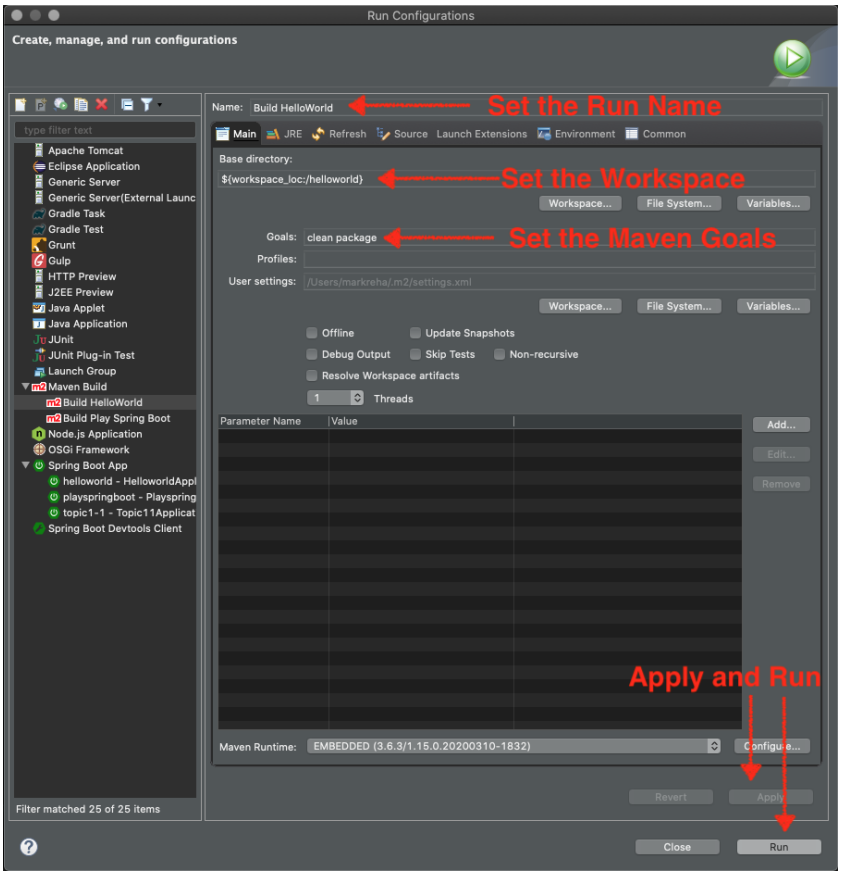
Goal and Directions:

In this activity you will learn about the Maven build and dependency management system. You will also build an application from Part 1 using Maven.

Execution

Execute this assignment according to the following guidelines:

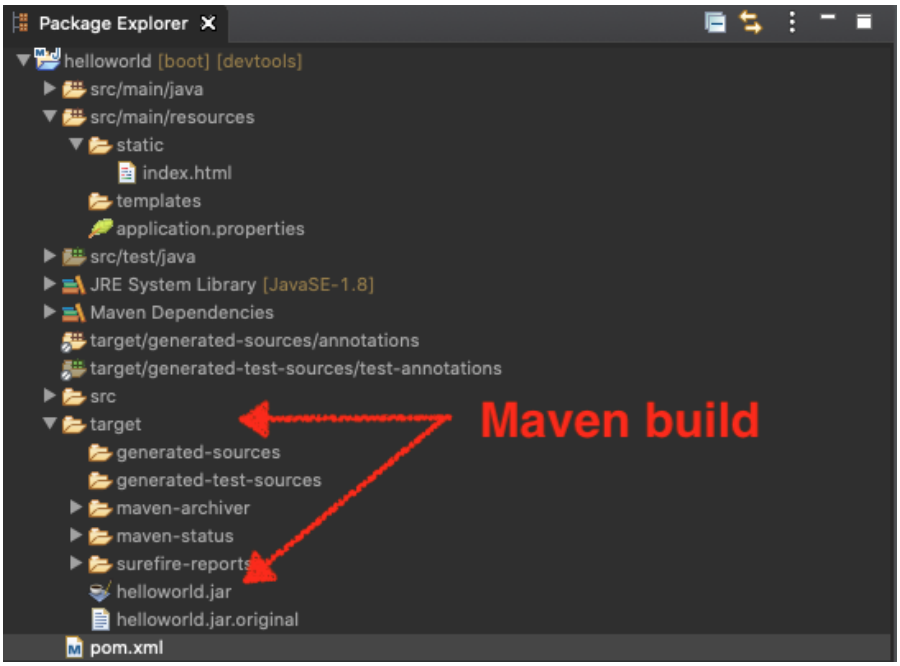
1. Complete the following tutorials on Maven:
    a. Maven In 5 Minutes
    b. What Is Maven
    c. Maven Tutorial from TutorialsPoint
2. Inspect the Project Directory Structure and the basic Maven POM file (*pom.xml*) located in the root of your project.
3. Update the java.version property in the Maven POM file (*pom.xml*) to a value of 15.
4. Use the Hello World application built in Part 1 and built the application within Spring Tool Suite using Maven.
    a. Create a Maven Configuration run script by selecting the Run > Run Configurations.
    b. Select the Maven Build type and click the New icon.
    c. Set the Name field to 'Build HelloWorld'. Set the Base directory to the root of your workspace, set the goals to 'clean package'.
    d. Click the Apply button.
    e. Click the Run button to run a build. You should see a BULD SUCCESS message in the console.

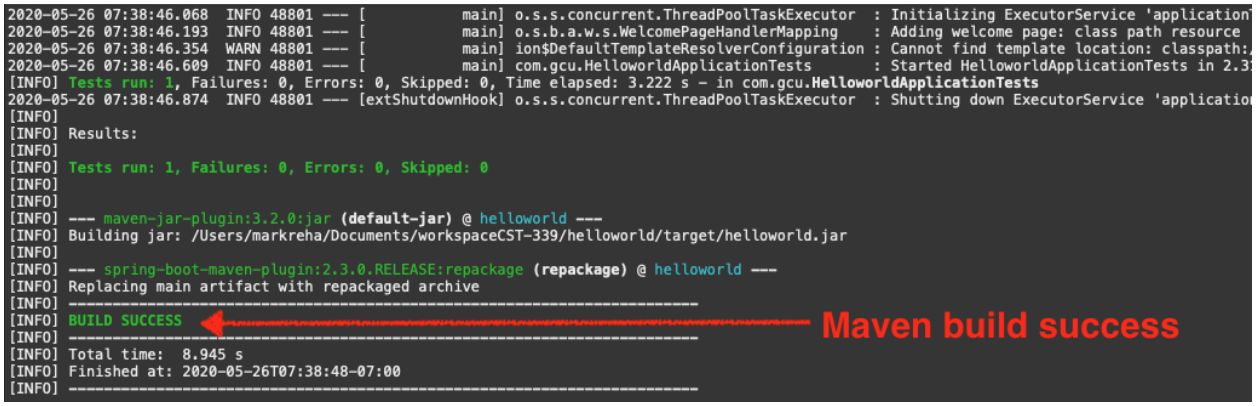


5. Add a <*finalName*> tag within the <*build*> tag of the POM file to set the output JAR filename. Right click on your project and select the Maven > Update Project menu option to update the project.

```
<build>
    <finalName>helloworld</finalName>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>
```

6. Run a Maven build by selecting the Run Configuration created in the previous steps by selecting the Run > Run Configurations menu option. The Maven build output will be displayed in the Console window. Ensure there are no build errors. A JAR file named *helloworld.jar* will be built in the target directory (you might have to right click on your project and select the Refresh menu option). Take a screenshot of the console output for the Maven build.





7. Copy the JAR file (*helloworld.jar*) to Desktop. Open a terminal window and run the JAR application by using the following command. NOTE: the path to the Java application can be found by going to the Spring Tool Suite IDE Properties, selecting Java > Installed JREs, copying the path listed under the JRE 15.0, and appending /*bin* to that path. Note, that if there are any spaces in the directory path then the Java command including the path should be entirely enclosed in double quotes.

    Mac: [PATH to Java 15 JRE Install]/bin/java -jar helloworld.jar

    Windows: [PATH to Java 15 JRE Install]\bin\java.exe -jar helloworld.jar

8. Open a browser and go to localhost:8080. The content of the index.html page should be displayed. Take a screenshot.

Deliverables:
The following needs to be submitted as this part of the Activity:

a. Screenshot of the Hello World *index.html* page.

Research Questions

1. Research Questions: For traditional ground students in a Word document answer the following questions:
    a. Research Spring Boot. Compare building dynamic web applications when using Spring Boot versus just using the Spring framework. How do they differ?
    b. Research Gradle, which is another popular build and dependency management tool. How does it differ from Maven?

Final Activity Submission

1. In a Microsoft Word document complete the following for the Activity Report:
    a. Cover Sheet with the name of this assignment, date, and your name.
    b. Section with a title that contains all the screenshots for each part of the Activity.
    c. Section with a title that contains the answers to the Research Questions (traditional ground students only).
2. Submit the Activity Report to the digital classroom.

Appendix A: Creating a default Spring Boot Application

During this course there will be a number of times where you will need to create a "boiler plate" Spring application. The following instructions can be used to create this application.

1. In the Spring Tool Suite use the built-in wizard by selecting the File>Spring Starter Project.
2. Select a Maven Project. Select the Java Programming Language. Select the version of Java as specified by the instructor. Select Spring Boot version as specified by the instructor.
3. Enter a desired Group and Package Name (like com.gcu), enter a desired Artifact and Name, and enter a brief description.
4. Add the Spring Boot DevTools (under Developer Tools), Spring Web (under Web), Thymeleaf and (under Template Engines) as Dependencies.
5. Click the Finish button.

Appendix B: Using the Microsoft Visual Studio Code IDE

The Microsoft Visual Studio Code IDE has adequate support for programming in Java. The following are the recommended extensions to be installed to support programming in Java for this IDE.

- Spring Boot Tools
- Spring Initializr
- Spring Boot Dashboard
- Java Extension Pack

The Microsoft Visual Studio Code IDE does not have the exact same concept of Workspaces like you would find in the IDE's based on the Eclipse IDE. However, to mimic the notion of projects in your Workspace simply create a top-level directory where you want all your projects to be held. Then use the *Add Folder to Workspace* menu option and then save your Workspace by selecting the *Save Workspace As* menu option. You can then create project folders underneath the top-level directory. Any number of projects can be placed in the Workspace. You can even copy one project to another by using the *Copy* and *Paste* menu options on the project directory while working in the IDE.

It should be noted that the Microsoft Visual Studio Code IDE does not have all the convenient wizards for creating Java packages, Java classes, Java interfaces, etc. that you will find in the Eclipse IDE. You will need to look up the alternative menu options to perform many of these common tasks. If you are not comfortable completing these tasks without the wizards, then it is highly recommended that you complete this course using the tried-and-true Eclipse IDE.

Until further notice, Java 8 is the only approved version of Java that is used in all the Java courses.