

Assisted Practice: 1.1 Environmental Setup

This section will guide you to:

- Set up Eclipse to work with Spring
- Create a pom.xml configuration file to add all the required components

Development Environment

- Eclipse IDE for Enterprise Java Developers v2019-03 (4.11.0)
- Apache Tomcat Server v9.0
- JRE: OpenJDK Runtime Environment 11.0.2
- All other software components are defined in the Project Object Model pom.xml

This lab has four subsections, namely:

- 1.1.1 Creating a Maven Project which is web enabled
- 1.1.2 Creating pom.xml for including the required components
- 1.1.3 Building the project
- 1.1.4 Pushing the code to your GitHub repositories

Step 1.1.1: Creating a Maven Project which is web enabled

- Open Eclipse
- Go to the **File** menu. Choose **New->Maven Project**
- Uncheck **Create a Simple Project** and check **Use Default Workspace Location** and click on **Next**
- From the **archetype** list, choose the row that has **Artifact Id** as **maven-archetype-webapp** and click on **Next**
- Enter **Group Id** as **com** and **Artifact Id** as **SpringSetup** and click on **Finish**
- This will create the project files in the Project Explorer

- Before building the project, we need to confirm that the **servlet.jar** has been added to the project
- In the Project Explorer, right click and choose **Properties**
- Select **Java Build Path** from the options on the left
- Click on **Libraries** tab on the right
- If there is an existing entry for the **servlet.jar**, then click on **Cancel** and exit the window
- If it is not there, then click on **Classpath** entry and click on **Add External JARs** button on the right
- From the **file** list, select **servlet.jar** file and click on **Ok**
- Click on **Apply and Close**

Step 1.1.2: Creating pom.xml for including the required components

- In the Project Explorer, expand **SpringSetup** and double click on **pom.xml**
- Add the following entries:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>SpringSetup</groupId>
  <artifactId>SpringSetup</artifactId>
  <packaging>war</packaging>
  <version>0.0.1-SNAPSHOT</version>
  <name>SpringSetup Maven Webapp</name>
  <url>http://maven.apache.org</url>

  <properties>
    <org.springframework.version>5.1.6.RELEASE</org.springframework.version>
  </properties>

  <dependencies>
```

```
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>3.8.1</version>
  <scope>test</scope>
</dependency>
```

```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-core</artifactId>
  <version>${org.springframework.version}</version>
</dependency>
```

```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-expression</artifactId>
  <version>${org.springframework.version}</version>
</dependency>
```

```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-beans</artifactId>
  <version>${org.springframework.version}</version>
</dependency>
```

```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context</artifactId>
  <version>${org.springframework.version}</version>
</dependency>
```

```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context-support</artifactId>
  <version>${org.springframework.version}</version>
```

```
</dependency>
```

```
<dependency>
```

```
  <groupId>org.springframework</groupId>
```

```
  <artifactId>spring-jdbc</artifactId>
```

```
  <version>${org.springframework.version}</version>
```

```
</dependency>
```

```
<dependency>
```

```
  <groupId>org.springframework</groupId>
```

```
  <artifactId>spring-orm</artifactId>
```

```
  <version>${org.springframework.version}</version>
```

```
</dependency>
```

```
<dependency>
```

```
  <groupId>org.springframework</groupId>
```

```
  <artifactId>spring-web</artifactId>
```

```
  <version>${org.springframework.version}</version>
```

```
</dependency>
```

```
<dependency>
```

```
  <groupId>org.springframework</groupId>
```

```
  <artifactId>spring-webmvc</artifactId>
```

```
  <version>${org.springframework.version}</version>
```

```
</dependency>
```

```
<dependency>
```

```
  <groupId>javassist</groupId>
```

```
  <artifactId>javassist</artifactId>
```

```
  <version>3.12.1.GA</version>
```

```
</dependency>
```

```
<dependency>
  <groupId>taglibs</groupId>
  <artifactId>standard</artifactId>
  <version>1.1.2</version>
  <scope>runtime</scope>
</dependency>

<dependency>
  <groupId>commons-dbcp</groupId>
  <artifactId>commons-dbcp</artifactId>
  <version>1.4</version>
</dependency>

<dependency>
<groupId>javax.servlet</groupId>
<artifactId>jstl</artifactId>
<version>1.2</version>
</dependency>

<dependency>
<groupId>javax.servlet</groupId>
<artifactId>jstl</artifactId>
<version>1.2</version>
</dependency>

<dependency>
<groupId>javax.persistence</groupId>
<artifactId>persistence-api</artifactId>
<version>1.0.2</version>
<scope>provided</scope>
</dependency>

</dependencies>
<build>
```

```
<finalName>SpringSetup</finalName>  
</build>  
</project>
```

Step 1.1.3: Building the project

- From the **Project** menu at the top, click on **Build**
- If any compile errors are shown, fix them as required

Step 1.1.4: Pushing the code to your GitHub repositories

- Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

- Initialize your repository using the following command:

git init

- Add all the files to your git repository using the following command:

git add .

- Commit the changes using the following command:

git commit . -m "Changes have been committed."

- Push the files to the folder you initially created using the following command:

git push -u origin master

