

Aug 31st Lecture and Project spring-mvc-demo

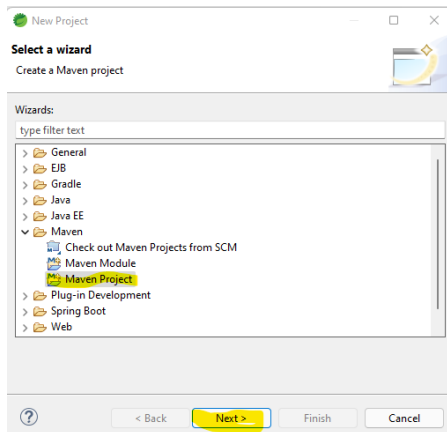
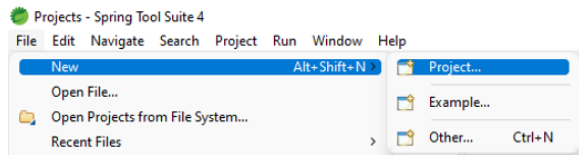
Create a Maven MVC project and deploy on an IDE Tomcat server.

Table of Contents

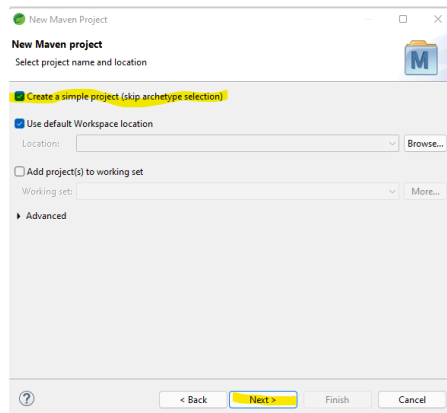
Create a Maven MVC project and deploy on an IDE Tomcat server	2
Create New Maven Project	2
Update the pom.xml	4
Update Java Version.....	4
Correct Error: Web.xml is missing.....	6
Correct web.xml error.....	10
Open Web.xml File	10
Update the web.xml file.....	12
Add DispatcherServlet to web.xml	12
Create an applicationContext.xml File	13
Update the applicationContext.xml File	14
Add other project dependencies to pom.xml	15
Complete the MVC demo project	18
Spring MVC Project Next Steps for Tomcat Server(s)	18

Create a Maven MVC project and deploy on an IDE Tomcat server

Create New Maven Project



Select “Create a simple project ...”
Click “Next >”



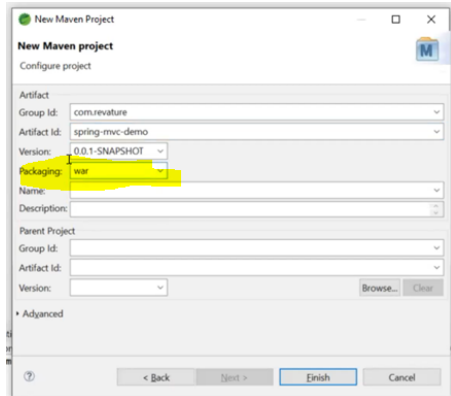
Enter the Group Id: com.revature

Enter the Artifact Id: spring-mvc-demo

Select Packaging: **war**

Note: packaging must be war to deploy on a Tomcat server.

Click Finish



New Maven Project

Configure project

Artifact

Group Id: com.revature

Artifact Id: spring-mvc-demo

Version: 0.0.1-SNAPSHOT

Packaging: war

Name:

Description:

Parent Project

Group Id:

Artifact Id:

Version:

Browse... Clear

Advanced

< Back Next > Finish Cancel

Update the pom.xml

Update Java Version

BEFORE PROCEEDING MAKE SURE THE PROJECT IS ON Java 1.8 in the pom.xml and has been rebuilt.

```
spring-mvc-demo/pom.xml
1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   <modelVersion>4.0.0</modelVersion>
3   <groupId>com.revature</groupId>
4   <artifactId>spring-mvc-demo</artifactId>
5   <version>0.0.1-SNAPSHOT</version>
6   <packaging>war</packaging>
7
8   <properties>
9     <maven.compiler.source>1.8</maven.compiler.source>
10    <maven.compiler.target>1.8</maven.compiler.target>
11  </properties>
12 </project>
```

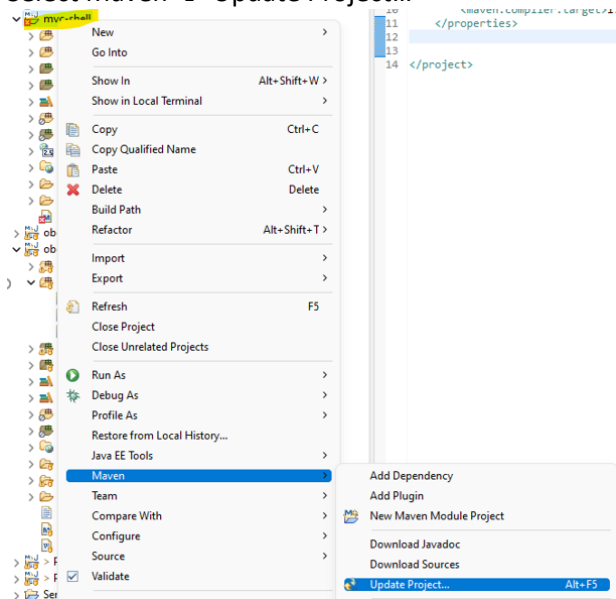
Add the Java 1.8 in the pom.xml

```
<properties>
  <maven.compiler.source>1.8</maven.compiler.source>
  <maven.compiler.target>1.8</maven.compiler.target>
</properties>
```

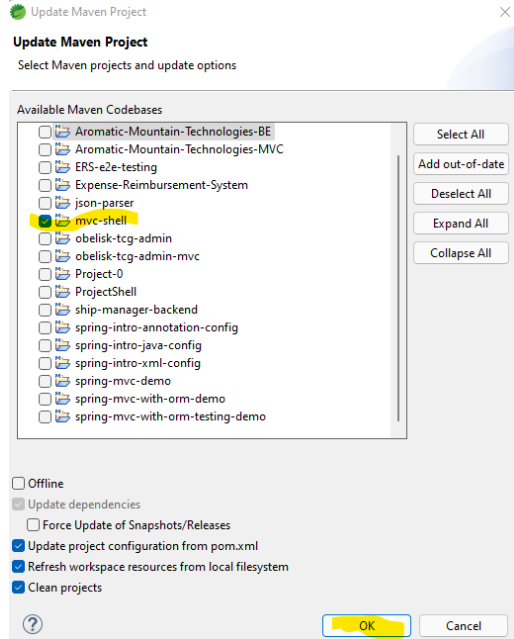
Rebuild the Maven project

Right click on the Project

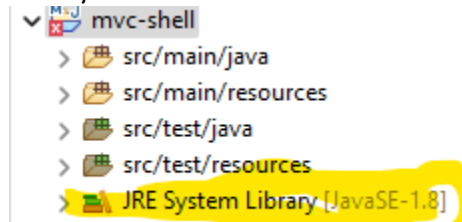
Select Maven → Update Project...



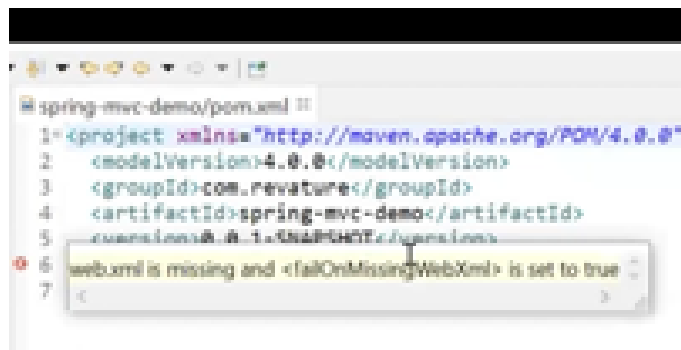
The project should already be selected.
Click “OK”



Library should now be at the new Java version



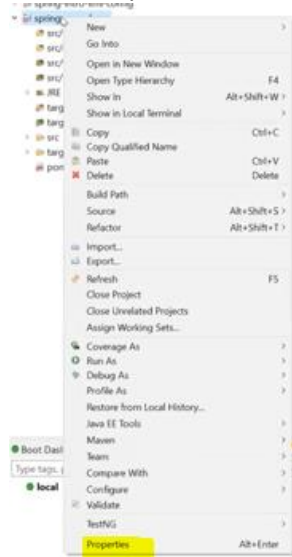
Correct Error: Web.xml is missing



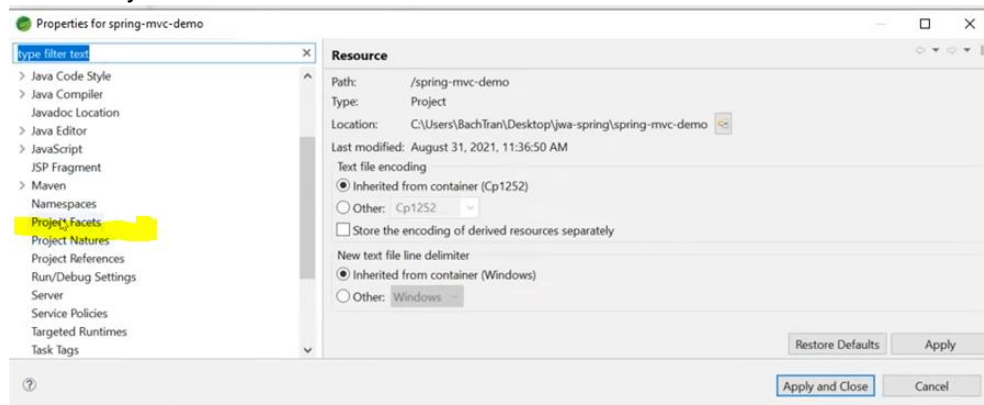
Need to generate Web.xml

Right click on project

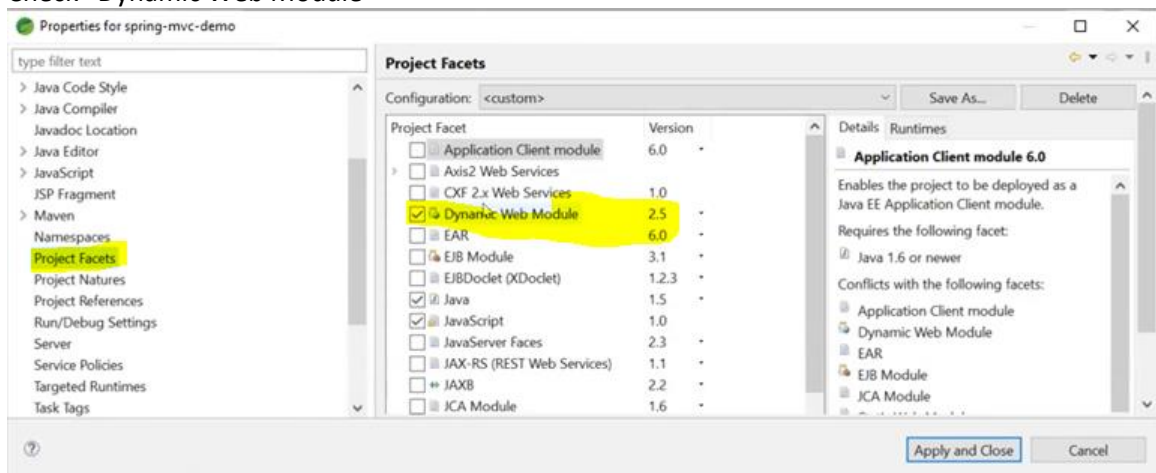
Select Properties



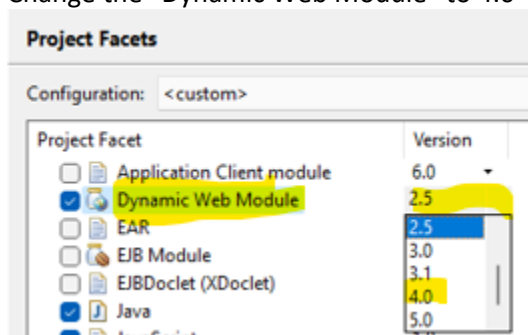
Select Project Facets



Check “Dynamic Web Module”

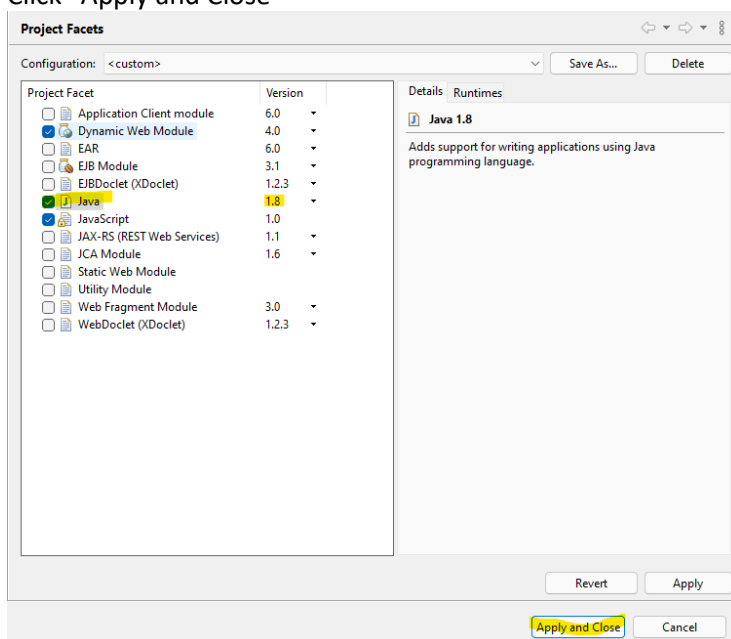


Change the “Dynamic Web Module” to 4.0



Change to new version Java 1.8 or newer version (if not already set)

Click “Apply and Close”

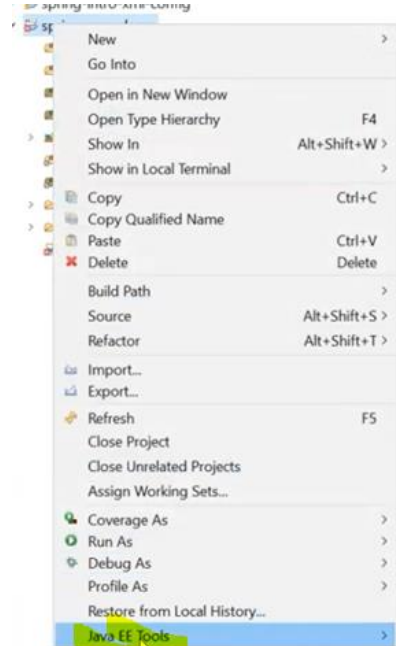


Install Java EE Tool if needed

Right click on project

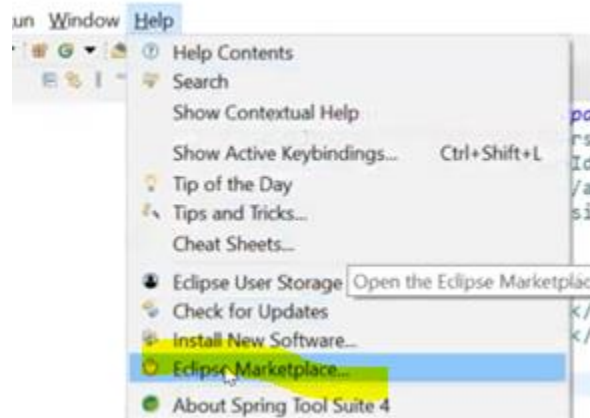
If “Java EE Tools” is in the menu list skip to **“Generate Deployment Descriptor Stub”**

If “Java EE Tools” is not in the menu list follow the next two steps



Step 1: Install via Eclipse Market Place under the Help menu

il Suite 4



Step 2: Search tab: enter enterprise

If not already installed, install the component

Once the installation completes restart the IDE for the changes to take affect

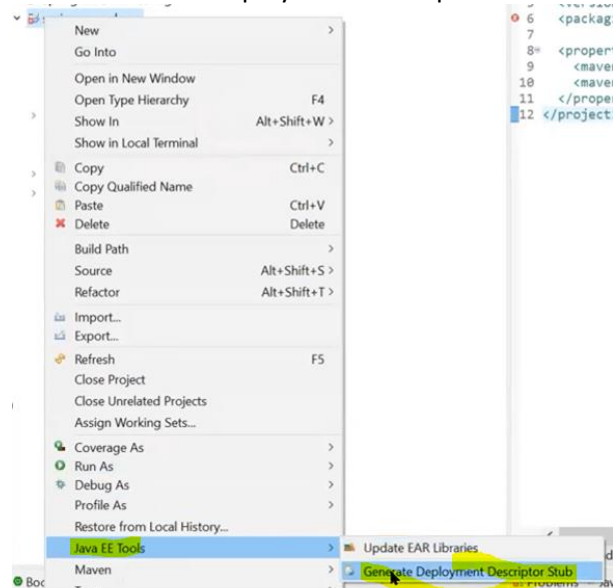


Generate Deployment Descriptor Stub

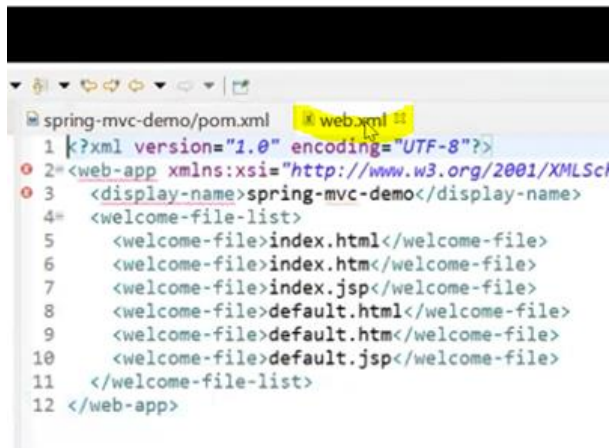
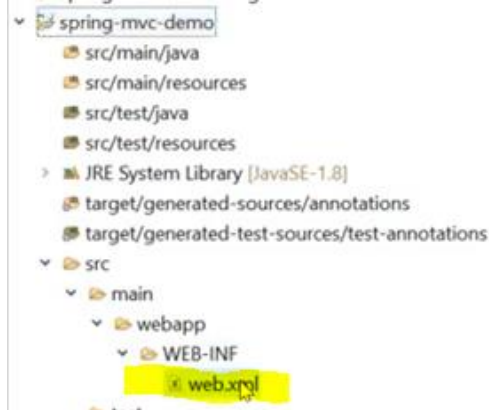
Right click on the project

Select Java EE Tools

Select Generate Deployment Descriptor Stub



The web.xml is generated in following location.
The pom.xml error should now be resolved.



Correct web.xml error

Open Web.xml File



Update outdated link:

From:

```
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" version="2.5">
```

To:

```
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd" version="4.0">
```

You might need to do spacing on the display name as well to get rid of the errors.

```
web.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml
3
4 <display-name>mvc-shell</display-name>
5
6 <welcome-file-list>
7   <welcome-file>index.html</welcome-file>
8   <welcome-file>index.htm</welcome-file>
9   <welcome-file>index.jsp</welcome-file>
10  <welcome-file>default.html</welcome-file>
11  <welcome-file>default.htm</welcome-file>
12  <welcome-file>default.jsp</welcome-file>
13 </welcome-file-list>
14 </web-app>
```

Update the web.xml file

The web.xml file is used for the Tomcat server configuration.

Add DispatcherServlet to web.xml

```
<!-- This is where we configure our DispatcherServlet (which comes from Spring Web) -->
<!-- The DispatcherServlet is the sole Servlet that receives HTTP requests through our Tomcat
server -->
<!-- It then routes the HTTP requests to the appropriate controller -->
<servlet>
    <servlet-name>DispatcherServlet</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

    <init-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>/WEB-INF/applicationContext.xml</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
    <servlet-name>DispatcherServlet</servlet-name>
    <url-pattern>/</url-pattern>
</servlet-mapping>
```

```
web.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml
3
4 <display-name>mvc-shell</display-name>
5
6 <welcome-file-list>
7 <welcome-file>index.html</welcome-file>
8 <welcome-file>index.htm</welcome-file>
9 <welcome-file>index.jsp</welcome-file>
10 <welcome-file>default.html</welcome-file>
11 <welcome-file>default.htm</welcome-file>
12 <welcome-file>default.jsp</welcome-file>
13 </welcome-file-list>
14
15 <!-- This is where we configure our DispatcherServlet (which comes from Spring Web) -->
16 <!-- The DispatcherServlet is the sole Servlet that receives HTTP requests through our Tomcat server -->
17 <!-- It then routes the HTTP requests to the appropriate controller -->
18 <servlet>
19 <servlet-name>DispatcherServlet</servlet-name>
20 <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
21
22 <init-param>
23 <param-name>contextConfigLocation</param-name>
24 <param-value>/WEB-INF/applicationContext.xml</param-value>
25 </init-param>
26 <load-on-startup>1</load-on-startup>
27 </servlet>
28
29 <servlet-mapping>
30 <servlet-name>DispatcherServlet</servlet-name>
31 <url-pattern>/</url-pattern>
32 </servlet-mapping>
33
34
35 </web-app>
```

All http request goes to the DispatcherServlet which in turn sends to the controller(s) endpoints.

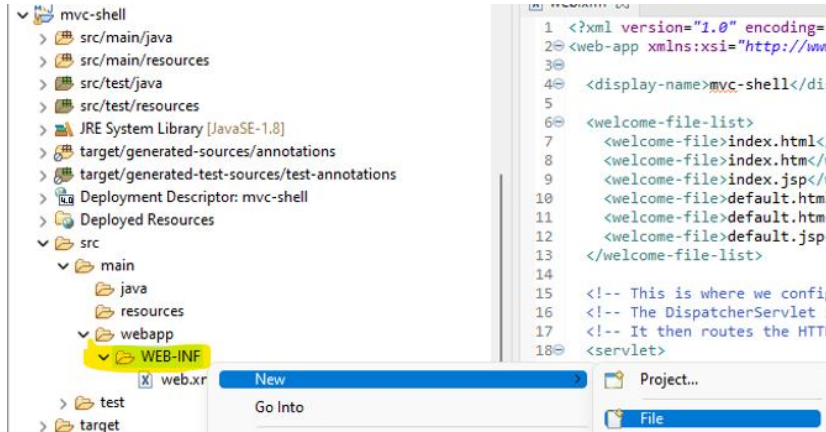
“Front Controller Design Pattern”

Create an applicationContext.xml File

Create the applicationContext.xml file in the directory:
spring-mvc-demo\src\main\webapp\WEB-INF\

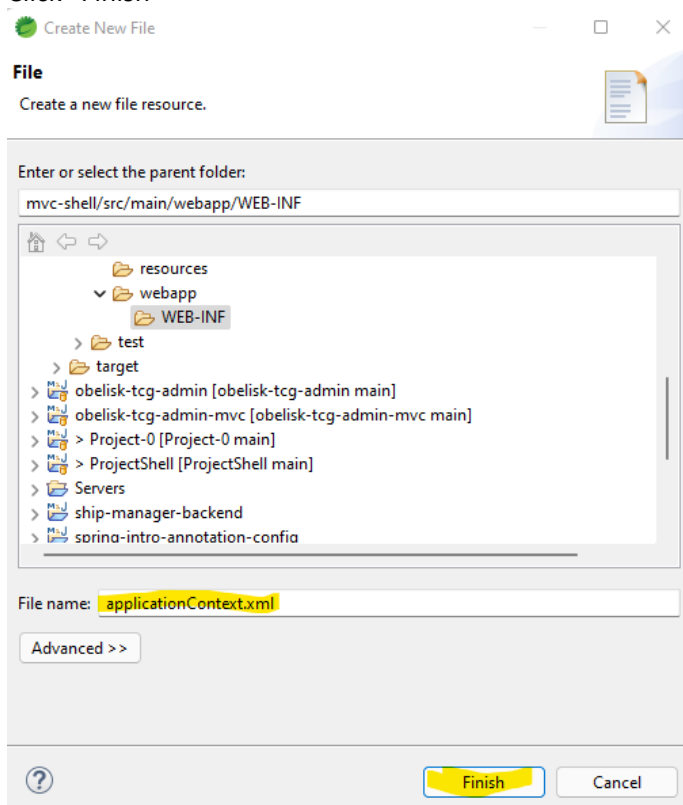
Right click on WEB-INF folder

Select New → File



Enter the "File name:" applicationContext.xml

Click "Finish"



Update the applicationContext.xml File

Add the following to the xml file:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:context="http://www.springframework.org/schema/context"
       xmlns:mvc="http://www.springframework.org/schema/mvc"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd
http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc.xsd">

    <context:component-scan base-package="com.revature"></context:component-scan>
    <mvc:annotation-driven></mvc:annotation-driven>

</beans>
```

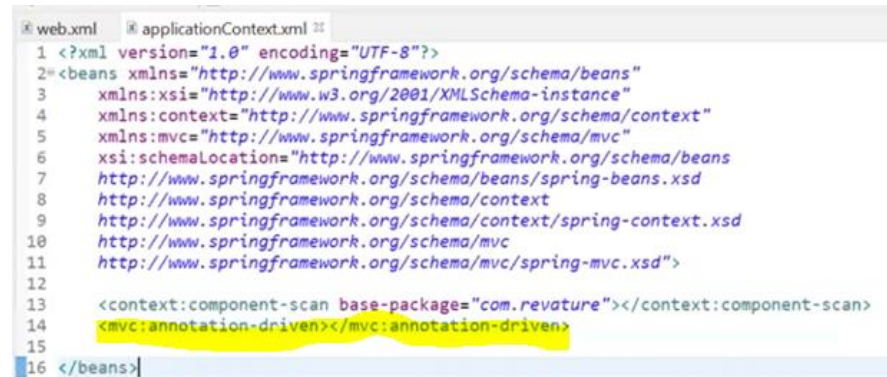


```
web.xml  applicationContext.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xmlns:context="http://www.springframework.org/schema/context"
5     xmlns:mvc="http://www.springframework.org/schema/mvc"
6     xsi:schemaLocation="http://www.springframework.org/schema/beans
7 http://www.springframework.org/schema/beans/spring-beans.xsd
8 http://www.springframework.org/schema/context
9 http://www.springframework.org/schema/context/spring-context.xsd
10 http://www.springframework.org/schema/mvc
11 http://www.springframework.org/schema/mvc/spring-mvc.xsd">
12
13 <context:component-scan base-package="com.revature"></context:component-scan>
14
15 </beans>
```

Make sure to update the base package: com.revature to the project's definition.

Make sure to add the <mvc:annotation> tags

mvc:annotation allows for the @Component, @Service and @RestController tags.



```
web.xml  applicationContext.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xmlns:context="http://www.springframework.org/schema/context"
5     xmlns:mvc="http://www.springframework.org/schema/mvc"
6     xsi:schemaLocation="http://www.springframework.org/schema/beans
7 http://www.springframework.org/schema/beans/spring-beans.xsd
8 http://www.springframework.org/schema/context
9 http://www.springframework.org/schema/context/spring-context.xsd
10 http://www.springframework.org/schema/mvc
11 http://www.springframework.org/schema/mvc/spring-mvc.xsd">
12
13 <context:component-scan base-package="com.revature"></context:component-scan>
14 <mvc:annotation-driven></mvc:annotation-driven>
15
16 </beans>
```

Add other project dependencies to pom.xml

Add dependencies tags

Between the dependencies tags

- Add dependencies for Spring Framework
- Add dependencies for Tomcat
- Add dependencies for other common project items (will vary depending on project):
 - hibernate
 - mariadb
 - junit.jupiter
 - h2database
 - mockito
 - logback

```
<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
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.mvc</groupId>
  <artifactId>mvc-shell</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>

  <properties>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
  </properties>

  <dependencies>
    <!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>javax.servlet-api</artifactId>
      <version>4.0.1</version>
      <scope>provided</scope>
    </dependency>
    <!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-
databind -->
    <dependency>
      <groupId>com.fasterxml.jackson.core</groupId>
      <artifactId>jackson-databind</artifactId>
      <version>2.12.5</version>
    </dependency>
    <!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-webmvc</artifactId>
      <version>5.3.9</version>
    </dependency>
    <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
    <dependency>
      <groupId>org.projectlombok</groupId>
      <artifactId>lombok</artifactId>
      <version>1.18.20</version>
      <scope>provided</scope>
    </dependency>

    <!-- Database Related dependencies -->
    <!-- https://mvnrepository.com/artifact/org.springframework/spring-orm -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-orm</artifactId>
      <version>5.3.9</version>
    </dependency>
    <!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->
```

```

<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-core</artifactId>
  <version>5.5.7.Final</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.apache.tomcat/tomcat-dbcp -->
<dependency>
  <groupId>org.apache.tomcat</groupId>
  <artifactId>tomcat-dbcp</artifactId>
  <version>10.0.10</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.mariadb.jdbc/mariadb-java-client -->
<dependency>
  <groupId>org.mariadb.jdbc</groupId>
  <artifactId>mariadb-java-client</artifactId>
  <version>2.7.4</version>
</dependency>

<!-- Testing related dependencies -->
<!-- https://mvnrepository.com/artifact/org.springframework/spring-test -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-test</artifactId>
  <version>5.3.9</version>
  <scope>test</scope>
</dependency>
<!-- https://mvnrepository.com/artifact/org.junit.jupiter/junit-jupiter-api -->
<dependency>
  <groupId>org.junit.jupiter</groupId>
  <artifactId>junit-jupiter-api</artifactId>
  <version>5.8.0-M1</version>
  <scope>test</scope>
</dependency>
<!-- https://mvnrepository.com/artifact/com.h2database/h2 -->
<dependency>
  <groupId>com.h2database</groupId>
  <artifactId>h2</artifactId>
  <version>1.4.200</version>
  <scope>test</scope>
</dependency>
<!-- https://mvnrepository.com/artifact/org.mockito/mockito-inline -->
<dependency>
  <groupId>org.mockito</groupId>
  <artifactId>mockito-inline</artifactId>
  <version>3.12.4</version>
  <scope>test</scope>
</dependency>
<!-- https://mvnrepository.com/artifact/org.mockito/mockito-junit-jupiter -->
<dependency>
  <groupId>org.mockito</groupId>
  <artifactId>mockito-junit-jupiter</artifactId>
  <version>3.12.4</version>
  <scope>test</scope>
</dependency>
<!-- https://mvnrepository.com/artifact/org.skyscreamer/jsonassert -->
<dependency>
  <groupId>org.skyscreamer</groupId>
  <artifactId>jsonassert</artifactId>
  <version>1.5.0</version>
  <scope>test</scope>
</dependency>

<!-- dependency definitions from logging-demo -->
<dependency>
  <groupId>ch.qos.logback</groupId>
  <artifactId>logback-classic</artifactId>
  <version>1.2.4</version>
</dependency>
<dependency>
  <groupId>org.slf4j</groupId>

```



```
        <artifactId>slf4j-api</artifactId>
        <version>1.7.32</version>
    </dependency>

</dependencies>

</project>
```

Complete the MVC demo project

The above is the basic configuration for any Spring MVC project.

To complete the spring-mvc-demo project add the following packages and file.

\spring-mvc-demo\src\main\java\

Package: com\revature\controller



TestController.java

TestController

Package: com\revature\dto



LoginDTO.java

LoginDTO

Spring MVC Project Next Steps for Tomcat Server(s)

After completing the code in this document, follow the steps in the “Tomcat 02 IDE Setup Aug 31st” Setup document which walks through the steps to create a Tomcat server in the Spring Tool IDE and run this MVC based applications.

Another Tomcat document “Tomcat 03 localhost Deploy war Sep 8th” detail Tomcat setup on a local computer, deploy and run an MVC application war file and connect to the application through localhost.

Another Tomcat document “Tomcat 04 AWS EC2 Sep 8th” detail Tomcat installation and setup on an AWS EC2 instance, deploy and run an MVC application war file and connect to the application through an AWS URL.