

Table of Contents

Aug 31 st Lecture and Project spring-mvc-demo.....	2
Create New Maven Project	2
BEFORE PROCEEDING MAKE SURE THE PROJECT IS ON Java 1.8 in the pom.xml and has been rebuilt.	2
Open pom.xml.....	3
Need to generate Web.xml.....	3
Select Java EE Tools.....	4
If Java EE Tools is not in the menu then:	5
Search tab: enter enterprise	5
Generate Deployment Descriptor Stub	6
Generates web.xml in following location	6
Web.xml File	7
web.xml is for the Tomcat server configuration.....	7
ApplicationContext.xml File	8
Add other project dependencies to pom.xml.....	10
The above is the basic configuration for any Spring MVC project.....	10
Spring MVC Project Next Steps for Tomcat Server(s)	11

Aug 31st Lecture and Project spring-mvc-demo

Create New Maven Project

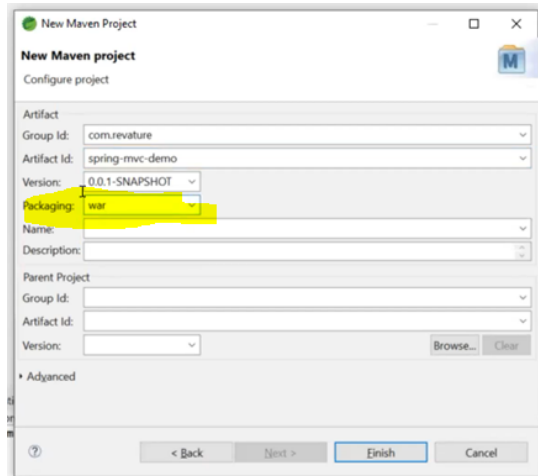
Enter the Group Id: com.revature

Enter the Artifact Id: spring-mvc-demo

Select Packaging: **war**

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

Click Finish



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

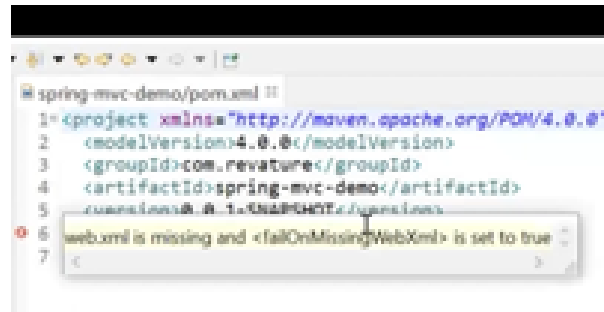
```
spring-mvc-demo/pom.xml ::
1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   <modelVersion>4.0.0</modelVersion>
3   <groupId>com.revature</groupId>
4   <artifactId>spring-mvc-demo</artifactId>
5   <version>0.0.1-SNAPSHOT</version>
6   <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

Rebuild the Maven project.

Open pom.xml

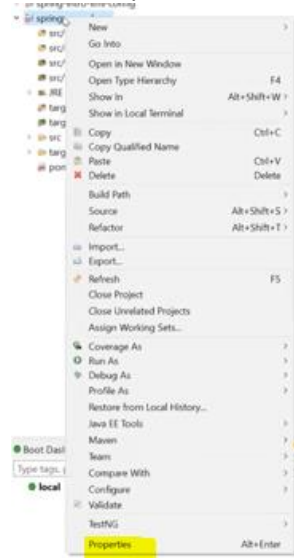
Error: Web.xml is missing



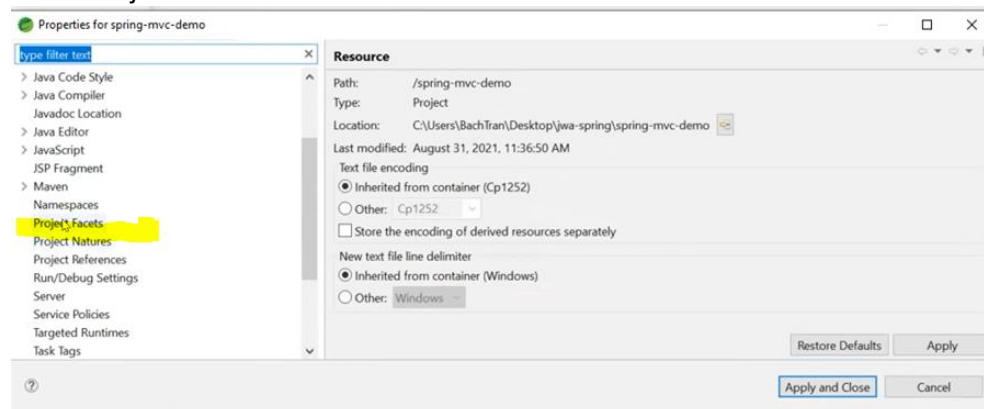
Need to generate Web.xml

Right click on project

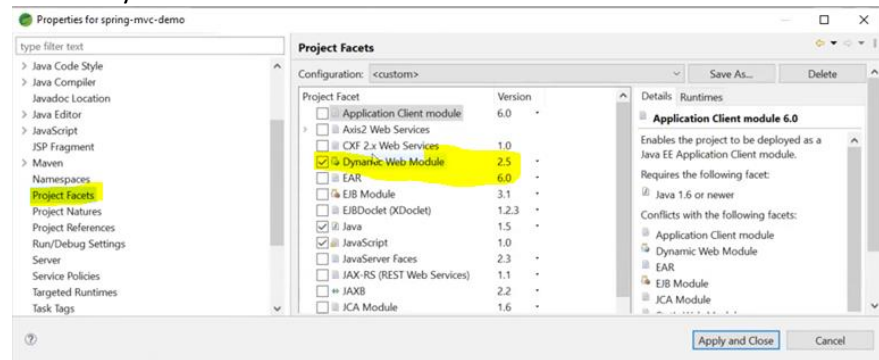
Select Properties



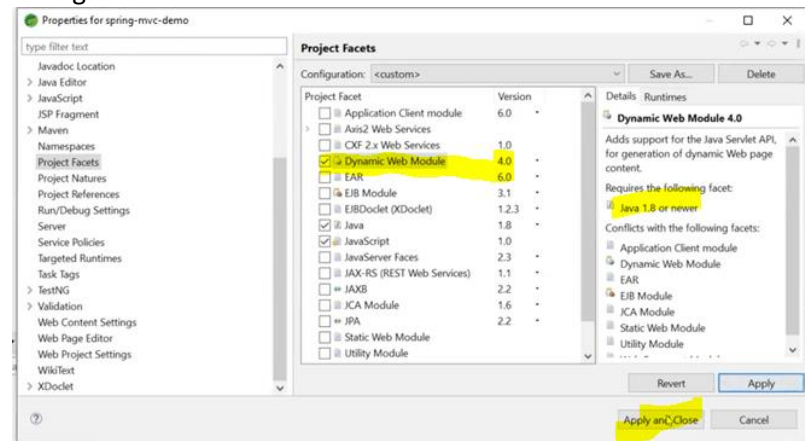
Select Project Facets



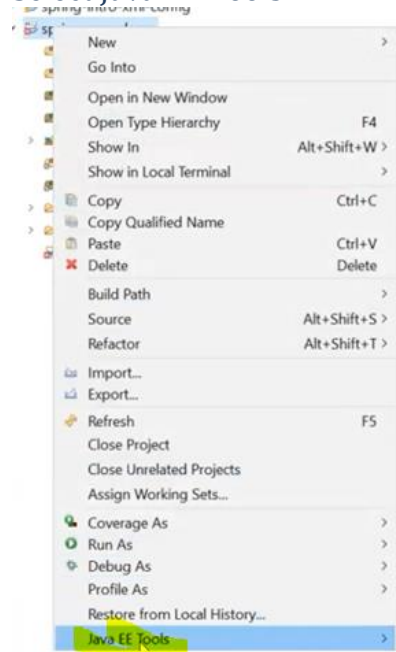
Check "Dynamic Web Module"



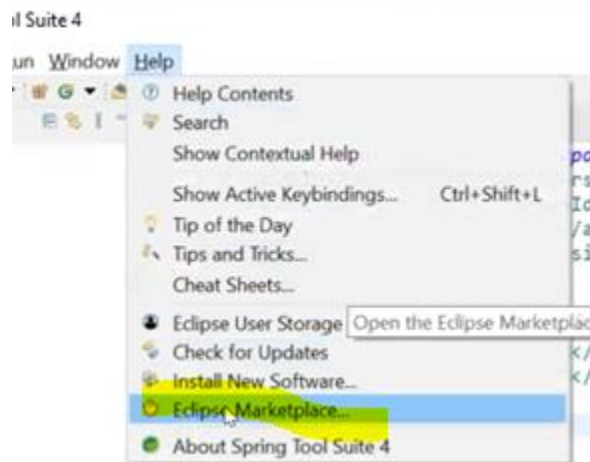
Change to new version Java 1.8 or newer version



Right click on project Select Java EE Tools



If Java EE Tools is not in the menu then:
Install via Eclipse Market Place under the Help menu



Search tab: enter enterprise
If not already installed, install the component
Once the installation completes restart the IDE for the changes to take affect

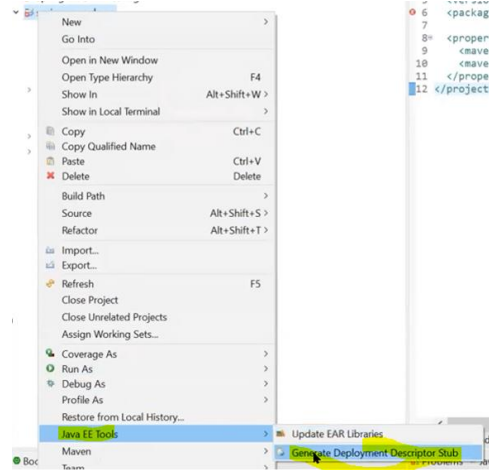


Generate Deployment Descriptor Stub

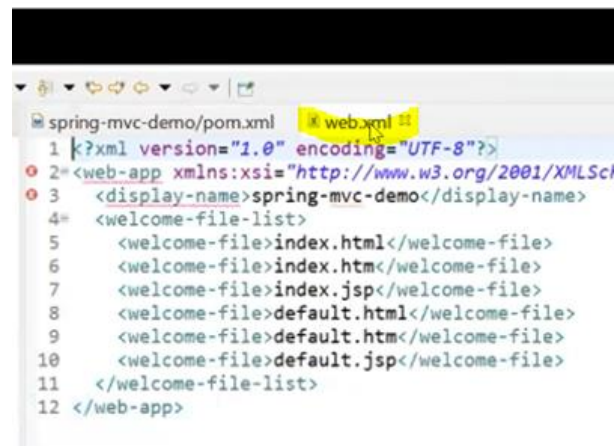
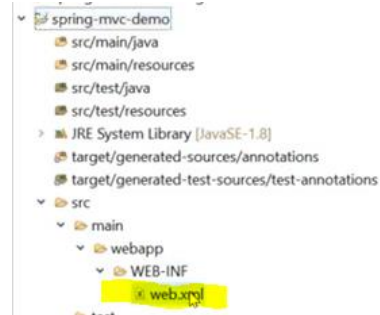
Right click on the project

Select Java EE Tools

Select Generate Deployment Descriptor Stub



Generates web.xml in following location



Web.xml File

Update outdated link:

```
<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">
```



```
spring-mvc-demo/pom.xml 1 *web.xml 2
1 encoding="UTF-8"?>
2 http://www.w3.org/2001/XMLSchema-instance xmlns="http://www.oracle.com/webfolder/technetwork/jsc/xml/n
3 ng-mvc-demo</display-name>
4 >
```

web.xml is 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>

28< <servlet-mapping>
29     <servlet-name>DispatcherServlet</servlet-name>
30     <url-pattern>/</url-pattern>
31 </servlet-mapping>
32
```

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

“Front Controller Design Pattern”

ApplicationContext.xml File

spring-mvc-demo\src\main\webapp\WEB-INF\applicationContext.xml

Add base package: com.revature

```
<?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>

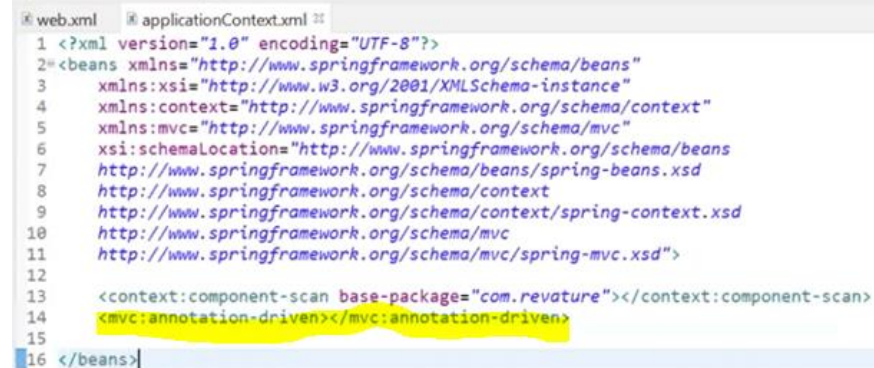
</beans>
```



```
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>
```

Add <mvc:annotation> tags

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



```
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

```
<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/xsd/maven-
4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.revature</groupId>
  <artifactId>spring-mvc-demo</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>
  </dependencies>
</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

Package: com\revature\dto



LoginDTO.java

Spring MVC Project Next Steps for Tomcat Server(s)

After completing the code follow the steps in the “Tomcat 02 IDE” 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 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 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.