

TECHNOLOGY



Caltech

Center for Technology & Management Education

Configuration for Java Maven
Backend Project

TECHNOLOGY



Caltech

Center for Technology & Management Education

Configure the Project with Dependencies

You Already Know

Before we begin, let's recall what we have covered till now:



JDBC



Servlets



JSP

Maven Project for Backend

- Created a Maven project with an archetype as a web app in Eclipse EE

Developed POJO Classes

- Created various classes for the Admin and End User Projects
- Developed POJO with constructors, getters, setters, and `toString`

DAO and Service

- Implemented the Service Layer to access DAO for DB operations

Build and Execute

- Built and executed the Maven Web App Project



A Day in the Life of a Full Stack Developer

As a full stack web developer, our key role is to develop both client and server software.



Angular and Node can be used to build front end of the web page.



Spring Boot, Java, and MySQL/MongoDB can be used to build at the back end.



A Day in the Life of a Full Stack Developer

Now, Bob needs to run his webapp on localhost. So, Bob brainstorms a bit and finds a solution.

Let me use Tomcat Server to run the web app on localhost.

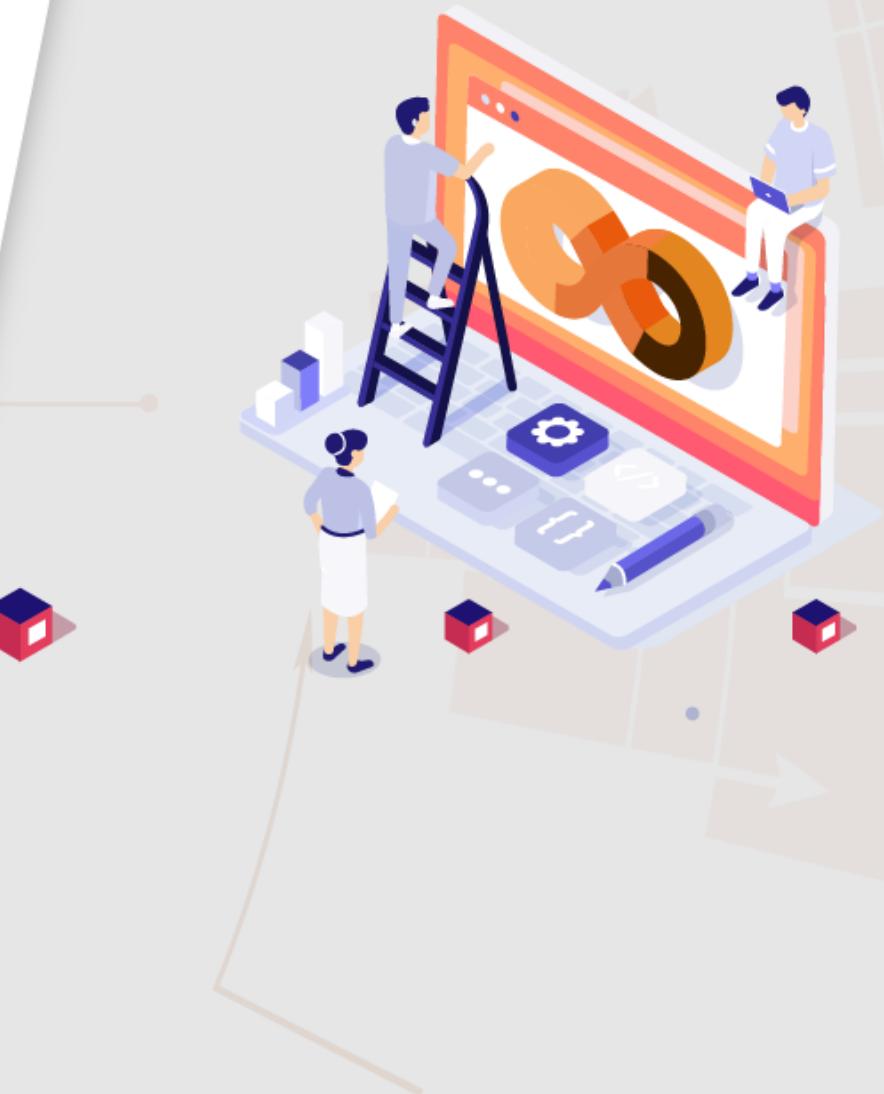


In this lesson, we will learn the skills Java and Maven to configure dependencies and the Tomcat Server to run the web app on localhost and help Bob to complete his task effectively and quickly.

Learning Objectives

By the end of this lesson, you will be able to:

- Configure Maven Project with Dependencies
- Configure Java Tomcat Server in Eclipse EE
- Develop JSP Page and Servlets with Front Controller Design Pattern
- Build the Project and Package Web Project using Maven



Configure Dev Environment for Java Backend with Maven

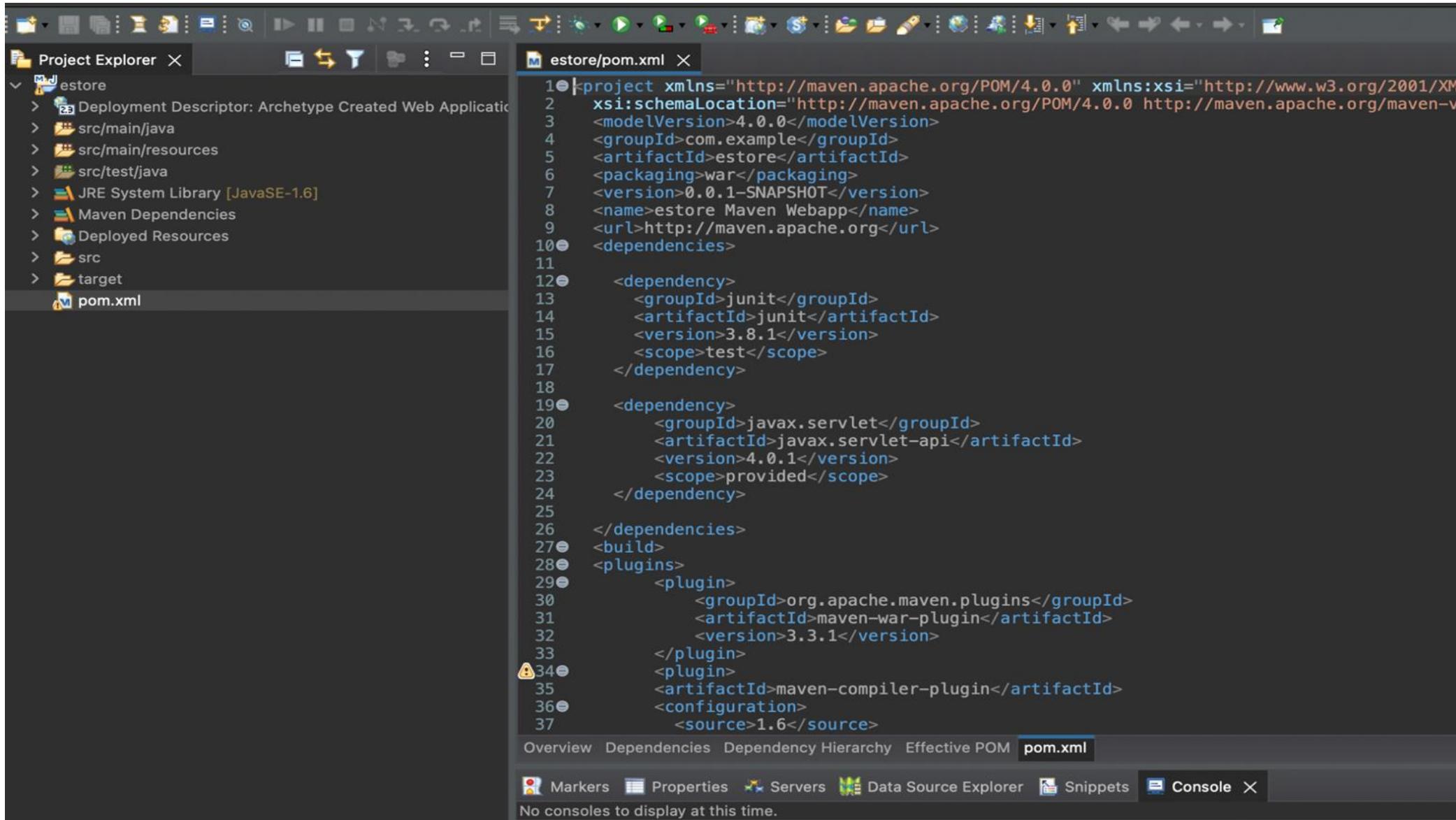
Web App for End User

Configure the pom.xml file for various dependencies and plugins for project to work properly.



Configure JDBC

Open the Eclipse EE IDE and navigate to the Maven Project.

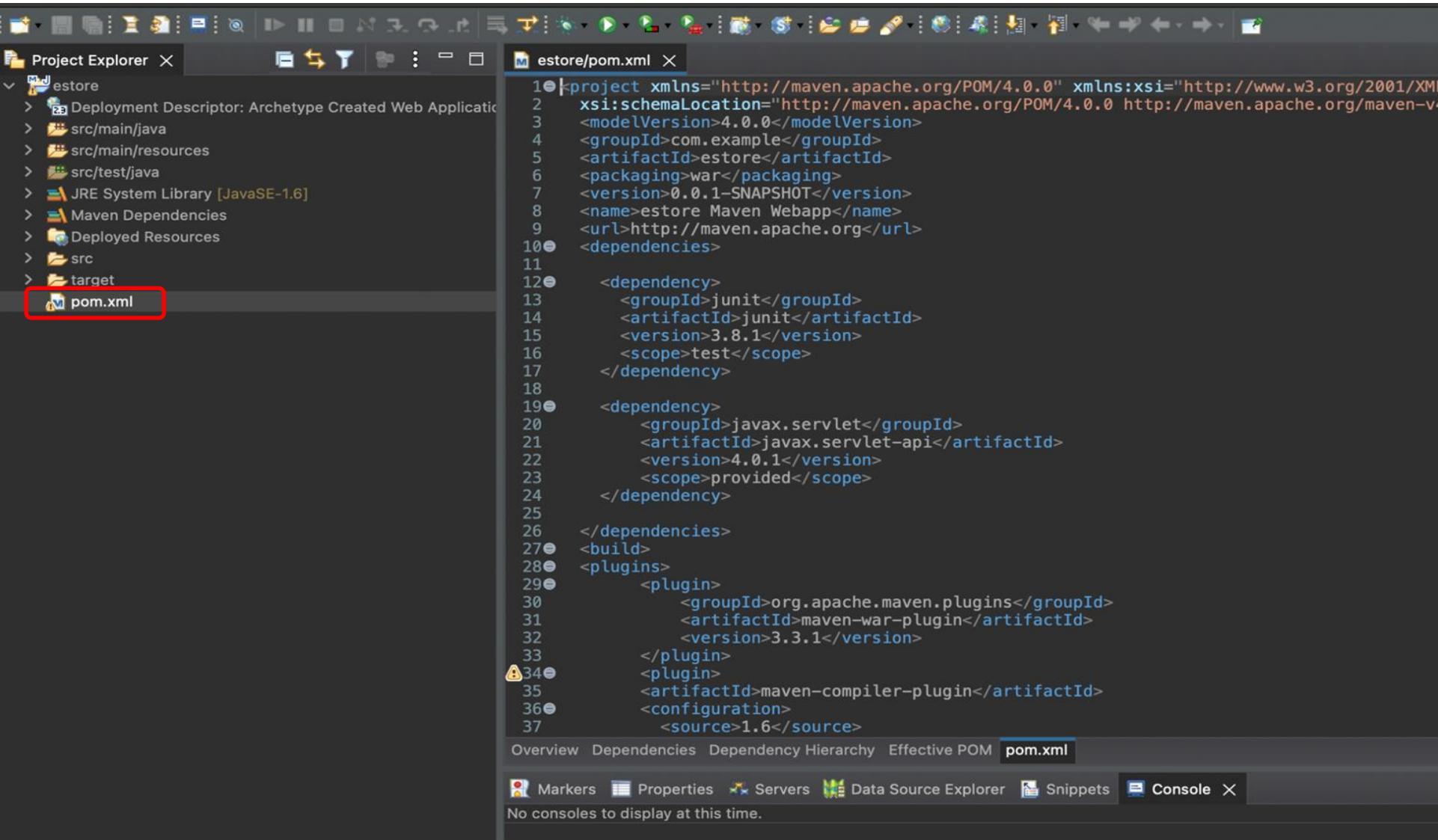


The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer view displays a Maven project named 'estore'. Inside the project, there are several source and resource folders, and a 'pom.xml' file which is currently selected and open in the central editor area. The editor shows the XML code for the project's Maven configuration. The code includes details such as the project's name ('estore Maven Webapp'), its URL ('http://maven.apache.org'), and its dependencies on JUnit and javax.servlet. It also specifies build plugins like maven-war-plugin and maven-compiler-plugin. A warning icon is visible near the bottom of the code editor, indicating a potential issue or note. The status bar at the bottom of the IDE window shows various toolbars and tabs, including 'Markers', 'Properties', 'Servers', 'Data Source Explorer', 'Snippets', and 'Console'.

```
<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>com.example</groupId>
  <artifactId>estore</artifactId>
  <packaging>war</packaging>
  <version>0.0.1-SNAPSHOT</version>
  <name>estore Maven Webapp</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>javax.servlet-api</artifactId>
      <version>4.0.1</version>
      <scope>provided</scope>
    </dependency>
  </dependencies>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-war-plugin</artifactId>
        <version>3.3.1</version>
      </plugin>
      <plugin>
        <artifactId>maven-compiler-plugin</artifactId>
        <configuration>
          <source>1.6</source>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

Configure JDBC

Open pom.xml file.

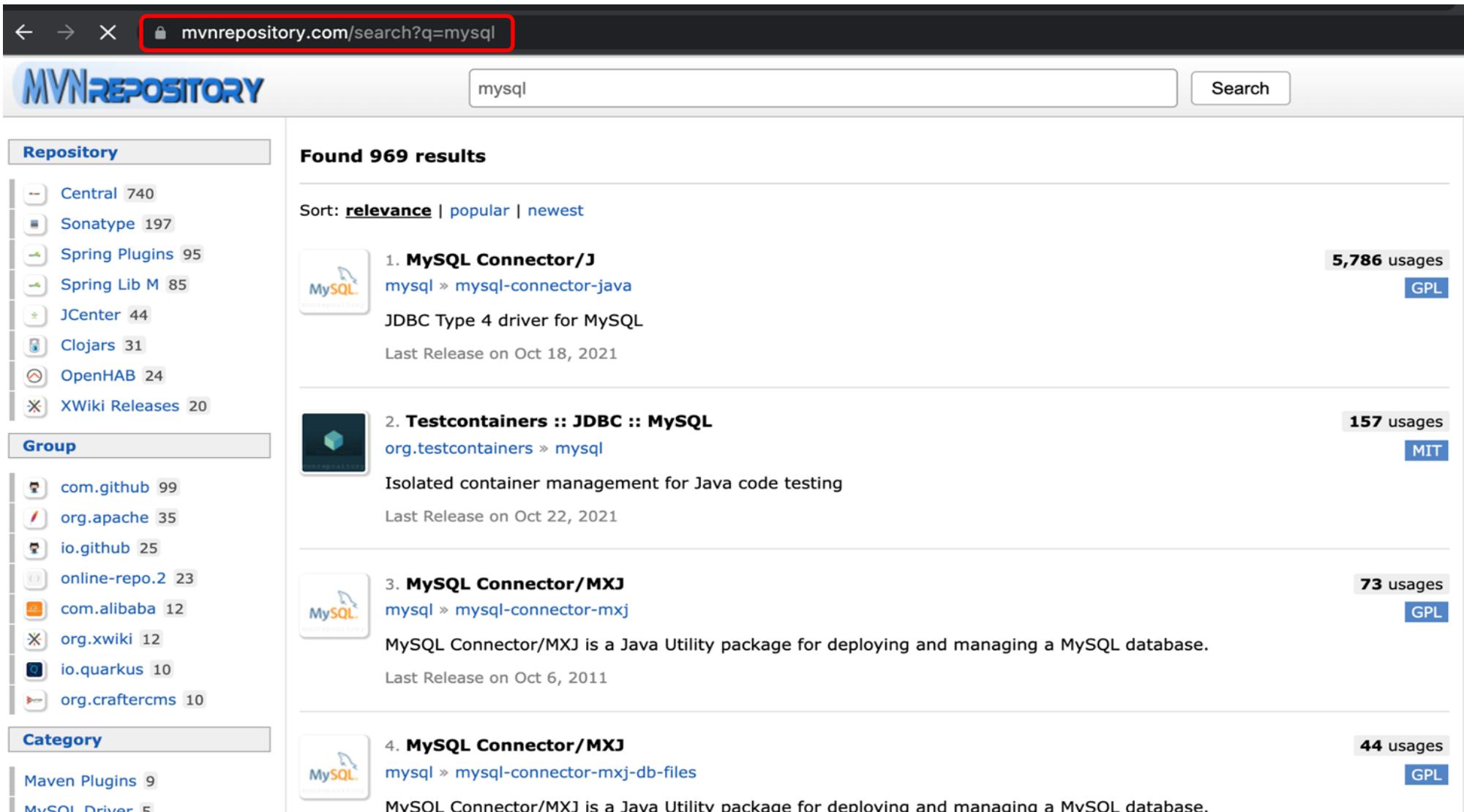


The screenshot shows the Eclipse IDE interface. The Project Explorer view on the left displays a project named 'estore' with various source and resource folders. The pom.xml file is selected and highlighted with a red border in the Project Explorer. The main editor area on the right shows the XML content of the pom.xml file. The XML code includes project metadata, dependencies (for JUnit and javax.servlet), and build plugins (maven-war-plugin and maven-compiler-plugin). A warning icon is visible near the bottom of the code area, indicating a potential issue. The status bar at the bottom shows tabs for Overview, Dependencies, Dependency Hierarchy, Effective POM, and pom.xml, with the latter being the active tab. Other tabs like Markers, Properties, Servers, Data Source Explorer, Snippets, and Console are also present in the status bar.

```
<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.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example</groupId>
  <artifactId>estore</artifactId>
  <packaging>war</packaging>
  <version>0.0.1-SNAPSHOT</version>
  <name>estore Maven Webapp</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>javax.servlet-api</artifactId>
      <version>4.0.1</version>
      <scope>provided</scope>
    </dependency>
  </dependencies>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-war-plugin</artifactId>
        <version>3.3.1</version>
      </plugin>
      <plugin>
        <artifactId>maven-compiler-plugin</artifactId>
        <configuration>
          <source>1.6</source>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

Configure JDBC

Navigate to the website mvnrepository.com and search for the MySQL.



The screenshot shows the Maven Repository search results for 'mysql'. The search bar at the top contains 'mvnrepository.com/search?q=mysql'. The results page displays 969 items. The first result is 'MySQL Connector/J' (mysql:mysql-connector-java), which is a JDBC Type 4 driver for MySQL. It has 5,786 usages, is licensed under GPL, and was last released on Oct 18, 2021. The second result is 'Testcontainers :: JDBC :: MySQL' (org.testcontainers:mysql), which is an isolated container management for Java code testing. It has 157 usages, is licensed under MIT, and was last released on Oct 22, 2021. The third result is 'MySQL Connector/MXJ' (mysql:mysql-connector-mxj), which is a Java Utility package for deploying and managing a MySQL database. It has 73 usages, is licensed under GPL, and was last released on Oct 6, 2011. The fourth result is 'MySQL Connector/MXJ' (mysql:mysql-connector-mxj-db-files), which is a Java Utility package for deploying and managing a MySQL database. It has 44 usages, is licensed under GPL, and was last released on Oct 6, 2011. The left sidebar includes filters for 'Repository' (Central, Sonatype, Spring Plugins, Spring Lib M, JCenter, Clojars, OpenHAB, XWiki Releases), 'Group' (com.github, org.apache, io.github, online-repo.2, com.alibaba, org.xwiki, io.quarkus, org.craftercms), and 'Category' (Maven Plugins, MySQL Driver).

Rank	Dependency	Group ID	Usages	Licenses
1.	MySQL Connector/J	mysql > mysql-connector-java	5,786	GPL
2.	Testcontainers :: JDBC :: MySQL	org.testcontainers > mysql	157	MIT
3.	MySQL Connector/MXJ	mysql > mysql-connector-mxj	73	GPL
4.	MySQL Connector/MXJ	mysql > mysql-connector-mxj-db-files	44	GPL

Configure JDBC

Search for the MySQL.

The screenshot shows the Maven Repository search results for 'mysql'. The search bar at the top contains 'mysql' and is highlighted with a red box. The results page displays 969 found items. The first result is 'MySQL Connector/J' by mysql, which is a JDBC Type 4 driver for MySQL. It has 5,786 usages and is under the GPL license. The second result is 'Testcontainers :: JDBC :: MySQL' by org.testcontainers, which is an isolated container management for Java code testing. It has 157 usages and is under the MIT license. The third result is 'MySQL Connector/MXJ' by mysql, which is a Java Utility package for deploying and managing a MySQL database. It has 73 usages and is under the GPL license. The fourth result is 'MySQL Connector/MXJ' by mysql, which is a Java Utility package for deploying and managing a MySQL database. It has 44 usages and is under the GPL license. The left sidebar includes sections for Repository (Central, Sonatype, Spring Plugins, Spring Lib M, JCenter, Clojars, OpenHAB, XWiki Releases), Group (com.github, org.apache, io.github, online-repo.2, com.alibaba, org.xwiki, io.quarkus, org.craftercms), and Category (Maven Plugins, MySQL Driver).

mvnrepository.com/search?q=mysql

MVNREPOSITORY

mysql

Repository

- Central 740
- Sonatype 197
- Spring Plugins 95
- Spring Lib M 85
- JCenter 44
- Clojars 31
- OpenHAB 24
- XWiki Releases 20

Group

- com.github 99
- org.apache 35
- io.github 25
- online-repo.2 23
- com.alibaba 12
- org.xwiki 12
- io.quarkus 10
- org.craftercms 10

Category

- Maven Plugins 9
- MySQL Driver 5

Found 969 results

Sort: relevance | popular | newest

1. MySQL Connector/J
mysql » mysql-connector-java
JDBC Type 4 driver for MySQL
Last Release on Oct 18, 2021

5,786 usages
GPL

2. Testcontainers :: JDBC :: MySQL
org.testcontainers » mysql
Isolated container management for Java code testing
Last Release on Oct 22, 2021

157 usages
MIT

3. MySQL Connector/MXJ
mysql » mysql-connector-mxj
MySQL Connector/MXJ is a Java Utility package for deploying and managing a MySQL database.
Last Release on Oct 6, 2011

73 usages
GPL

4. MySQL Connector/MXJ
mysql » mysql-connector-mxj-db-files
MySQL Connector/MXJ is a Java Utility package for deploying and managing a MySQL database.

44 usages
GPL

Configure JDBC

In the results, we will get MySQL JDBC Connector as the first result.

The screenshot shows the Maven Repository search results for the query 'mysql'. The search bar at the top has 'mysql' entered. On the left, there are three navigation panels: 'Repository' (Central, Sonatype, Spring Plugins, Spring Lib M, JCenter, Clojars, OpenHAB, XWiki Releases), 'Group' (com.github, org.apache, io.github, online-repo.2, com.alibaba, org.xwiki, io.quarkus, org.craftercms), and 'Category' (Maven Plugins, MySQL Driver). The main area displays 'Found 969 results' sorted by relevance. The first result is highlighted with a red box:

- 1. MySQL Connector/J**
mysql > mysql-connector-java
JDBC Type 4 driver for MySQL
Last Release on Oct 18, 2021
- 2. Testcontainers :: JDBC :: MySQL**
org.testcontainers > mysql
Isolated container management for Java code testing
Last Release on Oct 22, 2021
- 3. MySQL Connector/MXJ**
mysql > mysql-connector-mxj
MySQL Connector/MXJ is a Java Utility package for deploying and managing a MySQL database.
Last Release on Oct 6, 2011
- 4. MySQL Connector/MXJ**
mysql > mysql-connector-mxj-db-files
MySQL Connector/MXJ is a Java Utility package for deploying and managing a MySQL database.

Each result entry includes usage statistics (e.g., 5,786 usages for MySQL Connector/J) and a license indicator (e.g., GPL for MySQL Connector/J).

Configure JDBC

Click on the same link and open the MySQL Connector Page.

The screenshot shows the Maven Repository website at mvnrepository.com/artifact/mysql/mysql-connector-java/8.0.27. The page displays information for the MySQL Connector/J artifact, specifically version 8.0.27. Key details include:

- MySQL Connector/J > 8.0.27**: The main title, highlighted with a red border.
- License**: GPL 2.0
- Categories**: MySQL Drivers
- Organization**: Oracle Corporation
- HomePage**: <http://dev.mysql.com/doc/connector-j/en/>
- Date**: (Oct 18, 2021)
- Files**: pom (2 KB) | jar (2.4 MB) | View All
- Repositories**: Central
- Used By**: 5,786 artifacts
- Vulnerabilities**: Vulnerabilities from dependencies: CVE-2021-22569

On the left sidebar, there is a chart titled "Indexed Artifacts (25.2M)" showing the growth of indexed projects from 2006 to 2018, and a list of "Popular Categories" including Aspect Oriented, Actor Frameworks, Application Metrics, Build Tools, Bytecode Libraries, Command Line Parsers, Cache Implementations, Cloud Computing, Code Analyzers, Collections, Configuration Libraries, Core Utilities, Date and Time Utilities, Dependency Injection, and Embedded SQL Databases.

At the bottom of the page, there is a code snippet for Maven dependency configuration:

```
<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.27</version>
</dependency>
```

Configure JDBC

Paste the configuration in pom.xml file.

The screenshot shows the Maven Repository website at mvnrepository.com/artifact/mysql/mysql-connector-java/8.0.27. The page displays information for the MySQL Connector/J version 8.0.27, including its license (GPL 2.0), categories (MySQL Drivers), organization (Oracle Corporation), homepage (<http://dev.mysql.com/doc/connector-j/en/>), and a warning about vulnerabilities (CVE-2021-22569). Below the details, there are tabs for Maven, Gradle, Gradle (Short), Gradle (Kotlin), SBT, Ivy, Gape, Leiningen, and Buildr. A red box highlights the Maven tab, and a blue callout bubble with a red arrow points to the dependency configuration code in the Maven tab's content area.

MySQL Connector/J » 8.0.27
JDBC Type 4 driver for MySQL

License	GPL 2.0
Categories	MySQL Drivers
Organization	Oracle Corporation
HomePage	http://dev.mysql.com/doc/connector-j/en/
Date	(Oct 18, 2021)
Files	pom (2 KB) jar (2.4 MB) View All
Repositories	Central
Used By	5,786 artifacts
Vulnerabilities	Vulnerabilities from dependencies: CVE-2021-22569

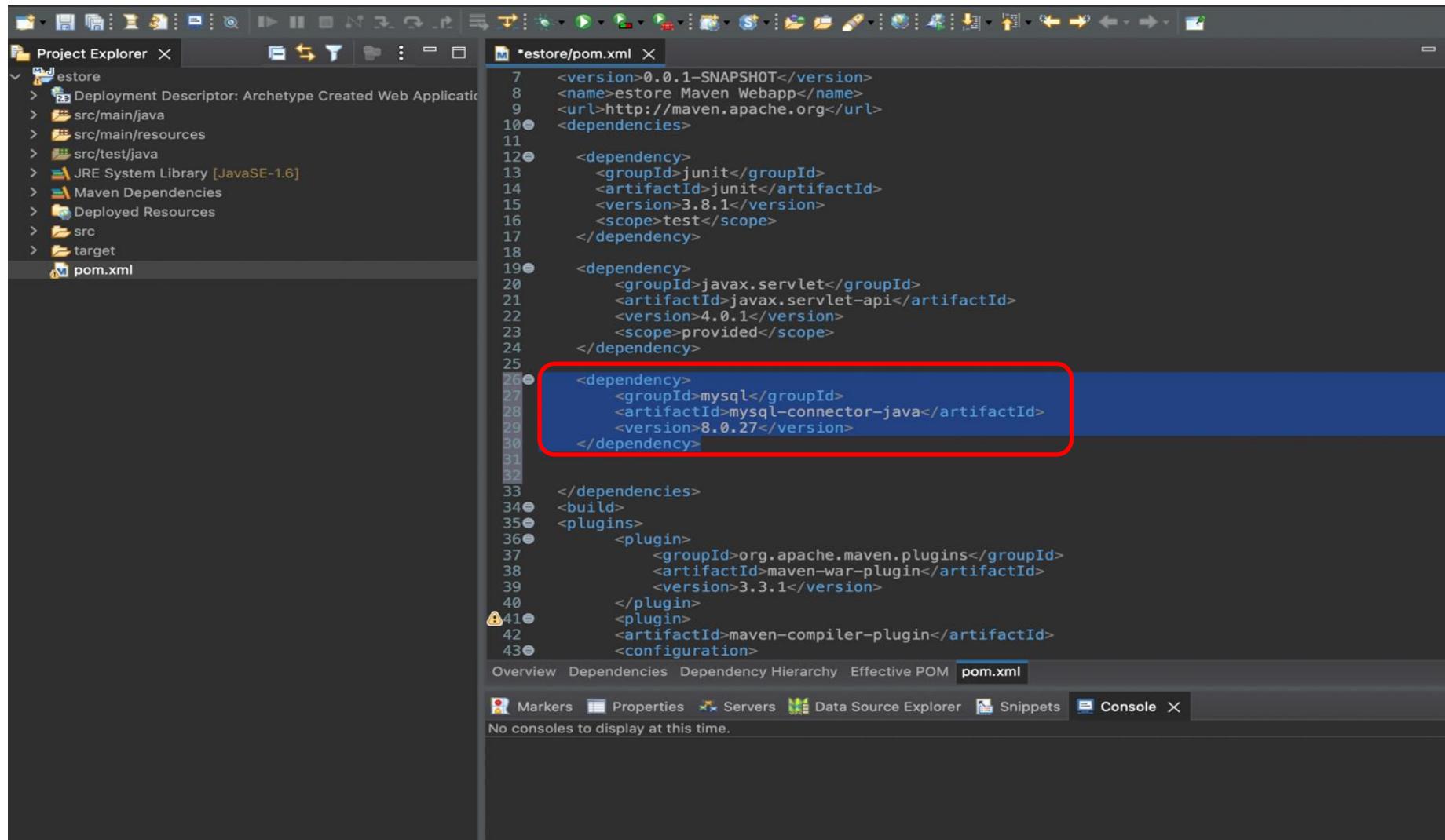
Maven Gradle Gradle (Short) Gradle (Kotlin) SBT Ivy Gape Leiningen Buildr

```
<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.27</version>
</dependency>
```

Copy the dependency configuration.

Configure JDBC

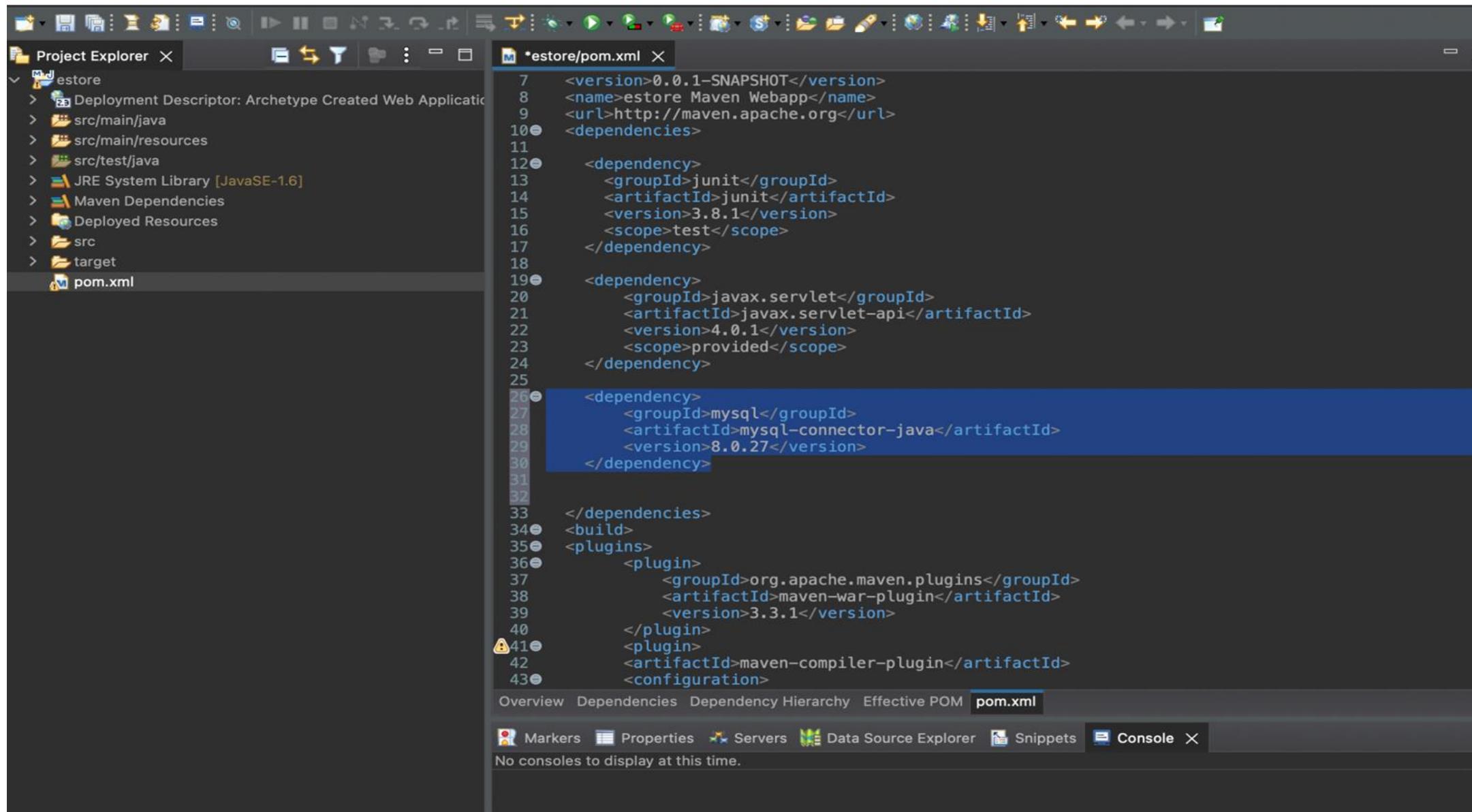
Paste the copied dependency from mvnrepository.com in the pom.xml file under the dependencies.



```
<version>0.0.1-SNAPSHOT</version>
<name>estore Maven Webapp</name>
<url>http://maven.apache.org</url>
<dependencies>
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>3.8.1</version>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>javax.servlet</groupId>
        <artifactId>javax.servlet-api</artifactId>
        <version>4.0.1</version>
        <scope>provided</scope>
    </dependency>
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>8.0.27</version>
    </dependency>
</dependencies>
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-war-plugin</artifactId>
            <version>3.3.1</version>
        </plugin>
        <plugin>
            <artifactId>maven-compiler-plugin</artifactId>
            <configuration>
```

Configure JDBC

Save pom.xml, and JDBC MySQL Library shall be synced to project.



```
7   <version>0.0.1-SNAPSHOT</version>
8   <name>estore Maven Webapp</name>
9   <url>http://maven.apache.org</url>
10  <dependencies>
11    <dependency>
12      <groupId>junit</groupId>
13      <artifactId>junit</artifactId>
14      <version>3.8.1</version>
15      <scope>test</scope>
16    </dependency>
17    <dependency>
18      <groupId>javax.servlet</groupId>
19      <artifactId>javax.servlet-api</artifactId>
20      <version>4.0.1</version>
21      <scope>provided</scope>
22    </dependency>
23    <dependency>
24      <groupId>mysql</groupId>
25      <artifactId>mysql-connector-java</artifactId>
26      <version>8.0.27</version>
27    </dependency>
28
29  </dependencies>
30  <build>
31    <plugins>
32      <plugin>
33        <groupId>org.apache.maven.plugins</groupId>
34        <artifactId>maven-war-plugin</artifactId>
35        <version>3.3.1</version>
36      </plugin>
37      <plugin>
38        <artifactId>maven-compiler-plugin</artifactId>
39        <configuration>
40
41  </configuration>
42  </plugin>
43</plugins>
44</build>
45</project>
```

Configure Servlets

Configure Servlet

Navigate back to the website mvnrepository.com.

The screenshot shows the Maven Repository search results for the query 'servlet'. The search bar at the top contains 'mvnrepository.com/search?q=servlet'. The search results page has a sidebar on the left with sections for 'Repository' and 'Group'. The 'Repository' section lists Central (1.9k), Sonatype (706), Spring Plugins (539), Spring Lib M (533), JBoss Releases (200), Geomajas (115), IBiblio (101), and Spring Lib Release (79). The 'Group' section lists org.mobicens (283), org.apache (180), com.github (90), org.jboss (81), org.eclipse (54), org.glassfish (47), com.aoapps (38), and org.wso2 (32). The main content area displays 'Found 2616 results' and lists three items:

- 1. Java Servlet API**
javax.servlet » javax.servlet-api
Java Servlet API
Last Release on Apr 20, 2018
15,497 usages
GPL | CDDL
- 2. JavaServlet(TM) Specification**
javax.servlet » servlet-api
JavaServlet(TM) Specification
Last Release on Apr 17, 2008
11,424 usages
GPL | GPL | CDDL
- 3. Jetty :: Servlet Handling**
org.eclipse.jetty » jetty-servlet
Jetty Servlet Container
Last Release on Oct 12, 2021
2,190 usages
EPL | Apache

Configure Servlet

Search for the servlets.

The screenshot shows the Maven Repository search results for the query 'servlet'. The search bar at the top contains the text 'servlet'. The results page displays three main entries:

- 1. Java Servlet API**: javax.servlet > javax.servlet-api. It has 15,497 usages. License: GPL | CDDL. Last Release on Apr 20, 2018.
- 2. JavaServlet(TM) Specification**: javax.servlet > servlet-api. It has 11,424 usages. License: GPL | CDDL. Last Release on Apr 17, 2008.
- 3. Jetty :: Servlet Handling**: org.eclipse.jetty > jetty-servlet. It has 2,190 usages. License: EPL | Apache. Last Release on Oct 12, 2021.

The left sidebar includes sections for 'Repository' (Central, Sonatype, Spring Plugins, Spring Lib M, JBoss Releases, Geomajas, IBiblio, Spring Lib Release) and 'Group' (org.mobicens, org.apache, com.github, org.jboss, org.eclipse, org.glassfish, com.aoapps, org.wso2).

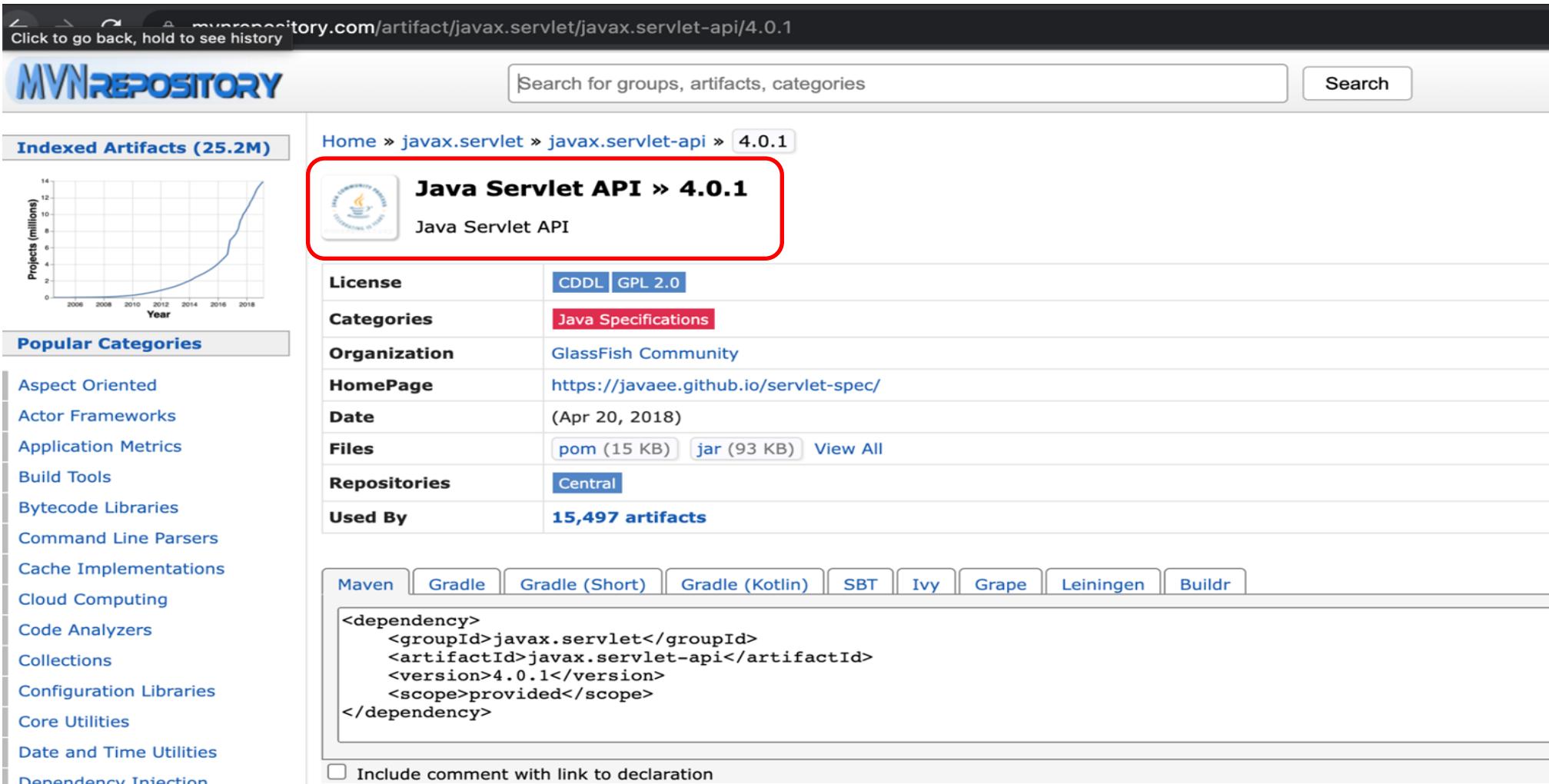
Configure Servlet

In the results, we get Servlet API.

The screenshot shows the Maven Repository search results for the query 'servlet'. The search bar at the top contains 'servlet'. The results section is titled 'Found 2616 results' and is sorted by relevance. The first result is '1. Java Servlet API' (javax.servlet > javax.servlet-api), which has 15,497 usages and is licensed under GPL | CDDL. This result is highlighted with a red box. The second result is '2. JavaServlet(TM) Specification' (javax.servlet > servlet-api), which has 11,424 usages and is licensed under GPL | GPL | CDDL. The third result is '3. Jetty :: Servlet Handling' (org.eclipse.jetty > jetty-servlet), which has 2,190 usages and is licensed under EPL | Apache. The left sidebar shows repository filters like Central, Sonatype, Spring Plugins, etc., and group filters like org.mobicens, org.apache, com.github, etc.

Configure Servlet

Click on the same link and open the Servlet API Page.



The screenshot shows the Maven Repository website at mvnrepository.com/artifact/javax.servlet/javax.servlet-api/4.0.1. The page displays information for the Java Servlet API version 4.0.1. A red box highlights the artifact title "Java Servlet API > 4.0.1".

Java Servlet API > 4.0.1
Java Servlet API

License	CDDL GPL 2.0
Categories	Java Specifications
Organization	GlassFish Community
HomePage	https://javaee.github.io/servlet-spec/
Date	(Apr 20, 2018)
Files	pom (15 KB) jar (93 KB) View All
Repositories	Central
Used By	15,497 artifacts

Maven | Gradle | Gradle (Short) | Gradle (Kotlin) | SBT | Ivy | Gape | Leiningen | Buildr

```
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>4.0.1</version>
    <scope>provided</scope>
</dependency>
```

Include comment with link to declaration

Configure Servlet

Paste the configuration in pom.xml file.

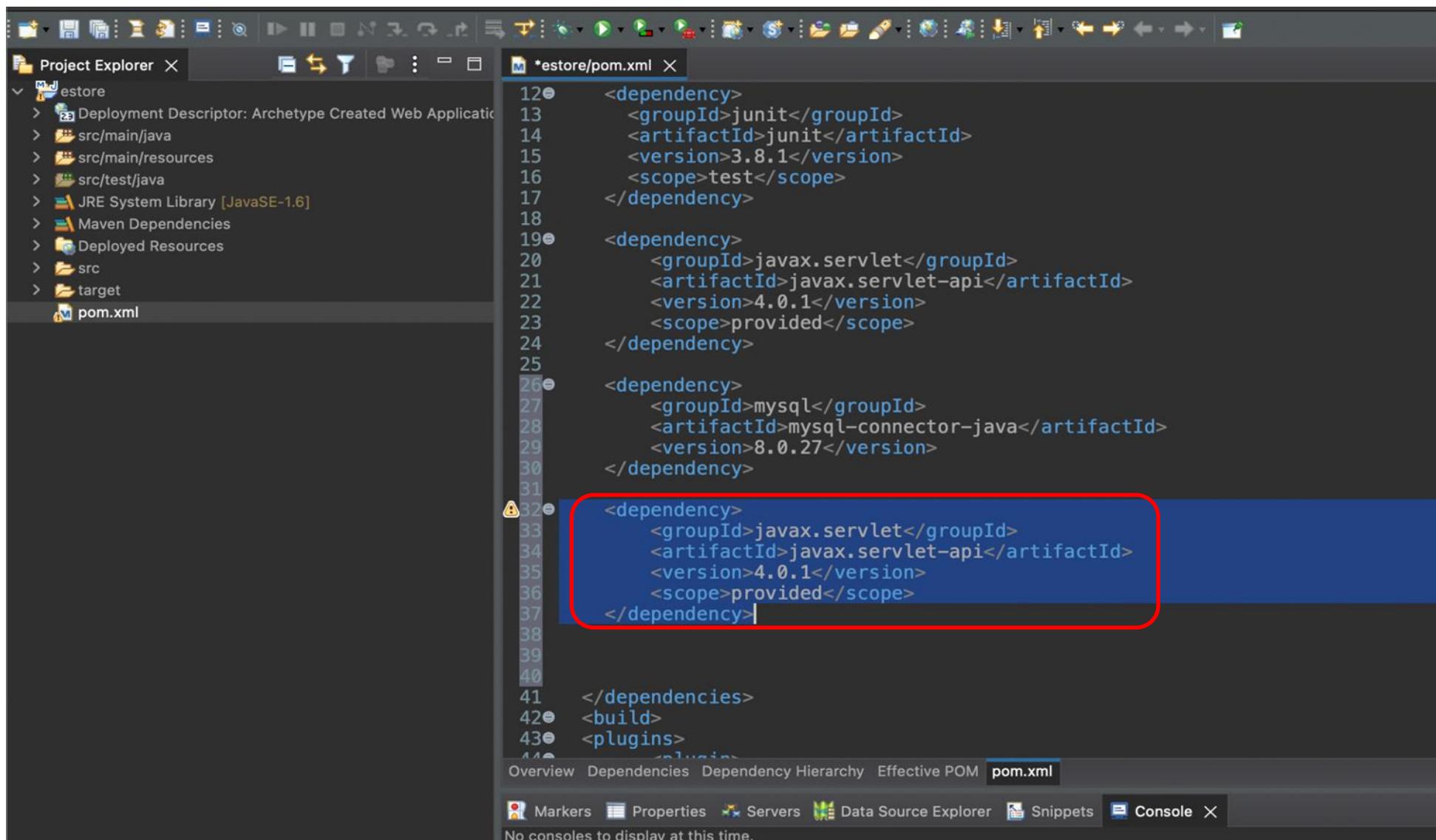
The screenshot shows the Maven Repository page for the Java Servlet API version 4.0.1. The URL in the address bar is `mavenrepository.com/artifact/javax.servlet/javax.servlet-api/4.0.1`. The page includes a search bar and a sidebar with popular categories like Aspect Oriented, Actor Frameworks, and Build Tools. The main content area displays the Java Servlet API's license (CDDL, GPL 2.0), categories (Java Specifications), organization (GlassFish Community), homepage (<https://javaee.github.io/servlet-spec/>), date (Apr 20, 2018), files (pom 15 KB, jar 93 KB, View All), repositories (Central), and usage statistics (15,497 artifacts). Below this, there are tabs for Maven, Gradle, Gradle (Short), Gradle (Kotlin), SBT, Ivy, Gape, Leiningen, and Buildr. A red box highlights the Maven dependency code:

```
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>4.0.1</version>
    <scope>provided</scope>
</dependency>
```

A blue callout bubble with the text "Copy the dependency configuration." points to this highlighted code.

Configure Servlet

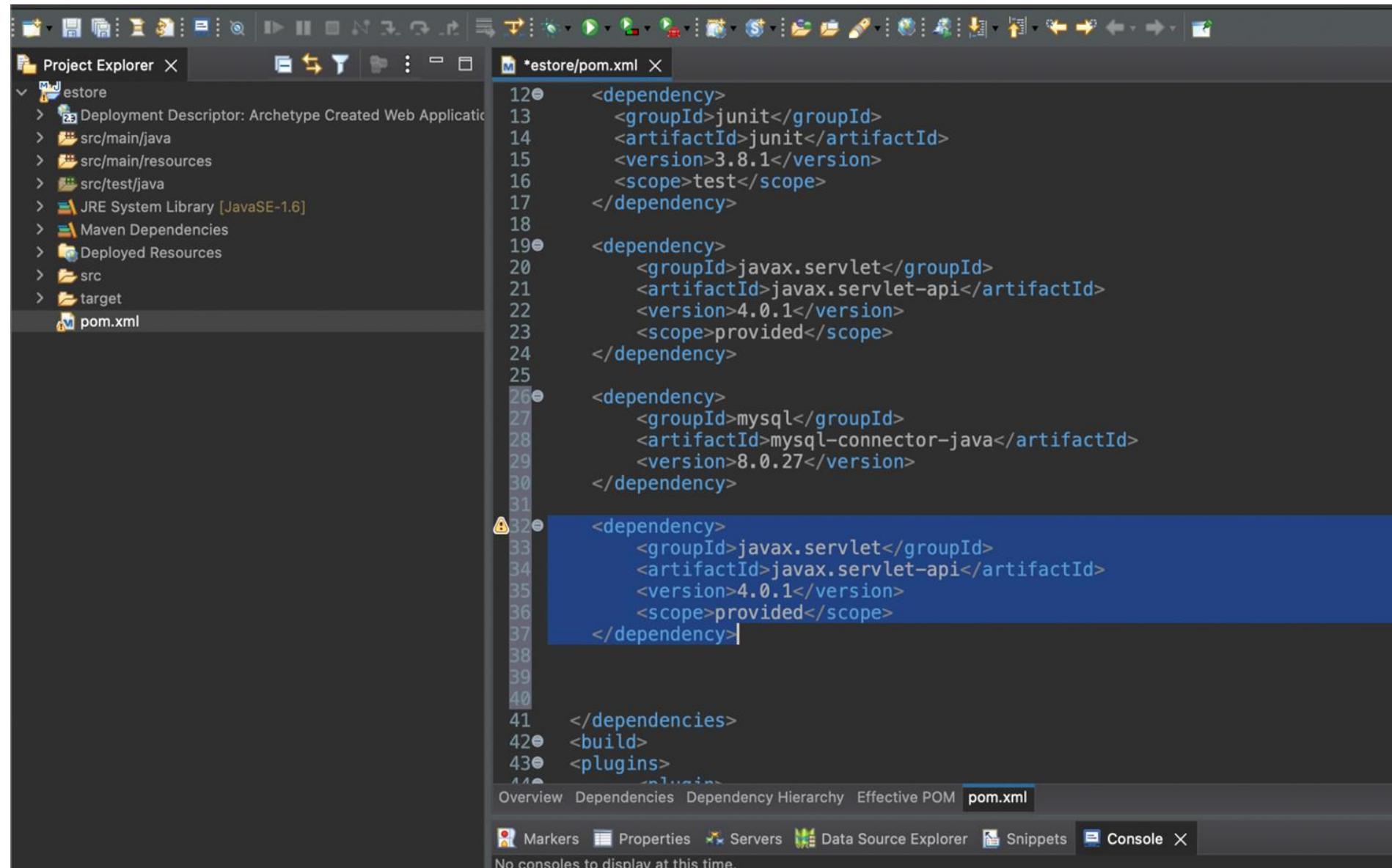
Paste the copied dependency from mvnrepository.com.



```
12<dependency>
13    <groupId>junit</groupId>
14    <artifactId>junit</artifactId>
15    <version>3.8.1</version>
16    <scope>test</scope>
17</dependency>
18<dependency>
19    <groupId>javax.servlet</groupId>
20    <artifactId>javax.servlet-api</artifactId>
21    <version>4.0.1</version>
22    <scope>provided</scope>
23</dependency>
24<dependency>
25    <groupId>mysql</groupId>
26    <artifactId>mysql-connector-java</artifactId>
27    <version>8.0.27</version>
28</dependency>
29<dependency>
30    <groupId>javax.servlet</groupId>
31    <artifactId>javax.servlet-api</artifactId>
32    <version>4.0.1</version>
33    <scope>provided</scope>
34</dependency>
35<dependency>
36    <groupId>javax.servlet</groupId>
37    <artifactId>javax.servlet-api</artifactId>
38    <version>4.0.1</version>
39    <scope>provided</scope>
40</dependency>
41</dependencies>
42<build>
43<plugins>
```

Configure Servlet

Save the pom.xml, and Servlet Library shall be synced to project.



```
12    <dependency>
13        <groupId>junit</groupId>
14        <artifactId>junit</artifactId>
15        <version>3.8.1</version>
16        <scope>test</scope>
17    </dependency>
18
19    <dependency>
20        <groupId>javax.servlet</groupId>
21        <artifactId>javax.servlet-api</artifactId>
22        <version>4.0.1</version>
23        <scope>provided</scope>
24    </dependency>
25
26    <dependency>
27        <groupId>mysql</groupId>
28        <artifactId>mysql-connector-java</artifactId>
29        <version>8.0.27</version>
30    </dependency>
31
32    <dependency>
33        <groupId>javax.servlet</groupId>
34        <artifactId>javax.servlet-api</artifactId>
35        <version>4.0.1</version>
36        <scope>provided</scope>
37    </dependency>
38
39
40
41    </dependencies>
42    <build>
43        <plugins>
44            <plugin>
```

Configure JSP

Configure JSP

Navigate back to the website mvnrepository.com.

The screenshot shows the Maven Repository search results for the query 'jsp'. The search bar at the top contains 'mvnrepository.com/search?q=jsp'. The results page displays 767 items. The first result is 'JavaServer Pages(TM) API' from javax.servlet.jsp, with 859 usages, released on Aug 1, 2018. The second result is also 'JavaServer Pages(TM) API' from javax.servlet.jsp, with 1,464 usages, released on Jan 10, 2011. The third result is 'JSP API' from javax.servlet, with 858 usages, released on Nov 8, 2005. The fourth result is 'Jetty :: Glassfish JSP Implementation' from org.eclipse.jetty, with 216 usages, released on an unspecified date. The left sidebar includes sections for Repository (Central, Sonatype, Spring Plugins, Spring Lib M, Grails Core, IBiblio, Apache Releases, JCenter), Group (org.apache, taglibs, org.eclipse, com.github, tagLibs, com.liferay, javax.servlet, org.mortbay), and Category (Web App).

Rank	Artifact	Description	Usages	Licenses
1.	JavaServer Pages(TM) API	javax.servlet.jsp > javax.servlet.jsp-api	859 usages	GPL CDDL
2.	JavaServer Pages(TM) API	javax.servlet.jsp > jsp-api	1,464 usages	GPL GPL CDDL
3.	JSP API	javax.servlet > jsp-api	858 usages	GPL CDDL
4.	Jetty :: Glassfish JSP Implementation	org.eclipse.jetty > jetty-jsp	216 usages	EPL Apache

Configure JSP

Search for the JSP.

The screenshot shows the mvnrepository.com search interface with the query 'jsp' entered in the search bar. A red box highlights the search bar. The results page displays 767 found items, sorted by relevance. The first result is 'JavaServer Pages(TM) API' with 859 usages, licensed under GPL | CDDL. The second result is also 'JavaServer Pages(TM) API' with 1,464 usages, licensed under GPL | GPL | CDDL. The third result is 'JSP API' with 858 usages, licensed under GPL | CDDL. The fourth result is 'Jetty :: Glassfish JSP Implementation' with 216 usages, licensed under EPL | Apache. On the left sidebar, there are sections for Repository (Central, Sonatype, Spring Plugins, Spring Lib M, Grails Core, IBiblio, Apache Releases, JCenter), Group (org.apache, taglibs, org.eclipse, com.github, tagLibs, com.liferay, javax.servlet, org.mortbay), and Category (Web App).

← → C mvnrepository.com/search?q=jsp

MVNREPOSITORY

Repository

- Central 546
- Sonatype 146
- Spring Plugins 140
- Spring Lib M 136
- Grails Core 48
- IBiblio 34
- Apache Releases 26
- JCenter 26

Group

- org.apache 78
- taglibs 58
- org.eclipse 46
- com.github 45
- tagLibs 42
- com.liferay 21
- javax.servlet 18
- org.mortbay 18

Category

- Web App 60

Found 767 results

Sort: [relevance](#) | [popular](#) | [newest](#)

1. **JavaServer Pages(TM) API**
javax.servlet.jsp » javax.servlet.jsp-api
JavaServer Pages(TM) API
Last Release on Aug 1, 2018
859 usages
GPL | CDDL

2. **JavaServer Pages(TM) API**
javax.servlet.jsp » jsp-api
JavaServer Pages(TM) API
Last Release on Jan 10, 2011
1,464 usages
GPL | GPL | CDDL

3. **JSP API**
javax.servlet » jsp-api
JSP API
Last Release on Nov 8, 2005
858 usages
GPL | CDDL

4. **Jetty :: Glassfish JSP Implementation**
org.eclipse.jetty » jetty-jsp
216 usages
EPL | Apache

Configure JSP

We get JSP as result.

The screenshot shows the Maven Repository search results for the query 'jsp'. The search bar at the top contains 'mvnrepository.com/search?q=jsp' and the search term 'jsp'. The results are sorted by relevance, showing 767 results. The first result is highlighted with a red box:

- 1. JavaServer Pages(TM) API**
javax.servlet.jsp > javax.servlet.jsp-api
JavaServer Pages(TM) API
Last Release on Aug 1, 2018
- 2. JavaServer Pages(TM) API**
javax.servlet.jsp > jsp-api
JavaServer Pages(TM) API
Last Release on Jan 10, 2011
- 3. JSP API**
javax.servlet > jsp-api
JSP API
Last Release on Nov 8, 2005
- 4. Jetty :: Glassfish JSP Implementation**
org.eclipse.jetty > jetty-jsp
Jetty

On the left sidebar, there are three sections: 'Repository' (Central 546, Sonatype 146, Spring Plugins 140, Spring Lib M 136, Grails Core 48, IBiblio 34, Apache Releases 26, JCenter 26), 'Group' (org.apache 78, taglibs 58, org.eclipse 46, com.github 45, tagLibs 42, com.liferay 21, javax.servlet 18, org.mortbay 18), and 'Category' (Web App 60).

Configure JSP

Click on the same link and open the Servlet API Page.

The screenshot shows the Maven Repository website at mvnrepository.com/artifact/javax.servlet.jsp/javax.servlet.jsp-api/2.3.3. The page displays information for the JavaServer Pages(TM) API version 2.3.3. A red box highlights the breadcrumb navigation: Home > javax.servlet.jsp > javax.servlet.jsp-api > 2.3.3. Below the breadcrumb, the title "JavaServer Pages(TM) API > 2.3.3" is shown. The page includes a sidebar with "Indexed Artifacts (25.2M)" and a line graph showing the number of projects over time. The main content area lists details such as License (CDDL | GPL 2.0), Categories (Java Specifications), Organization (Oracle), HomePage (<https://javaee.github.io/javaee-jsp-api>), Date (Aug 01, 2018), Files (jar (52 KB) | View All), Repositories (Central | ImageJ Public | OneBusAway Pub), and Used By (859 artifacts). At the bottom, there are links for various build tools (Maven, Gradle, etc.) and a snippet of the dependency XML code:

```
<dependency>
    <groupId>javax.servlet.jsp</groupId>
    <artifactId>javax.servlet.jsp-api</artifactId>
    <version>2.3.3</version>
    <scope>provided</scope>
</dependency>
```

Configure JSP

Paste the configuration in pom.xml file.

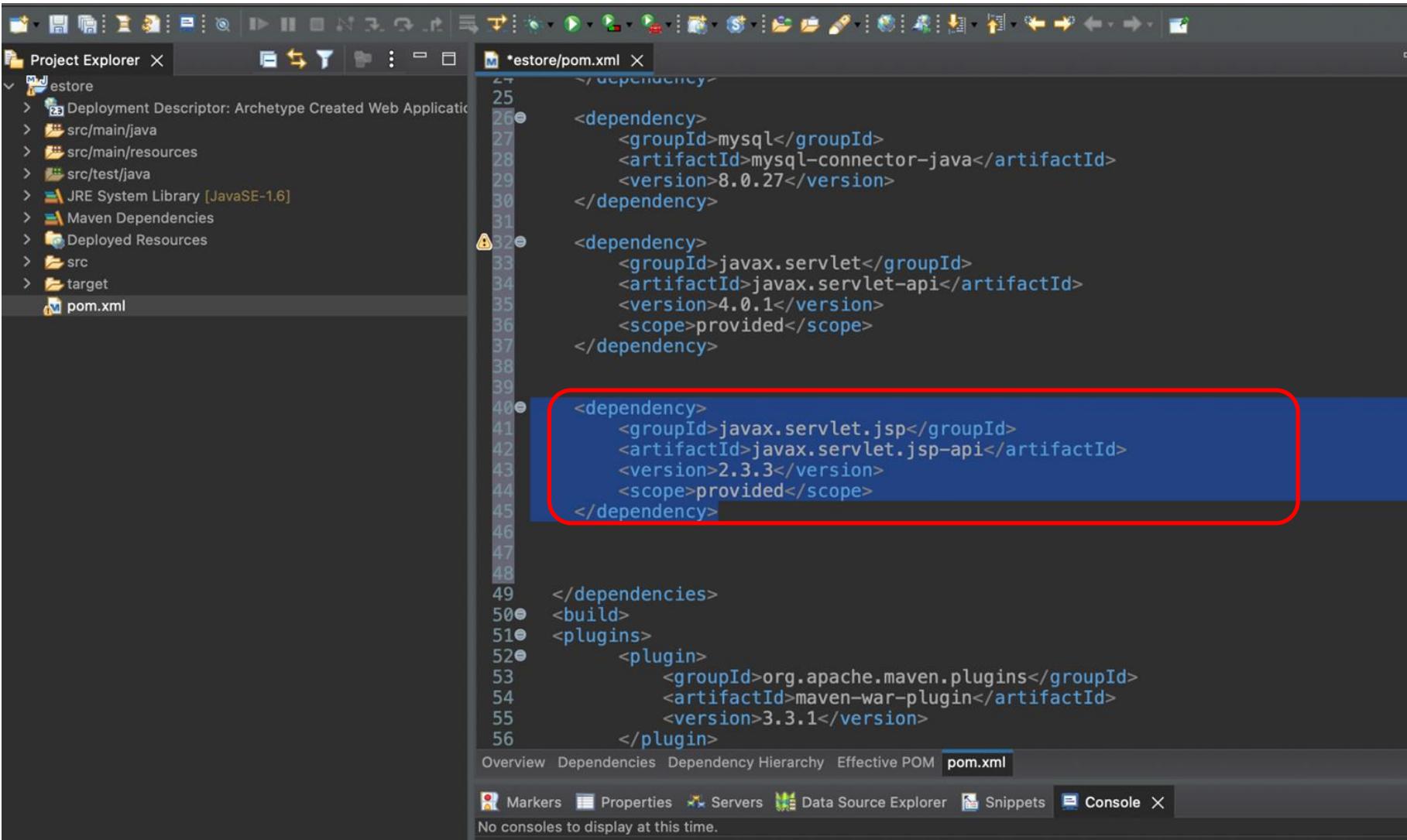
The screenshot shows the Maven Repository website at mvnrepository.com/artifact/javax.servlet.jsp/javax.servlet.jsp-api/2.3.3. The page title is "JavaServer Pages(TM) API » 2.3.3". The dependency code highlighted by a red box is:

```
<dependency>
    <groupId>javax.servlet.jsp</groupId>
    <artifactId>javax.servlet.jsp-api</artifactId>
    <version>2.3.3</version>
    <scope>provided</scope>
</dependency>
```

A blue callout bubble on the right side of the highlighted code contains the text "Copy the dependency configuration.".

Configure JSP

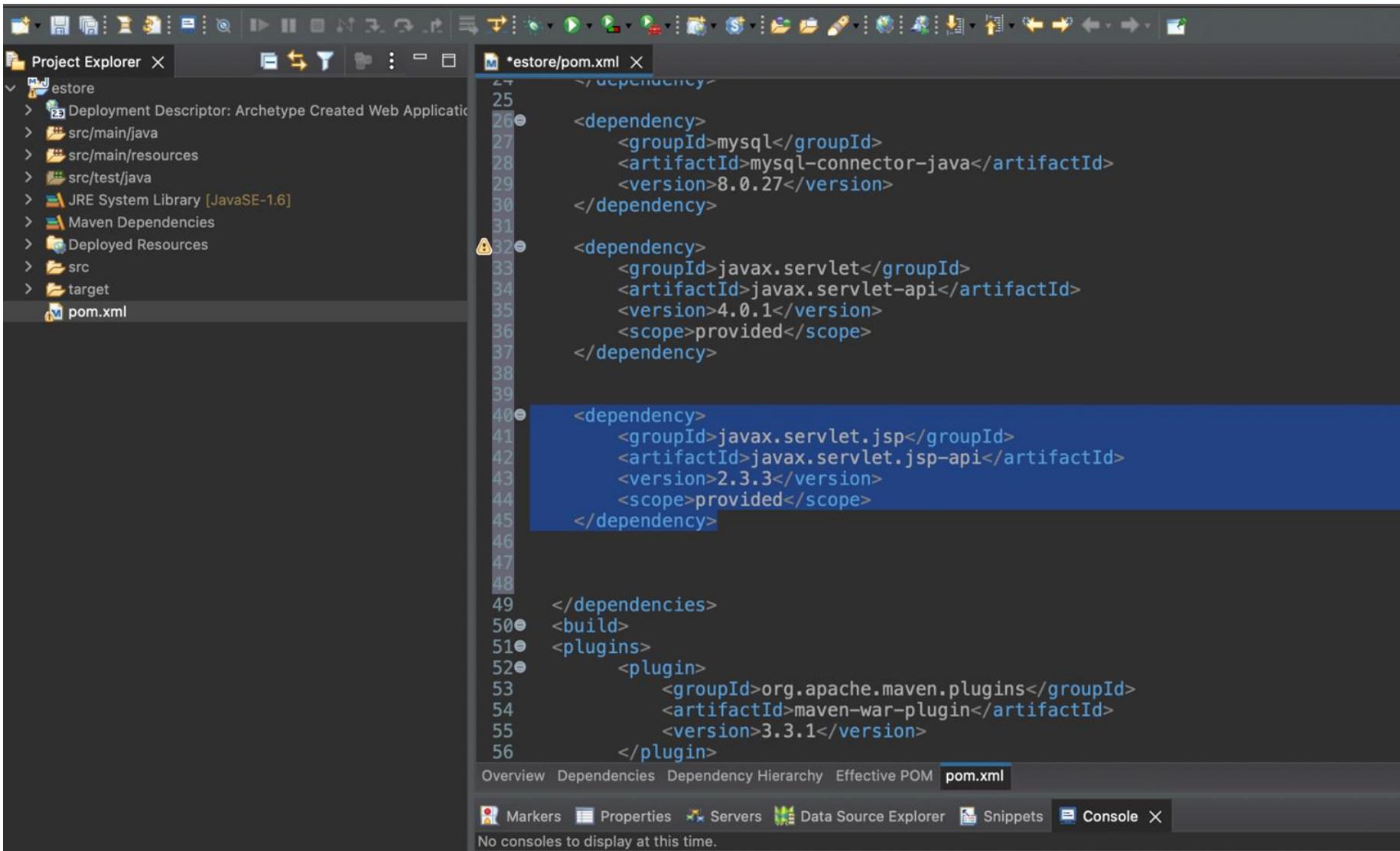
Paste the copied dependency from mvnrepository.com.



```
<dependency>
    <groupId>javax.servlet.jsp</groupId>
    <artifactId>javax.servlet.jsp-api</artifactId>
    <version>2.3.3</version>
    <scope>provided</scope>
</dependency>
```

Configure JSP

Save the pom.xml, and JSP Library shall be synced to project.



```
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.27</version>
</dependency>

<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>4.0.1</version>
    <scope>provided</scope>
</dependency>

<dependency>
    <groupId>javax.servlet.jsp</groupId>
    <artifactId>javax.servlet.jsp-api</artifactId>
    <version>2.3.3</version>
    <scope>provided</scope>
</dependency>

</dependencies>
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-war-plugin</artifactId>
            <version>3.3.1</version>
        </plugin>
    </plugins>
</build>

```

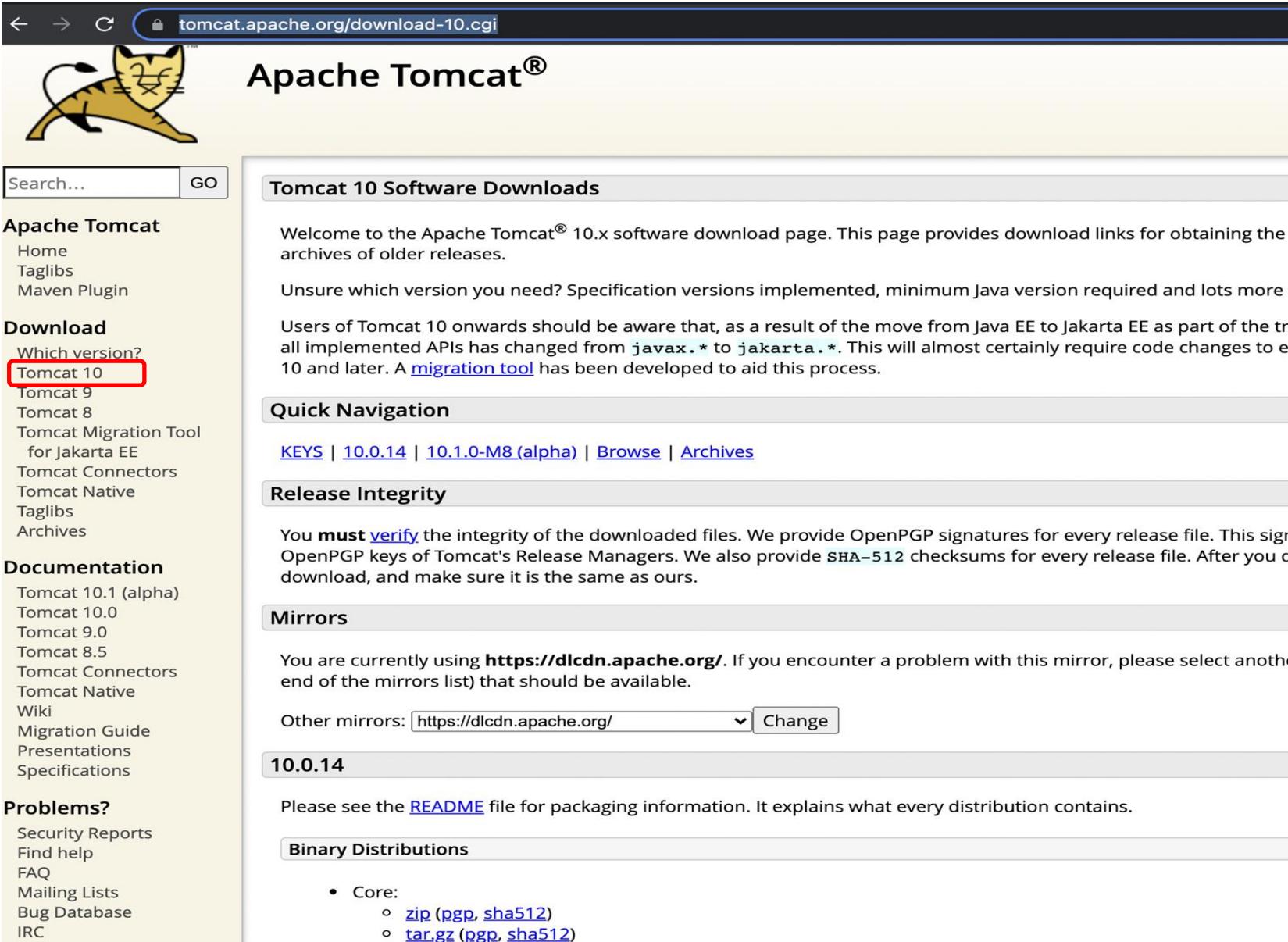
Configure Apache Tomcat

Configure Tomcat

Navigate back to the website tomcat.apache.org/download-10.cgi

Configure Tomcat

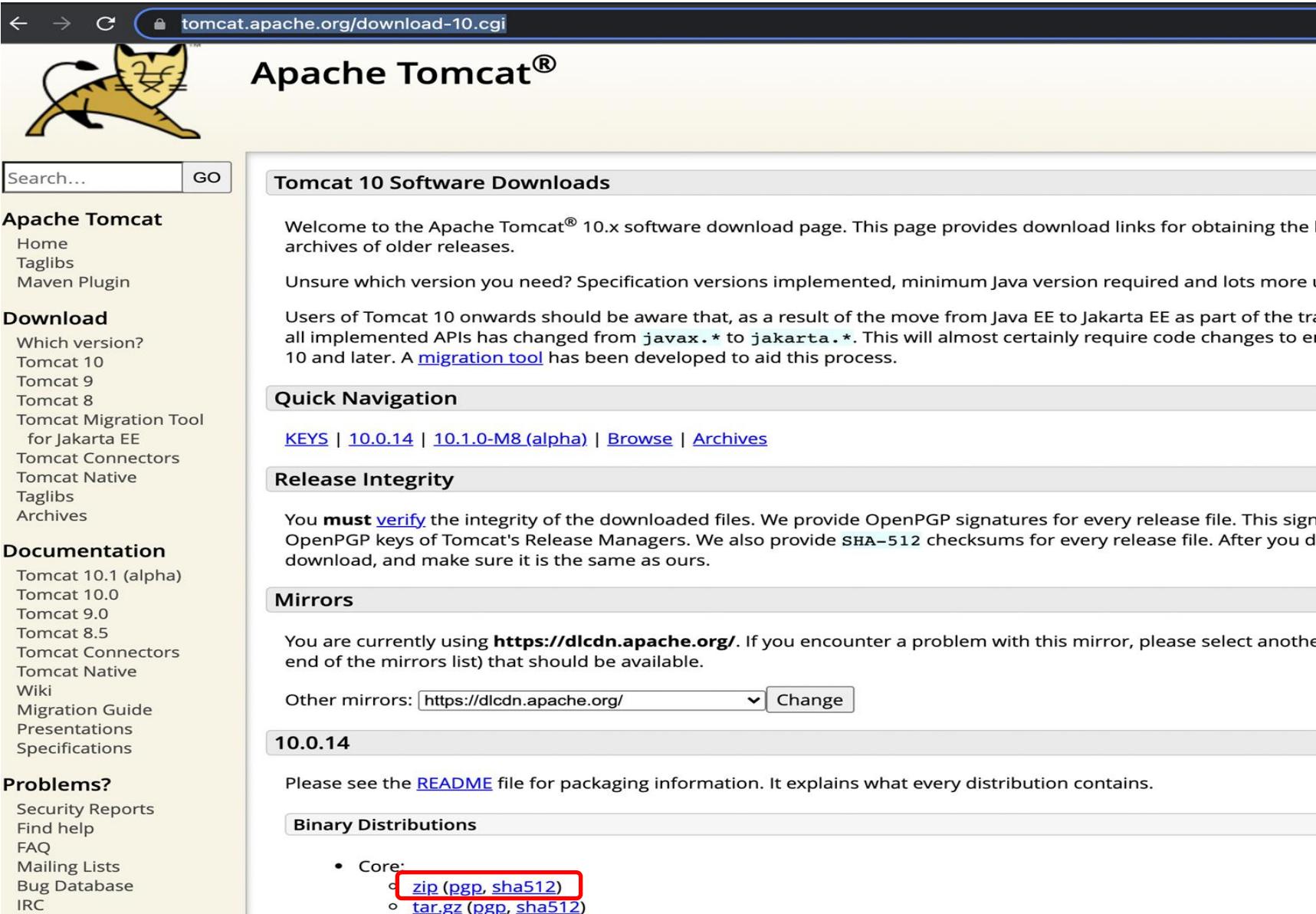
Download the latest Tomcat Server Version 10.



The screenshot shows the Apache Tomcat download page at tomcat.apache.org/download-10.cgi. The page features a logo of a yellow cat and a search bar. On the left, there's a sidebar with links for Apache Tomcat, Download (with 'Tomcat 10' selected), Documentation, and Problems?. The main content area is titled 'Tomcat 10 Software Downloads' and includes sections for 'Quick Navigation' (KEYS, 10.0.14, 10.1.0-M8 (alpha), Browse, Archives), 'Release Integrity' (instructions to verify files using OpenPGP signatures and SHA-512 checksums), 'Mirrors' (a dropdown menu set to https://dlcdn.apache.org/), and '10.0.14' (instructions to see the README file for packaging information). At the bottom, there's a 'Binary Distributions' section with a list including 'Core' items: zip (pgp, sha512) and tar.gz (pgp, sha512).

Configure Tomcat

Download the zip version of the server.

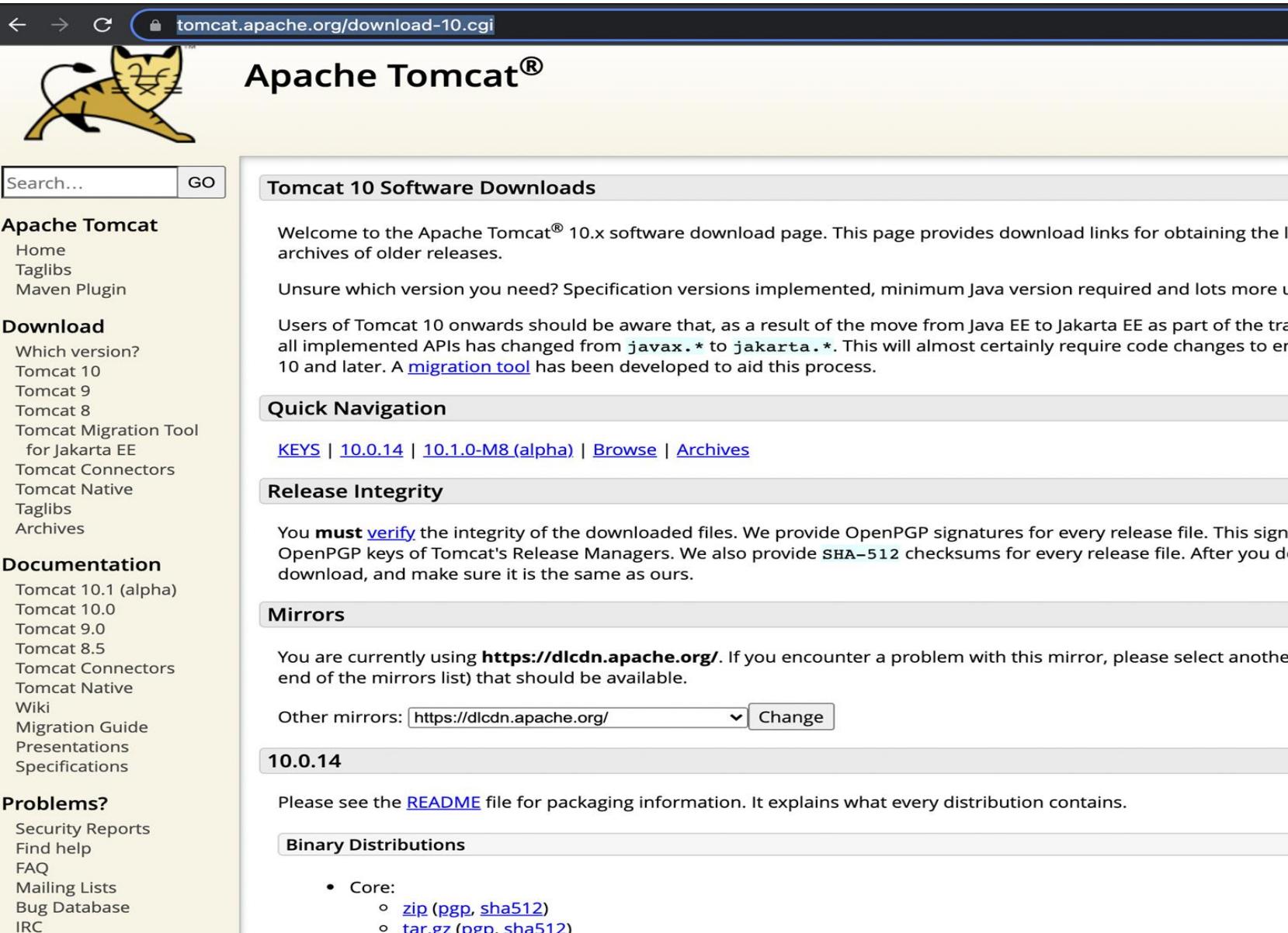


The screenshot shows the Apache Tomcat 10 Software Downloads page at tomcat.apache.org/download-10.cgi. The page features a logo of a yellow cat and a search bar. The main content area is titled "Tomcat 10 Software Downloads". It includes a welcome message, navigation links for "Quick Navigation" (KEYS, 10.0.14, 10.1.0-M8 (alpha), Browse, Archives), and "Release Integrity" instructions. A "Mirrors" section shows the current mirror as <https://dlcdn.apache.org/>. The "10.0.14" section contains a note about the README file and a "Binary Distributions" table. The "Core" distribution row has two options: "zip (pgp, sha512)" and "tar.gz (pgp, sha512)", where "zip (pgp, sha512)" is highlighted with a red box.

Distribution	File Type	PGP Signature	SHA-512 Hash
Core	zip	(pgp)	sha512
Core	tar.gz	(pgp)	sha512

Configure Tomcat

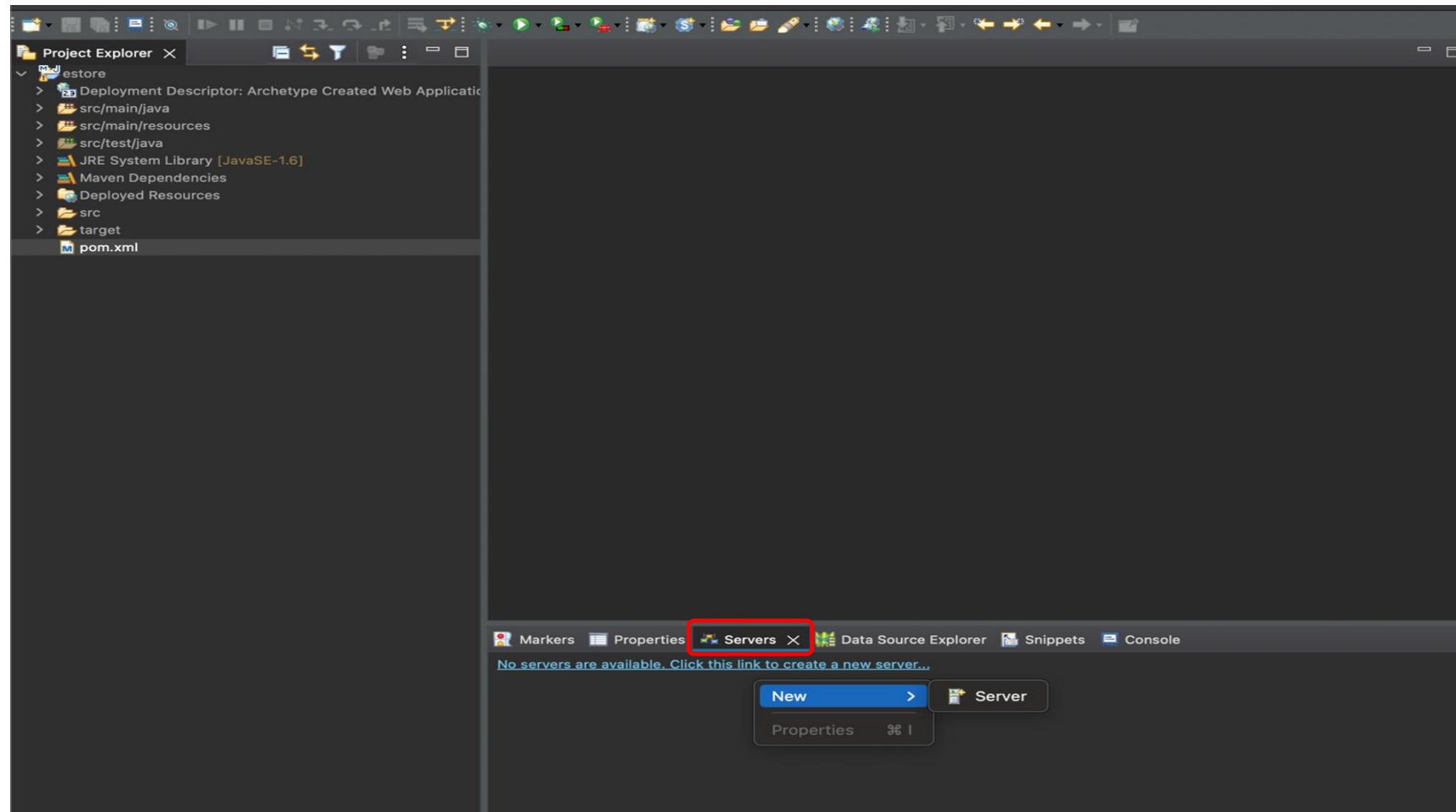
Unzip the downloaded file at some of the location on computer.



The screenshot shows the Apache Tomcat 10 Software Downloads page. At the top left is the Apache logo (a cartoon cat). The title "Apache Tomcat®" is centered above a search bar with fields "Search..." and "GO". On the left, a sidebar contains links for "Apache Tomcat" (Home, Taglibs, Maven Plugin), "Download" (Which version?, Tomcat 10, 9, 8, Tomcat Migration Tool for Jakarta EE, Connectors, Native, Taglibs, Archives), "Documentation" (Tomcat 10.1 (alpha), 10.0, 9.0, 8.5, Connectors, Native, Wiki, Migration Guide, Presentations, Specifications), and "Problems?" (Security Reports, Find help, FAQ, Mailing Lists, Bug Database, IRC). The main content area is titled "Tomcat 10 Software Downloads" and includes sections for "Welcome" (mentioning the move from Java EE to Jakarta EE), "Quick Navigation" (links to KEYS, 10.0.14, 10.1.0-M8 (alpha), Browse, Archives), "Release Integrity" (instructions for verifying files using OpenPGP signatures and SHA-512 checksums), "Mirrors" (a dropdown menu set to https://dlcdn.apache.org/ with a "Change" button), "10.0.14" (instructions to see the README file for packaging information), and "Binary Distributions" (a list including "Core": zip (pgp, sha512) and tar.gz (pgp, sha512)).

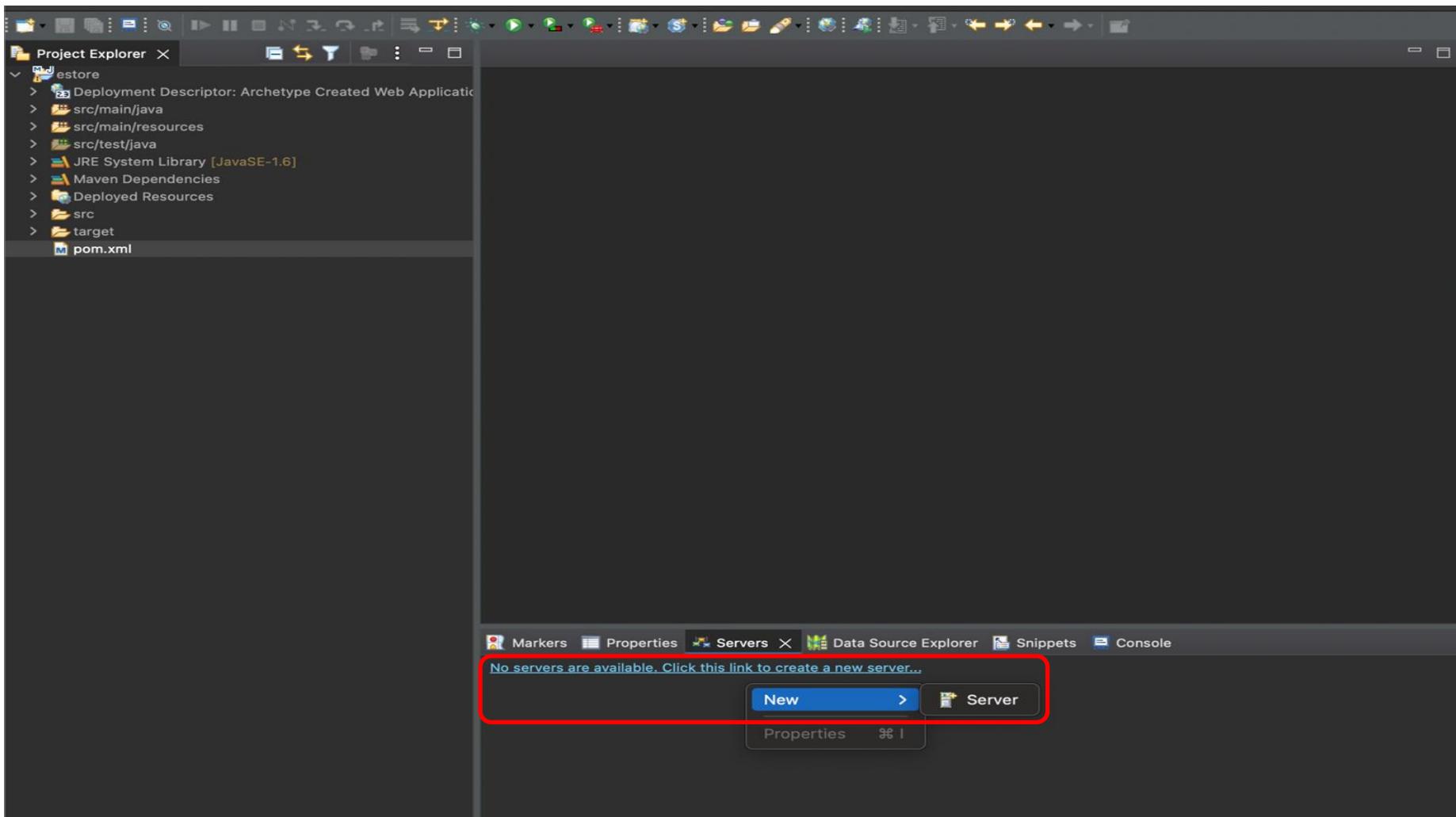
Configure Tomcat

Open Eclipse EE IDE and Servers Tab as given below.



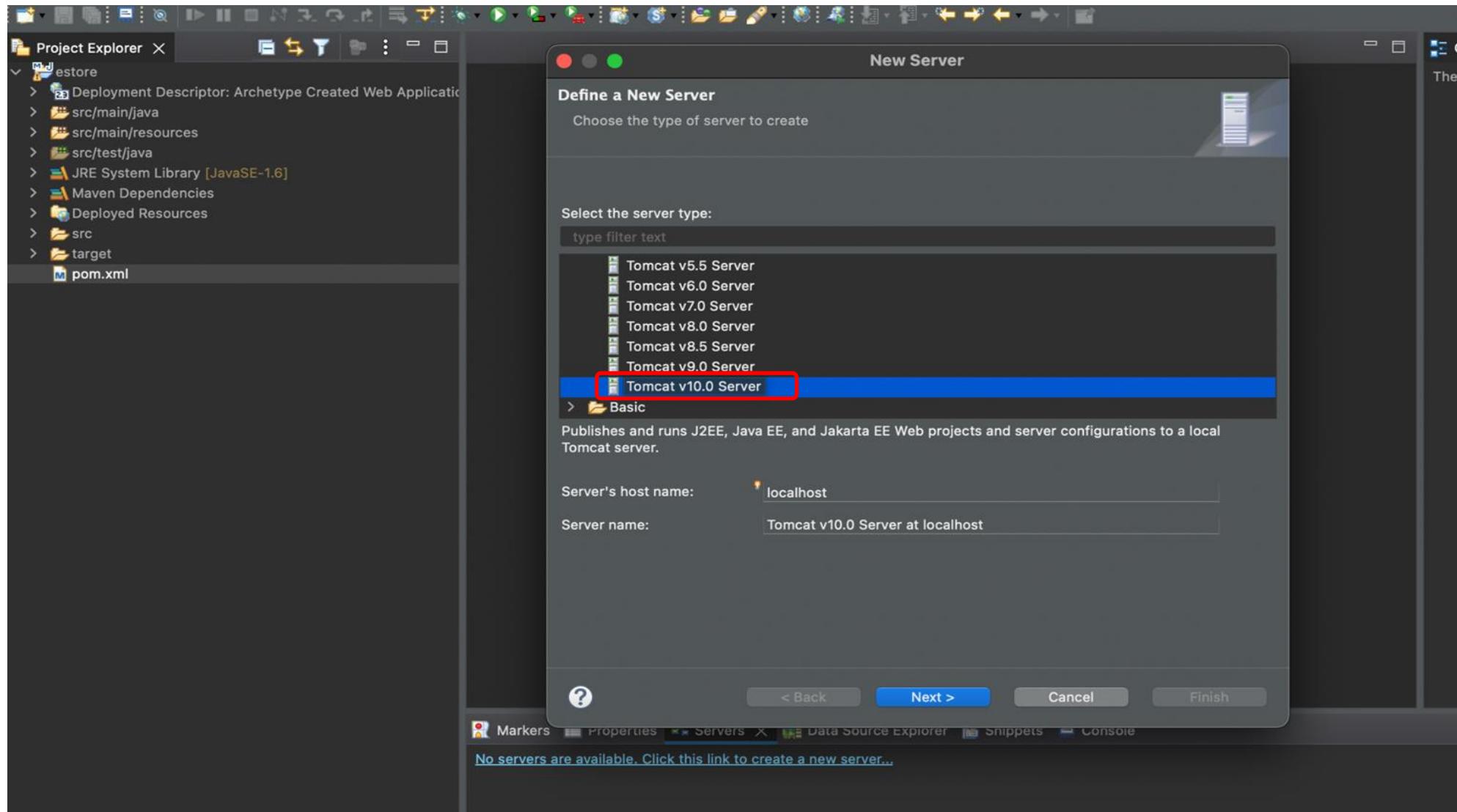
Configure Tomcat

Click on the link to add a new server or right-click to add a new server on the servers tab.



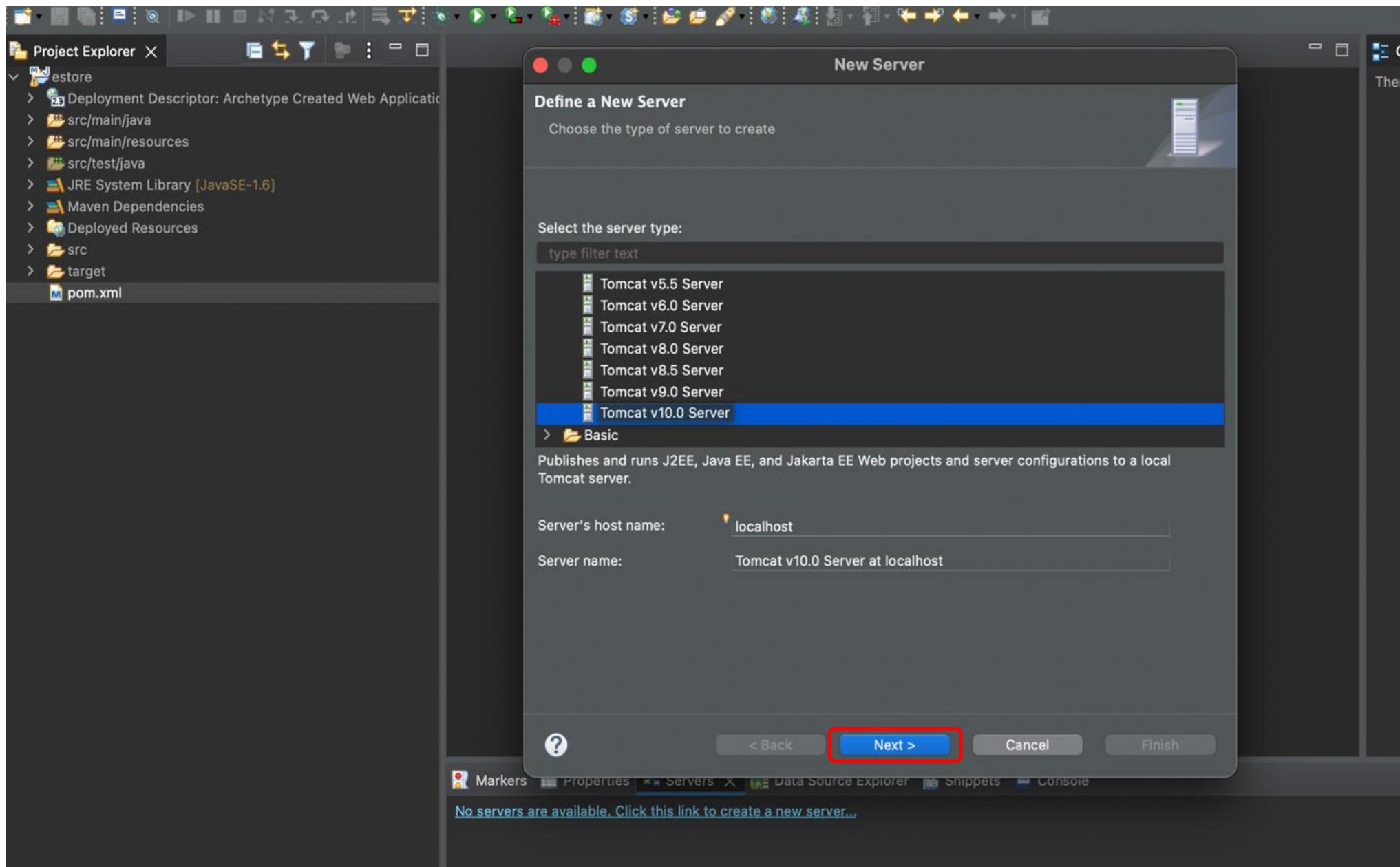
Configure Tomcat

Choose the server as Tomcat 10.



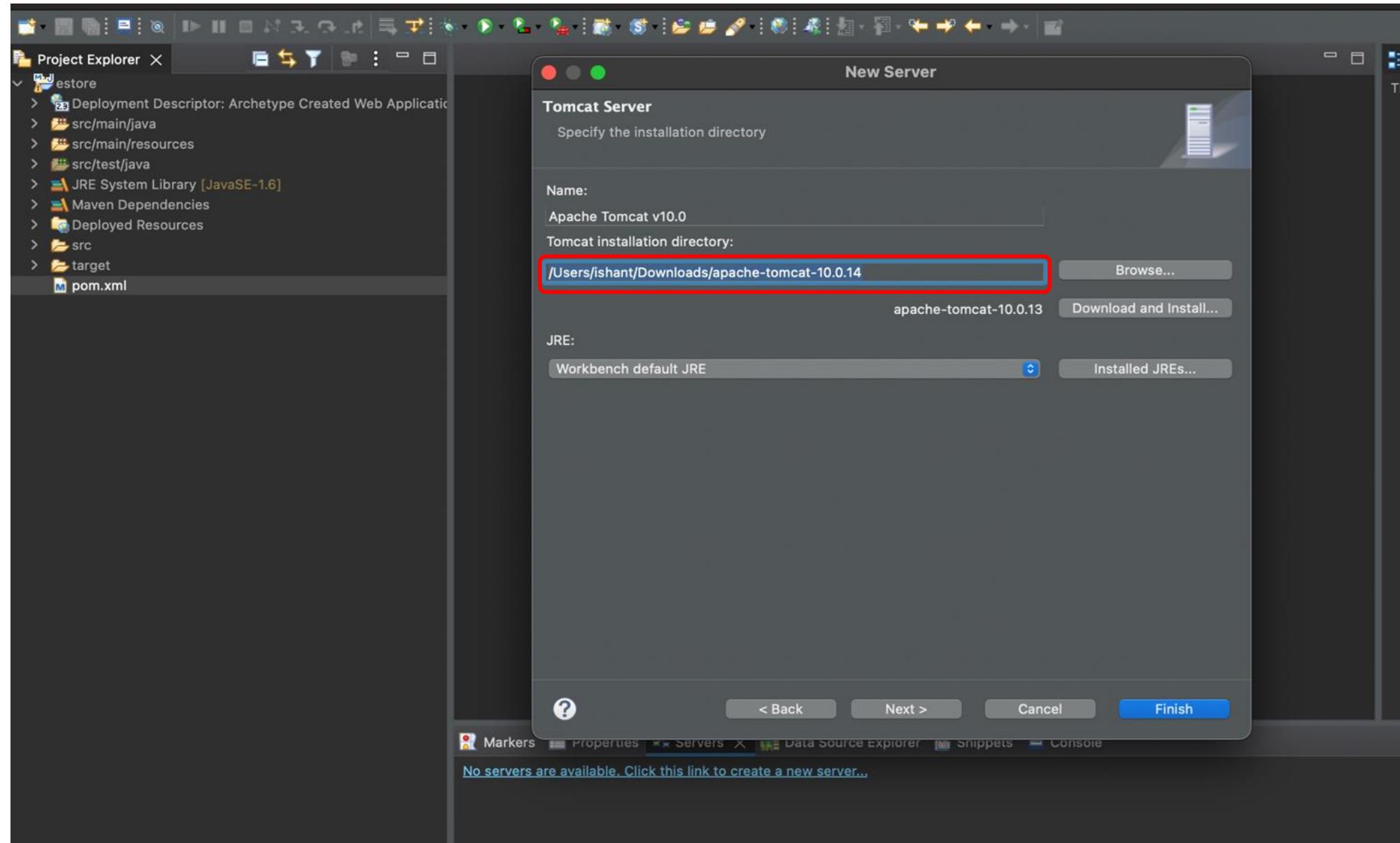
Configure Tomcat

Click on Next.



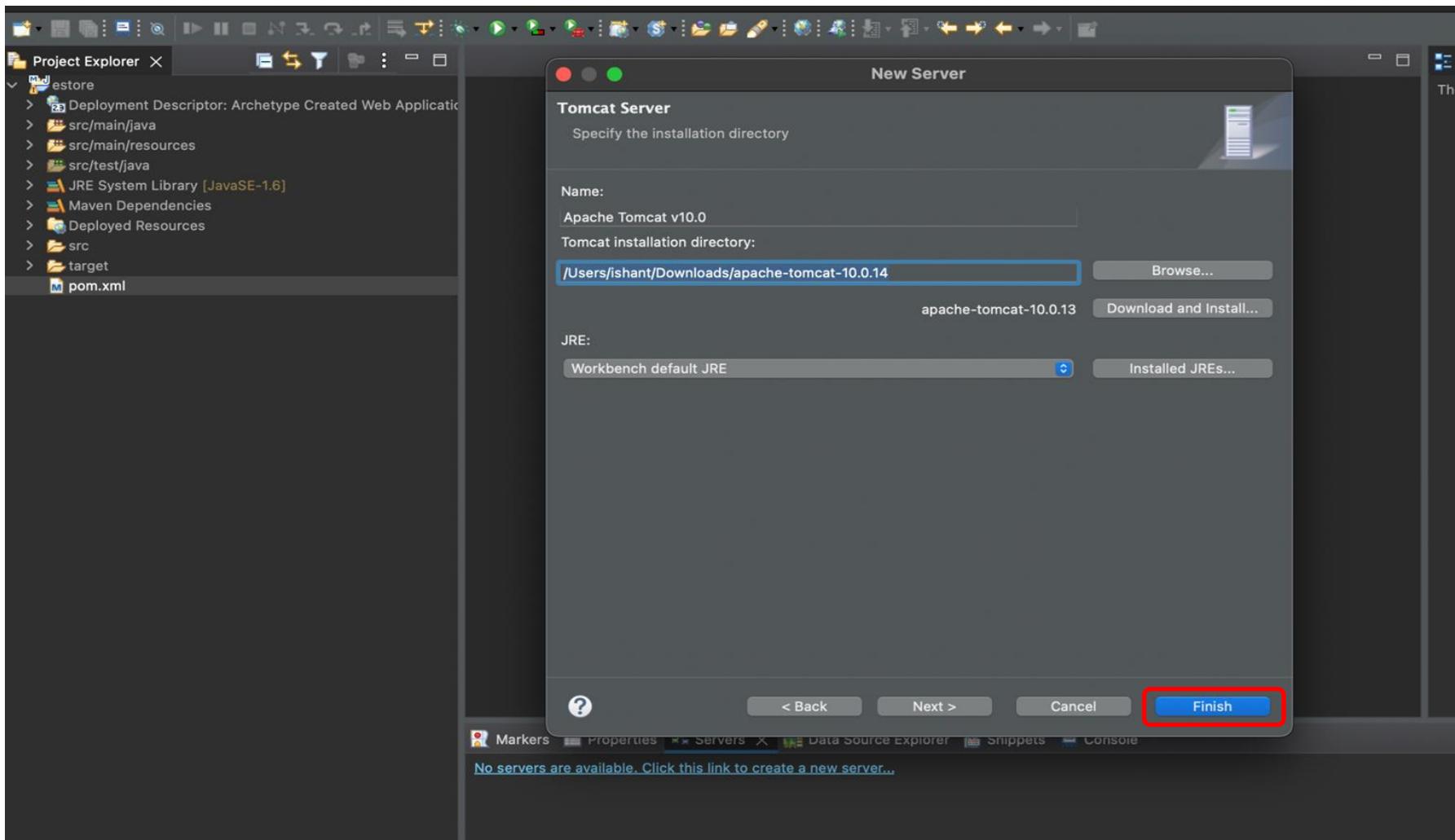
Configure Tomcat

Browse the path to the unzipped Tomcat 10 director.



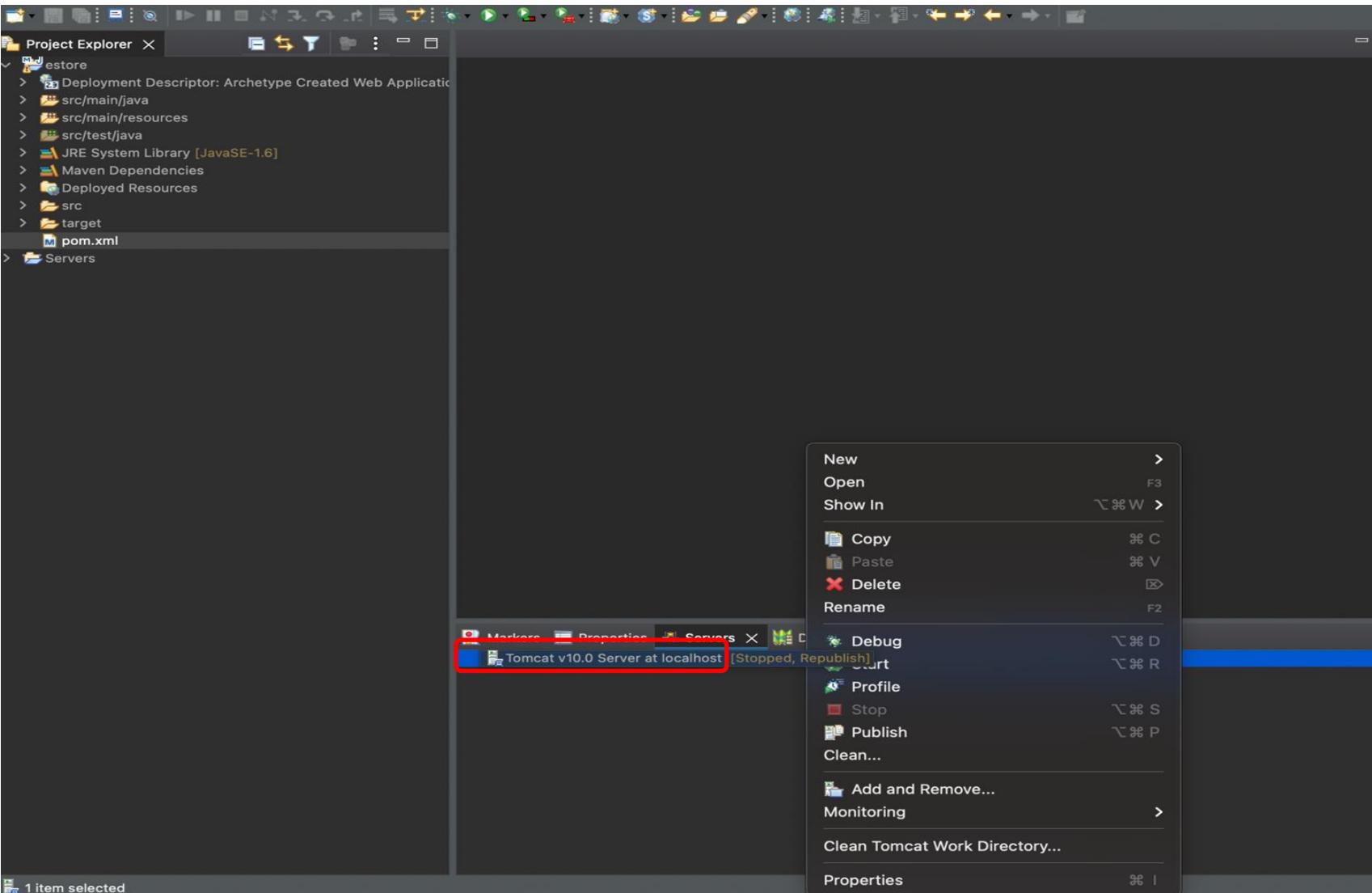
Configure Tomcat

Click on Finish.



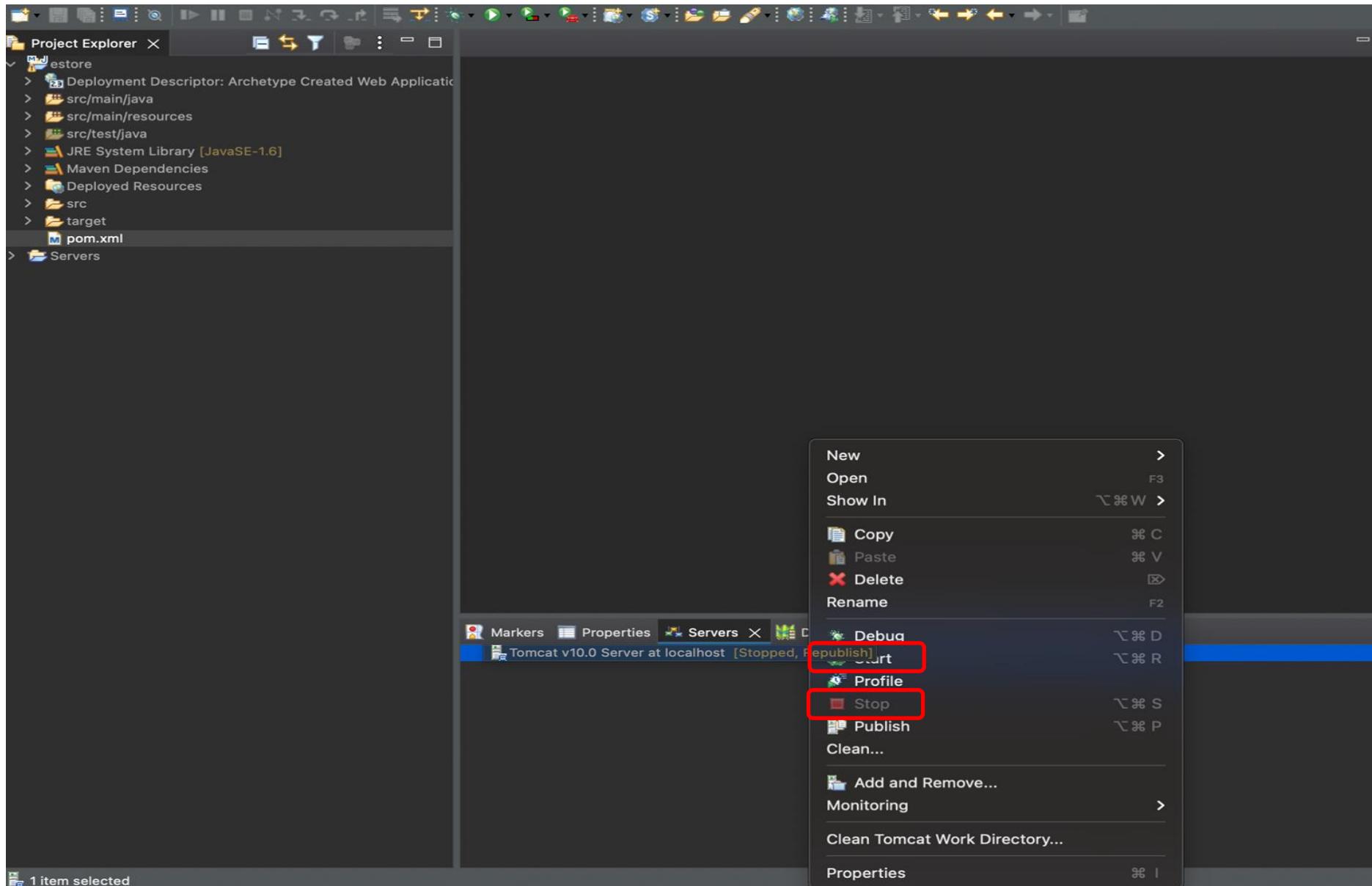
Configure Tomcat

The added server will now reflect in the servers tab.



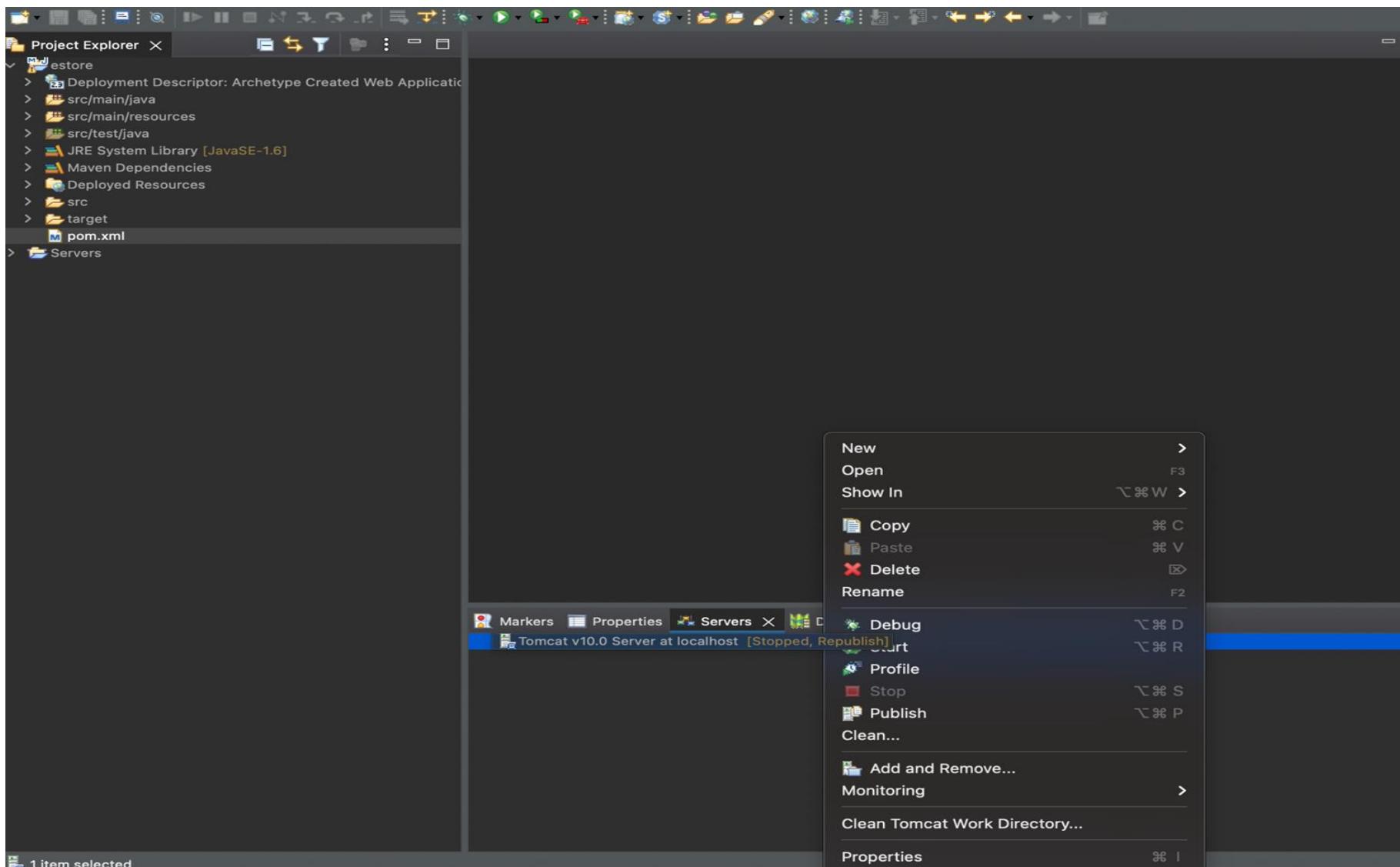
Configure Tomcat

Right-click and get options to start and stop the server.



Configure Tomcat

Successfully configured the apache tomcat 10 is successfully configured in eclipse EE IDE.



TECHNOLOGY



Caltech

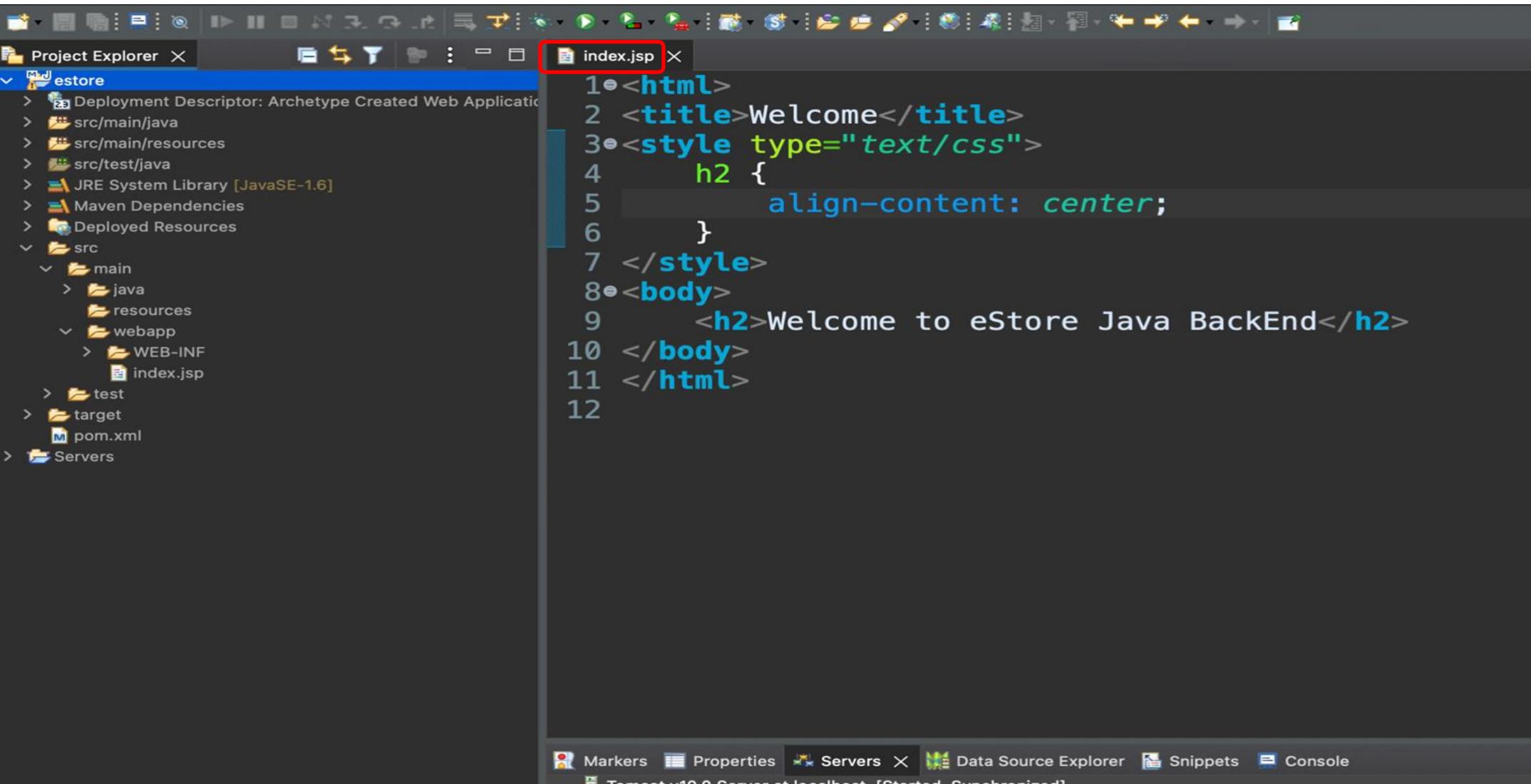
Center for Technology & Management Education

Develop, Build, and Package the Project

Create the Project

Develop index.jsp Page

In web.xml file, index.jsp page is welcome page.

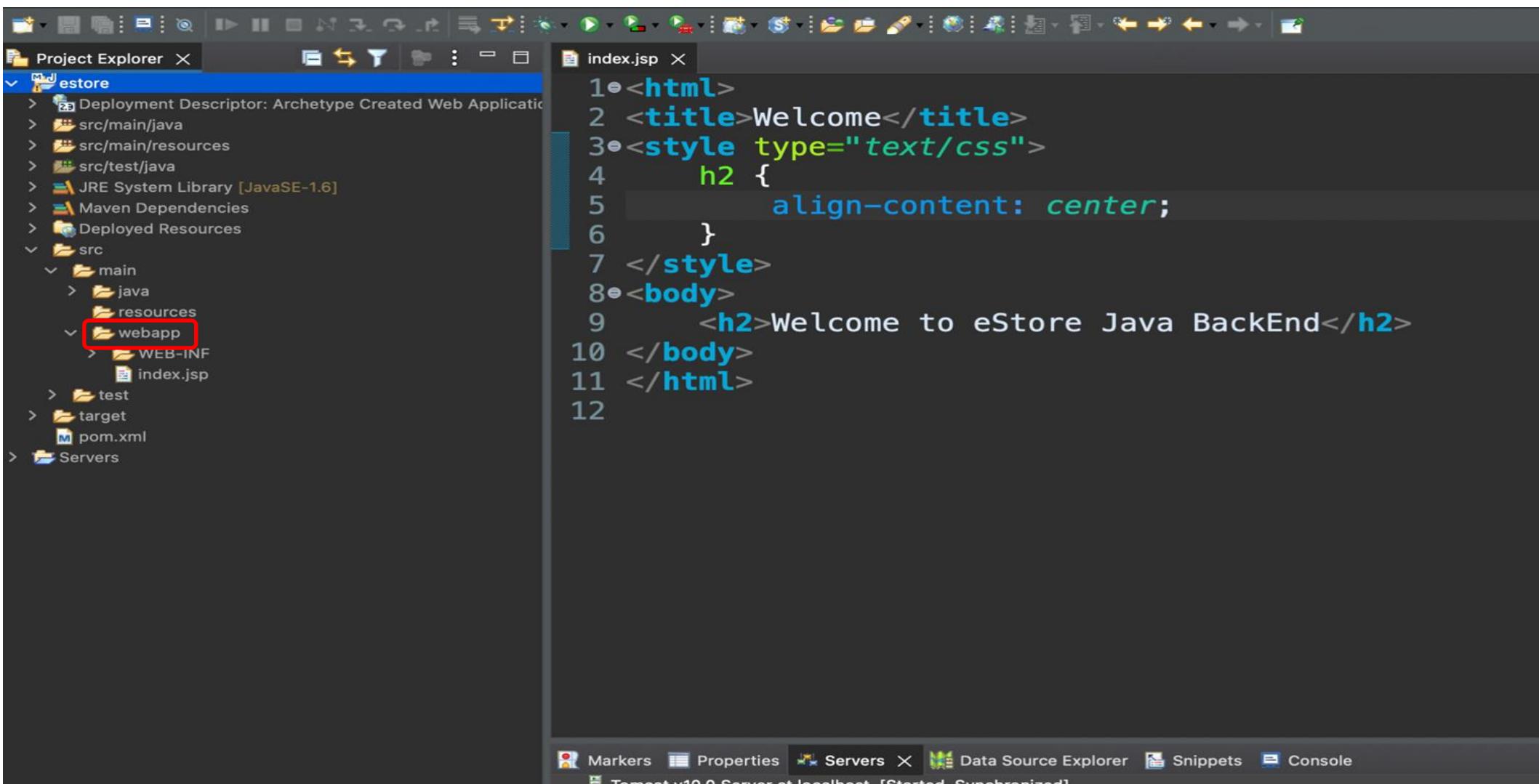


The screenshot shows the Eclipse IDE interface. The Project Explorer view on the left lists the project structure for 'eStore'. The Java Editor view on the right displays the code for 'index.jsp'.

```
1<html>
2 <title>Welcome</title>
3<style type="text/css">
4   h2 {
5     align-content: center;
6   }
7 </style>
8<body>
9   <h2>Welcome to eStore Java BackEnd</h2>
10 </body>
11 </html>
12
```

Develop index.jsp Page

Navigate to the webapp directory (src>main>webapp).



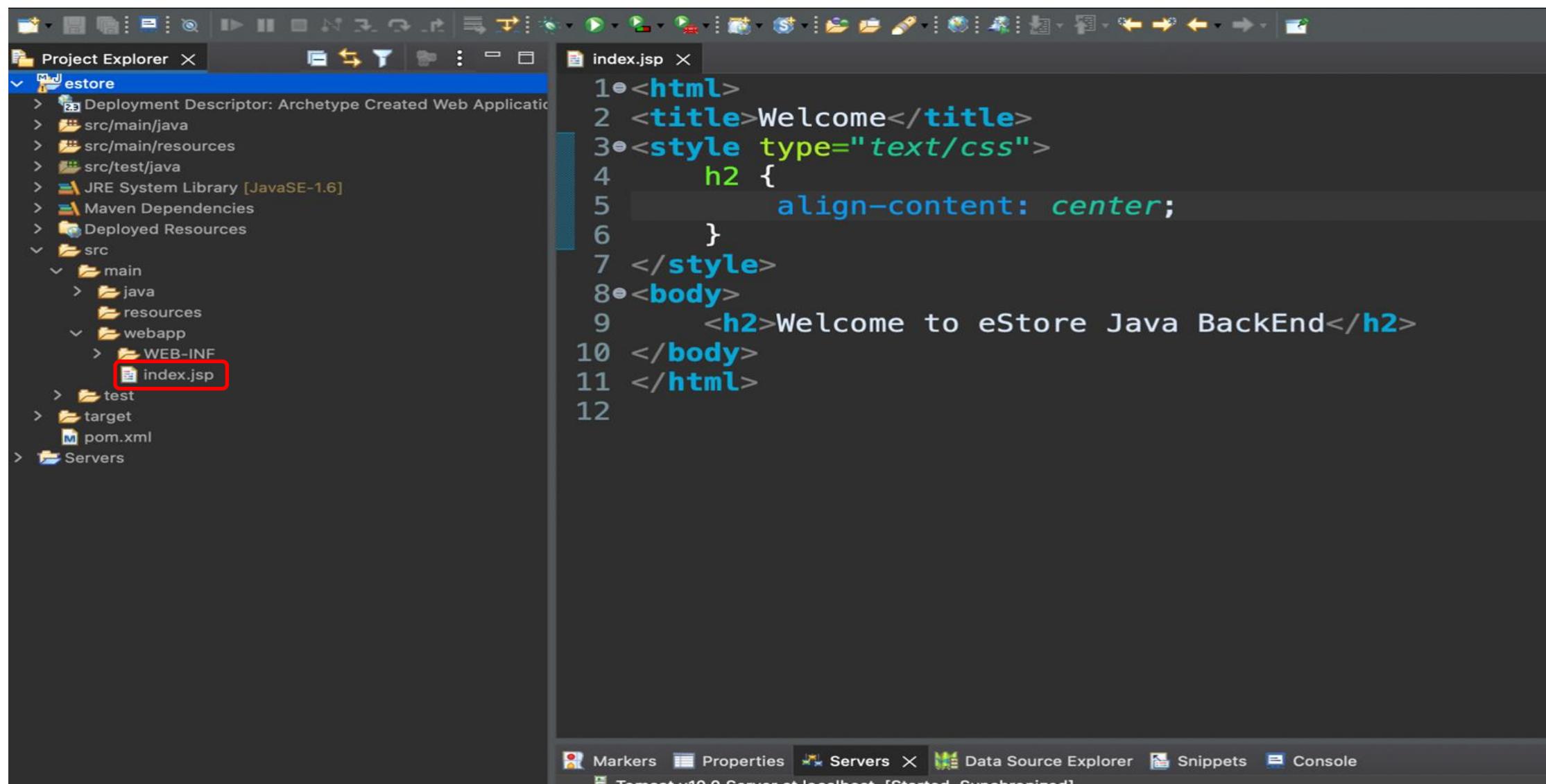
The screenshot shows a Java-based IDE interface. On the left, the Project Explorer displays a project structure for 'eStore'. The 'src' folder contains 'main', 'resources', and 'webapp'. The 'webapp' folder is highlighted with a red box. Inside 'webapp' are 'WEB-INF' and 'index.jsp'. The right side of the interface is a code editor window titled 'index.jsp' containing the following JSP code:

```
1<html>
2 <title>Welcome</title>
3<style type="text/css">
4   h2 {
5     align-content: center;
6   }
7 </style>
8<body>
9   <h2>Welcome to eStore Java BackEnd</h2>
10 </body>
11 </html>
12
```

The code editor also shows standard Eclipse-style toolbars at the top and various tabs like 'Markers', 'Properties', 'Servers', 'Data Source Explorer', 'Snippets', and 'Console' at the bottom.

Develop index.jsp Page

Open the index.jsp.



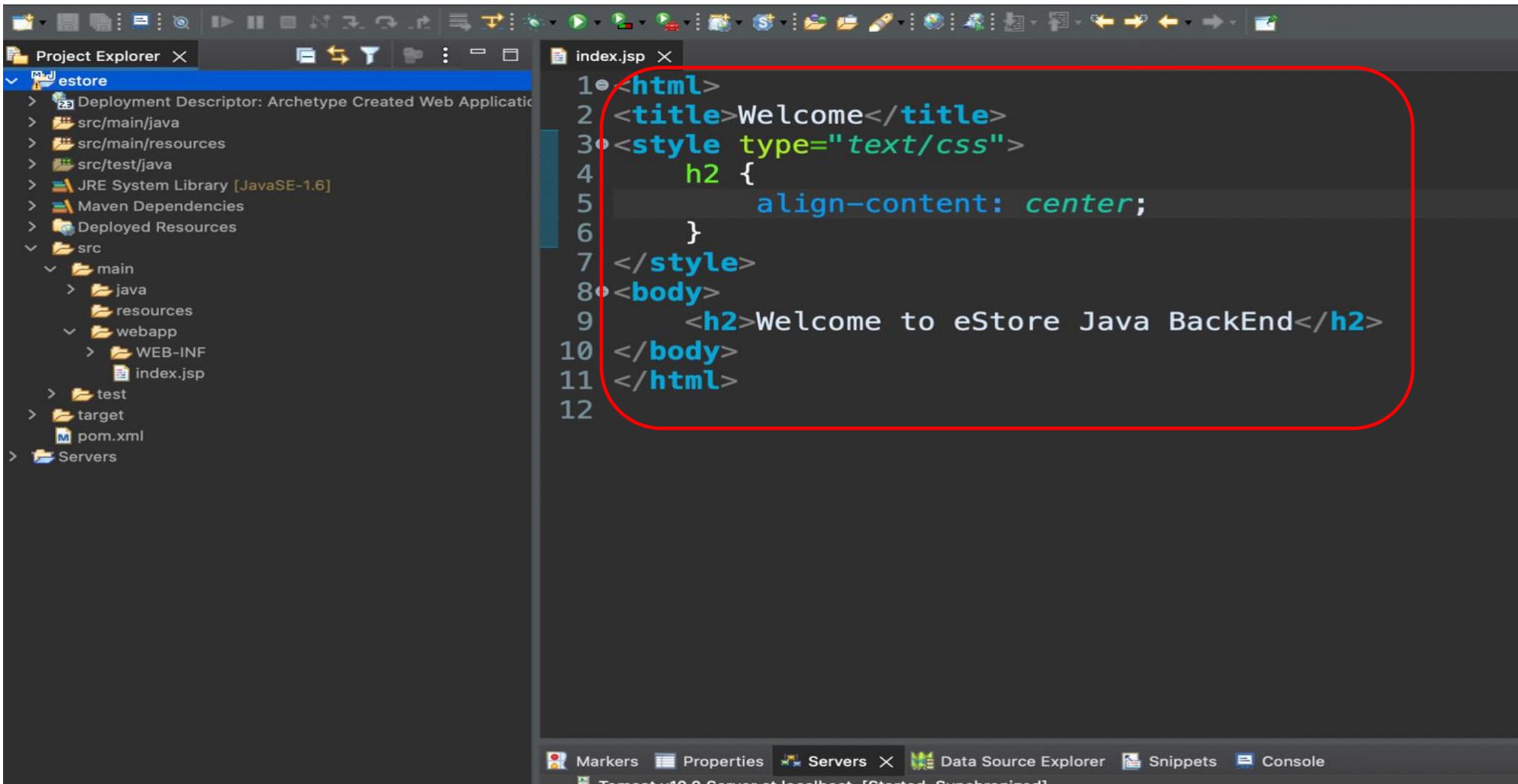
The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer view displays the project structure for 'eStore'. Inside the 'src' folder, there is a 'main' folder containing 'java', 'resources', and 'webapp'. The 'webapp' folder contains 'WEB-INF' and 'index.jsp'. The 'index.jsp' file is selected in the editor on the right. The code in 'index.jsp' is:

```
1<html>
2 <title>Welcome</title>
3<style type="text/css">
4   h2 {
5     align-content: center;
6   }
7 </style>
8<body>
9   <h2>Welcome to eStore Java BackEnd</h2>
10 </body>
11 </html>
12
```

The 'index.jsp' file in the Project Explorer is highlighted with a red box.

Develop index.jsp Page

Write a welcome message to the index page.

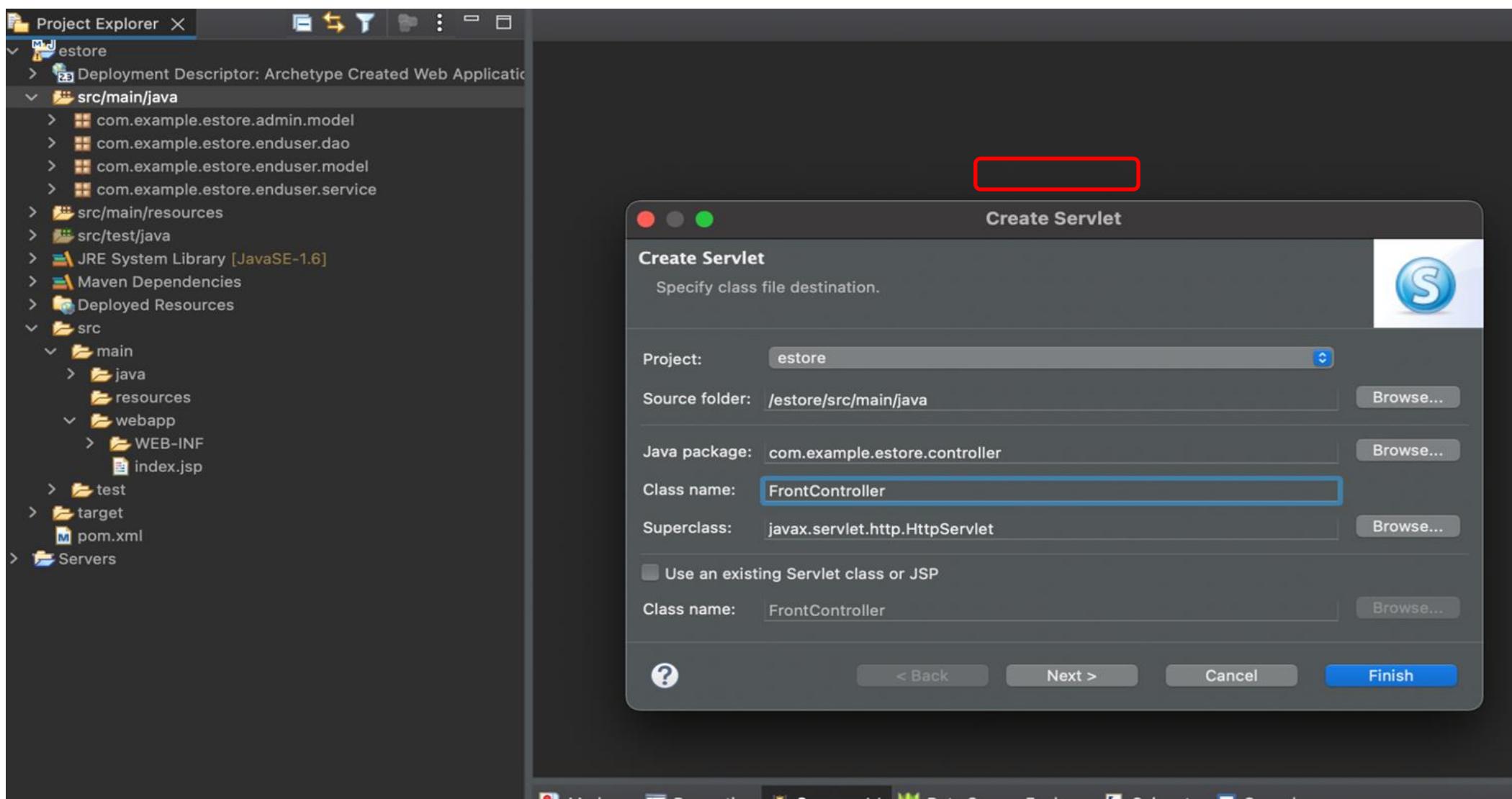


The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer view displays the project structure for 'eStore'. The 'src' folder contains 'main' and 'test' folders, with 'main' further divided into 'java', 'resources', and 'webapp'. Inside 'webapp', there is a 'WEB-INF' folder containing an 'index.jsp' file. On the right, the editor view shows the content of 'index.jsp'. The code includes an HTML structure with a title, a CSS style block for centering the content, and a body containing a welcome message. A red oval highlights the CSS code (lines 3-6). The bottom of the screen shows the Eclipse toolbar and various perspectives like Markers, Properties, Servers, Data Source Explorer, Snippets, and Console.

```
1<html>
2<title>Welcome</title>
3<style type="text/css">
4    h2 {
5        align-content: center;
6    }
7</style>
8<body>
9    <h2>Welcome to eStore Java BackEnd</h2>
10</body>
11</html>
12
```

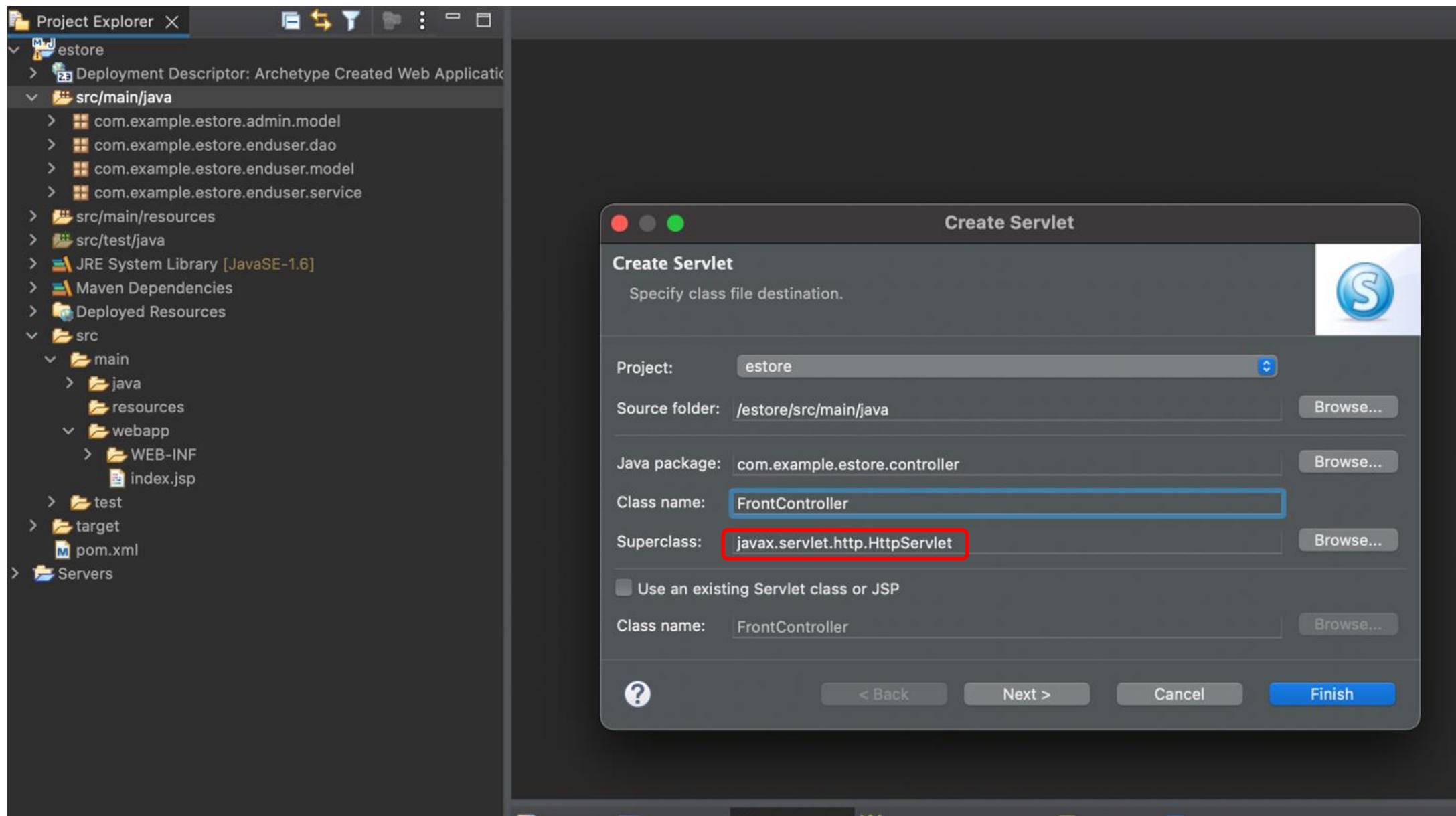
Develop FrontController Design Pattern

Create a servlet by doing right-clicking on the java src.



Develop FrontController Design Pattern

Choose the method as service to override which can handle both http get and post requests.



Develop FrontController Design Pattern

Create a Servlet by doing right-clicking on the java-src.

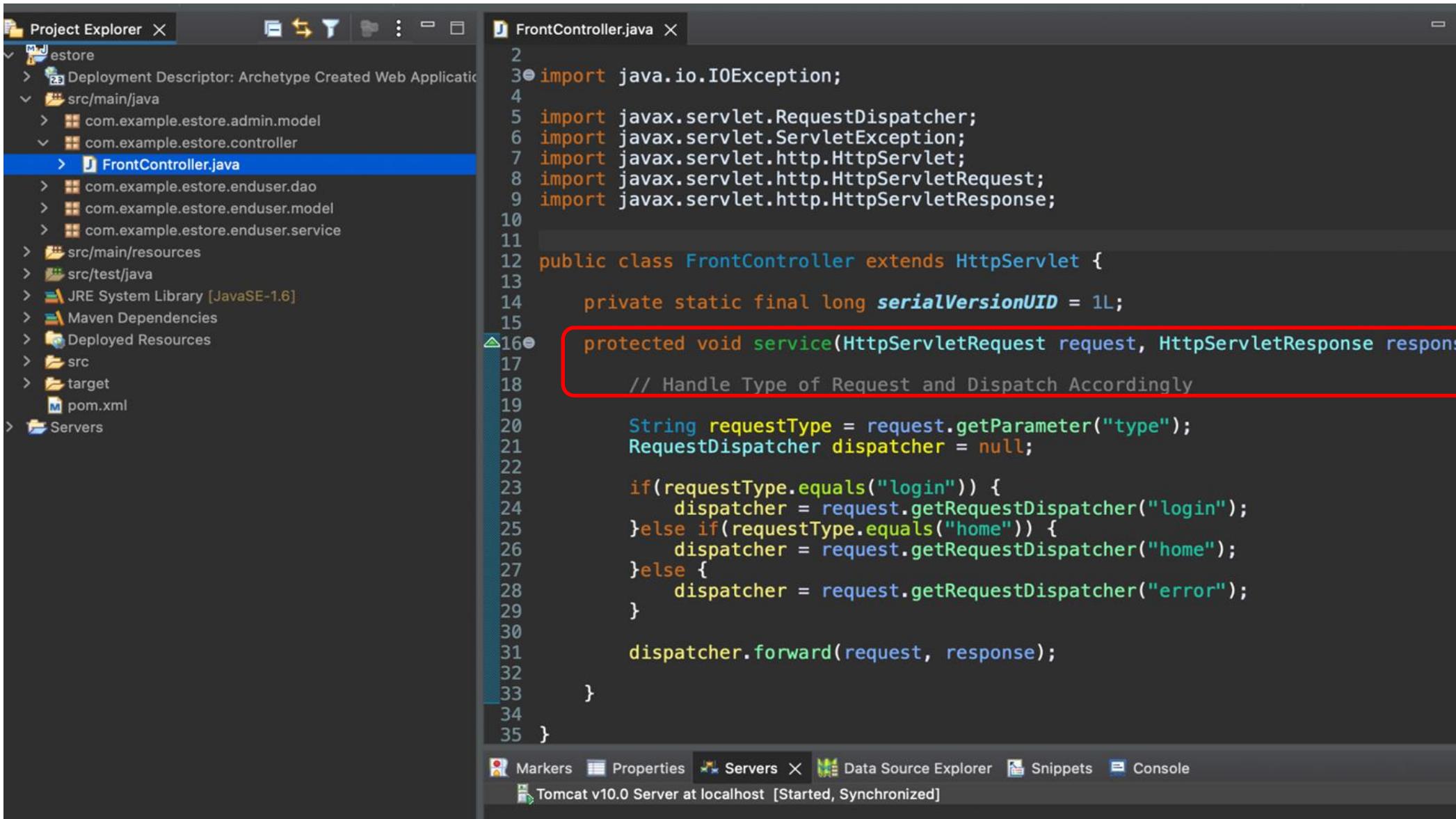
The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer view displays a project structure for 'estore'. The 'src/main/java' folder contains several packages: 'com.example.estore.admin.model', 'com.example.estore.controller', 'com.example.estore.enduser.dao', 'com.example.estore.enduser.model', and 'com.example.estore.enduser.service'. A file named 'FrontController.java' is selected and highlighted with a red border in the Project Explorer. On the right, the Java code editor shows the content of 'FrontController.java'. The code implements a servlet that dispatches requests based on their type ('login', 'home', or 'error').

```
2
3 import java.io.IOException;
4
5 import javax.servlet.RequestDispatcher;
6 import javax.servlet.ServletException;
7 import javax.servlet.http.HttpServlet;
8 import javax.servlet.http.HttpServletRequest;
9 import javax.servlet.http.HttpServletResponse;
10
11 public class FrontController extends HttpServlet {
12
13     private static final long serialVersionUID = 1L;
14
15     protected void service(HttpServletRequest request, HttpServletResponse response) {
16
17         // Handle Type of Request and Dispatch Accordingly
18
19         String requestType = request.getParameter("type");
20         RequestDispatcher dispatcher = null;
21
22         if(requestType.equals("login")) {
23             dispatcher = request.getRequestDispatcher("login");
24         }else if(requestType.equals("home")) {
25             dispatcher = request.getRequestDispatcher("home");
26         }else {
27             dispatcher = request.getRequestDispatcher("error");
28         }
29
30         dispatcher.forward(request, response);
31
32     }
33
34 }
35 }
```

Below the code editor, the Eclipse status bar shows 'Tomcat v10.0 Server at localhost [Started, Synchronized]'. The bottom navigation bar includes tabs for 'Markers', 'Properties', 'Servers', 'Data Source Explorer', 'Snippets', and 'Console'.

Develop FrontController Design Pattern

Choose the method as service to override that can handle both http get and post requests.

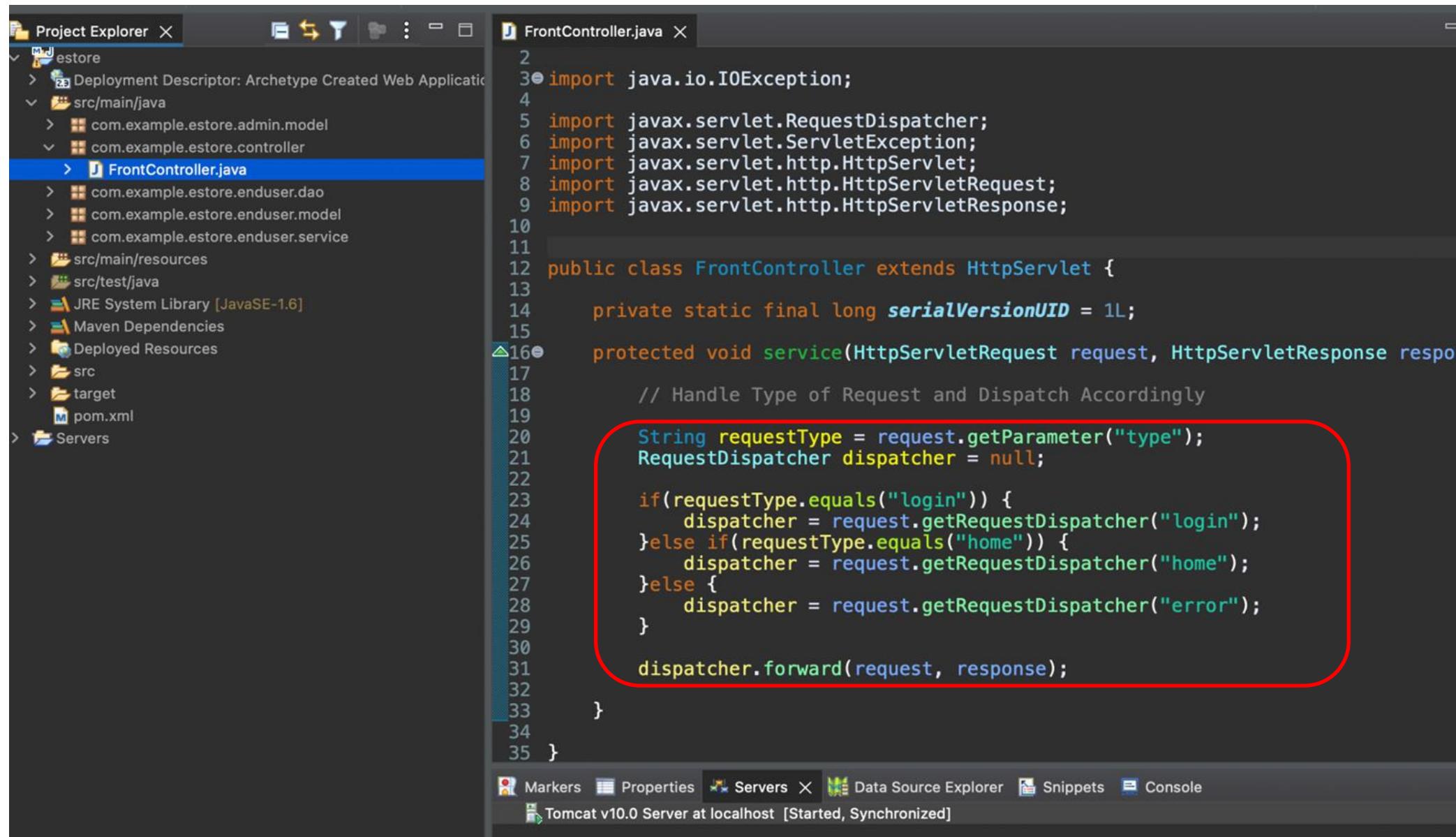


The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer view displays a project structure for a web application named 'estore'. The 'src/main/java' folder contains several packages: 'com.example.estore.admin.model', 'com.example.estore.controller', 'com.example.estore.enduser.dao', 'com.example.estore.enduser.model', and 'com.example.estore.enduser.service'. The file 'FrontController.java' is selected in the Project Explorer and is shown in the code editor on the right. The code implements the 'FrontController' class, which extends 'HttpServlet'. It overrides the 'service' method to handle both 'HttpServletRequest' and 'HttpServletResponse'. The 'service' method checks the 'request.getParameter("type")' value to determine which dispatcher to use ('login', 'home', or 'error') and then forwards the request to that dispatcher. A red box highlights the 'service' method definition.

```
2 import java.io.IOException;
3 import javax.servlet.RequestDispatcher;
4 import javax.servlet.ServletException;
5 import javax.servlet.http.HttpServlet;
6 import javax.servlet.http.HttpServletRequest;
7 import javax.servlet.http.HttpServletResponse;
8
9 public class FrontController extends HttpServlet {
10
11     private static final long serialVersionUID = 1L;
12
13     protected void service(HttpServletRequest request, HttpServletResponse response) {
14         // Handle Type of Request and Dispatch Accordingly
15
16         String requestType = request.getParameter("type");
17         RequestDispatcher dispatcher = null;
18
19         if(requestType.equals("login")) {
20             dispatcher = request.getRequestDispatcher("login");
21         }else if(requestType.equals("home")) {
22             dispatcher = request.getRequestDispatcher("home");
23         }else {
24             dispatcher = request.getRequestDispatcher("error");
25         }
26
27         dispatcher.forward(request, response);
28     }
29
30 }
31
32 }
33
34 }
35 }
```

Develop FrontController Design Pattern

FrontController Servlet will handle different types of request from the client.



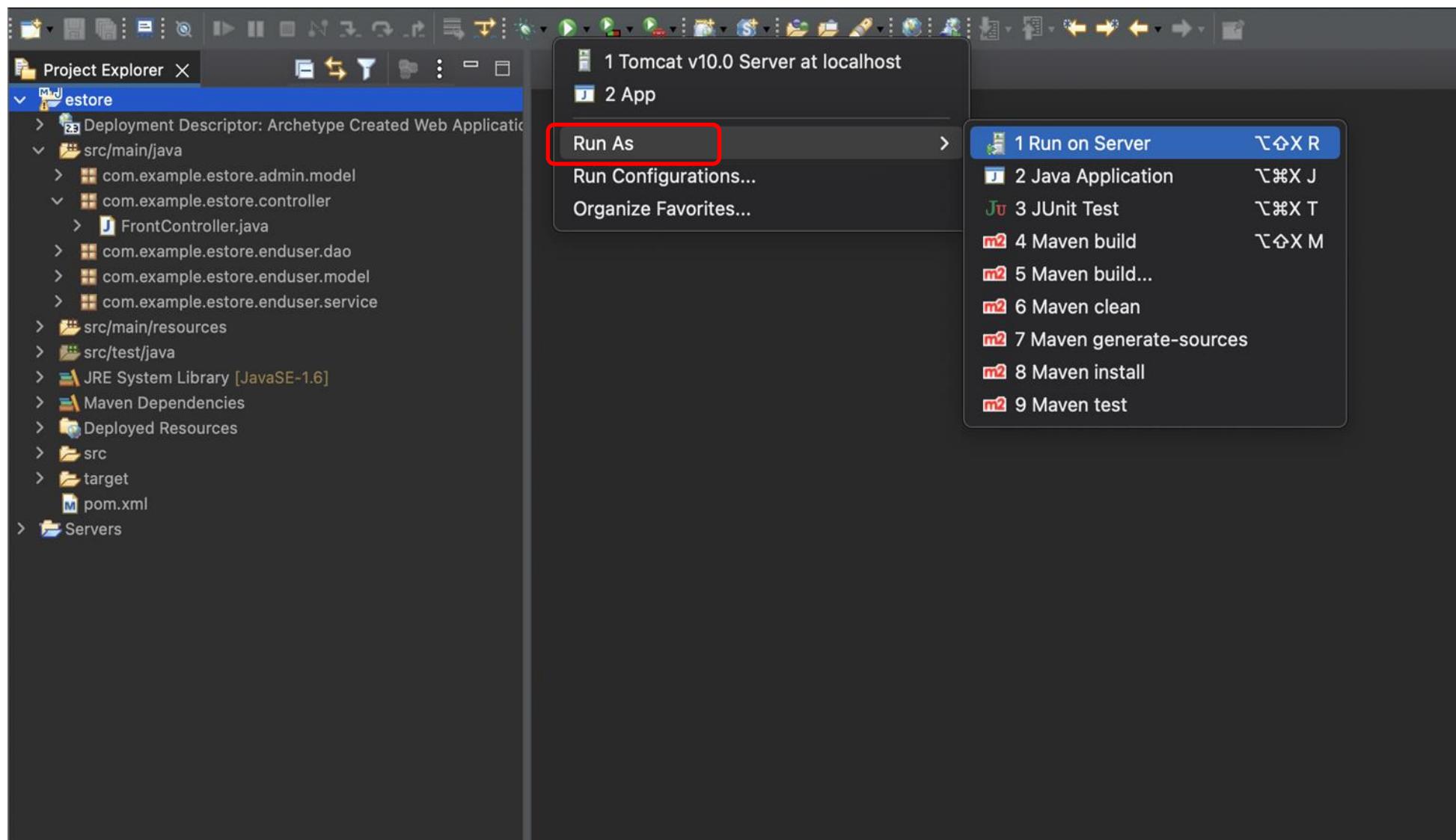
The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer view displays a project named 'estore' with its structure: Deployment Descriptor, src/main/java (containing com.example.estore.admin.model, com.example.estore.controller, and com.example.estore.enduser.dao), src/main/resources, src/test/java, JRE System Library [JavaSE-1.6], Maven Dependencies, Deployed Resources, src, target, and pom.xml. The file 'FrontController.java' is selected in the Project Explorer and is also open in the main code editor window. The code editor shows the following Java code:

```
2 import java.io.IOException;
3 import javax.servlet.RequestDispatcher;
4 import javax.servlet.ServletException;
5 import javax.servlet.http.HttpServlet;
6 import javax.servlet.http.HttpServletRequest;
7 import javax.servlet.http.HttpServletResponse;
8
9 public class FrontController extends HttpServlet {
10
11     private static final long serialVersionUID = 1L;
12
13     protected void service(HttpServletRequest request, HttpServletResponse response) {
14         // Handle Type of Request and Dispatch Accordingly
15
16         String requestType = request.getParameter("type");
17         RequestDispatcher dispatcher = null;
18
19         if(requestType.equals("login")) {
20             dispatcher = request.getRequestDispatcher("login");
21         } else if(requestType.equals("home")) {
22             dispatcher = request.getRequestDispatcher("home");
23         } else {
24             dispatcher = request.getRequestDispatcher("error");
25         }
26
27         dispatcher.forward(request, response);
28     }
29
30 }
31
32
33 }
```

A red rounded rectangle highlights the dispatch logic starting at line 16, specifically the if-else block that handles 'login', 'home', and 'error' requests by getting the corresponding RequestDispatcher and then calling forward().

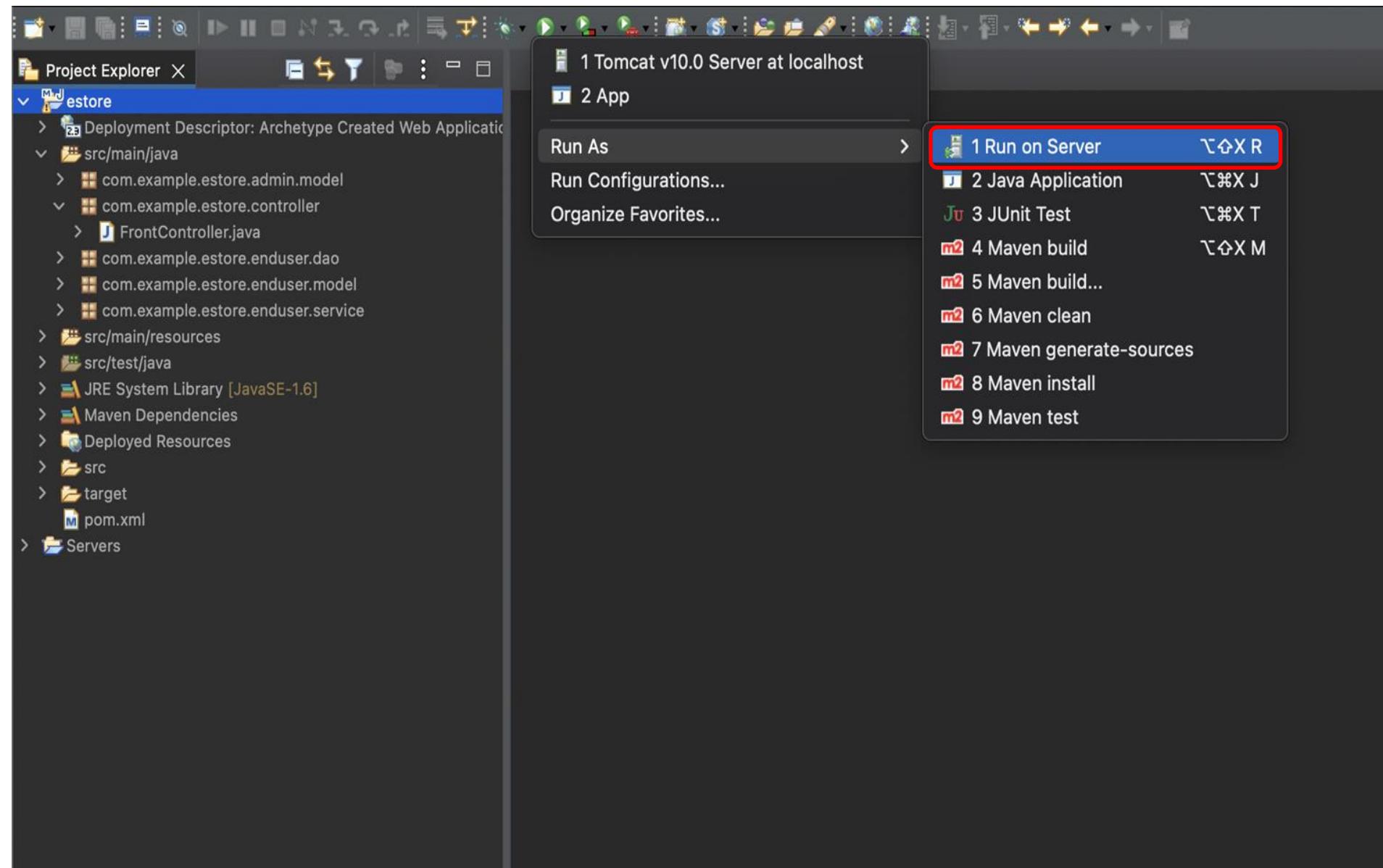
Run the Project

Finally, run the project to see if any errors exists.



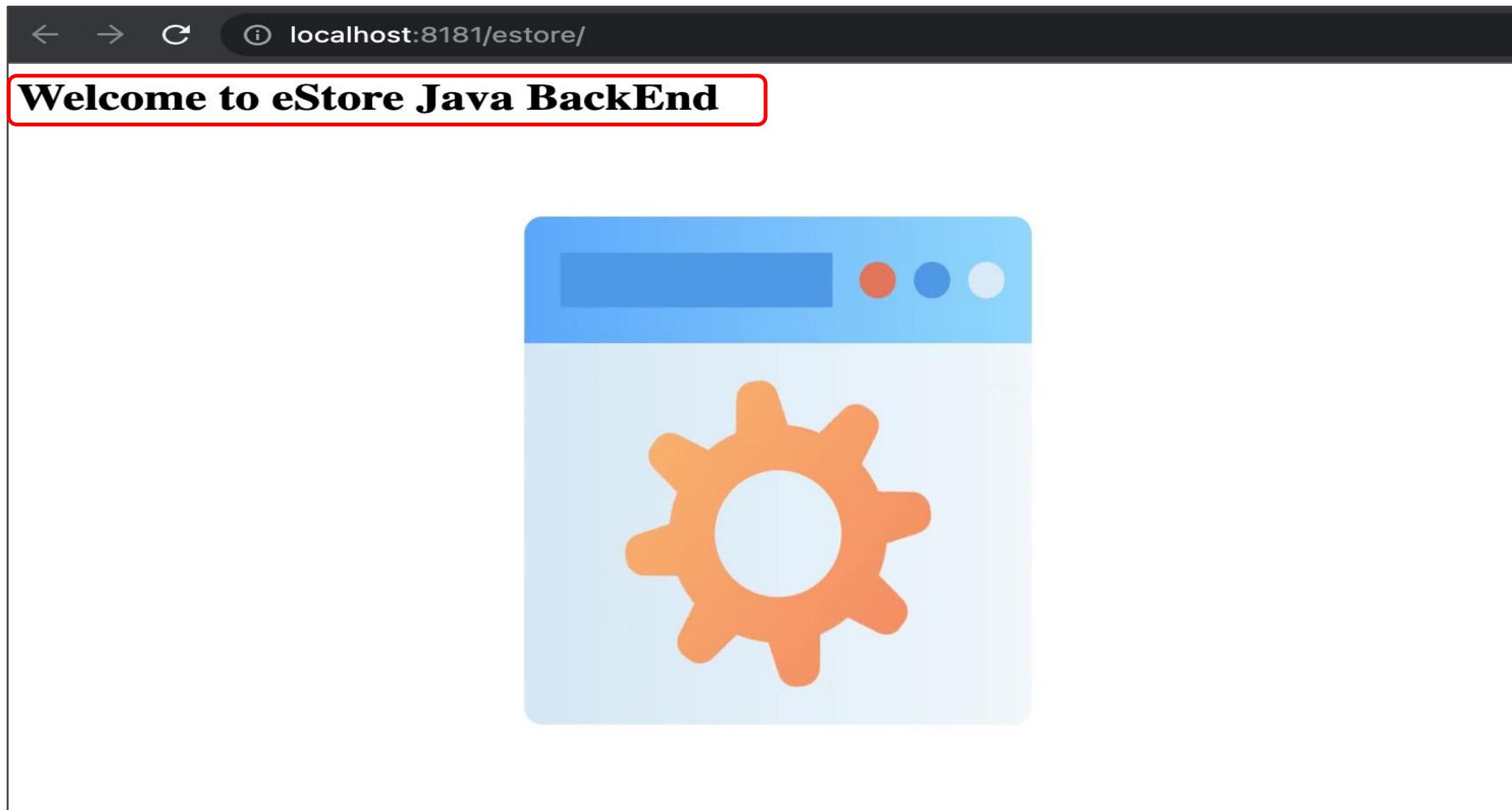
Run the Project

To run the project, select Run on Server.



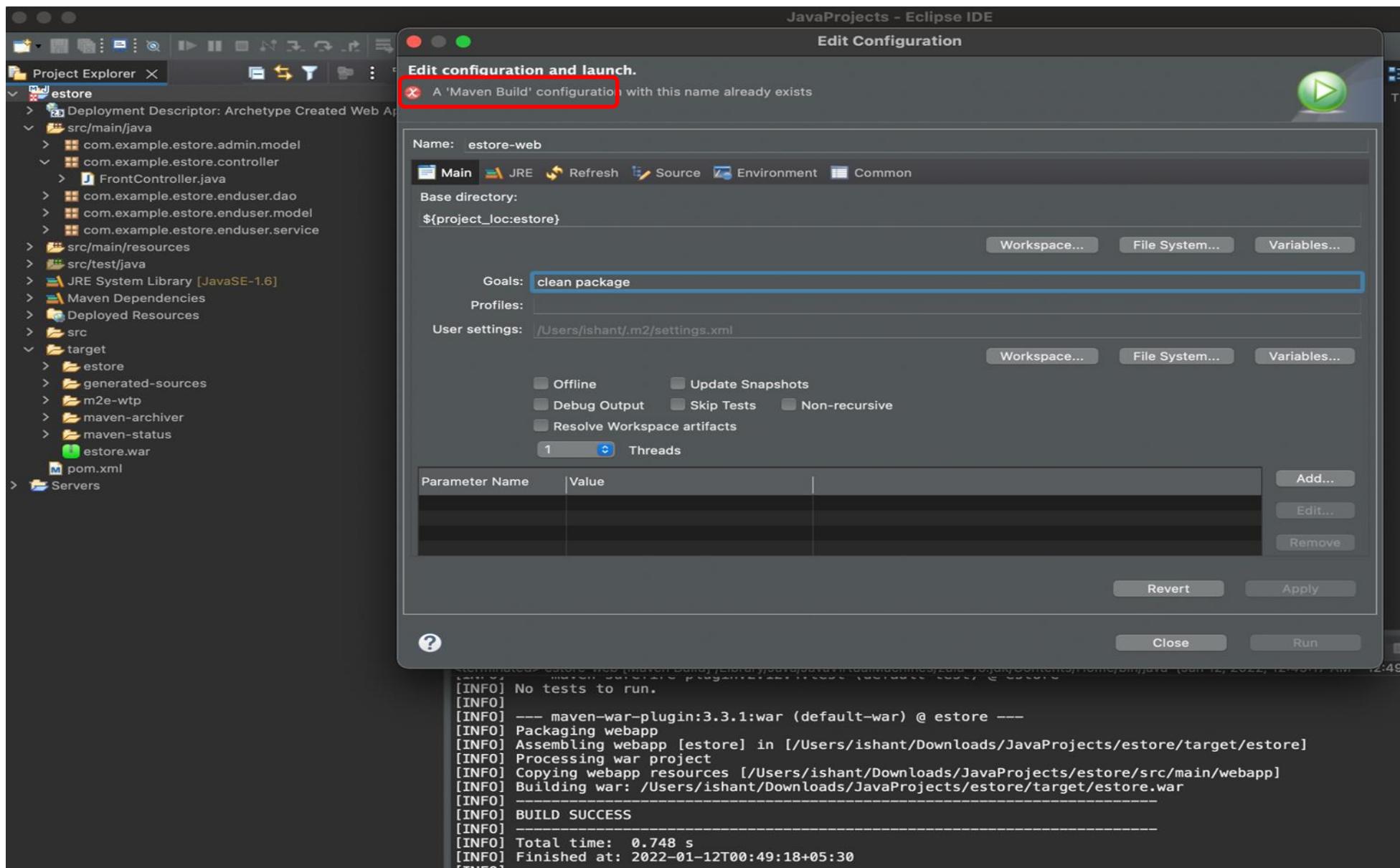
Run the Project on Localhost

After running the project, we will see index.jsp file is executed automatically.



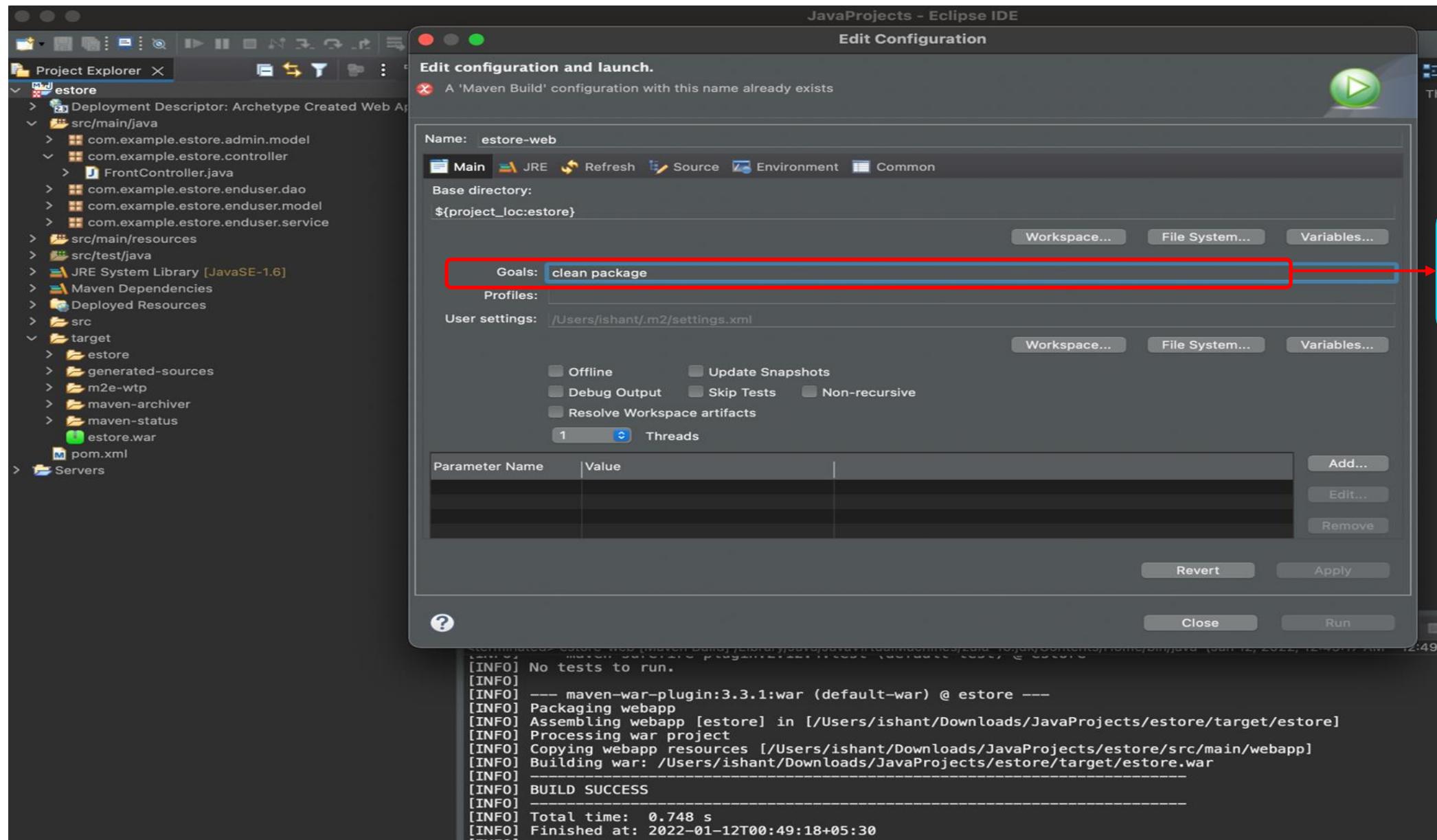
Package the Web App

Run as Maven Build in order to package the web app as .war (web archive).



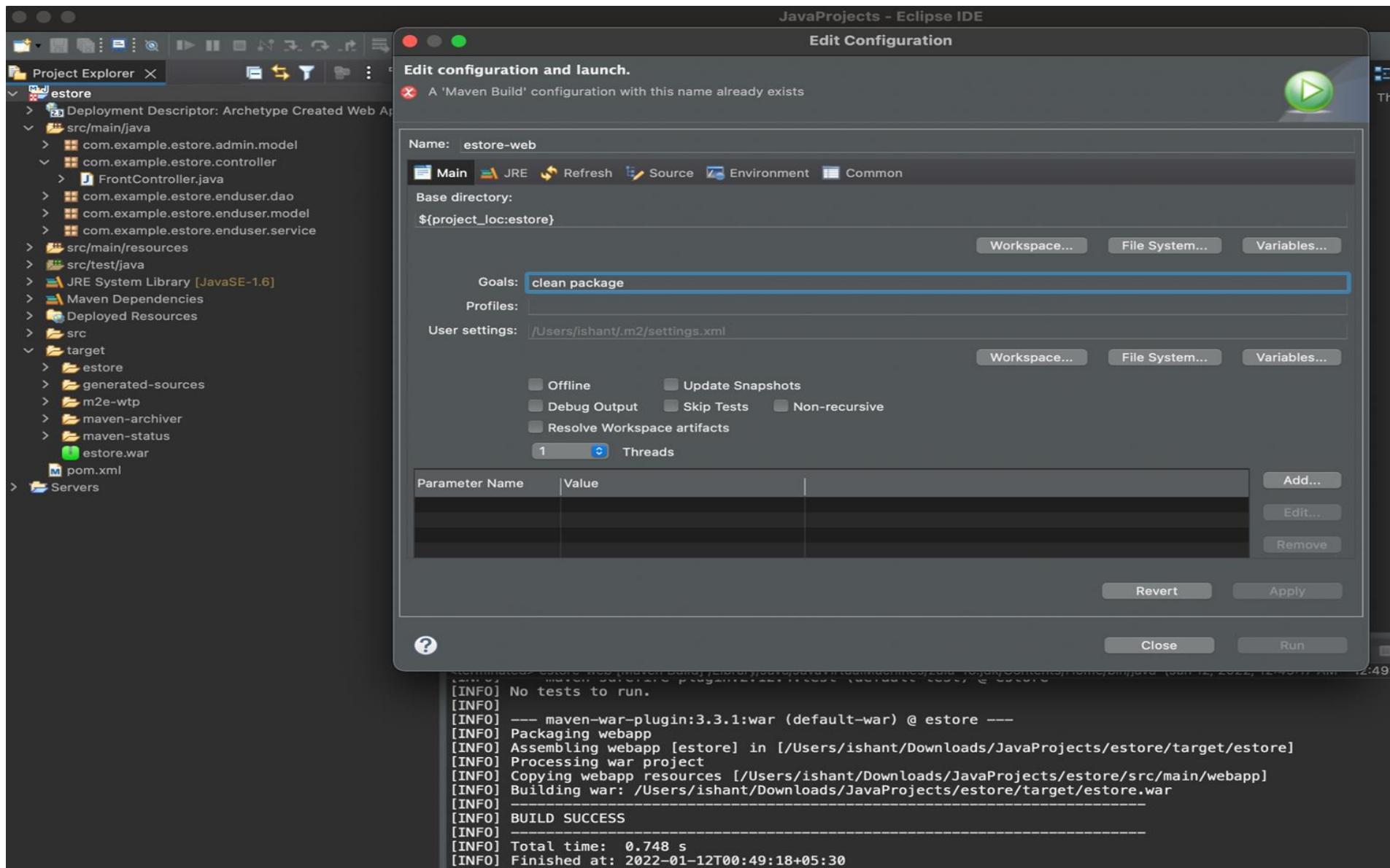
Package the Web App

To get estore.war file under the target directory of the project:



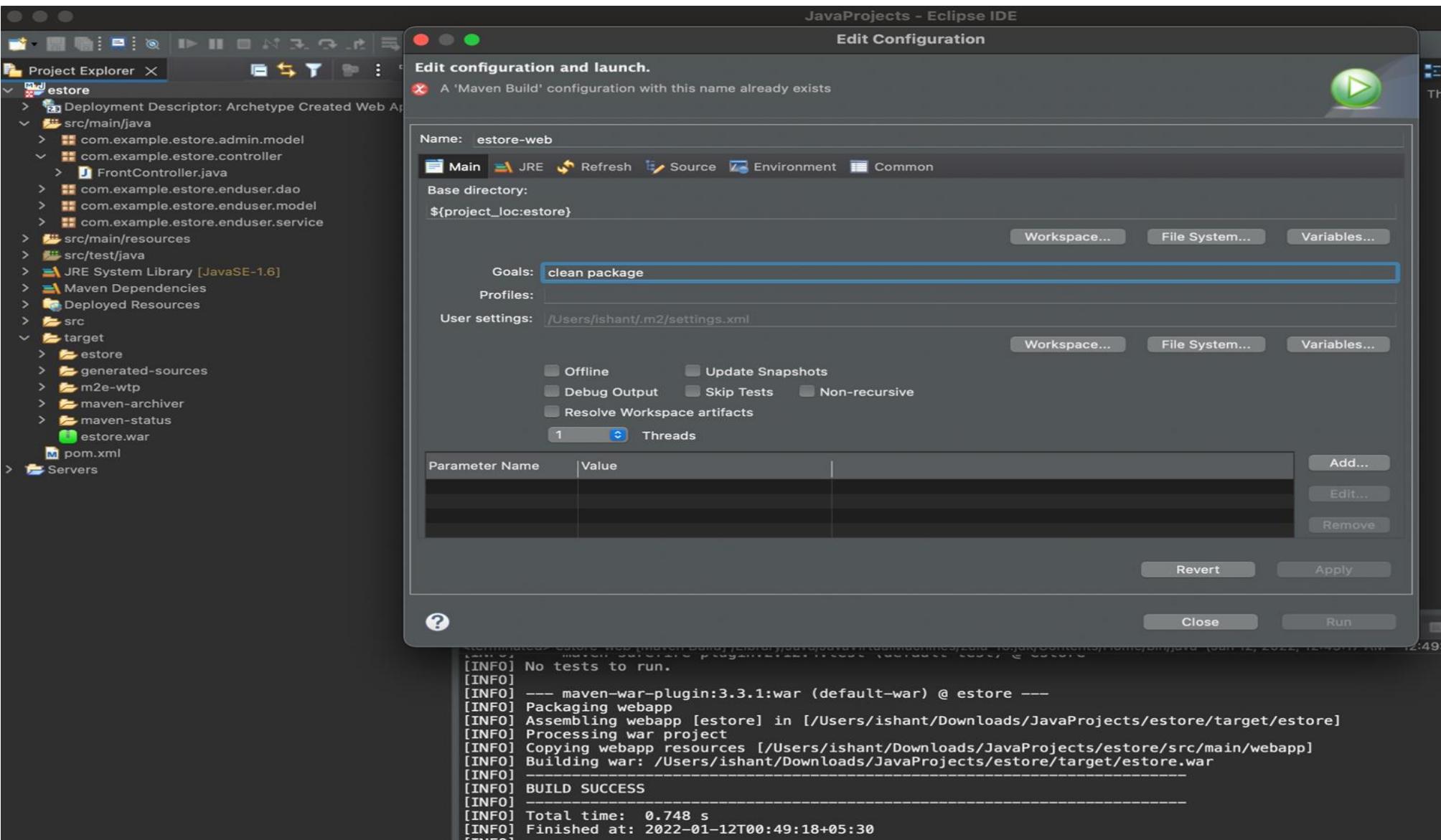
Package the Web App

Refresh the project if after the build is success but the war file cannot be seen.



Package the Web App

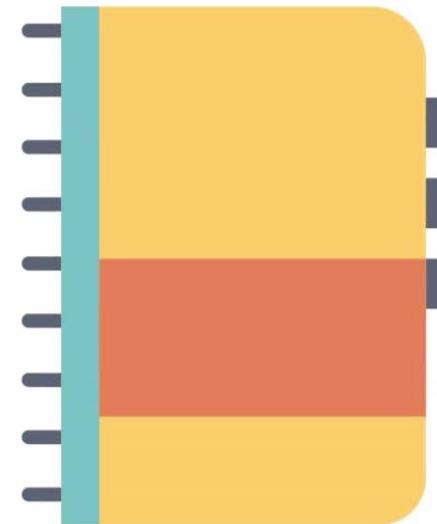
Plugins in pom.xml files are of latest version. Compiler plugin must be 1.8 or higher.



Package the Web App

Sample code for plugins in pom.xml

```
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-war-plugin</artifactId>
    <version>3.3.1</version>
</plugin>
<plugin>
    <artifactId>maven-compiler-plugin</artifactId>
<configuration>
    <source>1.8</source>
    <target>1.8</target>
</configuration>
</plugin>
```



Key Takeaways

- Configure the pom.xml file for various dependencies and plugins for project to work properly.
- Navigate to the website mvnrepository.com to configure JDBC, Servlets, JSP.
- Navigate to the website tomcat.apache.org/download-10.cgi to configure Tomcat.
- Navigate to the webapp directory (src>main>webapp to develop, build and package a project.



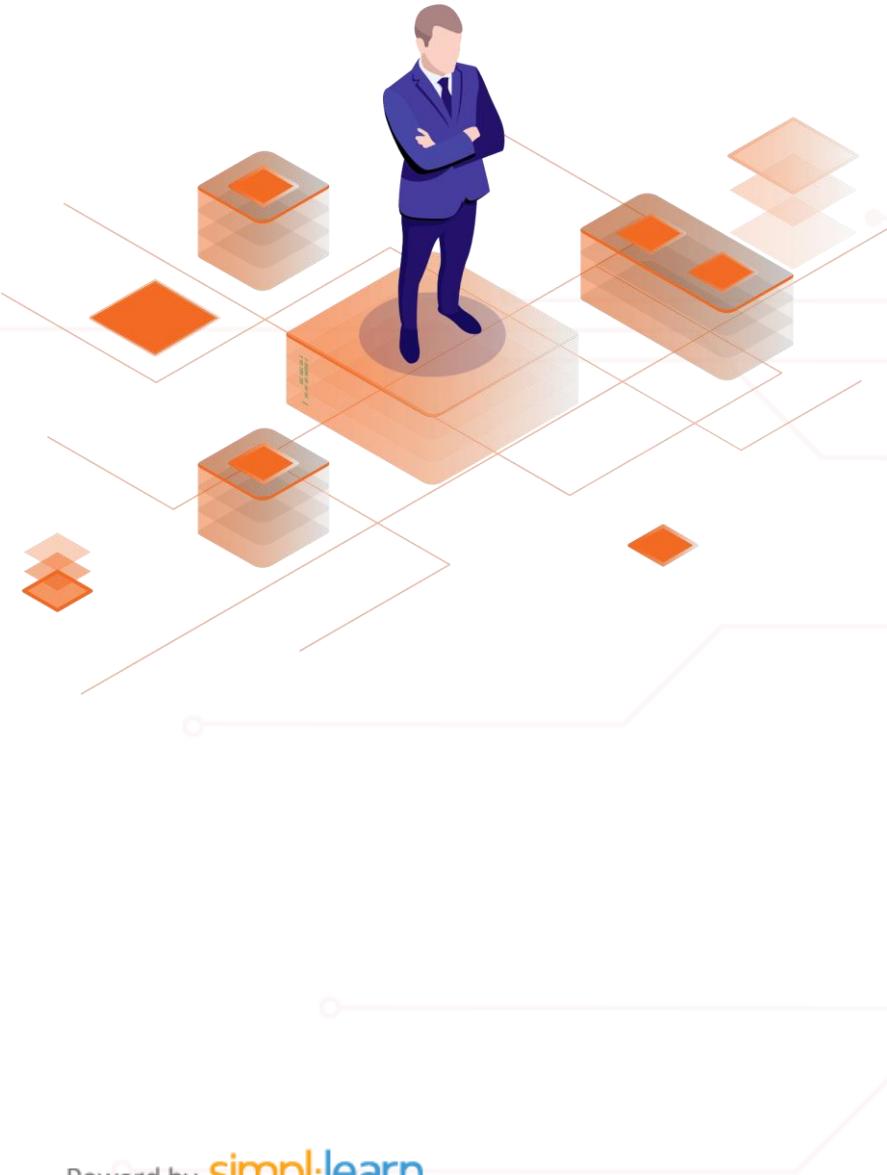
Before the Next Class

You have successfully completed this session. Before next discussion you should go through:

- MongoDB



What's Next?



Now, we have finished the configuration for our Backend Project. In our next live session You will :

- Explore how to create Servlets.
- See how to use JDBC with Servlets.
- Perform CRUD Operations with DB.
- Work with Design Patterns.