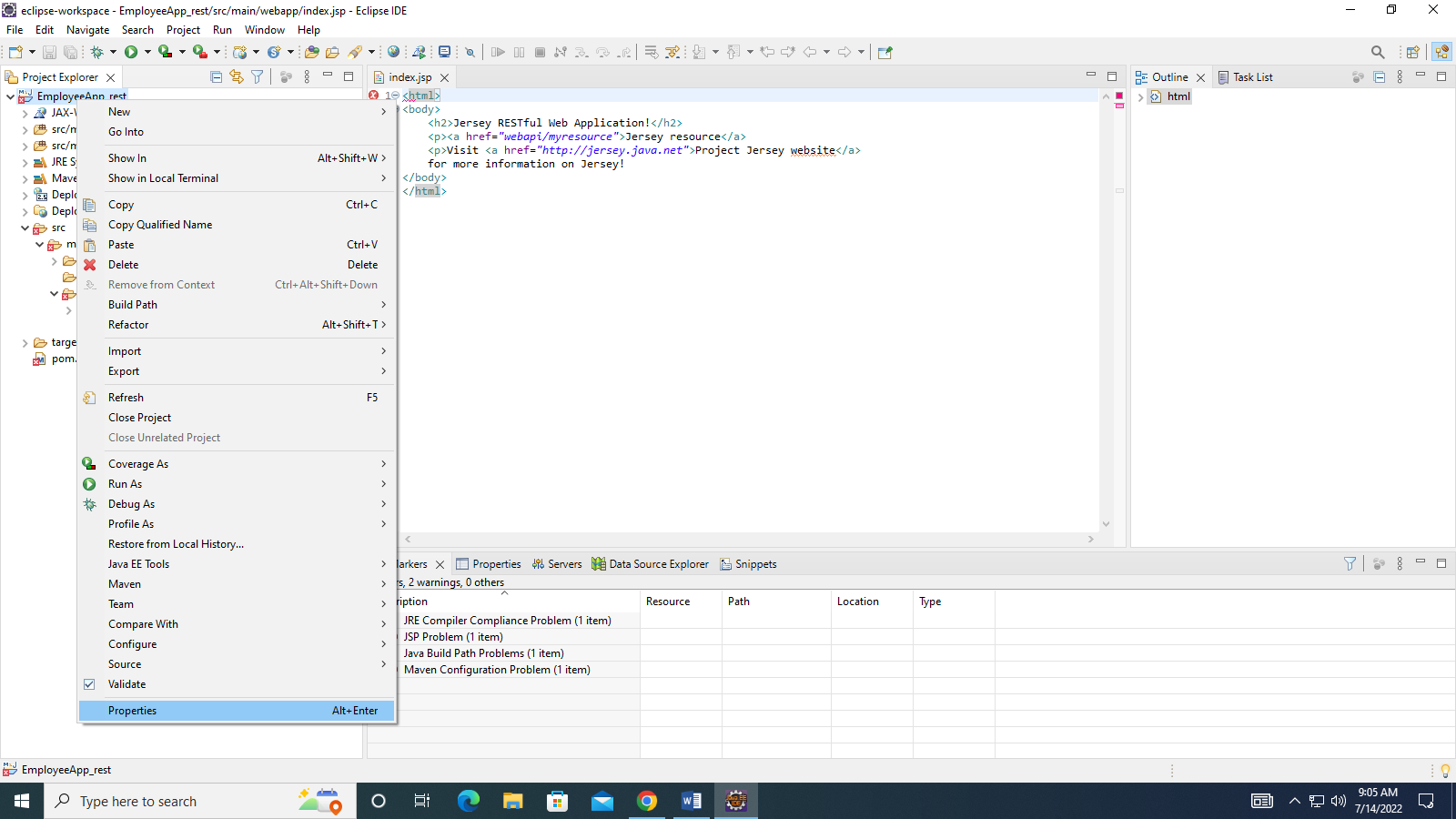
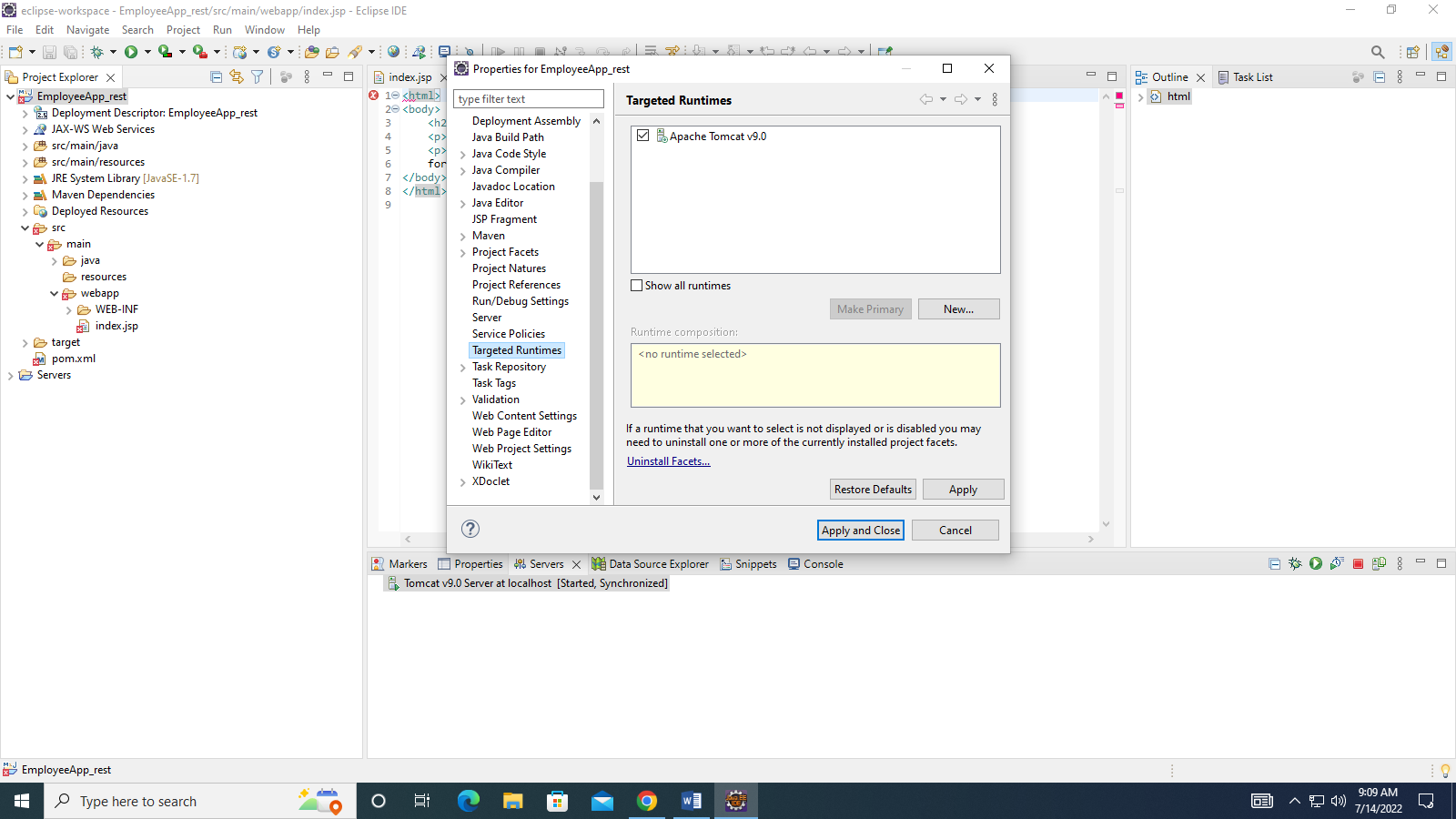
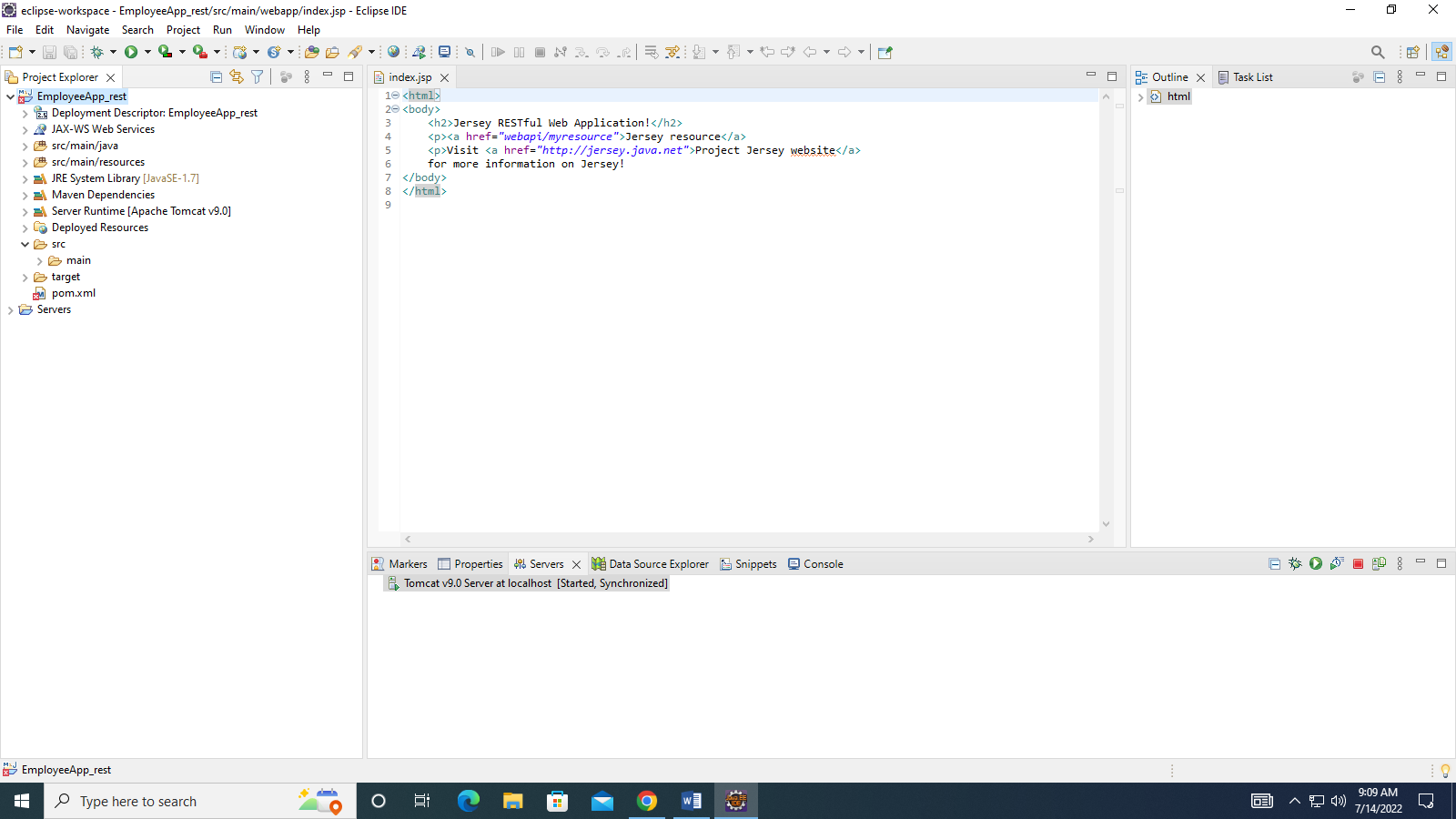


In order to get rid of error in JSP page right click om your project then go to properties and then Targeted Runtime .Check The Box for TomCat 9 Server and Apply & Close.

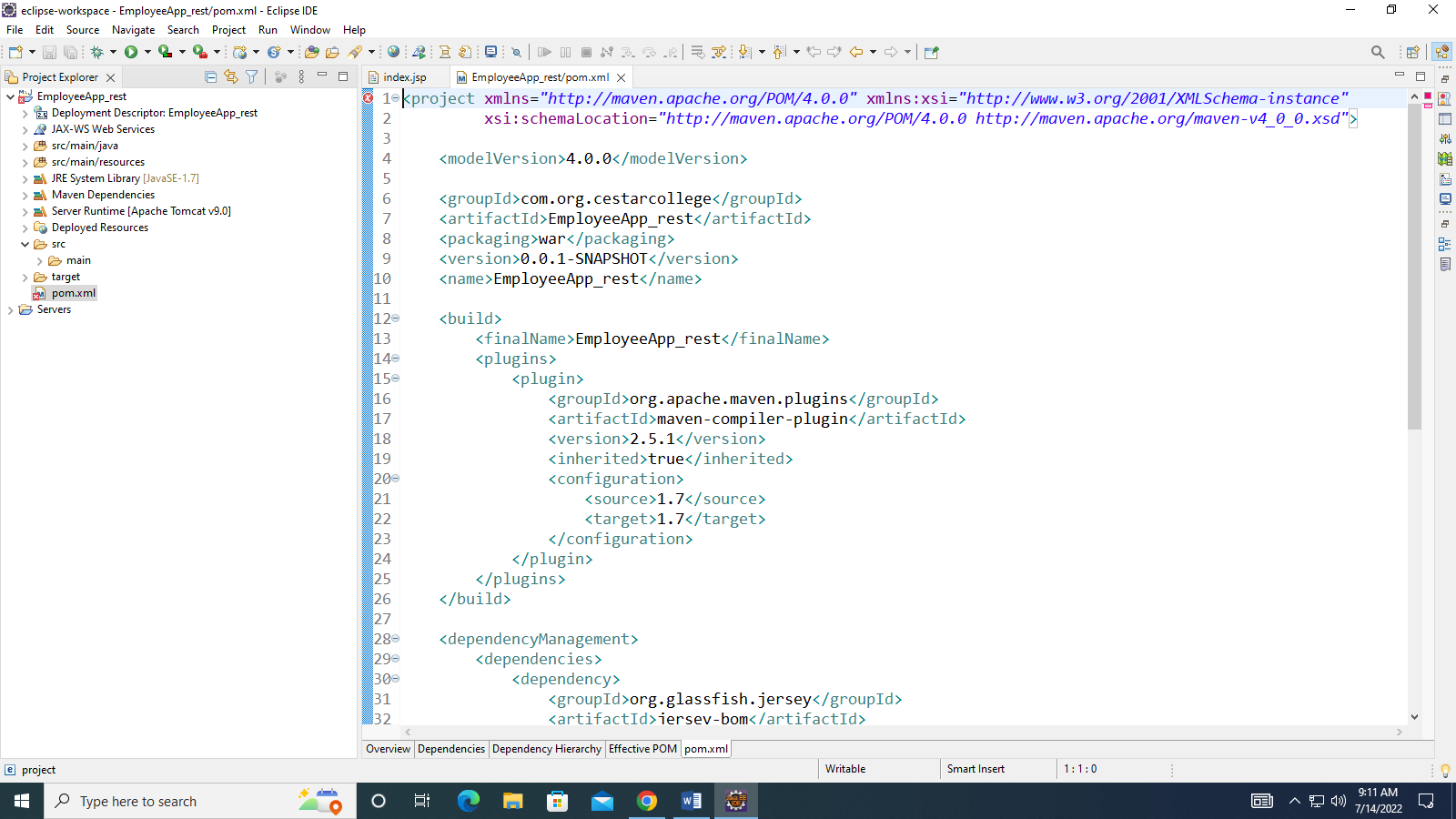


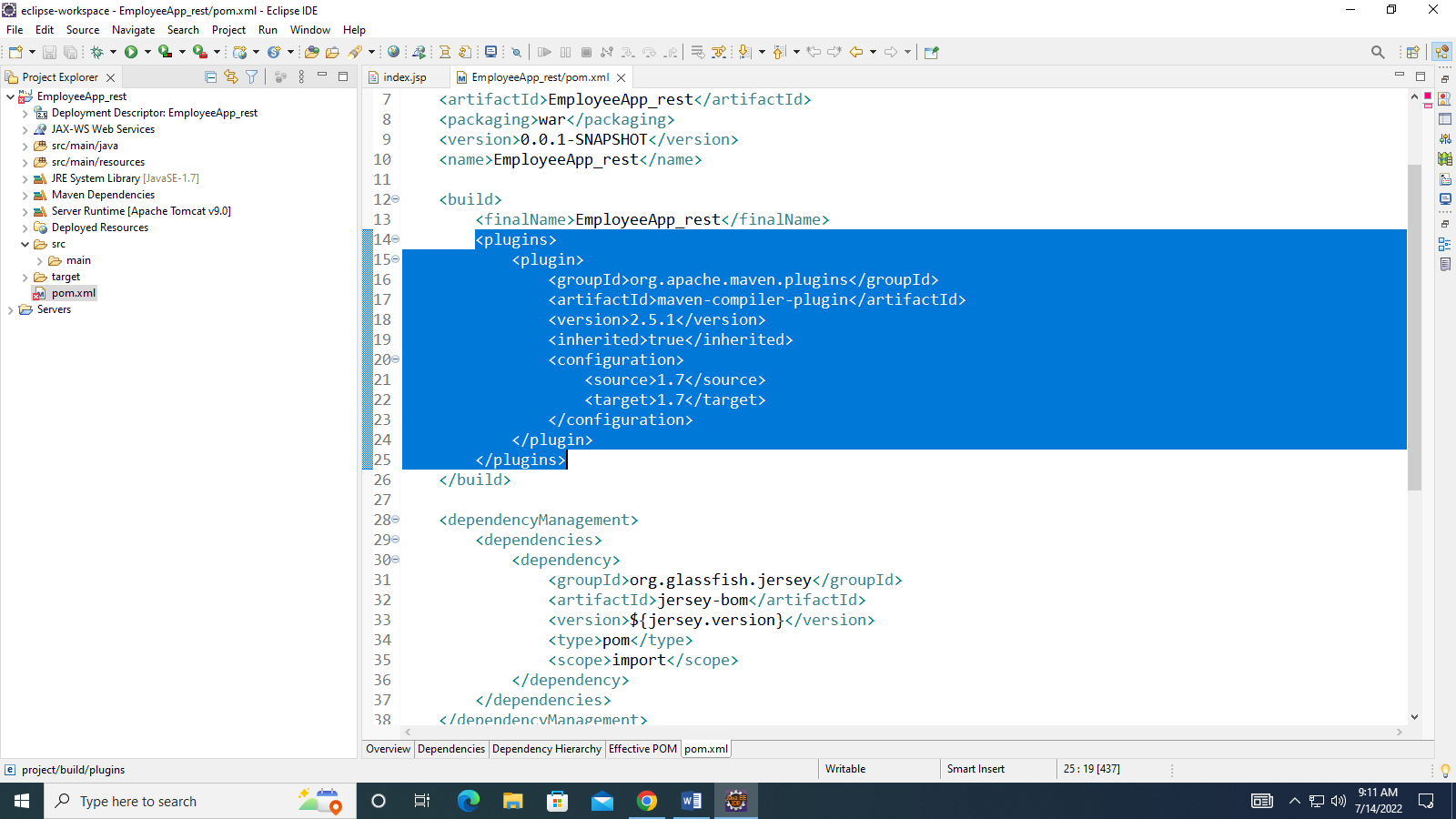


Error Mark disappears in JSP Page



Now Let us look at the error in pom.xml file





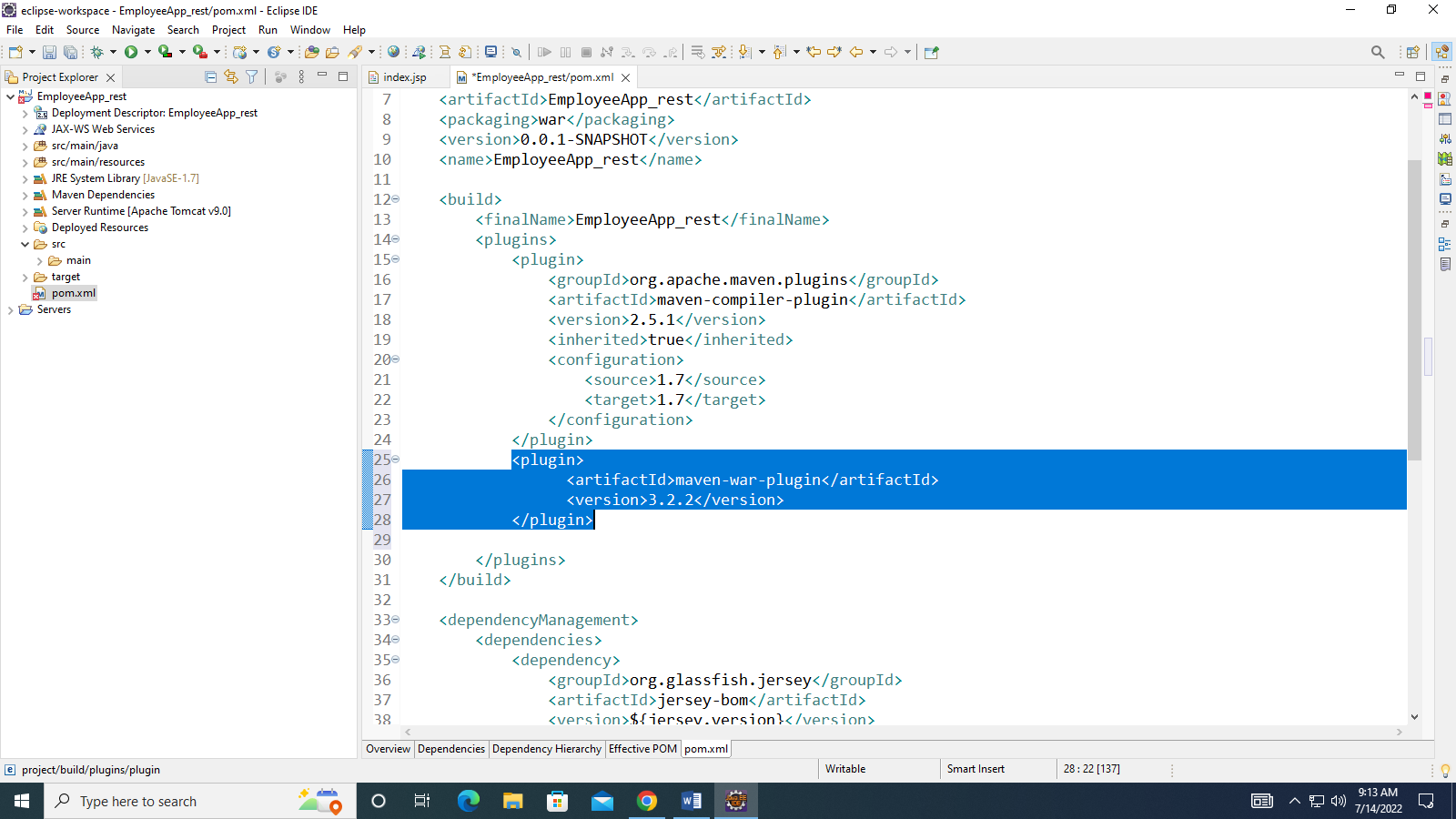
Just Before the Closing </plugins> tag paste this plugin

<plugin>

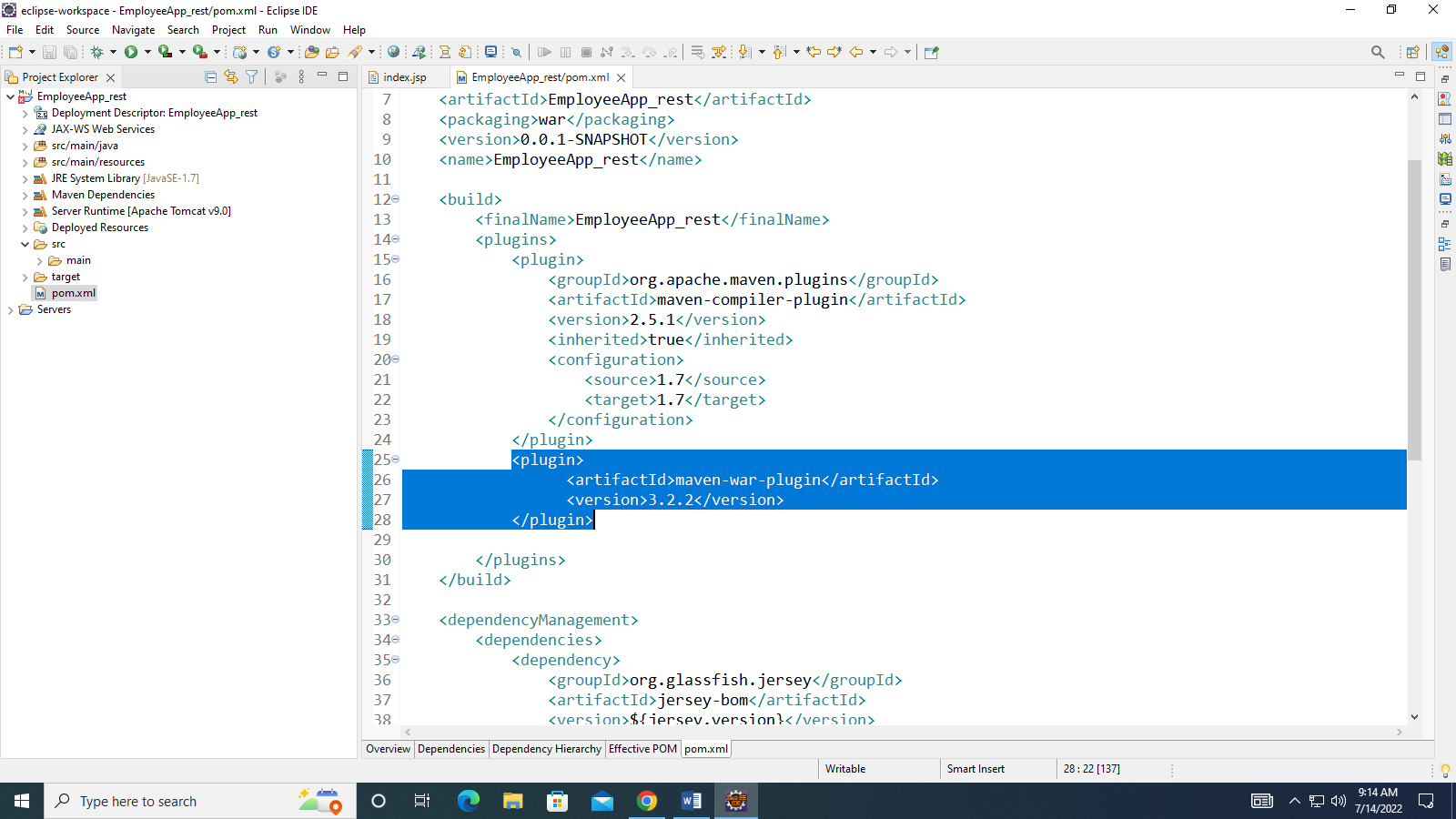
<artifactId>maven-war-plugin</artifactId>

<version>3.2.2</version>

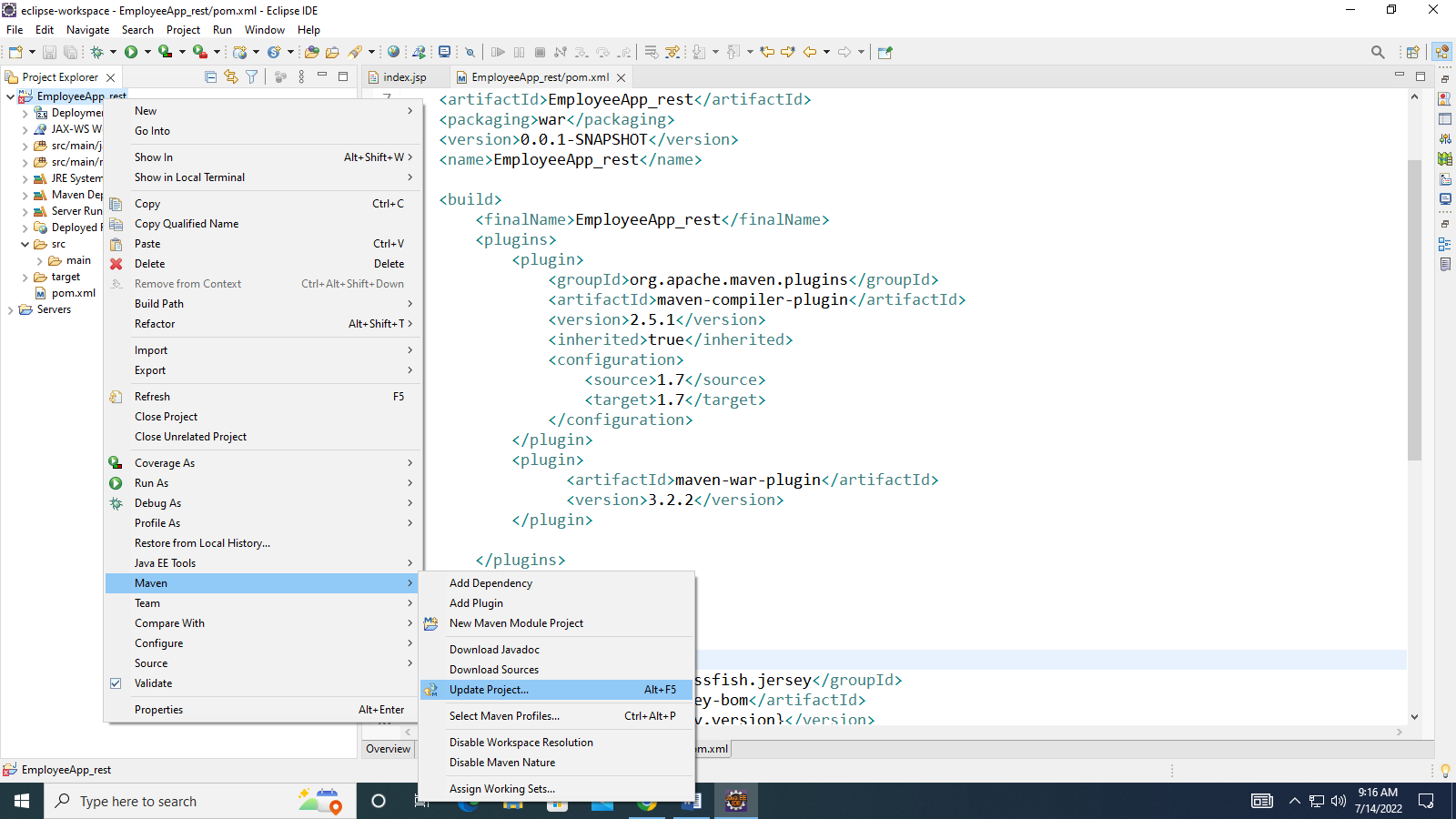
</plugin>

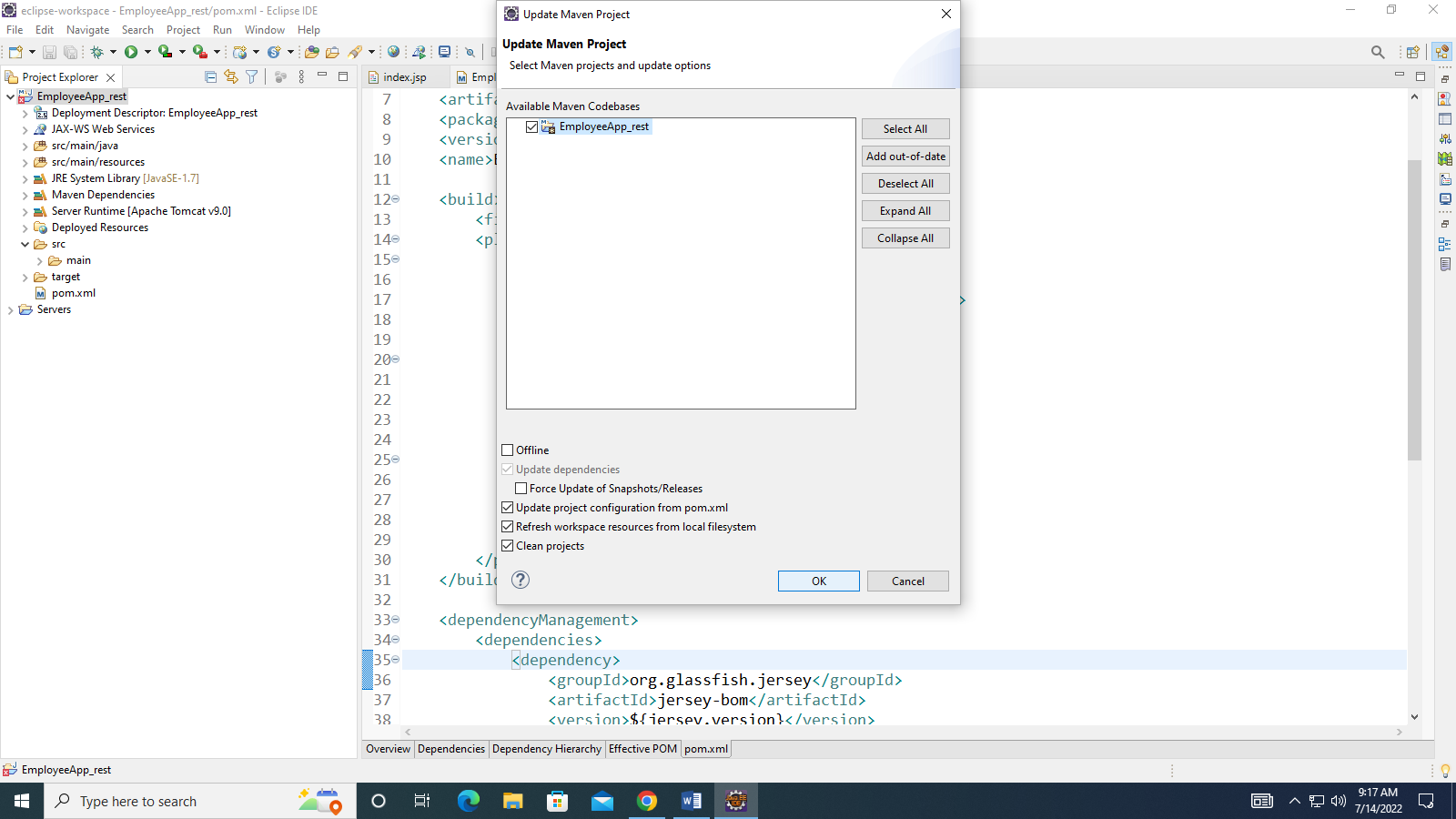


Save the File Using Ctrl+S and error mark disappears.

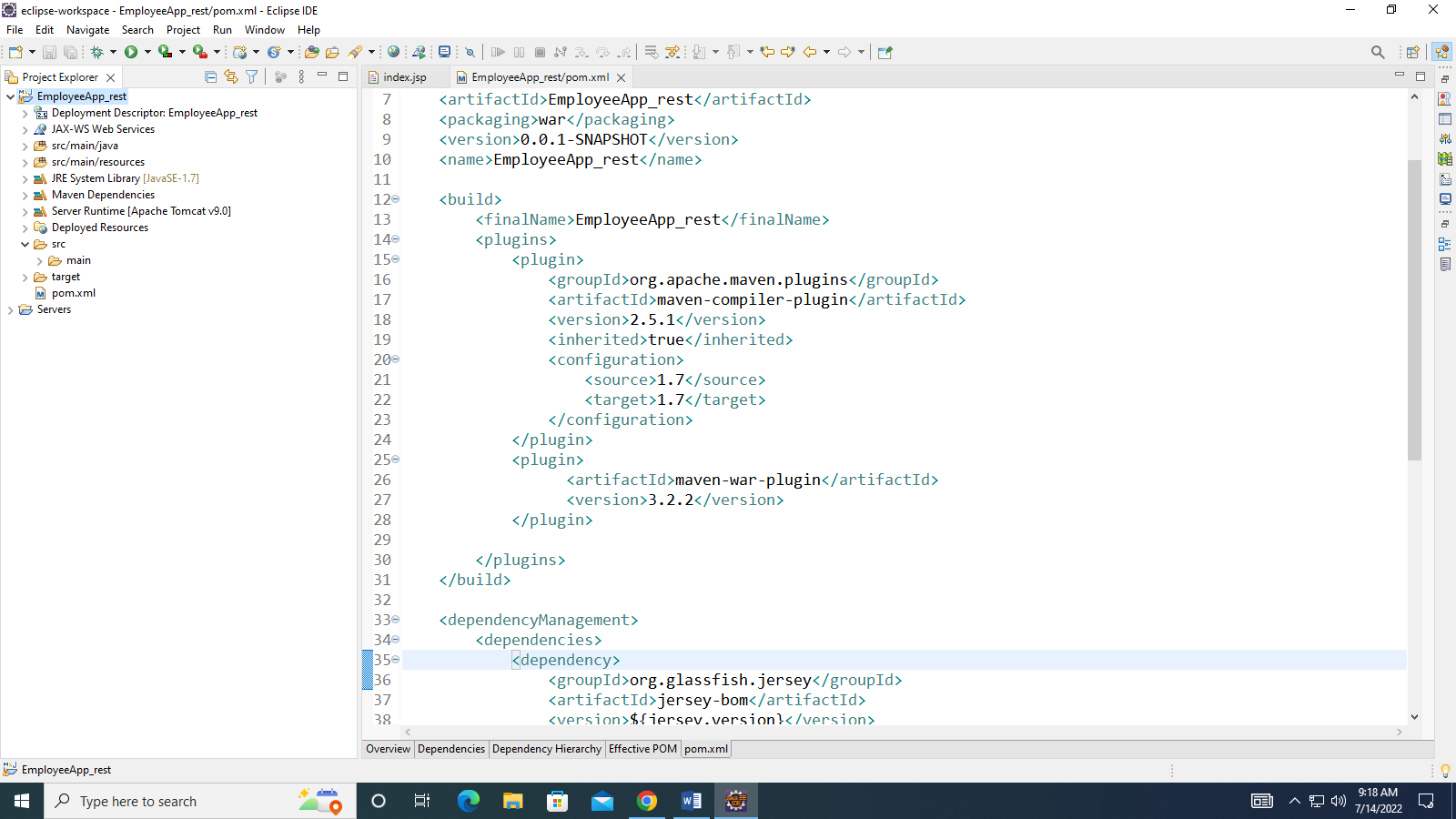


Now Update the Maven Project

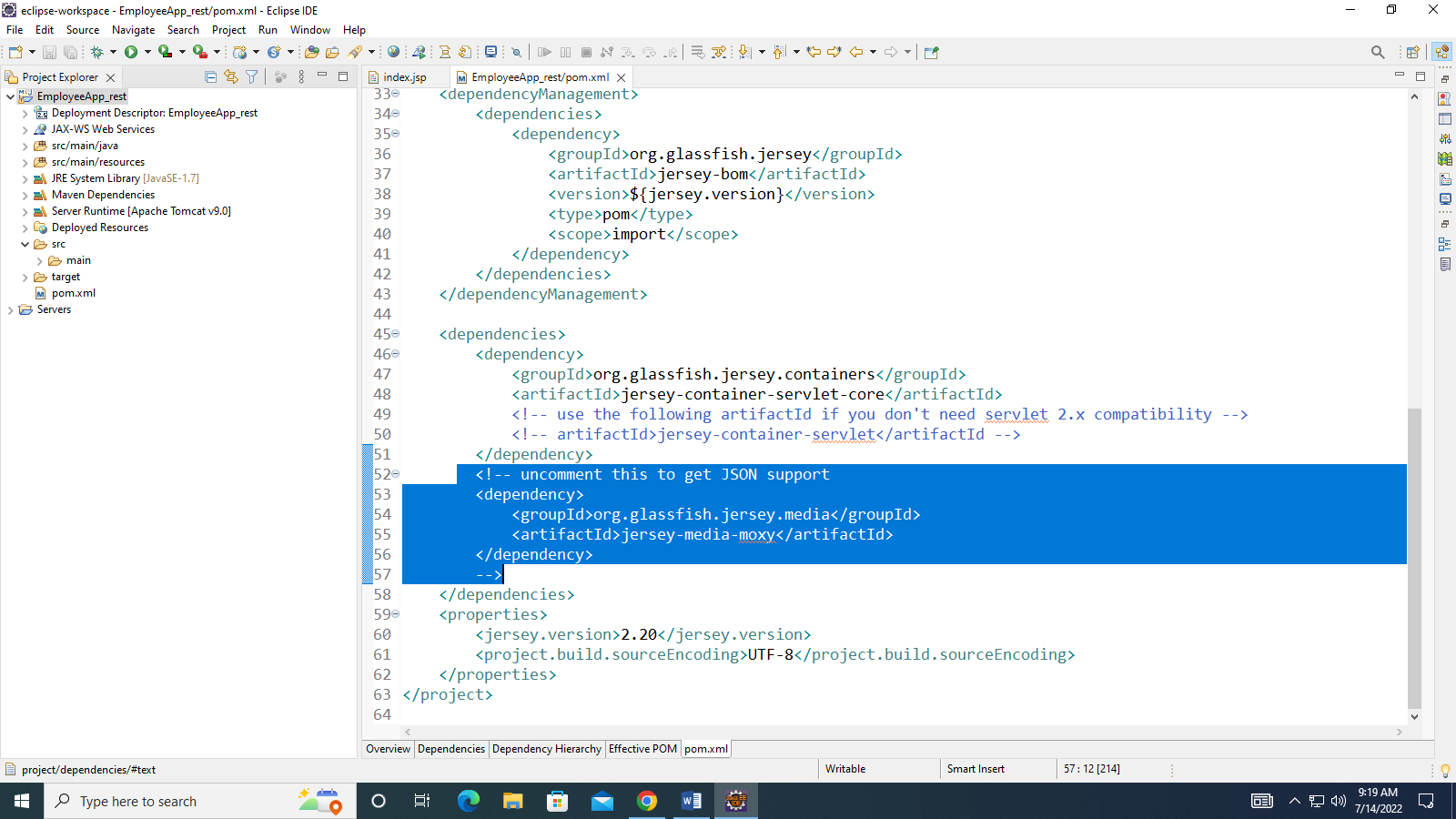




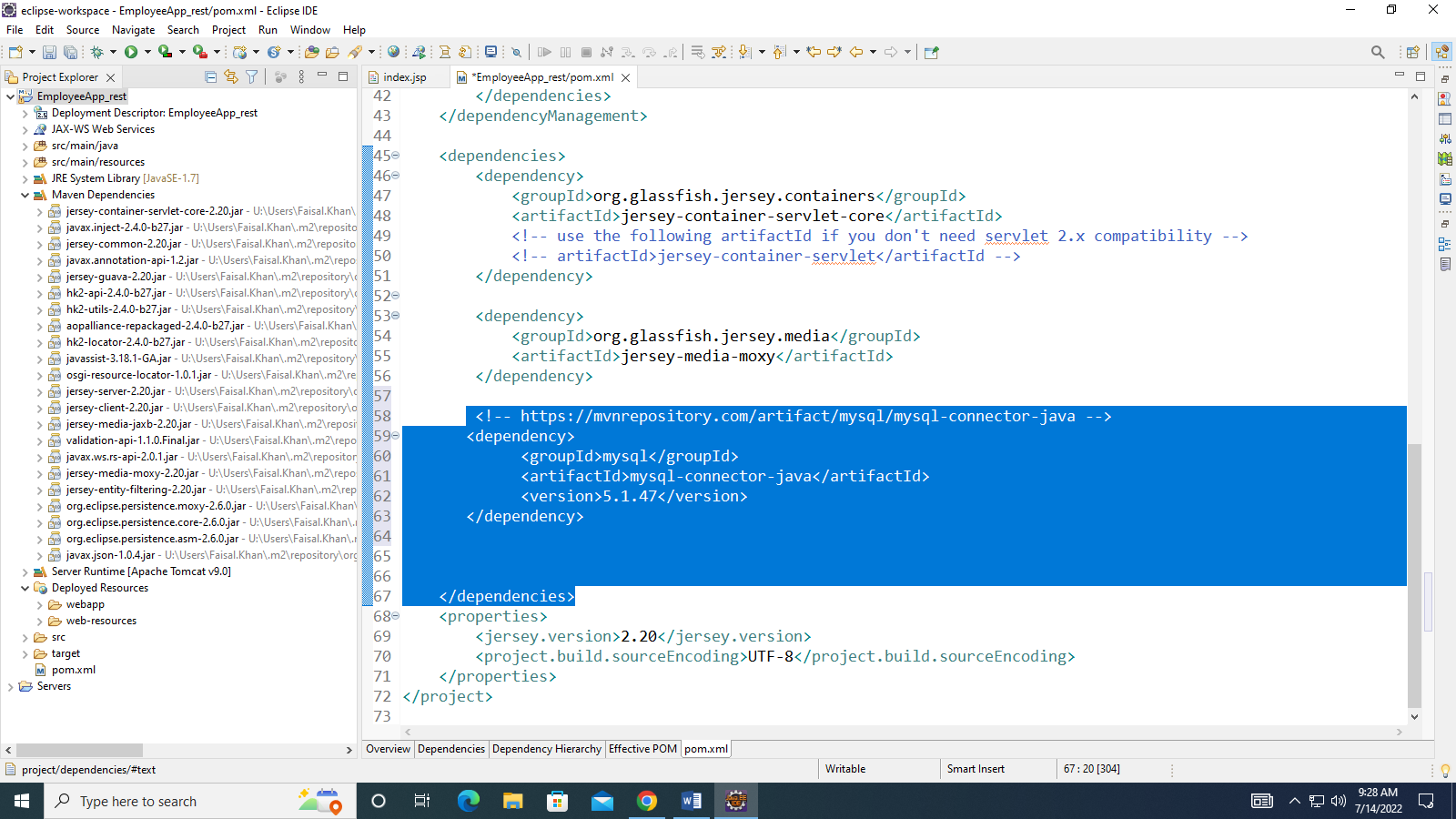
Just Click OK buttan and it will update your Maven Project and error Mark disappears from the Project Folder also



UnComment the Higlighted Dependency for JSON Support



Copy the mysql dependency from maven repository AND PASTE IT JUST BEFORE THE CLOSING </dependencies> tag



Now Copy the dependency for Jaxb API just before the closing dependencies tag

<dependency>

<groupId>javax.xml.bind</groupId>

<artifactId>jaxb-api</artifactId>

<version>2.3.1</version>

</dependency>

For XML Format Add these dependencies just before the

</dependencies> tag

<!-- API, java.xml.bind module -->

<dependency>

<groupId>jakarta.xml.bind</groupId>

<artifactId>jakarta.xml.bind-api</artifactId>

<version>2.3.2</version>

</dependency>

<!-- Runtime, com.sun.xml.bind module -->

<dependency>

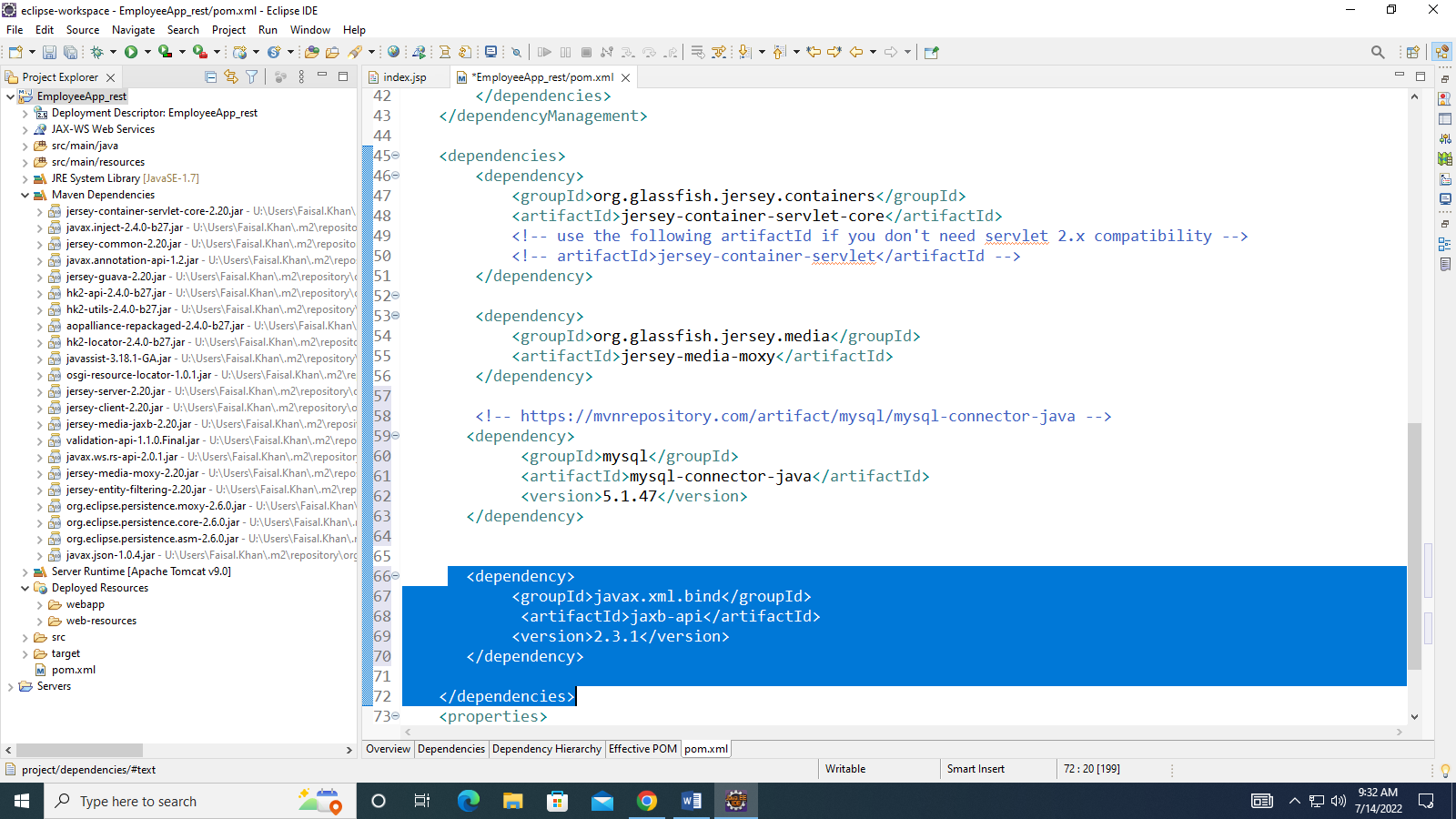
<groupId>org.glassfish.jaxb</groupId>

<artifactId>jaxb-runtime</artifactId>

<version>2.3.2</version>

</dependency>

==================================================================================================================



Now update the Project

Important points to create a web service using Rest Api

HTTP methods

Following four HTTP methods are commonly used in REST based architecture.

GET − Provides a read only access to a resource.

POST − Used to create a new resource.

DELETE − Used to remove a resource.

PUT − Used to update a existing resource or create a new resource.

HTTP Code

|  |  |
| --- | --- |
| Sr.No. | HTTP Code & Description |
| 1 | 200  OK − shows success. |
| 2 | 201  CREATED − when a resource is successfully created using POST or PUT request. Returns link to the newly created resource using the location header. |
| 3 | 204  NO CONTENT − when response body is empty. For example, a DELETE request. |
| 4 | 304  NOT MODIFIED − used to reduce network bandwidth usage in case of conditional GET requests. Response body should be empty. Headers should have date, location, etc. |
| 5 | 400  BAD REQUEST − states that an invalid input is provided. For example, validation error, missing data. |
| 6 | 401  UNAUTHORIZED − states that user is using invalid or wrong authentication token. |
| 7 | 403  FORBIDDEN − states that the user is not having access to the method being used. For example, Delete access without admin rights. |
| 8 | 404  NOT FOUND − states that the method is not available. |
| 9 | 409  CONFLICT − states conflict situation while executing the method. For example, adding duplicate entry. |
| 10 | 500  INTERNAL SERVER ERROR − states that the server has thrown some exception while executing the method. |

Following are the most commonly used annotations to map a resource as a web service resource.

|  |  |
| --- | --- |
| Sr.No. | Annotation & Description |
| 1 | @Path  Relative path of the resource class/method. |
| 2 | @GET  HTTP Get request, used to fetch resource. |
| 3 | @PUT  HTTP PUT request, used to update resource. |
| 4 | @POST  HTTP POST request, used to create a new resource. |
| 5 | @DELETE  HTTP DELETE request, used to delete resource. |
| 6 | @HEAD  HTTP HEAD request, used to get status of method availability. |
| 7 | @Produces  States the HTTP Response generated by web service. For example, APPLICATION/XML, TEXT/HTML, APPLICATION/JSON etc. |
| 8 | @Consumes  States the HTTP Request type. For example, application/x-www-formurlencoded to accept form data in HTTP body during POST request. |
| 9 | @PathParam  Binds the parameter passed to the method to a value in path. |
| 10 | @QueryParam  Binds the parameter passed to method to a query parameter in the path. |
| 11 | @MatrixParam  Binds the parameter passed to the method to a HTTP matrix parameter in path. |
| 12 | @HeaderParam  Binds the parameter passed to the method to a HTTP header. |
| 13 | @CookieParam  Binds the parameter passed to the method to a Cookie. |
| 14 | @FormParam  Binds the parameter passed to the method to a form value. |
| 15 | @DefaultValue  Assigns a default value to a parameter passed to the method. |
| 16 | @Context  Context of the resource. For example, HTTPRequest as a context. |

Steps to create a web service using RestApi

1-Create a Maven Project or add jars to a normal dynamic web project.

Creating a Jersey Project in Eclipse

Use Add Archetype Button

Group ID = org.glassfish.jersey.archetypes

Artifact Id = jersy-quickstart-webapp

Version = 2.20

Please copy this plugin in your pom.xml file inside <pulgins> </plugins>

<plugin>

<artifactId>maven-war-plugin</artifactId>

<version>3.2.2</version>

</plugin>

Please Add this Dependency inside <dependencies> </ dependencies> in pom.xml

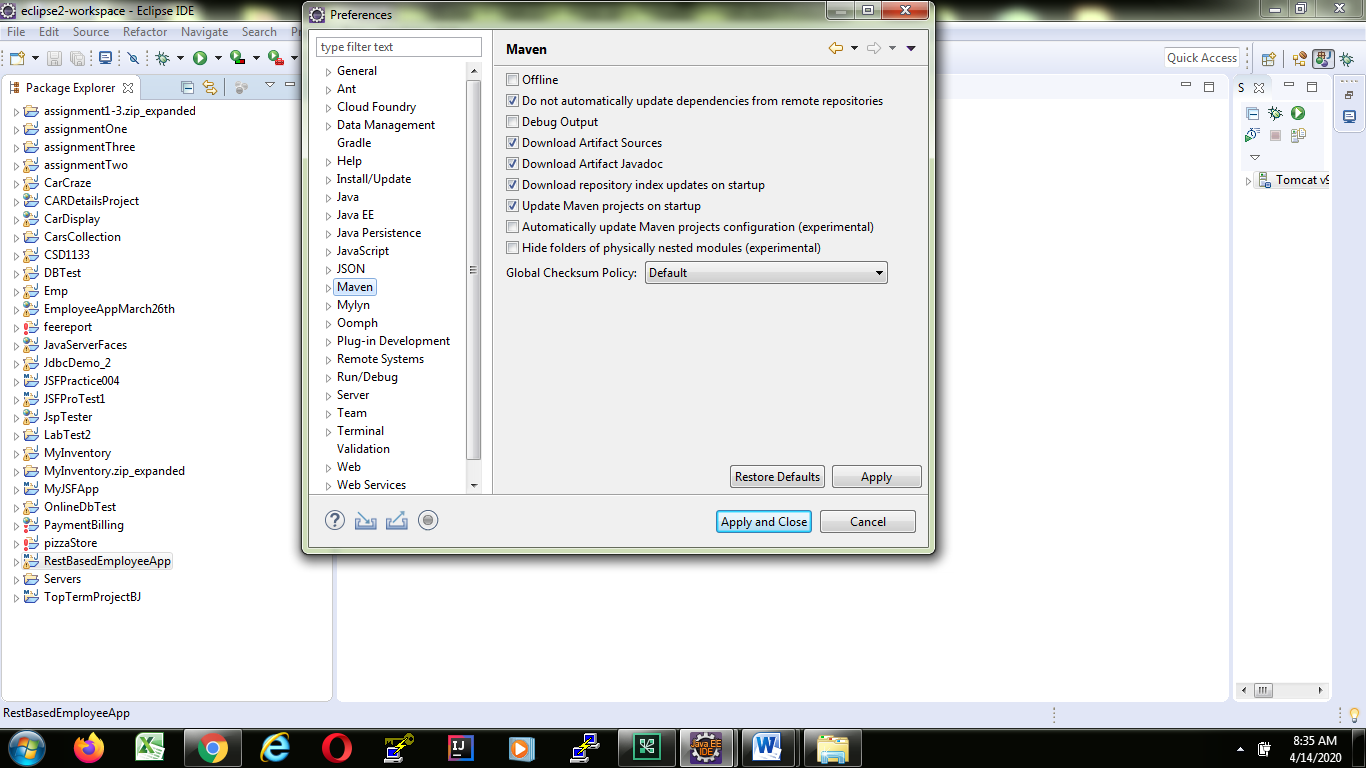
<dependency>

<groupId>javax.xml.bind</groupId>

<artifactId>jaxb-api</artifactId>

<version>2.3.1</version>

</dependency>



Use

2-We need to know what Jercy implementation is

3-We have to know what is the use of @url annotation

4-We have to know what a resource is.

* **GET** − Provides a read only access to a resource.
* **POST** − Used to create a new resource.
* **DELETE** − Used to remove a resource.
* **PUT** − Used to update a existing resource or create a new resource.

Download the latest version of Jersey framework binaries from the following link – <https://jersey.java.net/download.html>.

5-We have to what is a Rest Client.

6-How to install postman as a Rest Client.

7.What is the use @PathParam annotation

8-When do we use @Produces

9-When do we use @Consumes

10-Create the project both ways.

====================All Steps at One Place=========================================

1- Creatre a Jersey-quickstaet-webapp in maven.

2-Right click on your Project Folder abd select Tomcat 9 as your Targeted run time

3- Add this plugin before </plugins>

<plugin>

<artifactId>maven-war-plugin</artifactId>

<version>3.2.2</version>

</plugin>

4- Add these dependencies just before the </dependencies> tag

<!-- Adding the two dependencies below for XML support from Word Document -->

<!-- API, java.xml.bind module -->

<dependency>

<groupId>jakarta.xml.bind</groupId>

<artifactId>jakarta.xml.bind-api</artifactId>

<version>2.3.2</version>

</dependency>

<!-- Runtime, com.sun.xml.bind module -->

<dependency>

<groupId>org.glassfish.jaxb</groupId>

<artifactId>jaxb-runtime</artifactId>

<version>2.3.2</version>

</dependency>

<!-- Adding mysql Connector 5.1.47 from Maven Repository -->

<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.47</version>

</dependency>

<!-- Adding dependency for JaxB API from Word Document -->

<dependency>

<groupId>javax.xml.bind</groupId>

<artifactId>jaxb-api</artifactId>

<version>2.3.1</version>

</dependency>

=====================================

Use This Annotation at the top of your Employee Class

import javax.xml.bind.annotation.XmlRootElement;

//In order to get the Employee Object in XML Format make sure to use this annotation

// at the top of the class name

@XmlRootElement