CURSO DE JAVA EE

EXERCISE

WEB APPLICATION WITH SERVERS AND JPS IN JAVA EE

Global Mentoring

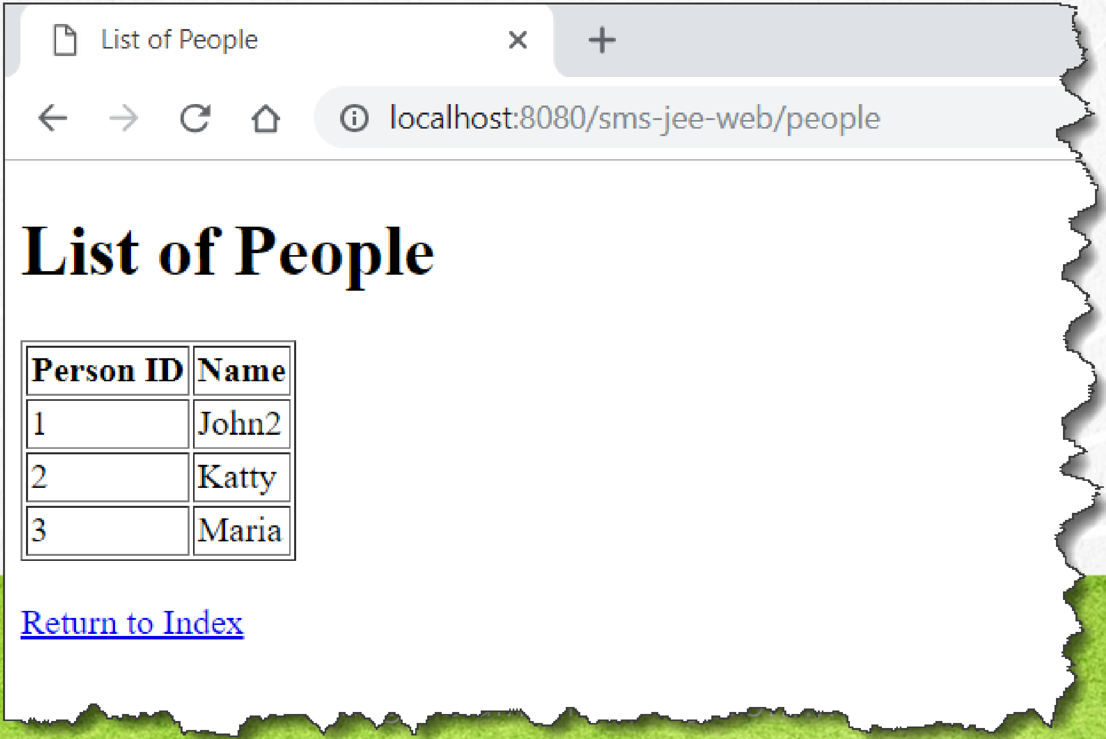
Experiencia y Conocimiento para tu vida

CURSO DE JAVA EE

www.globalmentoring.com.mx

EXERCISE OBJECTIVE

The objective of the exercise is to create a Web application that List People using the architecture of JSP, Servlets, EJB and JPA.



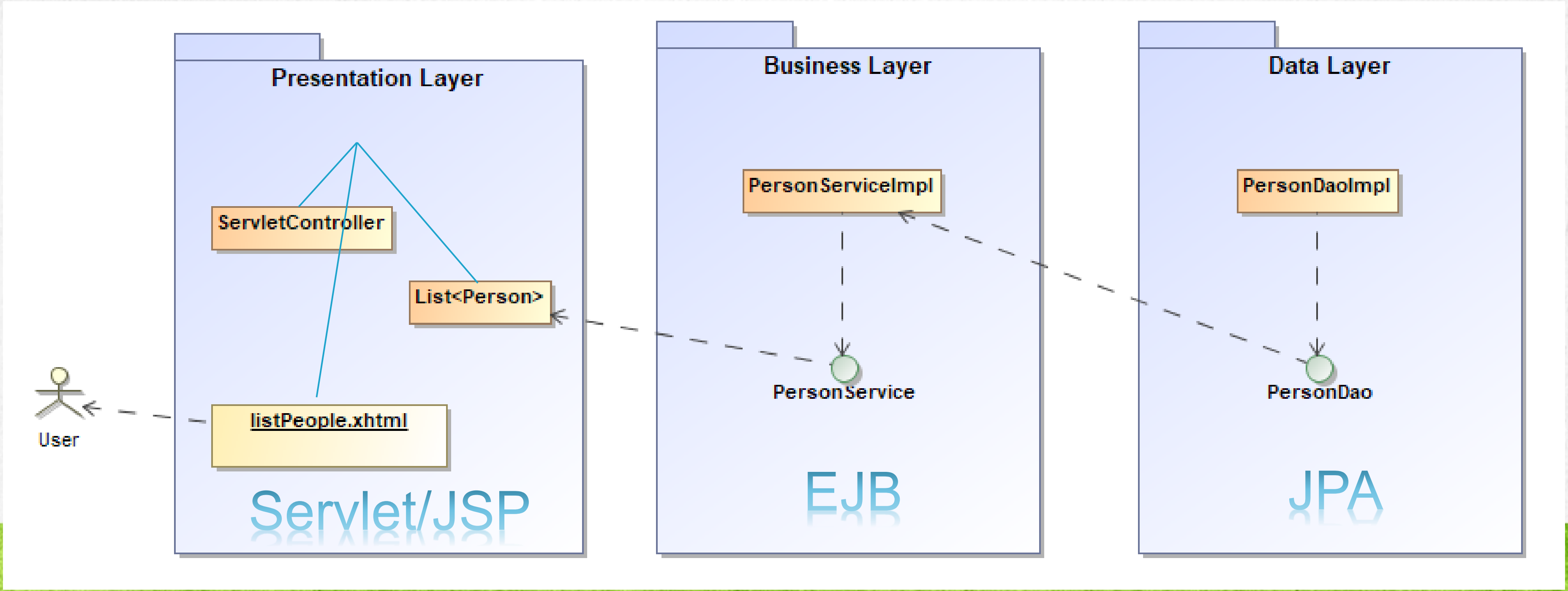
List of People

Person ID	Name
1	John2
2	Katty
3	Maria

Return to Index

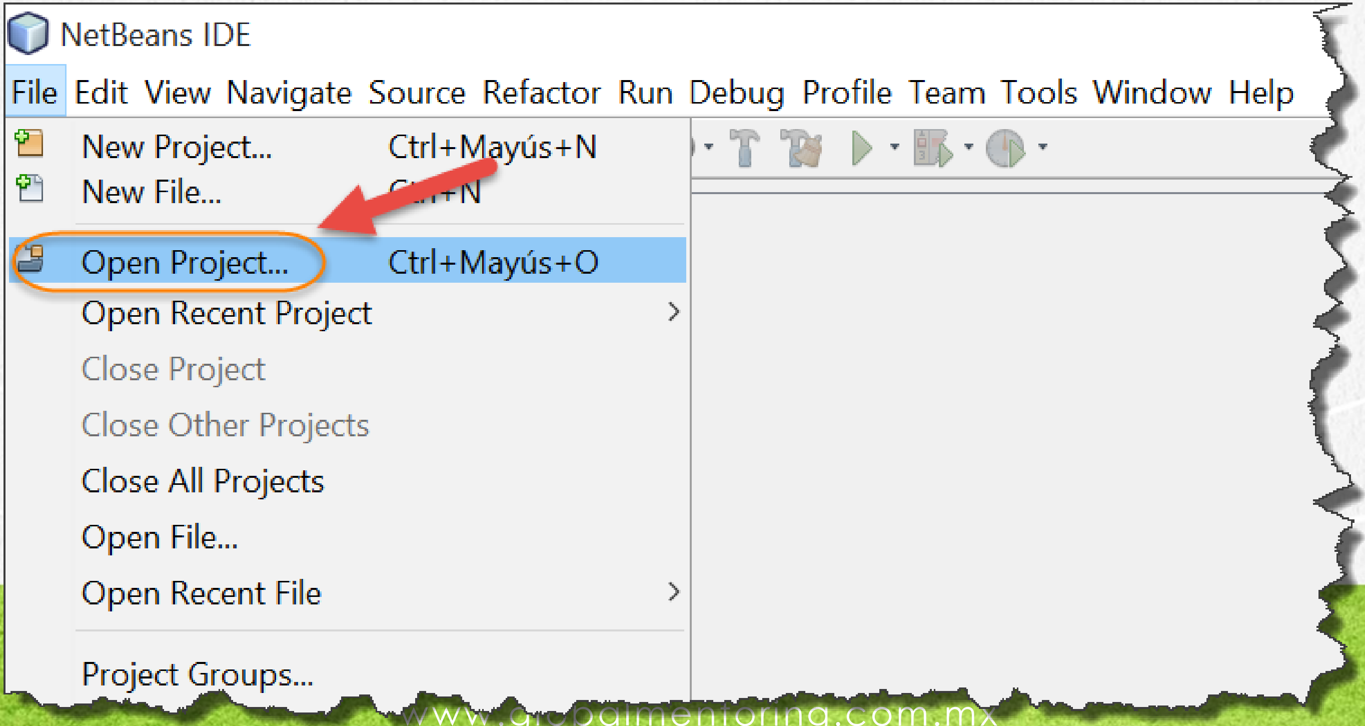
CLASS DIAGRAM

This is the Exercise Class Diagram, where you can see the 3-layer Architecture of our System.



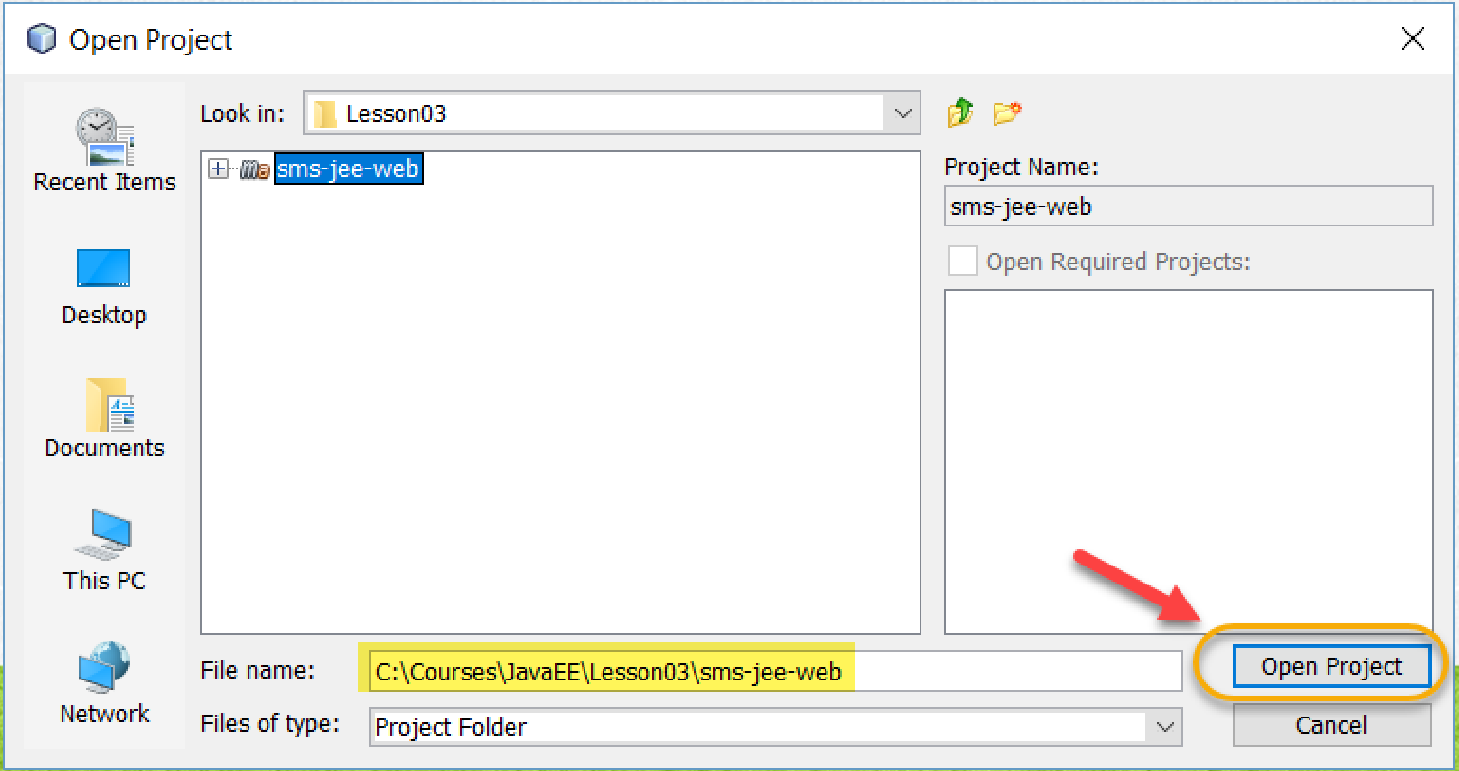
1. OPEN THE PROJECT

In case we do not have open the project sga-jee-web we open it (the latest version):



1. OPEN PROJECT

In case we do not have open the sga-jee-web project we open it:


Open Project

Look in: Lesson03

sms-jee-web

Project Name: sms-jee-web

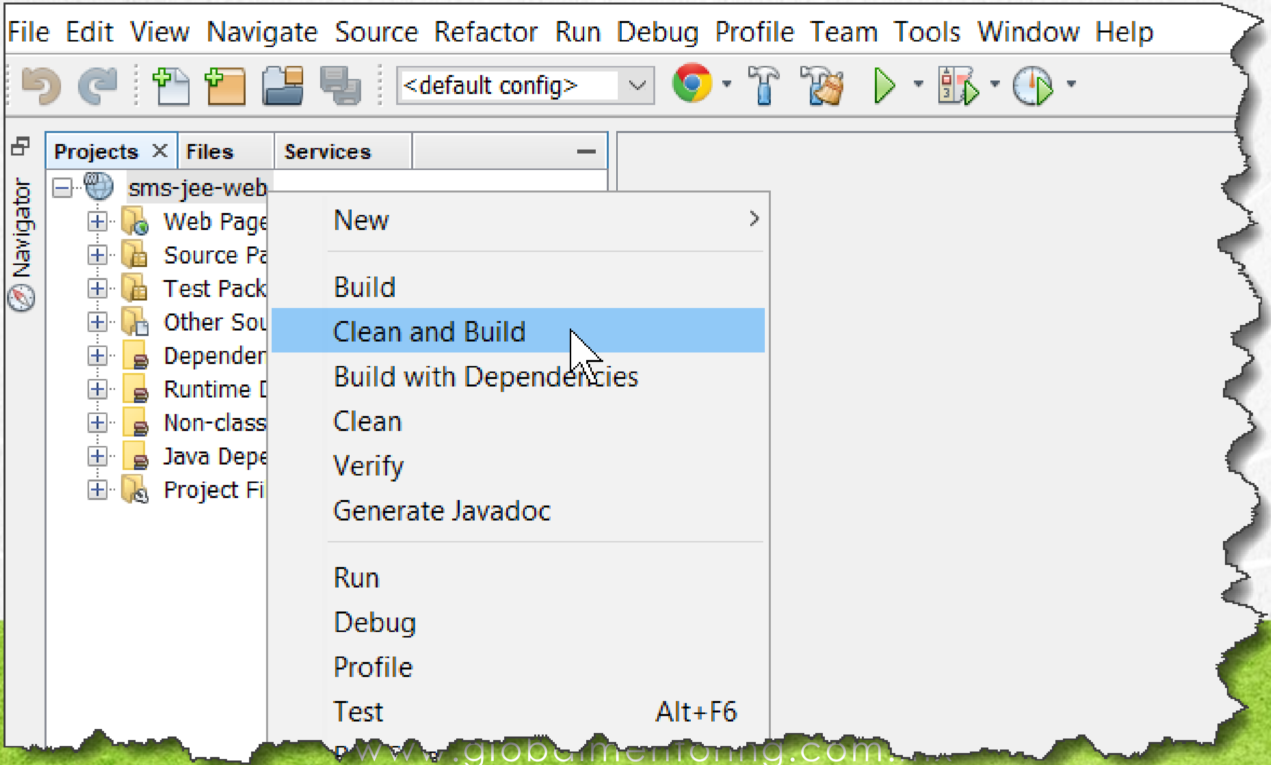
Open Required Projects:

File name: C:\Courses\JavaEE\Lesson03\sms-jee-web

Files of type: Project Folder

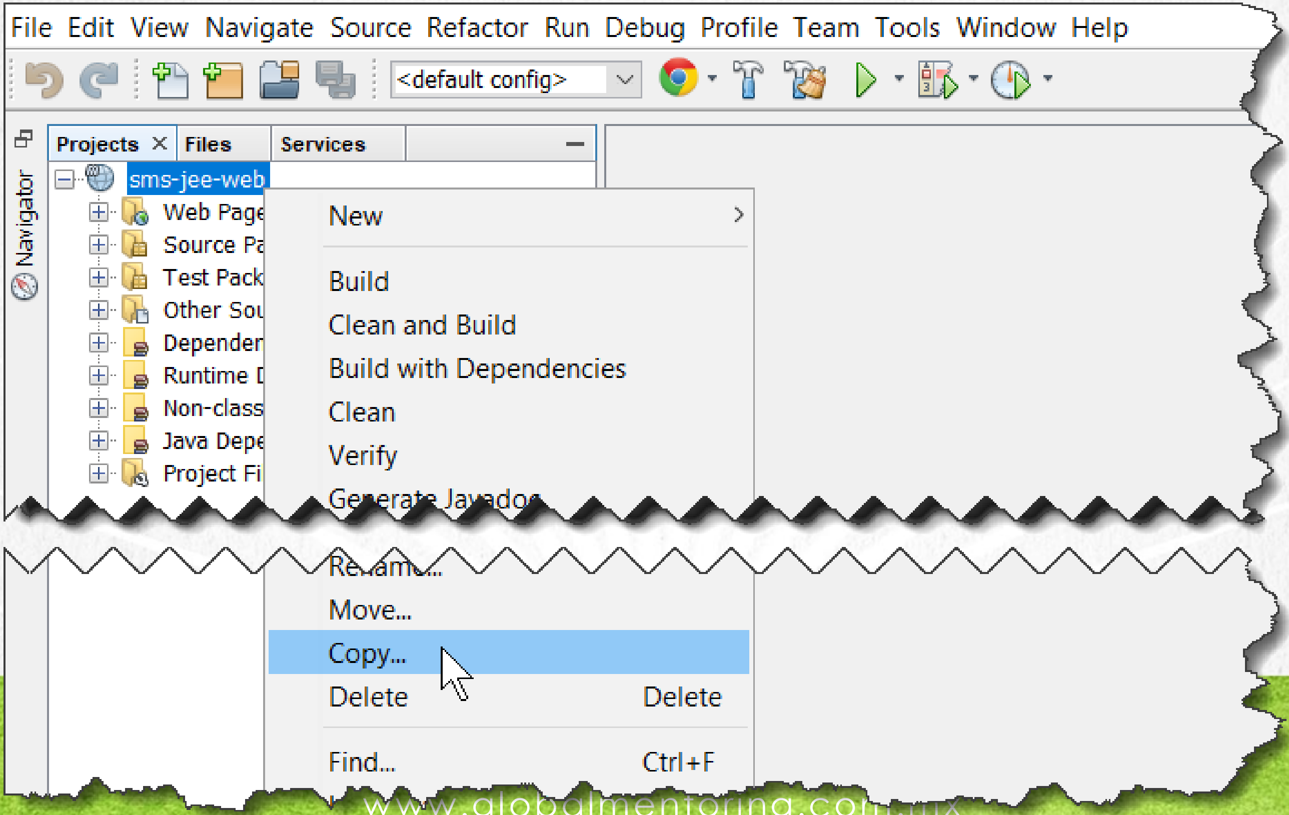
1. OPEN THE PROJECT

We wait for you to fully load the project. In case the project makes a mistake, we make a Clean & Build so that all the files are shown, this step is optional:



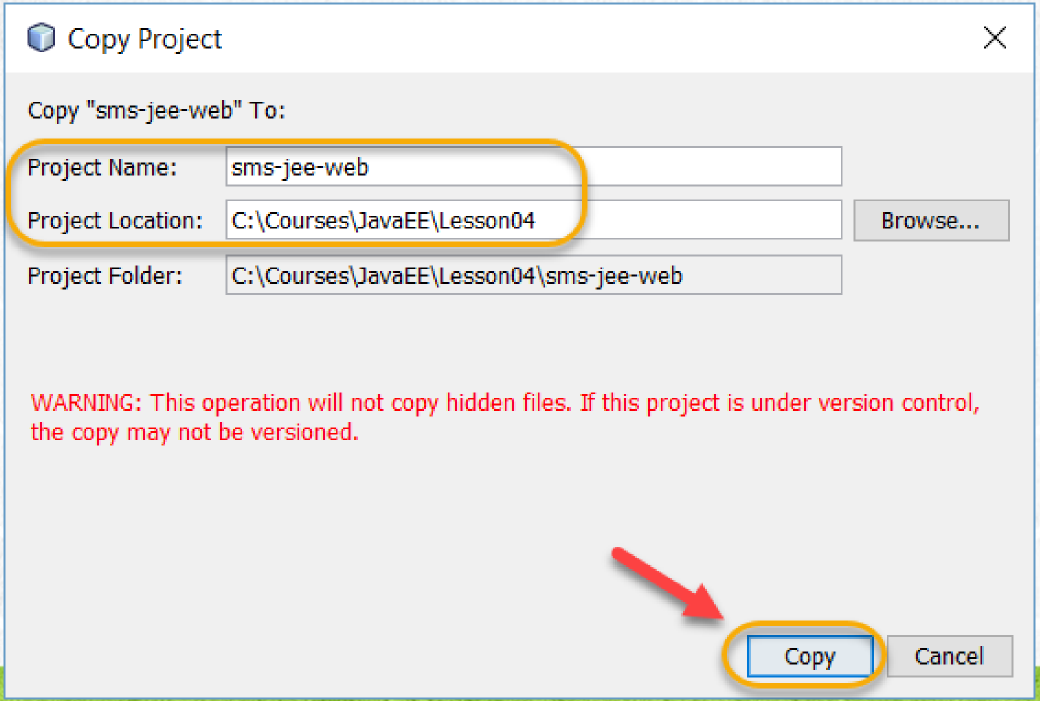
2. COPY THE PROJECT

We copy the project to put it in the new path:



2. COPY THE PROJECT

We copy the project to put it in the new path:



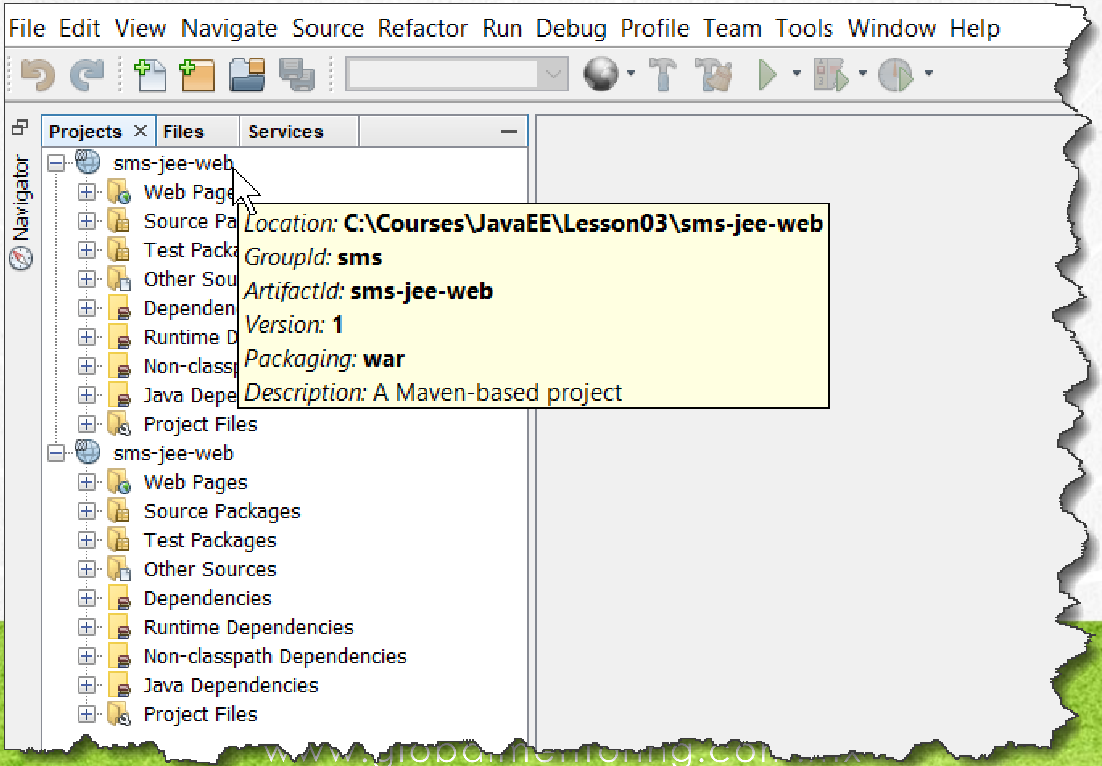
Copy "sms-jee-web" To:

Project Name:	sms-jee-web
Project Location:	C:\Courses\JavaEE\Lesson04
Project Folder:	C:\Courses\JavaEE\Lesson04\sms-jee-web

WARNING: This operation will not copy hidden files. If this project is under version control, the copy may not be versioned.

2. CLOSE THE PROJECT

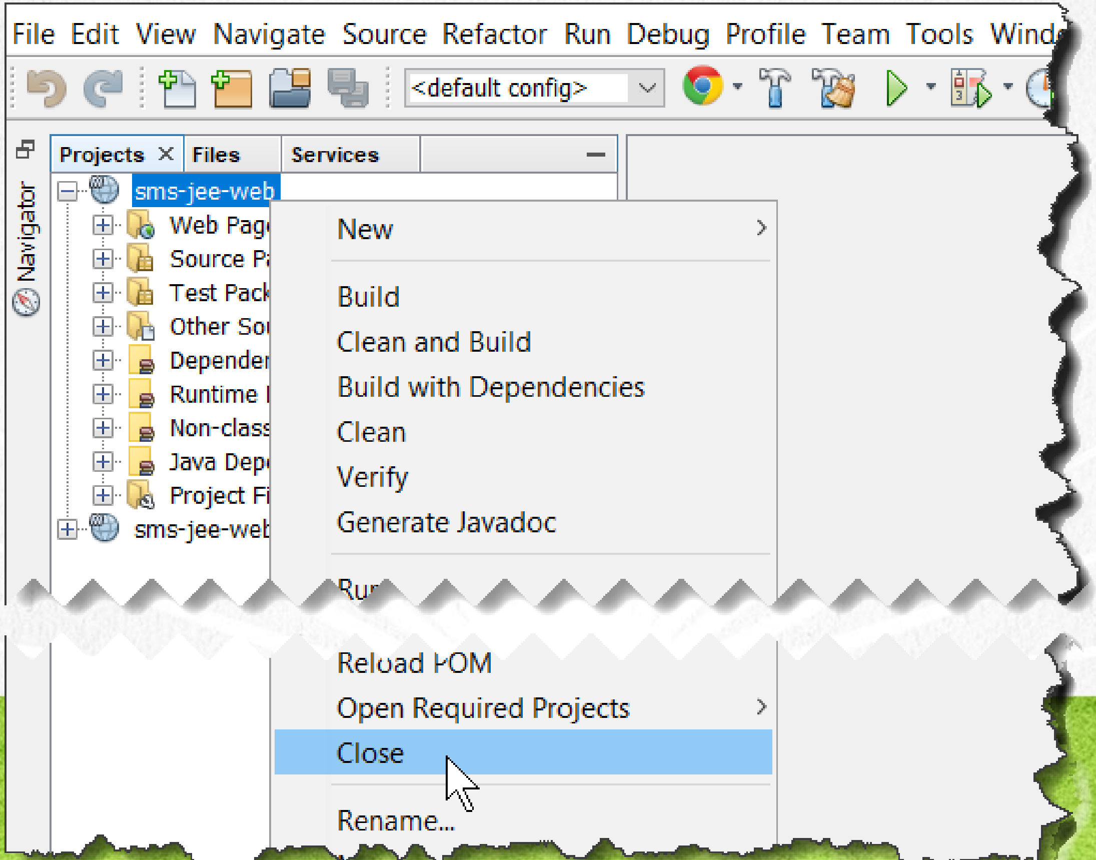
We closed the previous project and left only the new one. We identify the project to close by placing the cursor on the project:



Location: **C:\Courses\JavaEE\Lesson03\sms-jee-web**
GroupId: **sms**
ArtifactId: **sms-jee-web**
Version: **1**
Packaging: **war**
Description: A Maven-based project

2. CLOSE THE PROJECT

We closed the previous project and left only the new one:



MODIFY THE VIEW

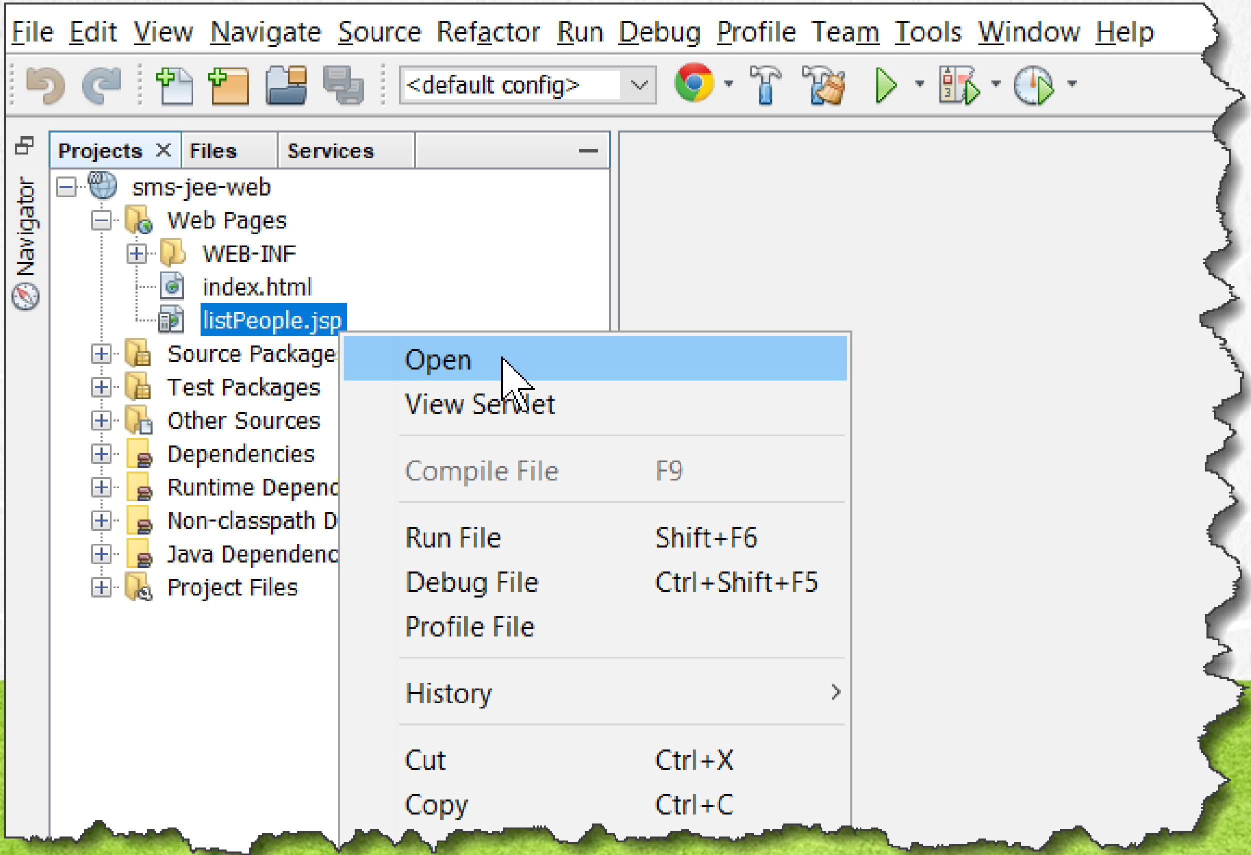
Because our project is already a Web project, we have already included everything necessary for the topic of Servlets and JSPs. We have created the controller that is a Servlets, the view that are the JSPs, and the Entity classes that are the Model.

With this we are formalizing the MVC (Model - View - Controller) design pattern in our Web application, so we are ready to add more features to our JSPs pages.

We are going to prepare our view to manage an HTML table that shows the list of People in the view. Let's see the necessary changes in our view or JSP.

3. Modify the JSP

Modify the listPeople.jsp:



3. MODIFY THE FILE

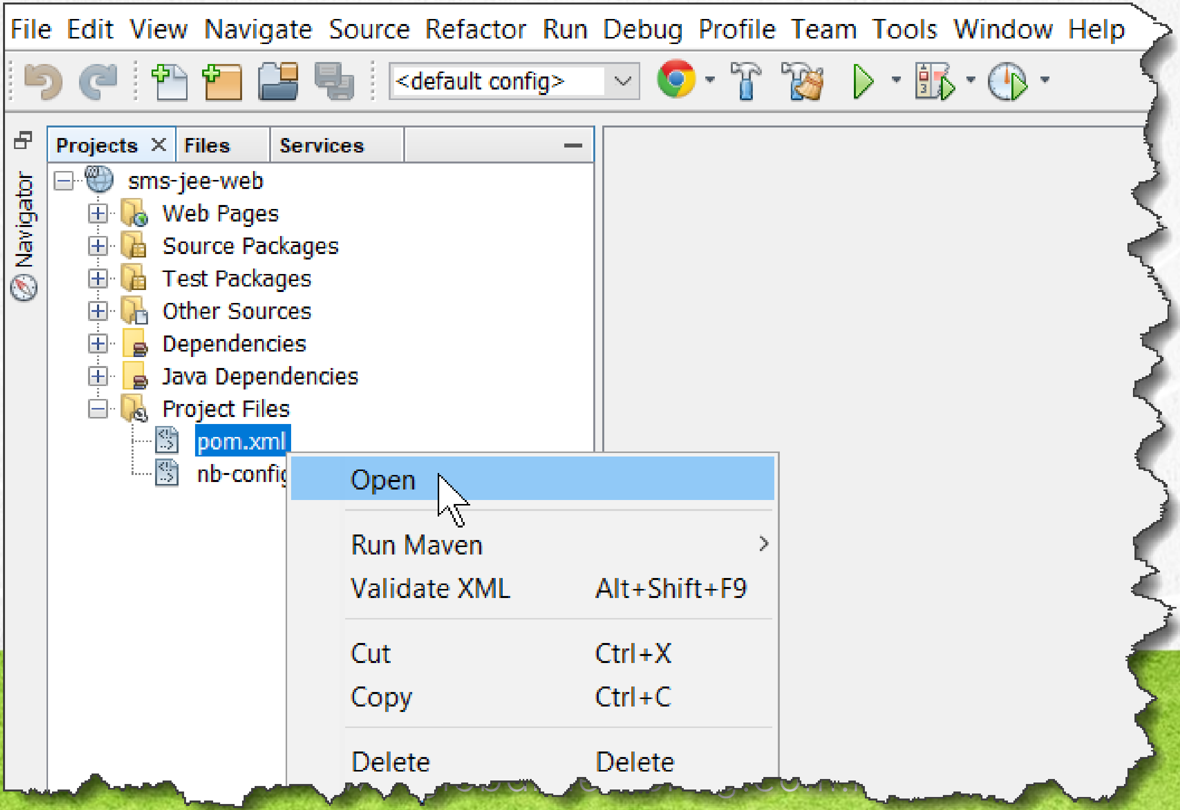
listPeople.jsp:

Click to download

```
<%@ taglib uri = "http://java.sun.com/jsp/jstl/core" prefix = "c" %>
<!DOCTYPE html>
<html>
    <head>
        <title>List of People</title>
    </head>
    <body>
        <h1>List of People</h1>
        <table border="1">
            <tr>
                <th>Person ID</th>
                <th>Name</th>
            </tr>
            <c:forEach var="person" items="${people}">
                <tr>
                    <td>${person.idPerson}</td>
                    <td>${person.name}</td>
                </tr>
            </c:forEach>
        </table>
        <br>
        <a href="index.html">Return to Index</a>
    </body>
</html>
```

4. MODIFY THE POM.XML FILE

Modify the pom.xml file to exclude the glassfish client jar:



4. MODIFY THE FILE

pom.xml:

Click to download

```
<?xml version="1.0" encoding="UTF-8"?>
<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>sms</groupId>
    <artifactId>sms-jee-web</artifactId>
    <version>1</version>
    <packaging>war</packaging>

    <name>sms-jee-web</name>

   <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>

    <dependencies>
        <dependency>
            <groupId>javax</groupId>
            <artifactId>javaee-api</artifactId>
            <version>8.0</version>
            <scope>provided</scope>
        </dependency>
    </dependencies>
```

4. MODIFY THE FILE

pom.xml:

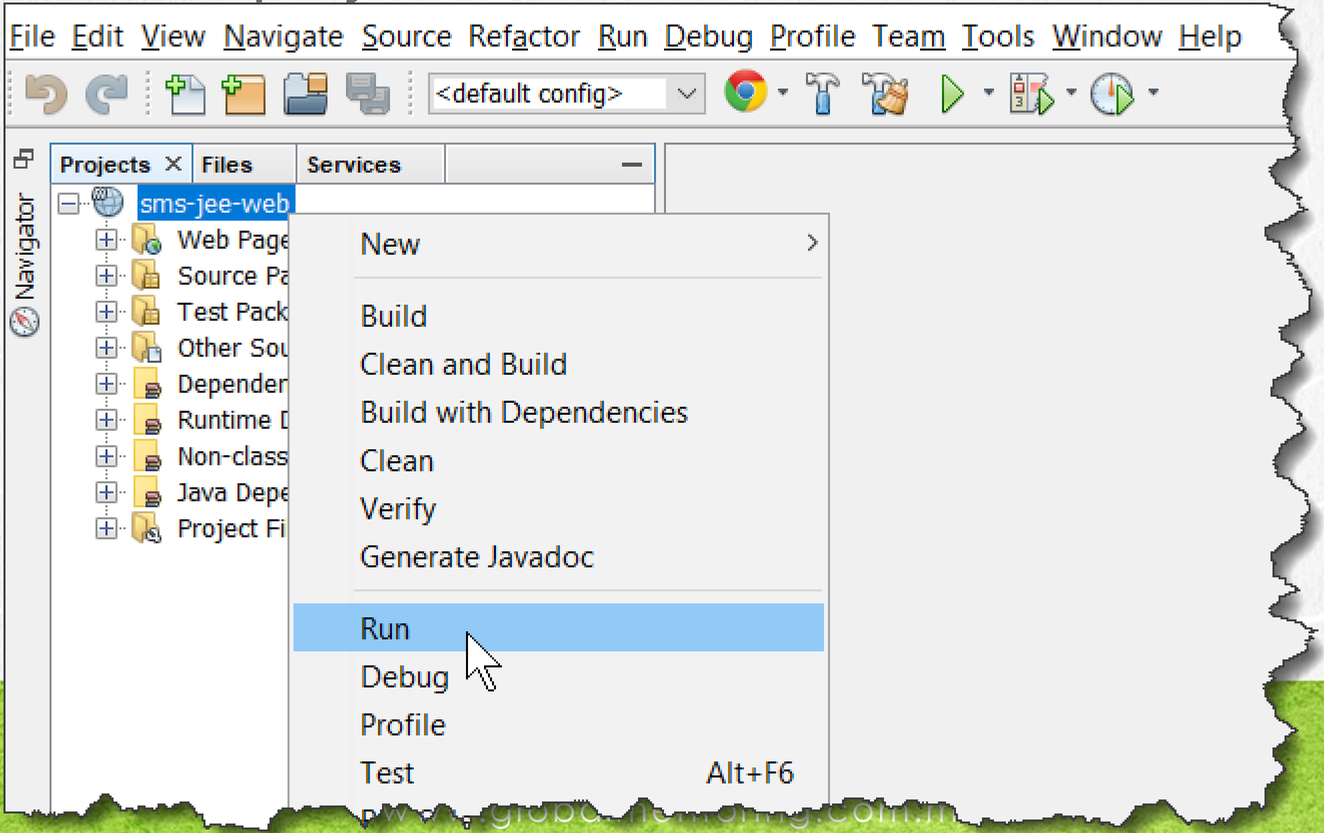
Click to download

```
    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-war-plugin</artifactId>
                <version>2.6</version>
                <configuration>
                    <failOnMissingWebXml>false</failOnMissingWebXml>
                </configuration>
            </plugin>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.8.0</version>
                <configuration>
                    <source>1.8</source>
                    <target>1.8</target>
                </configuration>
            </plugin>
        </plugins>
    </build>

</project>
```

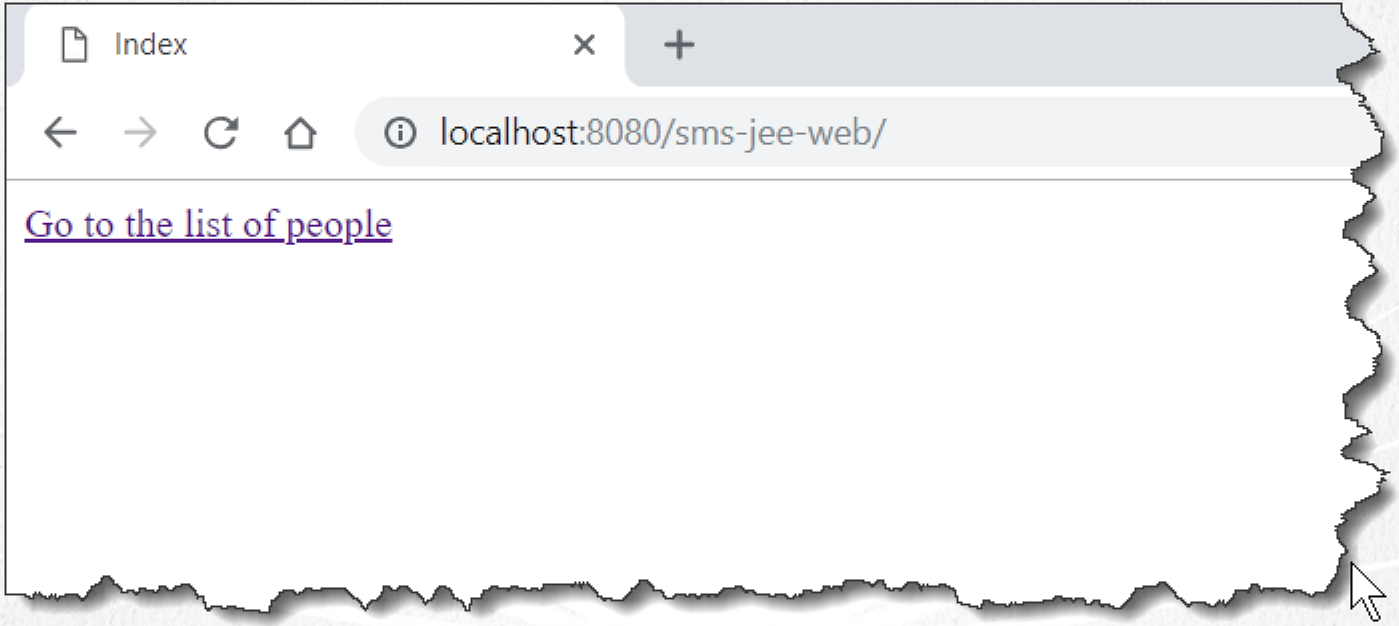
5. EXECUTE THE PROJECT

We execute the project:



5. EXECUTE THE PROJECT

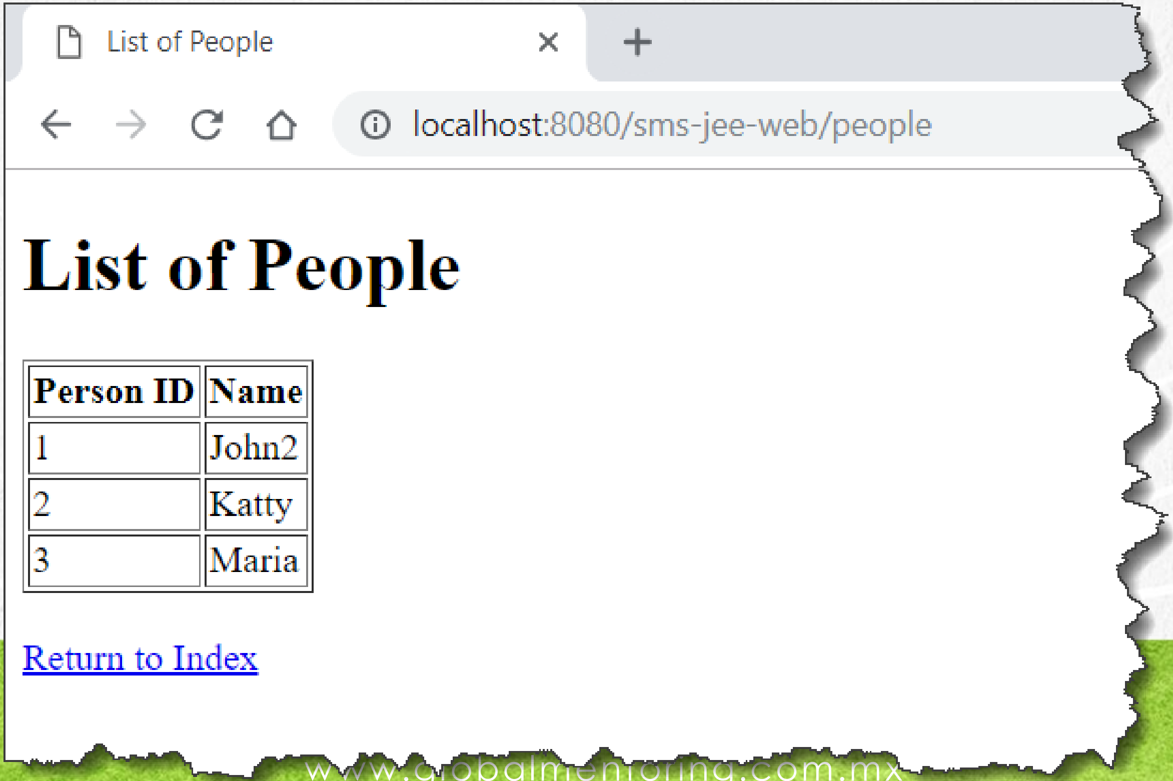
The result is as follows:



Go to the list of people

5. EXECUTE THE PROJECT

The result is as follows:



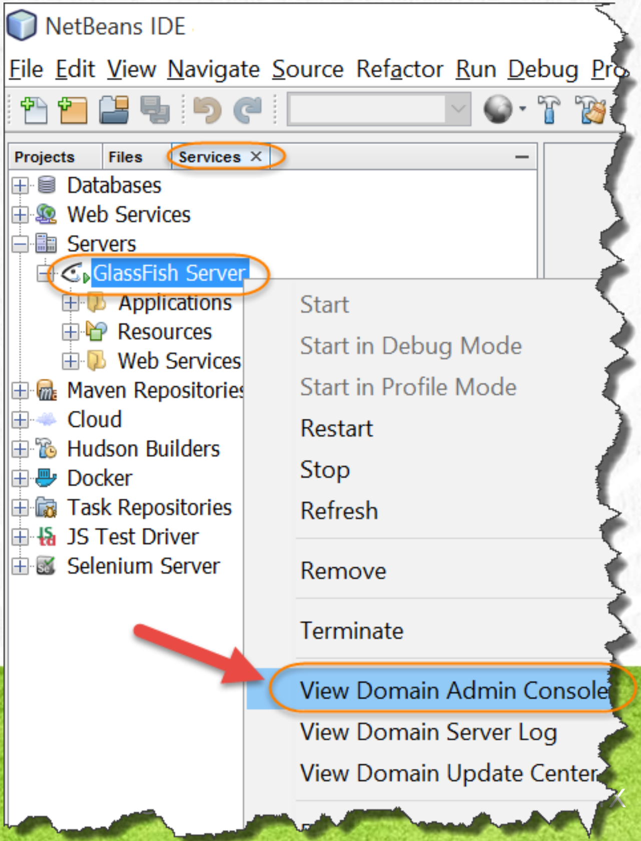
List of People

Person ID	Name
1	John2
2	Katty
3	Maria

Return to Index

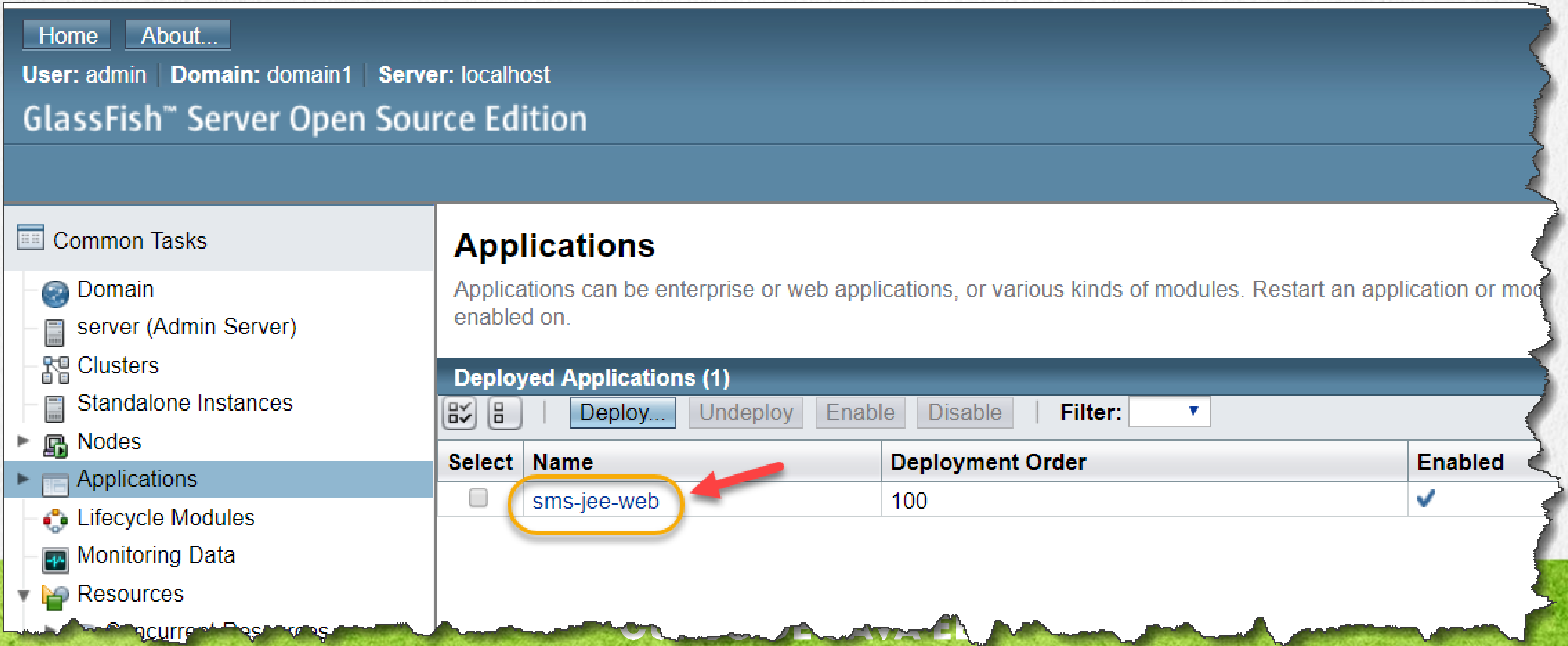
6. REVIEW THE COMPONENTS

We review the components deployed in the web application:



6. REVIEW THE COMPONENTS

We review the components deployed in the web application:



User: admin | **Domain:** domain1 | **Server:** localhost

GlassFish™ Server Open Source Edition

Applications

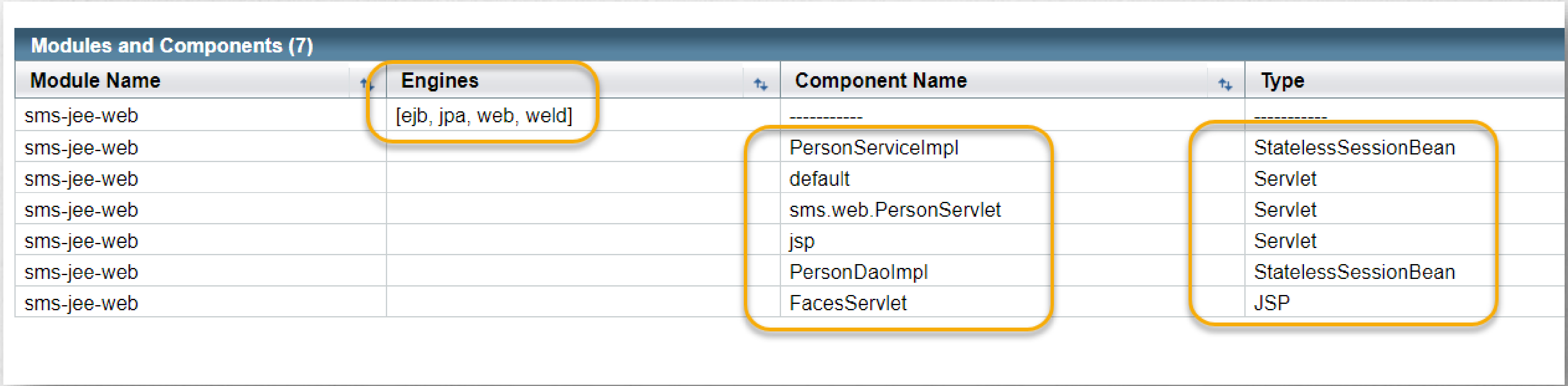
Applications can be enterprise or web applications, or various kinds of modules. Restart an application or mod
enabled on.

Deployed Applications (1)

Select	Name	Deployment Order	Enabled
☐	sms-jee-web	100	✓

6. REVIEW THE COMPONENTS

We review the components deployed in the web application:



Modules and Components (7)

Module Name	Engines	Component Name	Type
sms-jee-web	[ejb, jpa, web, weld]	-----------	-----------
sms-jee-web		PersonServiceImpl	StatelessSessionBean
sms-jee-web		default	Servlet
sms-jee-web		sms.web.PersonServlet	Servlet
sms-jee-web		jsp	Servlet
sms-jee-web		PersonDaoImpl	StatelessSessionBean
sms-jee-web		FacesServlet	JSP

EXERCISE CONCLUSION

With this exercise we could observe how to integrate the Web layer with the Service layer, the latter was already integrated with the data layer. In this way we have integrated the 3 layers in a single application.

The Servlets have facilitated the configuration and integration with the EJB due to the support of annotations that they provide, and with it we can perform tasks in a very simple way, without the need of configuration files xml.

In the next lesson we will integrate the JSF technology with EJB and JPA in a very similar way as we did in this exercise.

ONLINE COURSE

JAVA EE JAKARTA EE

By: Eng. Ubaldo Acosta

Global Mentoring

CURSO DE JAVA EE

www.globalmentoring.com.mx