

JAVA EE COURSE

EXERCISE

**WEB APPLICATION WITH
JSF/AJAX/PRIMEFACES/EJB/JPA**

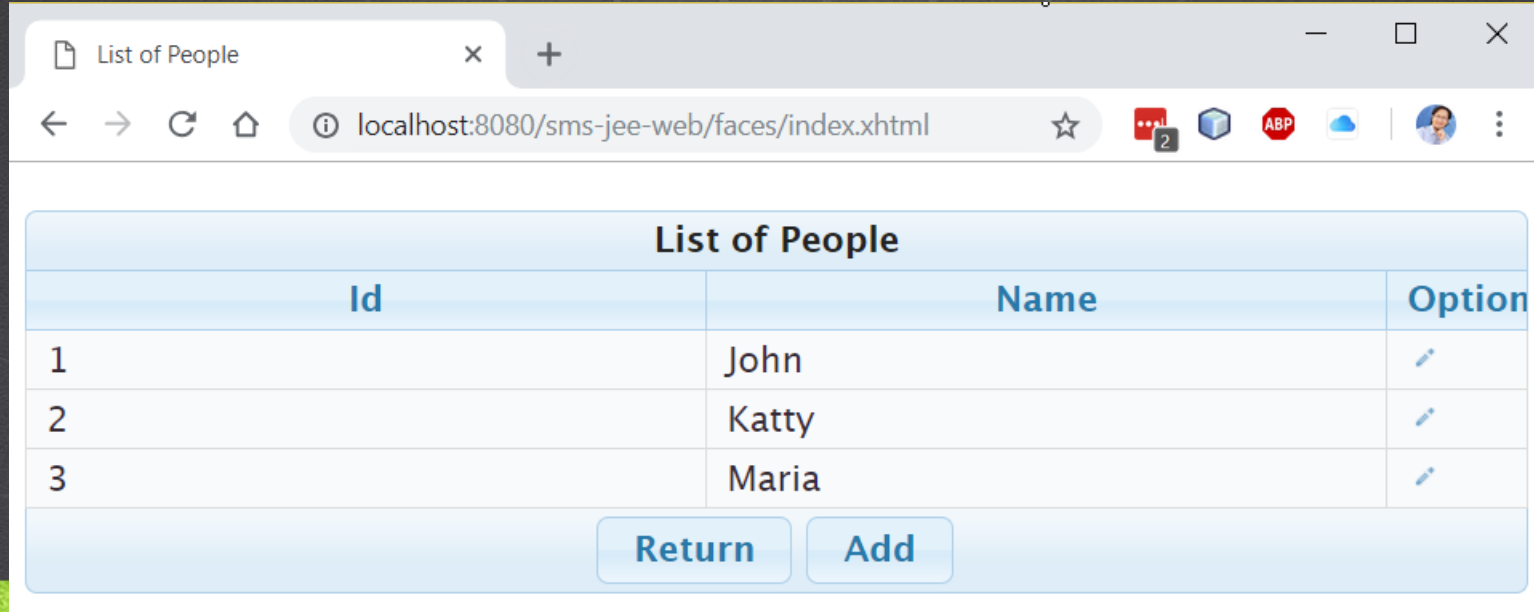





JAVA EE COURSE

www.globalmentoring.com.mx

EXERCISE COURSE

Create an application to integrate the technologies of JSF / Ajax / PrimeFaces, EJB and JPA. At the end we must observe:



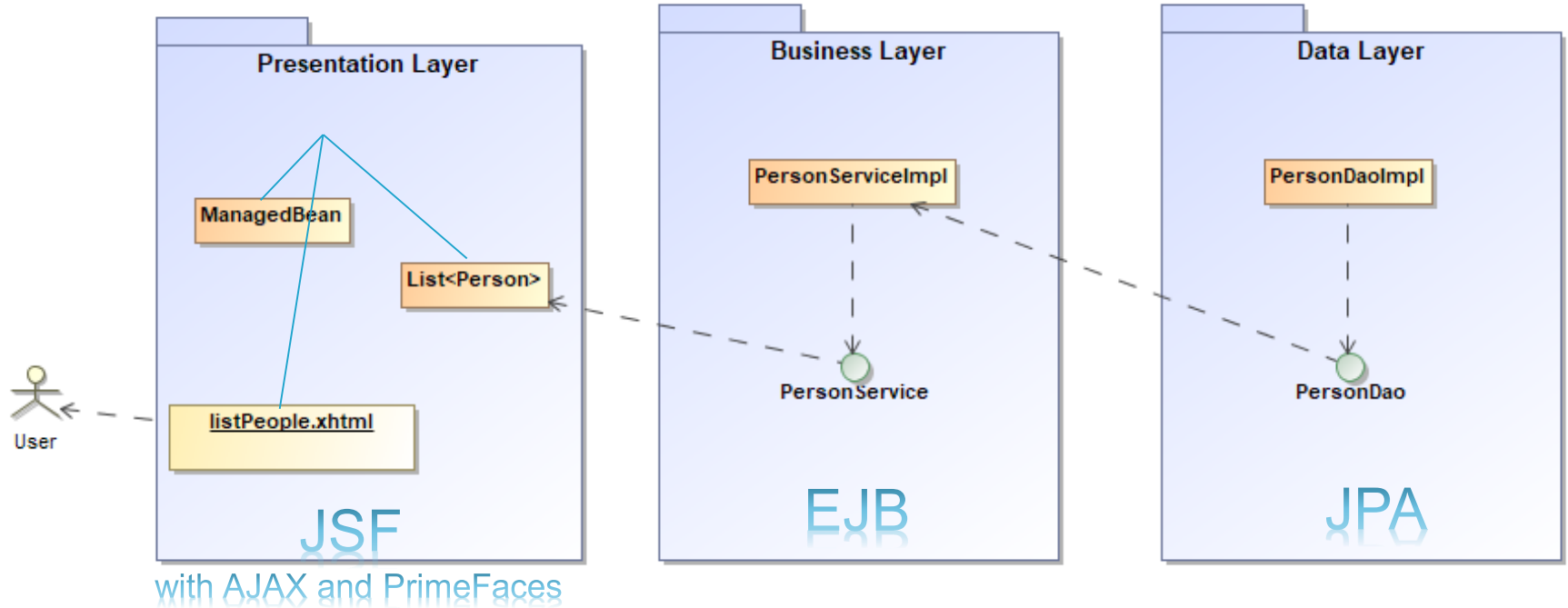
List of People		
Id	Name	Option
1	John	
2	Katty	
3	Maria	

ReturnAdd

JAVA EE COURSE

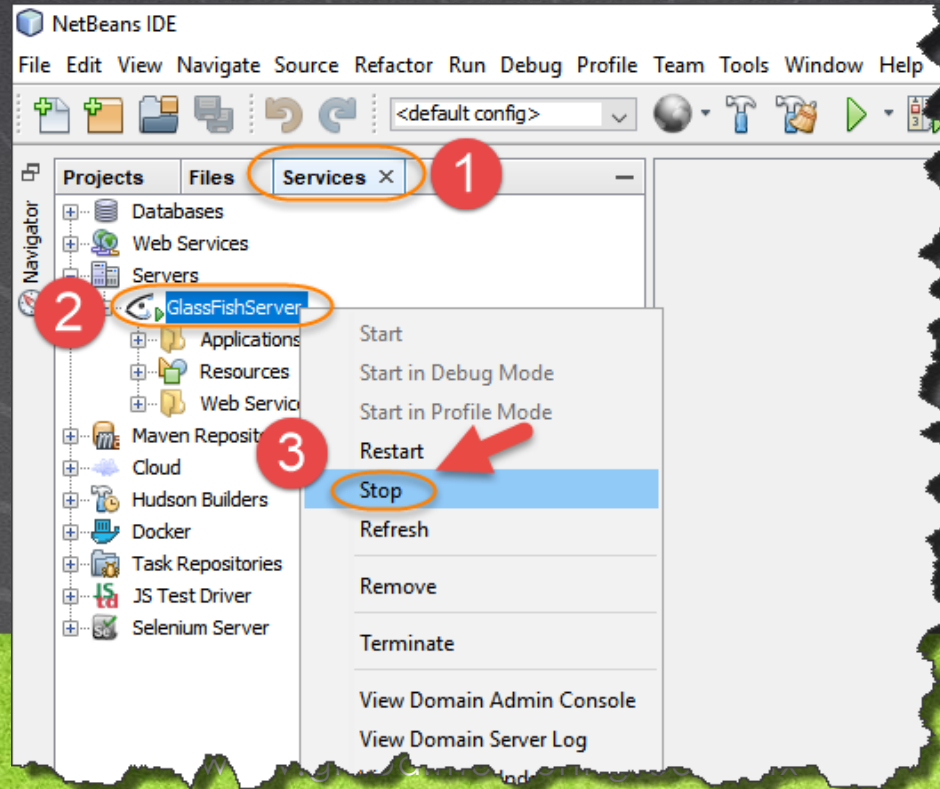
www.globalmentoring.com.mx

ENTERPRISE ARCHITECTURE JAVA EE



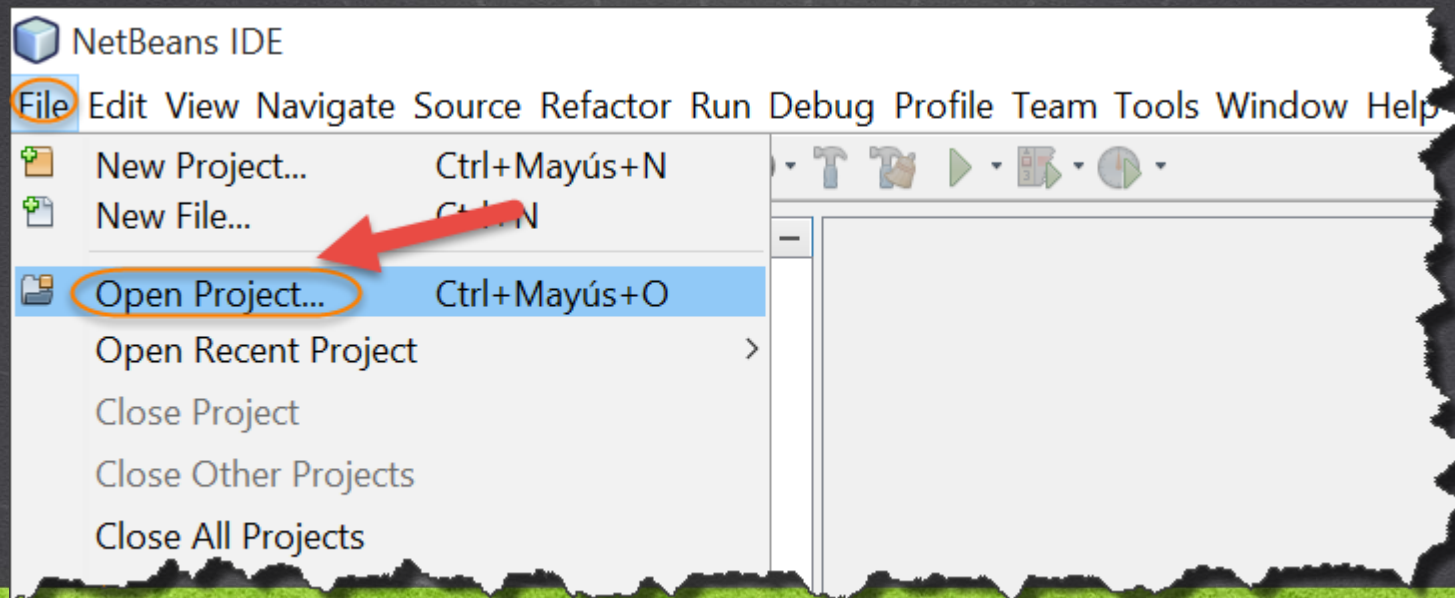
STOP GLASSFISH IF IT IS ACTIVE

Stop the Glassfish server if it was started:



1. OPEN THE PROJECT

We open the previous project sga-jee-web if we do not have it open:

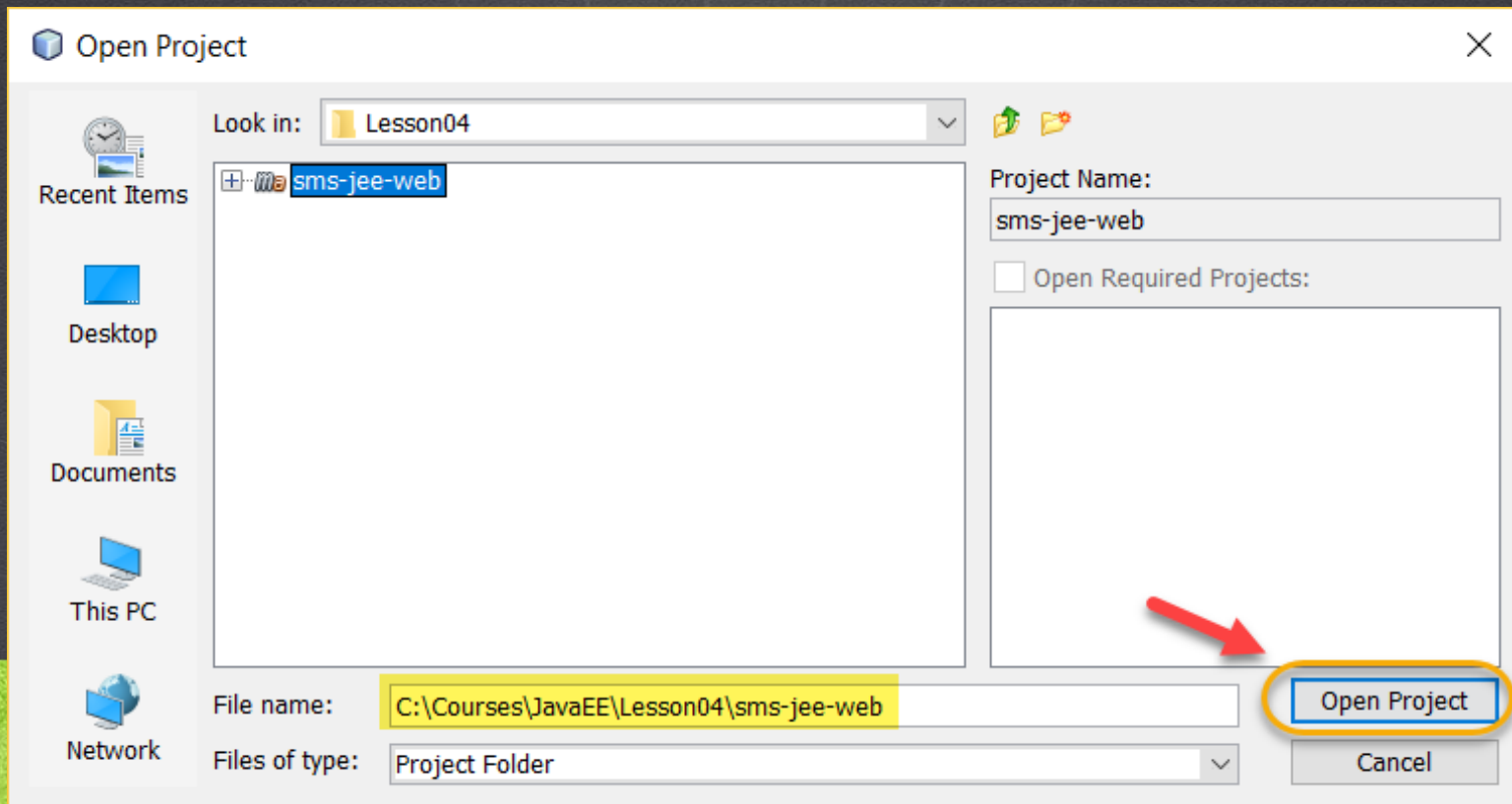


JAVA EE COURSE

www.globalmentoring.com.mx

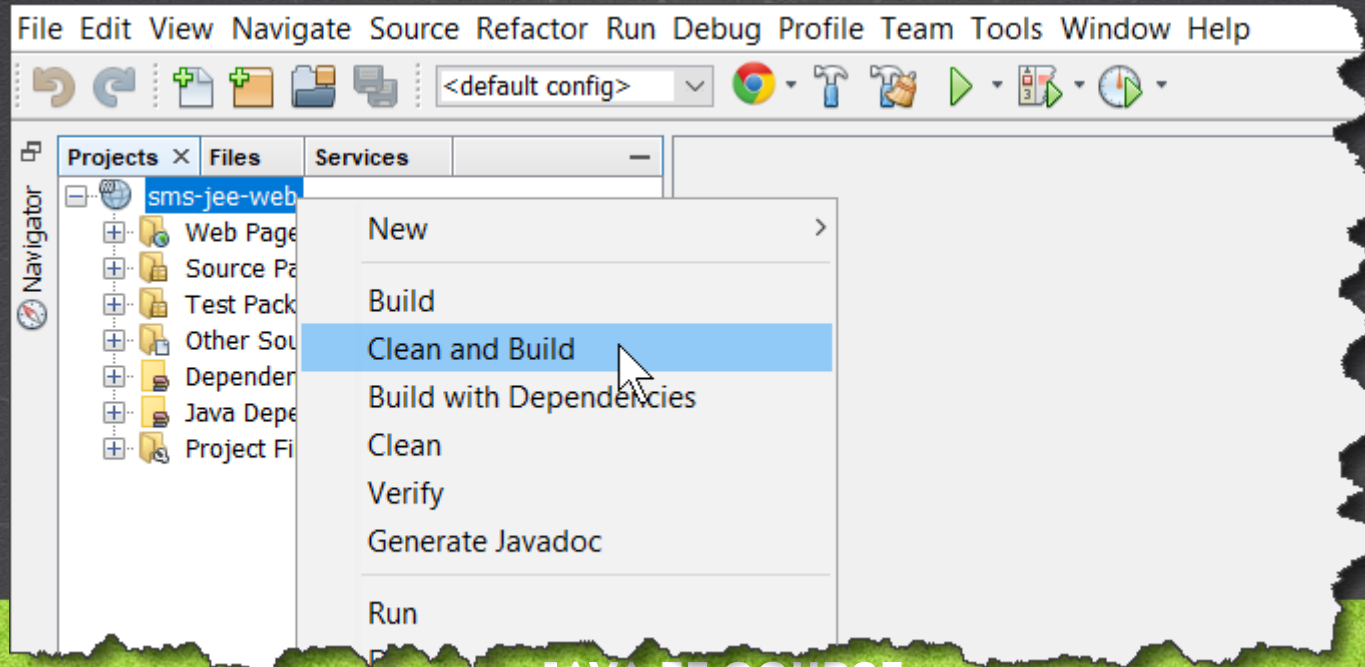
1. OPEN THE PROJECT

We open the previous project sga-jee-web if we do not have it open:



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:

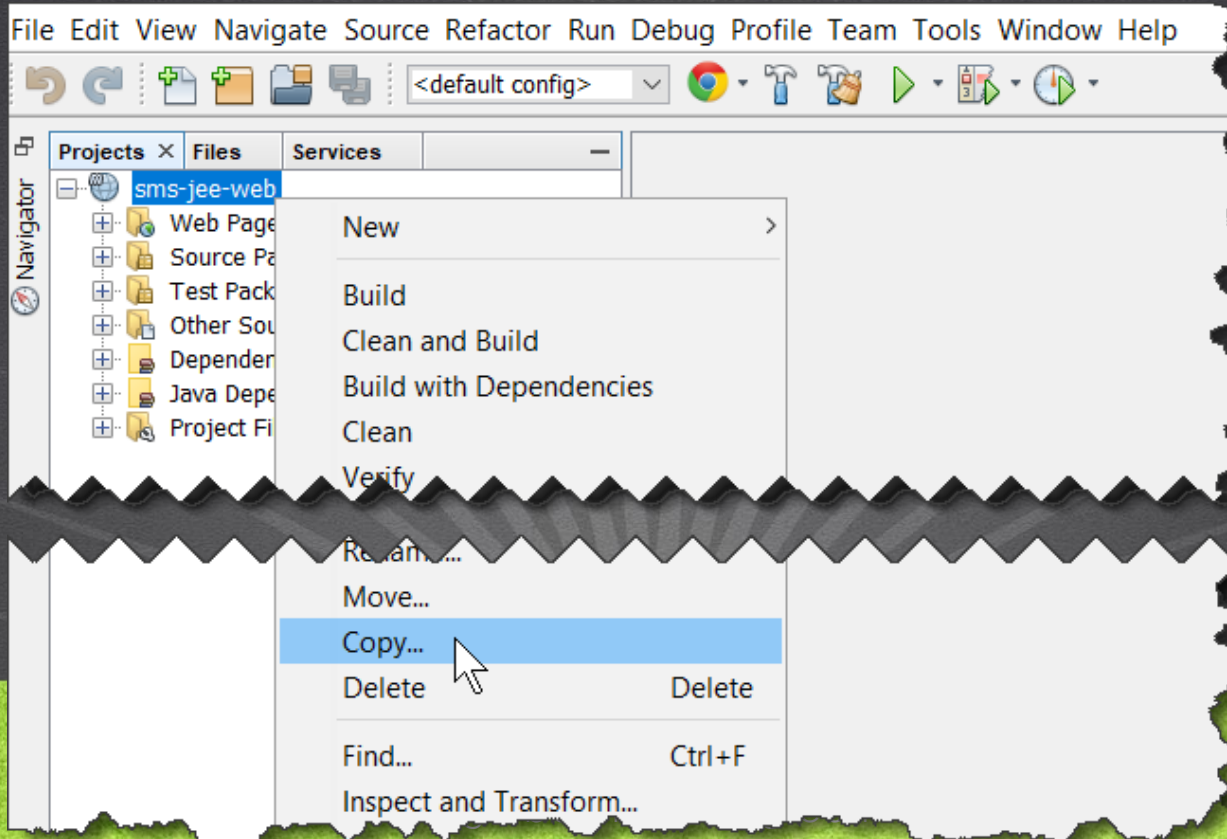


JAVA EE COURSE

www.globalmentoring.com.mx

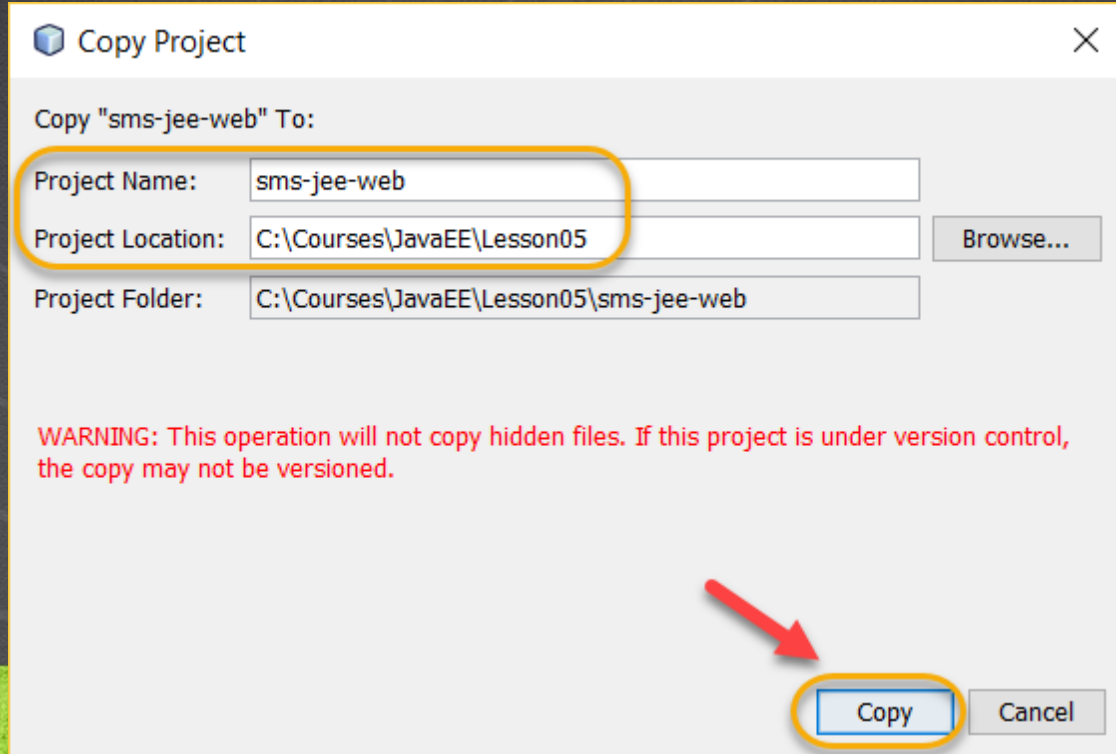
2. COPY THE PROJECT

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



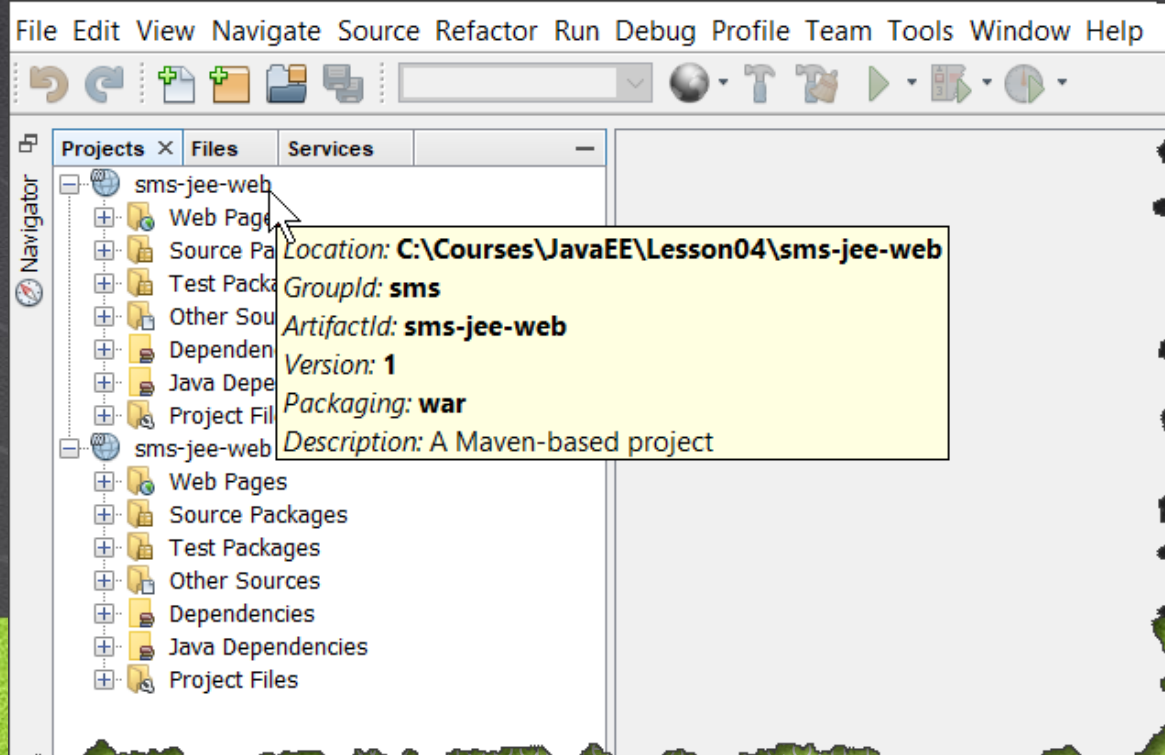
2. COPY THE PROJECT

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



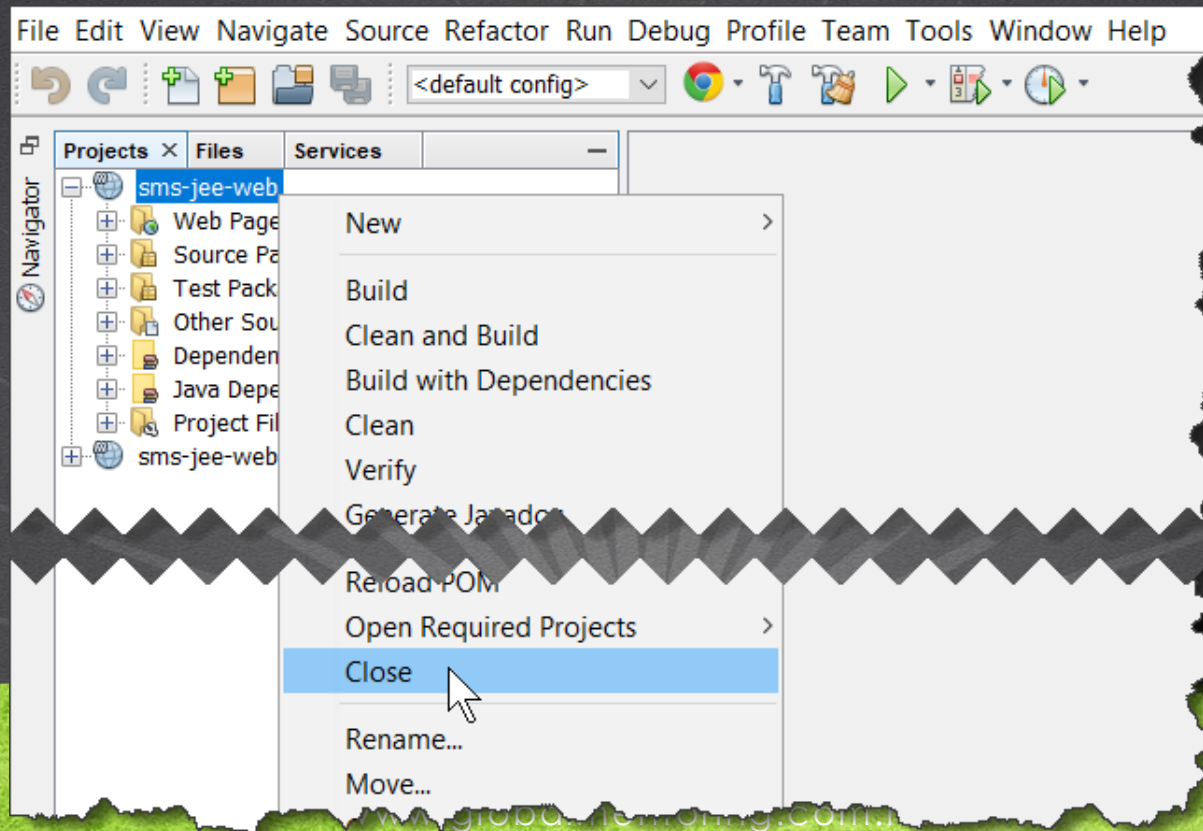
3. CLOSE THE PROJECT

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



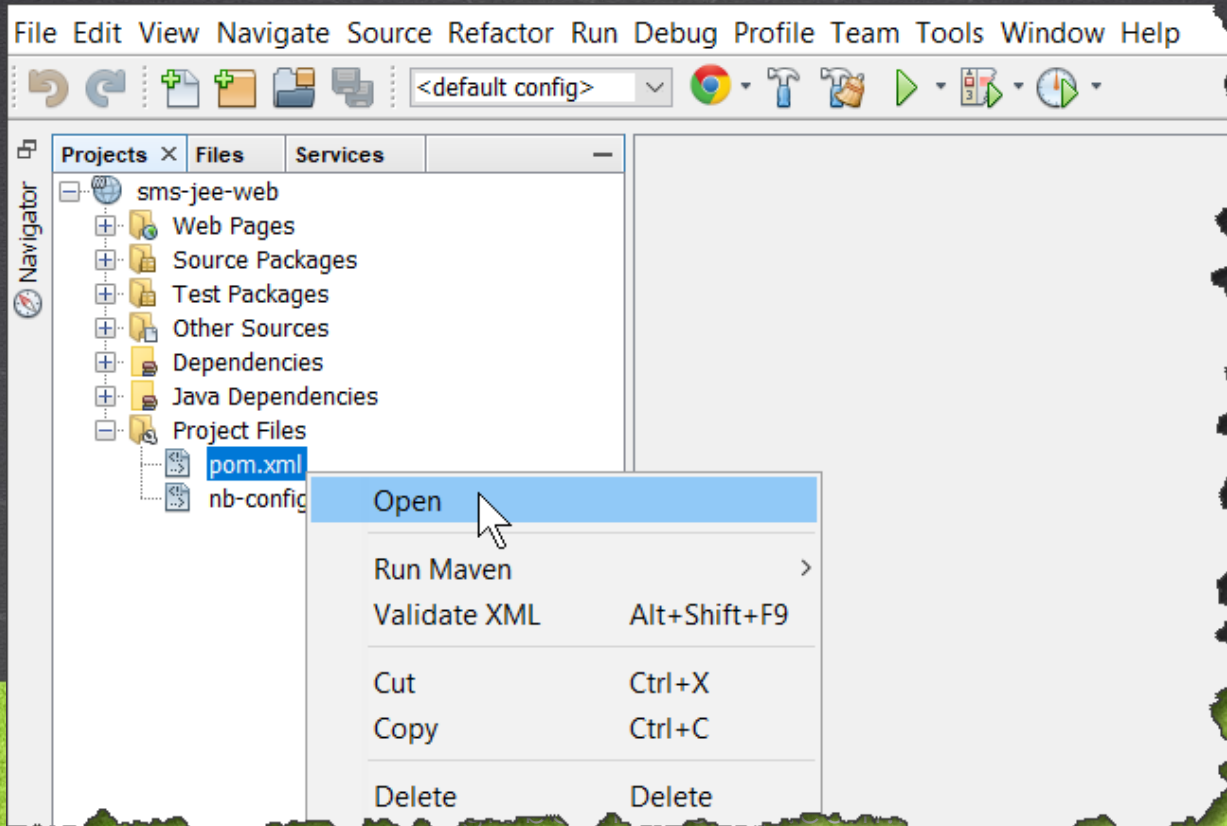
3. CLOSE THE PROJECT

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



4. MODIFY THE POM.XML

Modify the pom.xml file to add the dependency of primefaces:



4. MODIFY THE CODE

[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>
        <dependency>
            <groupId>org.primefaces</groupId>
            <artifactId>primefaces</artifactId>
            <version>6.2</version>
        </dependency>
    </dependencies>
</project>
```

JAVA EE COURSE

www.globalmentoring.com.mx

4. MODIFY THE CODE

[pom.xml:](#)

Click to download

```
<dependency>
  <groupId>org.primefaces.themes</groupId>
  <artifactId>all-themes</artifactId>
  <version>1.0.10</version>
</dependency>
</dependencies>
<repositories>
  <repository>
    <id>prime-repo</id>
    <name>PrimeFaces Maven Repository</name>
    <url>http://repository.primefaces.org</url>
    <layout>default</layout>
  </repository>
</repositories>
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-war-plugin</artifactId>
      <version>2.6</version>
      <configuration>
        <failOnMissingWebXml>false</failOnMissingWebXml>
      </configuration>
    </plugin>
```

JAVA EE COURSE

www.globalmentoring.com.mx

4. MODIFY THE CODE

[pom.xml:](#)

Click to download

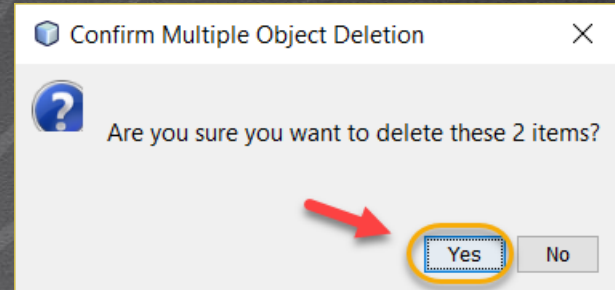
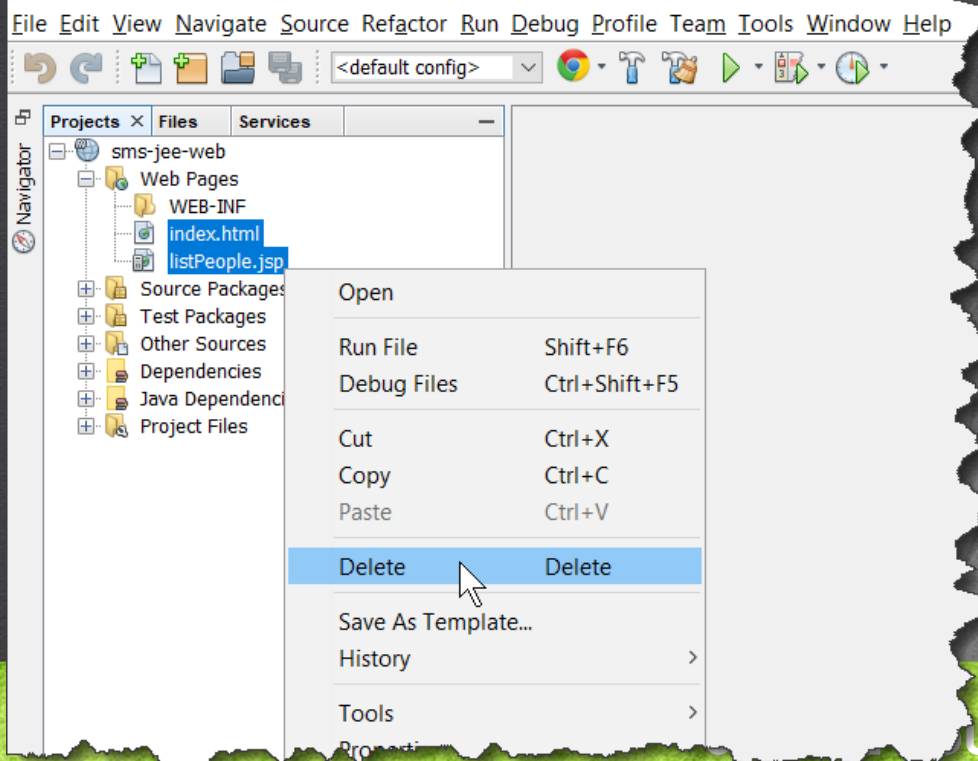
```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-compiler-plugin</artifactId>
  <version>3.8.0</version>
  <configuration>
    <source>1.8</source>
    <target>1.8</target>
  </configuration>
</plugin>
</plugins>
</build>
</project>
```

JAVA EE COURSE

www.globalmentoring.com.mx

5. DELETE SOME FILES

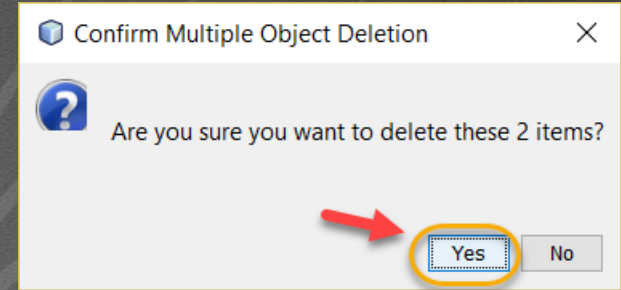
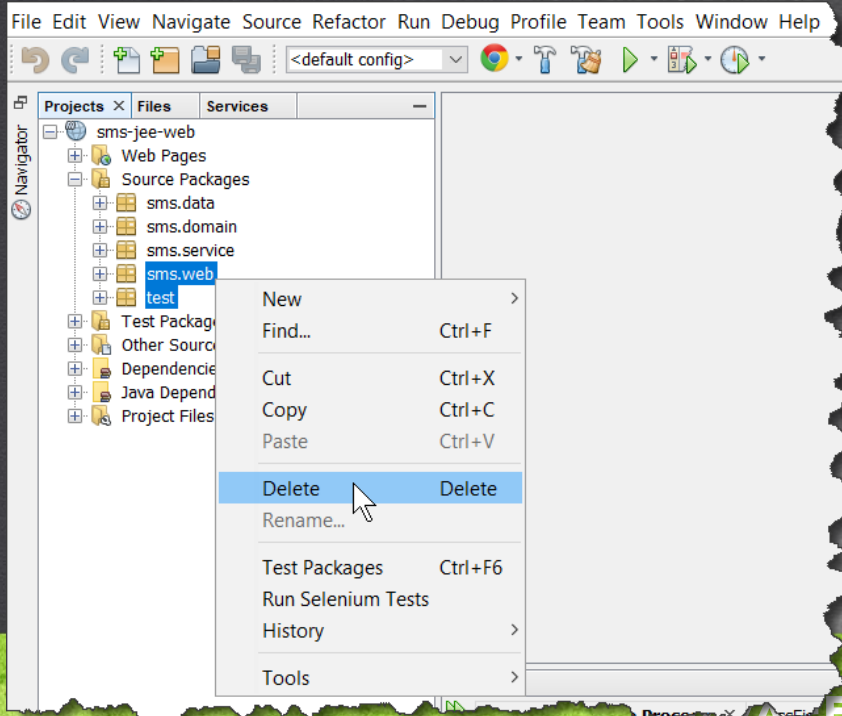
We delete several files that we will no longer use :



COURSE

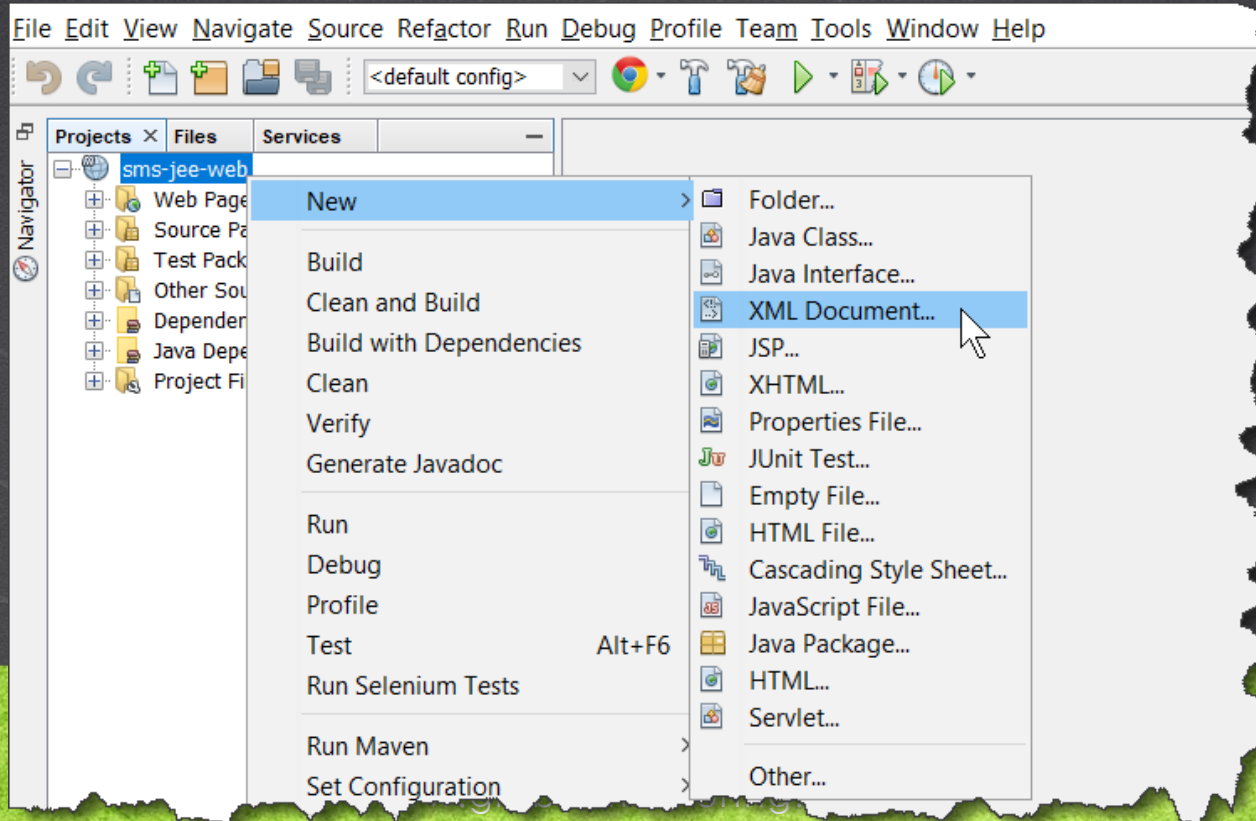
6. DELETE SOME FILES

We delete several files that we will no longer use :



7. CREATE AN XML FILE

We add the web.xml configuration file:



7. CREATE AN XML FILE

We add the web.xml configuration file:

New XML Document

Steps

1. Choose File Type
2. **Name and Location**
3. Select Document Type
4. ...

Name and Location

File Name: web

Project: sms-jee-web

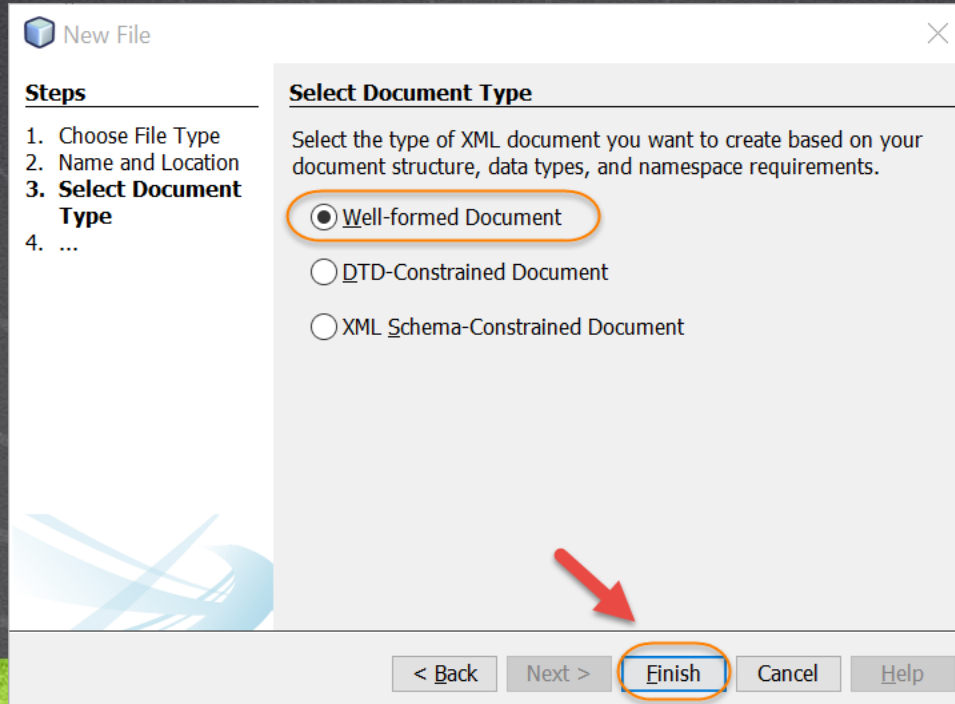
Folder: src/main/webapp/WEB-INF Browse...

Created File: C:\Courses\JavaEE\Lesson05\sms-jee-web\src/main/webapp\WEB-INF\web.xml

< Back **Next >** Finish Cancel Help

7. CREATE AN XML FILE

We add the web.xml configuration file:



JAVA EE COURSE

www.globalmentoring.com.mx

8. MODIFY THE CODE

web.xml:

Click to download

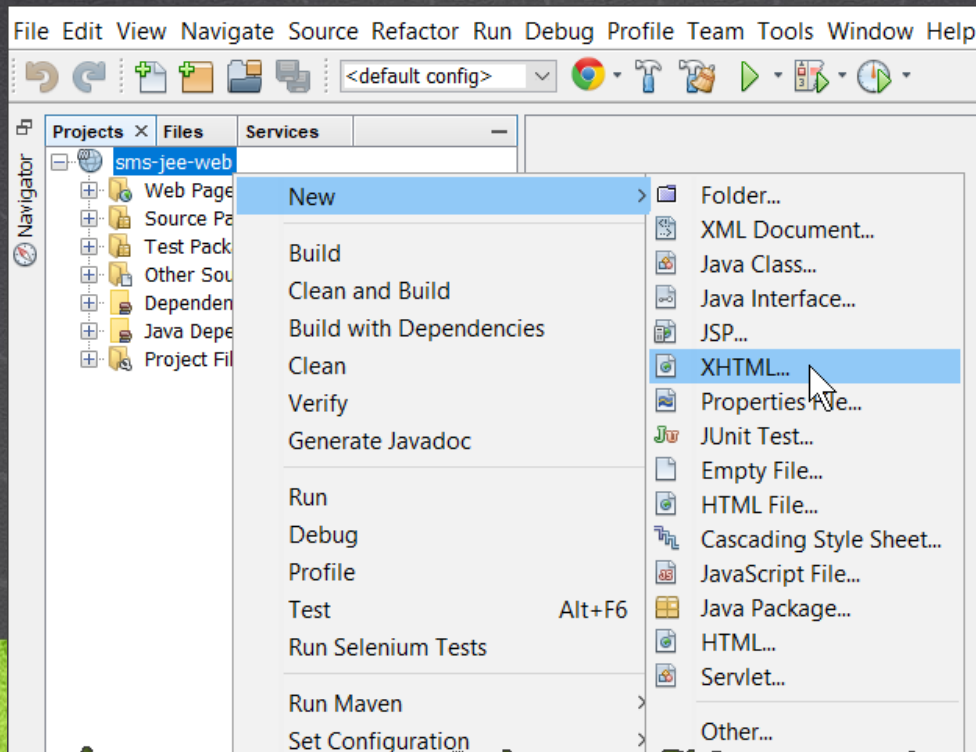
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
  version="4.0">
  <context-param>
    <param-name>primefaces.THEME</param-name>
    <param-value>cupertino</param-value>
  </context-param>
  <welcome-file-list>
    <welcome-file>faces/index.xhtml</welcome-file>
  </welcome-file-list>
</web-app>
```

JAVA EE COURSE

www.globalmentoring.com.mx

9. CREATE THE INDEX.XHTML FILE

We create the index.xhtml file:



9. CREATE THE INDEX.XHTML FILE

We create the index.xhtml file:

New XHTML

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

XHTML File Name:

Project:

Location:

Folder:

Created File:

JAVA EE COURSE

www.globalmentoring.com.mx

10. MODIFY THE FILE

[index.xhtml:](#)

Click to download

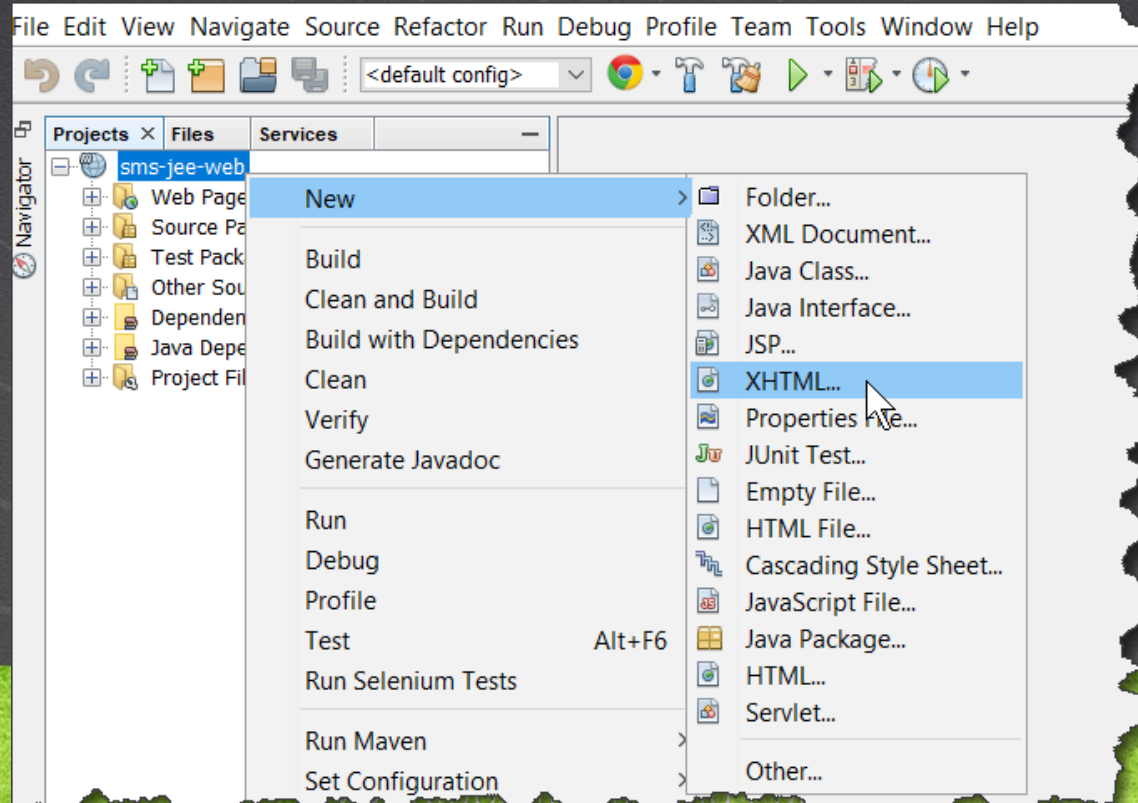
```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:p="http://primefaces.org/ui">
  <h:head>
    <title>SMS System</title>
  </h:head>
  <h:body>
    <h2>SMS System</h2>
    <h:form>
      <p:messages />
      <h:panelGrid columns="1">
        <h:commandButton value="List People" action="listPeople" />
      </h:panelGrid>
    </h:form>
  </h:body>
</html>
```

JAVA EE COURSE

www.globalmentoring.com.mx

11. CREATE AN XHTML FILE

We create the file listPeople.xhtml:



11. CREATE AN XHTML FILE

We create the file listPeople.xhtml:

New XHTML

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

XHTML File Name:

Project:

Location:

Folder: Browse...

Created File:

< Back Next > **Finish** Cancel Help

12. MODIFY THE FILE

[listPeople.xhtml](#):

Click to download

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:p="http://primefaces.org/ui">

    <h:head>
        <title>List of People</title>
    </h:head>
    <h:body>

        <p:ajaxStatus style="width:16px;height:16px;" id="ajaxStatusPanel">
            <f:facet name="start">
                <h:graphicImage library="images" name="ajaxloading.gif" />
            </f:facet>
            <f:facet name="complete">
                <h:outputText value="" />
            </f:facet>
        </p:ajaxStatus>

        <h:form id="PersonListForm">

            <p:growl id="messages" showDetail="true" />
            <p:messages />

        </h:form>

    </h:body>
</html>
```

12. MODIFY THE FILE

[listPeople.xhtml](#):

Click to download

```
<p:dataTable
  id="peopleDataTable"
  var="person"
  value="#{personBean.people}"
  editable="true"
  rowIndex="#{person.idPerson}"
  selection="#{personBean.selectedPerson}"
  selectionMode="single">

  <f:facet name="header">
    List of People
  </f:facet>

  <p:column headerText="Id">
    <h:outputText value="#{person.idPerson}" />
  </p:column>

  <p:column headerText="Name">
    <p:cellEditor>
      <f:facet name="output">
        <h:outputText value="#{person.name}" />
      </f:facet>
      <f:facet name="input">
        <p:inputText value="#{person.name}" />
      </f:facet>
    </p:cellEditor>
  </p:column>
```

12. MODIFY THE FILE

[listPeople.xhtml](#):

Click to download

```
<f:facet name="footer">
    <p:commandButton value="Return" action="index" />
    <p:spacer height="5px;" />
    <p:commandButton value="Add" actionListener="#{personBean.restartSelectedPerson}" update=":dialogForm"
oncomplete="PF('personDialog').show()" />
</f:facet>

<p:column headerText="Options" style="width:50px">
    <p:rowEditor />
</p:column>

<p:ajax event="rowEdit" listener="#{personBean.editListener}">
</p:ajax>
</p:dataTable>

<p:contextMenu for="peopleDataTable">
    <p:menuItem value="Delete" update="PersonListForm:peopleDataTable" icon="ui-icon-close"
actionListener="#{personBean.deletePerson}" />
</p:contextMenu>
</h:form>
```

JAVA EE COURSE

www.globalmentoring.com.mx

12. MODIFY THE FILE

[listPeople.xhtml](#):

Click to download

```
<p:dialog id="personDlg" widgetVar="personDialog" header="Add Person" modal="true" showEffect="fade"
hideEffect="fade" resizable="false">
  <h:form id="dialogForm">
    <p:outputPanel id="personDetail" style="text-align:center;">
      <p:panelGrid id="display" columns="2" columnClasses="label,value" >
        <h:outputText value="Name" />
        <h:inputText value="#{personBean.selectedPerson.name}" />

      <f:facet name="footer">
        <p:commandButton value="Add" actionListener="#{personBean.addPerson}"
update="display,:PersonListForm:peopleDataTable" onComplete="PF('personDialog').hide();" />
      </f:facet>

    </p:panelGrid>
  </p:outputPanel>
</h:form>
</p:dialog>

</h:body>

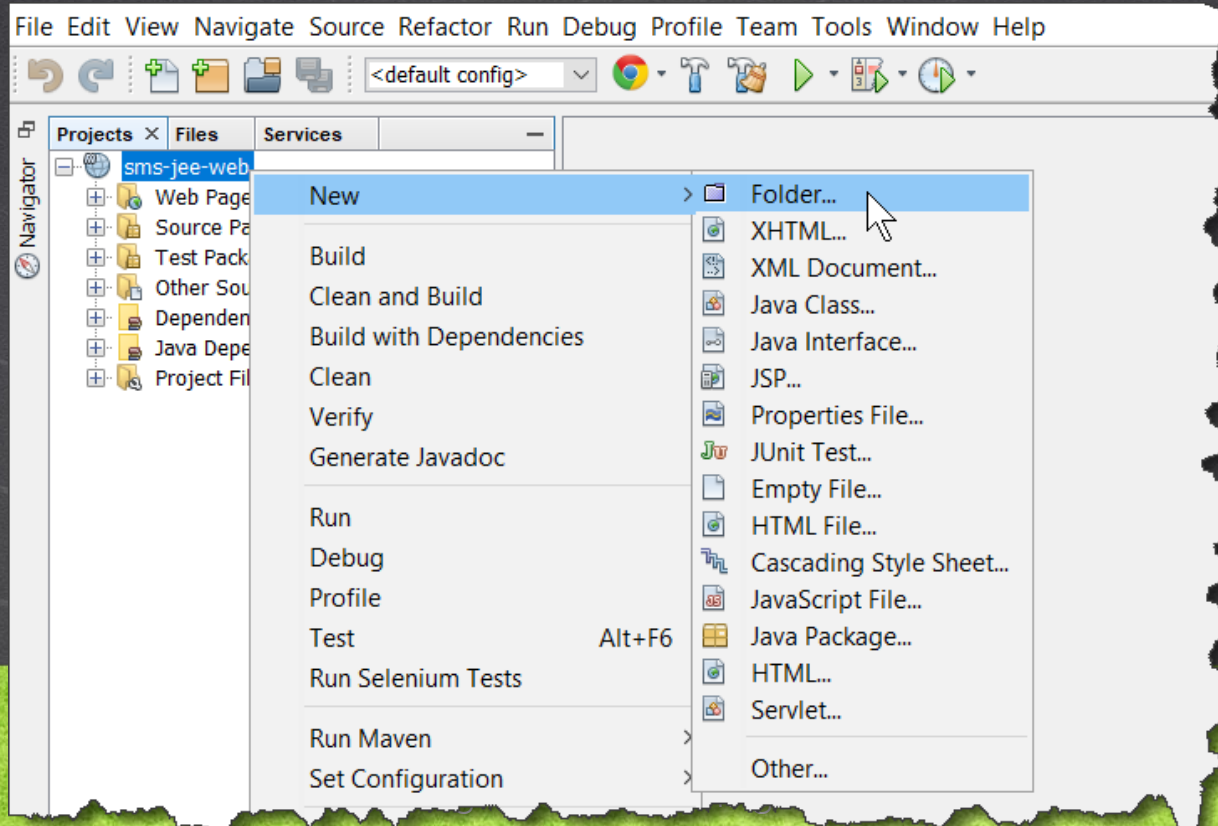
</html>
```

JAVA EE COURSE

www.globalmentoring.com.mx

13. DOWNLOAD GIF IMAGE

We create a folder in the web part as follows: resources/images:



13. DOWNLOAD GIF IMAGE

We create a folder in the web part as follows: resources/images

New Folder

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Folder Name: resources\images

Project: sms-jee-web

Parent Folder: src\main\webapp Browse...

Created Folder: C:\Courses\JavaEE\Lesson05\sms-jee-web\src\main\webapp\resources\images

< Back Next > **Finish** Cancel Help

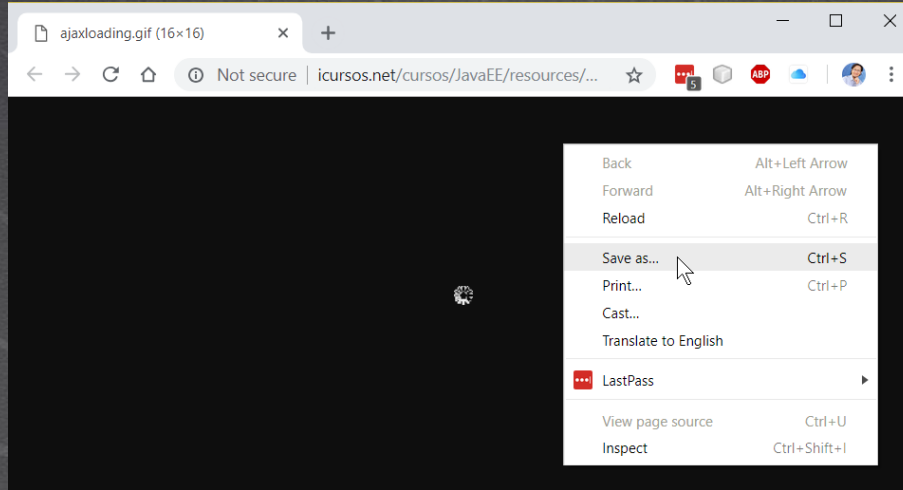
JAVA EE COURSE

www.globalmentoring.com.mx

13. DOWNLOAD GIF IMAGE

Download the ajaxloading.gif image:

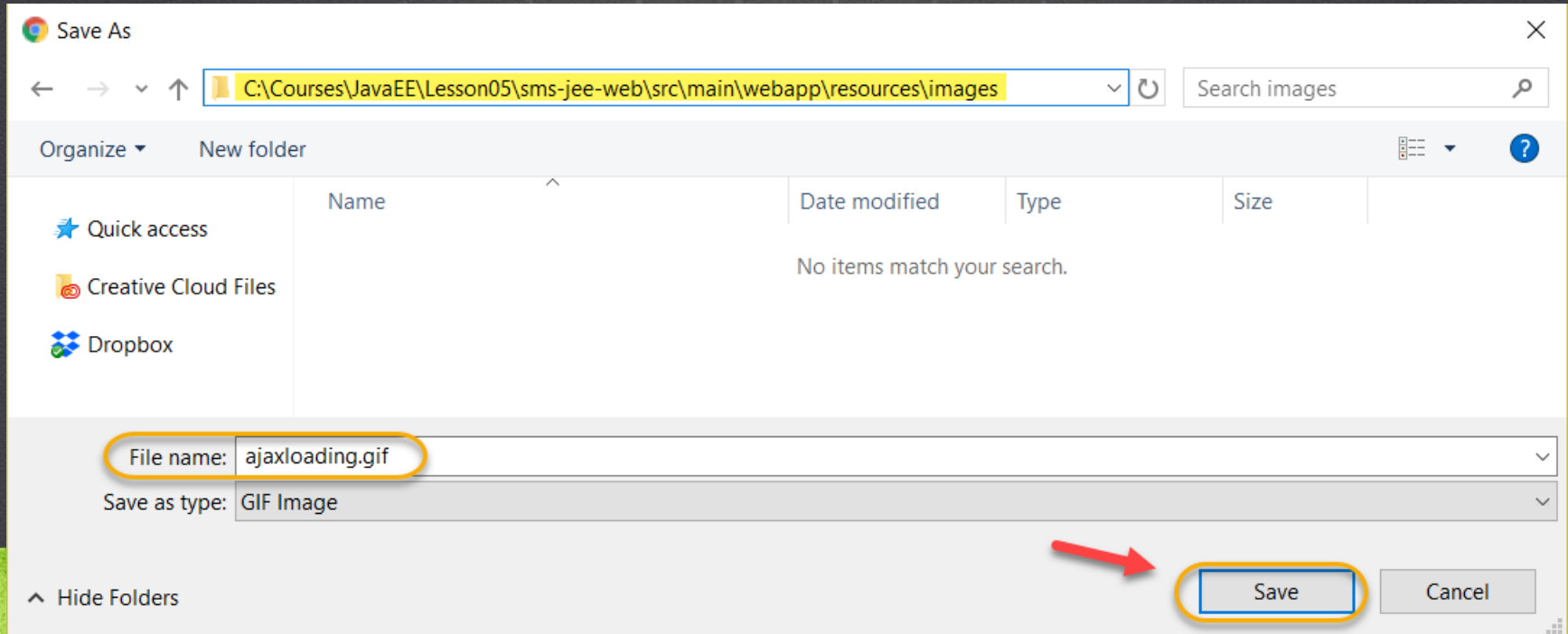
<http://icursos.net/cursos/JavaEE/resources/ajaxloading.gif>



We save the image in the created path of images:

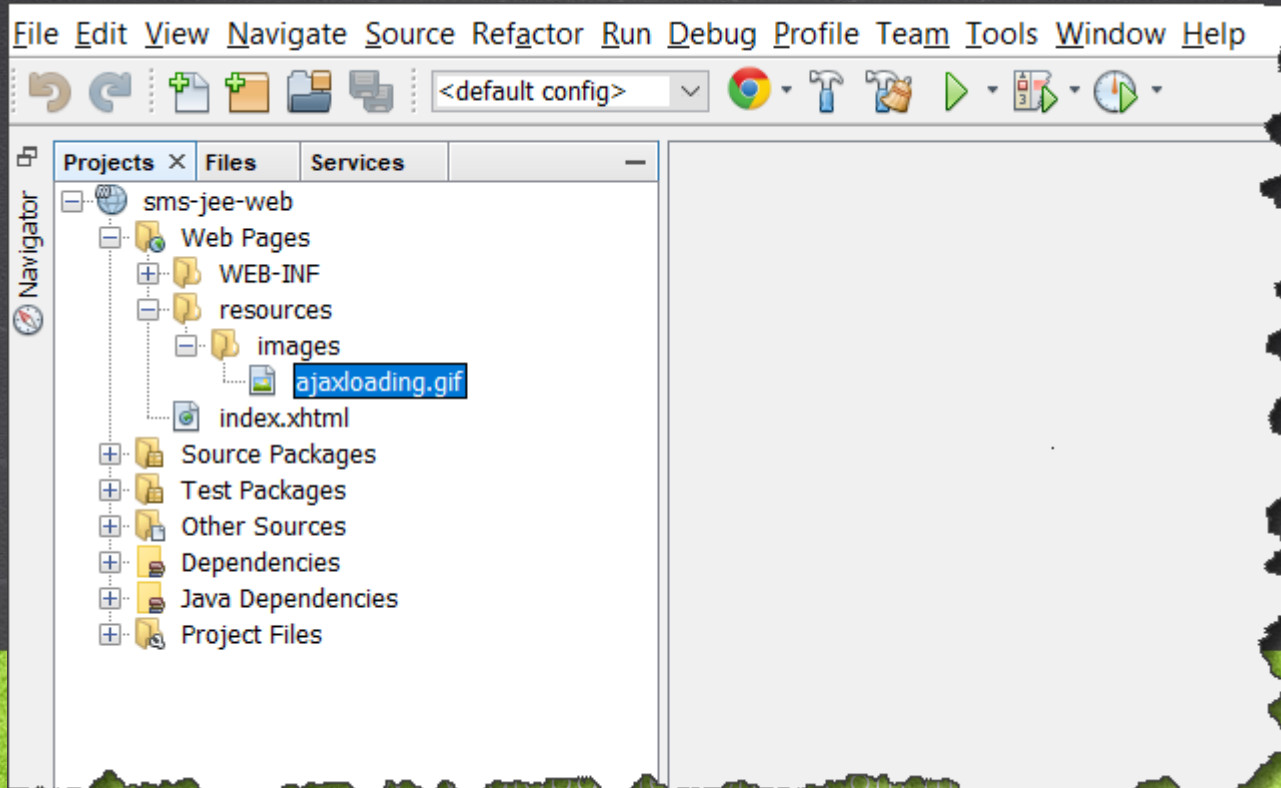
13. DOWNLOAD GIF IMAGE

We download the image in the indicated route, or according to where we created it:



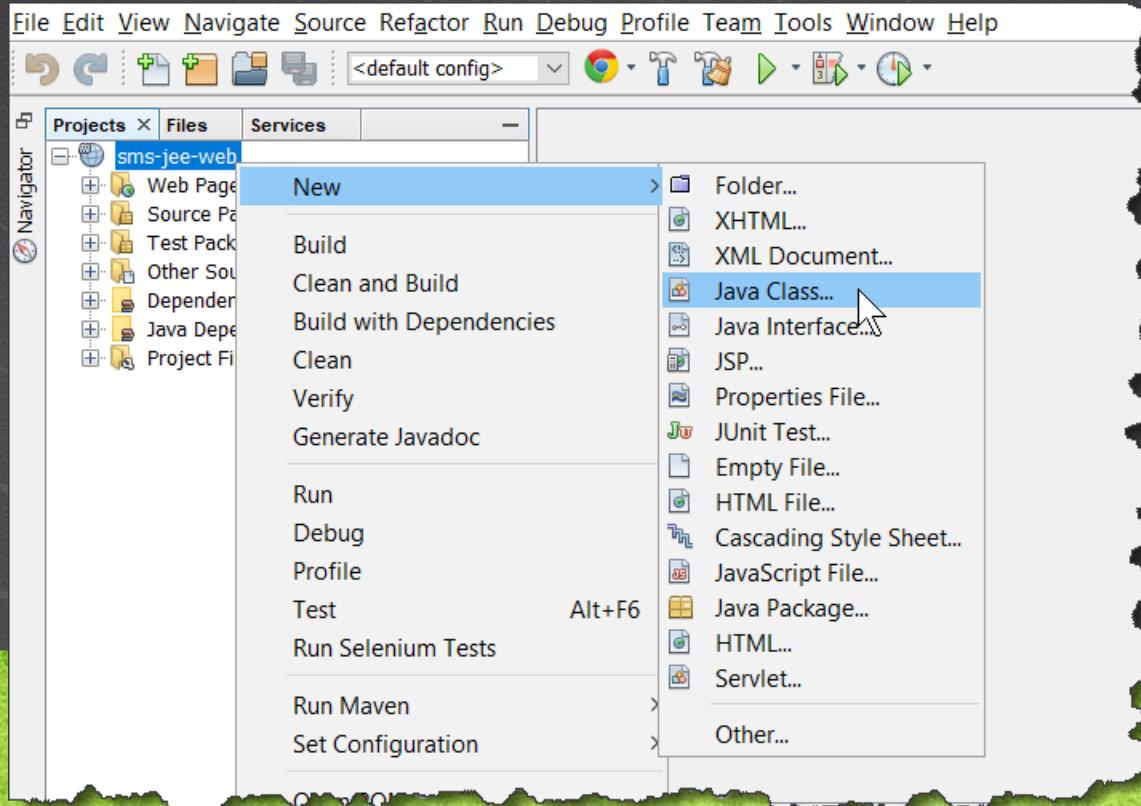
13. DOWNLOAD GIF IMAGE

This is how the downloaded image should look:



14. CREATE A JAVA CLASS

We create the PersonaBean.java class, which is a ManagedBean:



14. CREATE A JAVA CLASS

We create the PersonaBean.java class, which is a ManagedBean:

New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

< Back Next > **Finish** Cancel Help

15. MODIFY THE CODE

PersonBean.java:

Click to download

```
package sms.web;

import sms.domain.Person;
import java.util.List;
import javax.annotation.PostConstruct;
import javax.enterprise.context.RequestScoped;
import javax.inject.Inject;
import javax.inject.Named;
import org.primefaces.event.RowEditEvent;
import sms.service.PersonService;

@Named
@RequestScoped
public class PersonBean {

    @Inject
    private PersonService personService;

    private Person selectedPerson;

    List<Person> people;

    public PersonBean() {
        System.out.println("Starting ManagedBean PersonBean");
    }
}
```

JAVA EE COURSE

www.globalmentoring.com.mx

15. MODIFY THE CODE

PersonBean.java:

Click to download

```
@PostConstruct
public void init() {
    //We start the variables
    people = personService.listPeople();
    selectedPerson = new Person();
}

public void editListener(RowEditEvent event) {
    Person person = (Person) event.getObject();
    personService.modifyPerson(person);
}

public List<Person> getPeople() {
    return people;
}

public void setPeople(List<Person> people) {
    this.people = people;
}

public Person getSelectedPerson() {
    return selectedPerson;
}

public void setSelectedPerson(Person selectedPerson) {
    this.selectedPerson = selectedPerson;
}
```

15. MODIFY THE CODE

PersonBean.java:

Click to download

```
public void restartSelectedPerson() {
    this.selectedPerson = new Person();
}

public void addPerson() {
    personService.addPerson(this.selectedPerson);
    //We also add it to the view (refresh the Model)
    this.people.add(this.selectedPerson);
    this.selectedPerson = null;
}

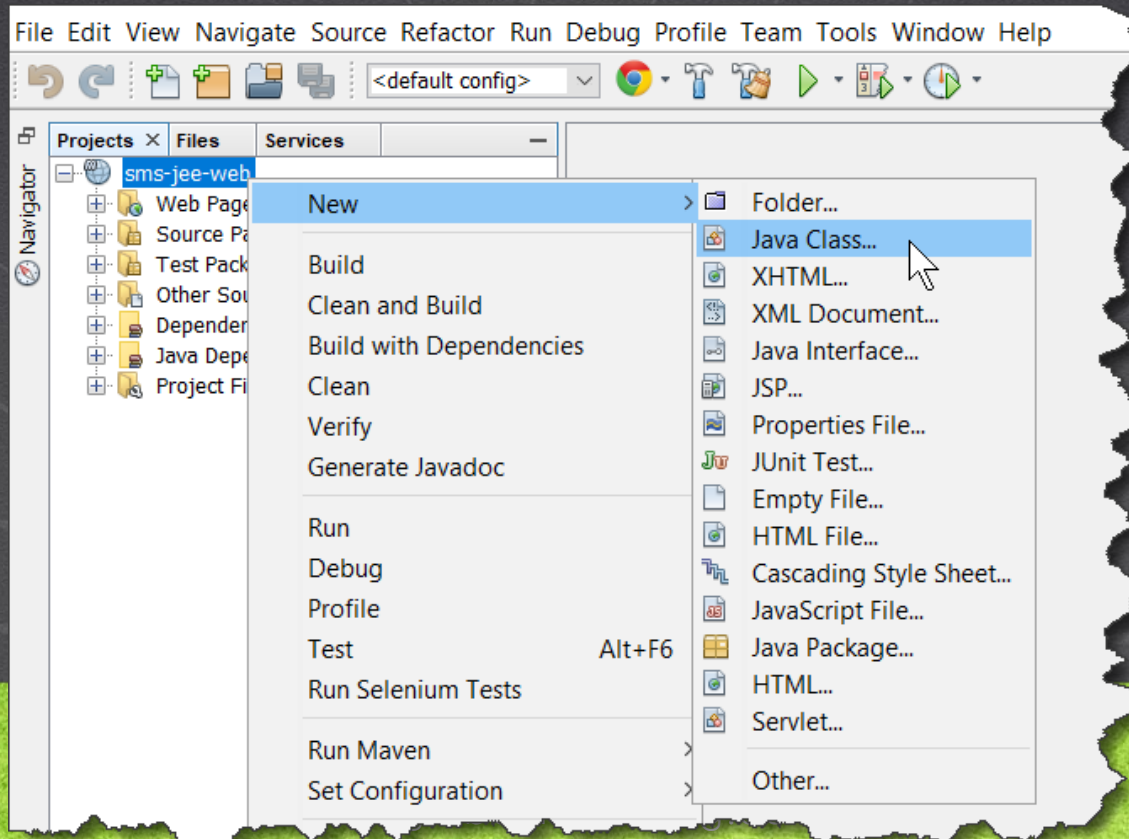
public void deletePerson() {
    personService.deletePerson(this.selectedPerson);
    //We also eliminate it from the view (refresh the Model)
    people.remove(this.selectedPerson);
    this.selectedPerson = null;
}
}
```

JAVA EE COURSE

www.globalmentoring.com.mx

16. CREATE A JAVA CLASS

We create the ConfigurationBean class to activate the latest version of JSF:



16. CREATE A JAVA CLASS

We create the ConfigurationBean class to activate the latest version of JSF:

New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name: ConfigurationBean

Project: sms-jee-web

Location: Source Packages

Package: sms.web

Created File: C:\Courses\JavaEE\Lesson05\sms-jee-web\src\main\java\sms\web\ConfigurationBean.java

< Back Next > **Finish** Cancel Help

JAVA EE COURSE

www.globalmentoring.com.mx

17. MODIFY THE FILE

[ConfigurationBean.java:](#)

Click to download

```
package mx.com.gm.sga.beans;

import javax.enterprise.context.ApplicationScoped;
import javax.faces.annotation.FacesConfig;
import static javax.faces.annotation.FacesConfig.Version.JSF_2_3;

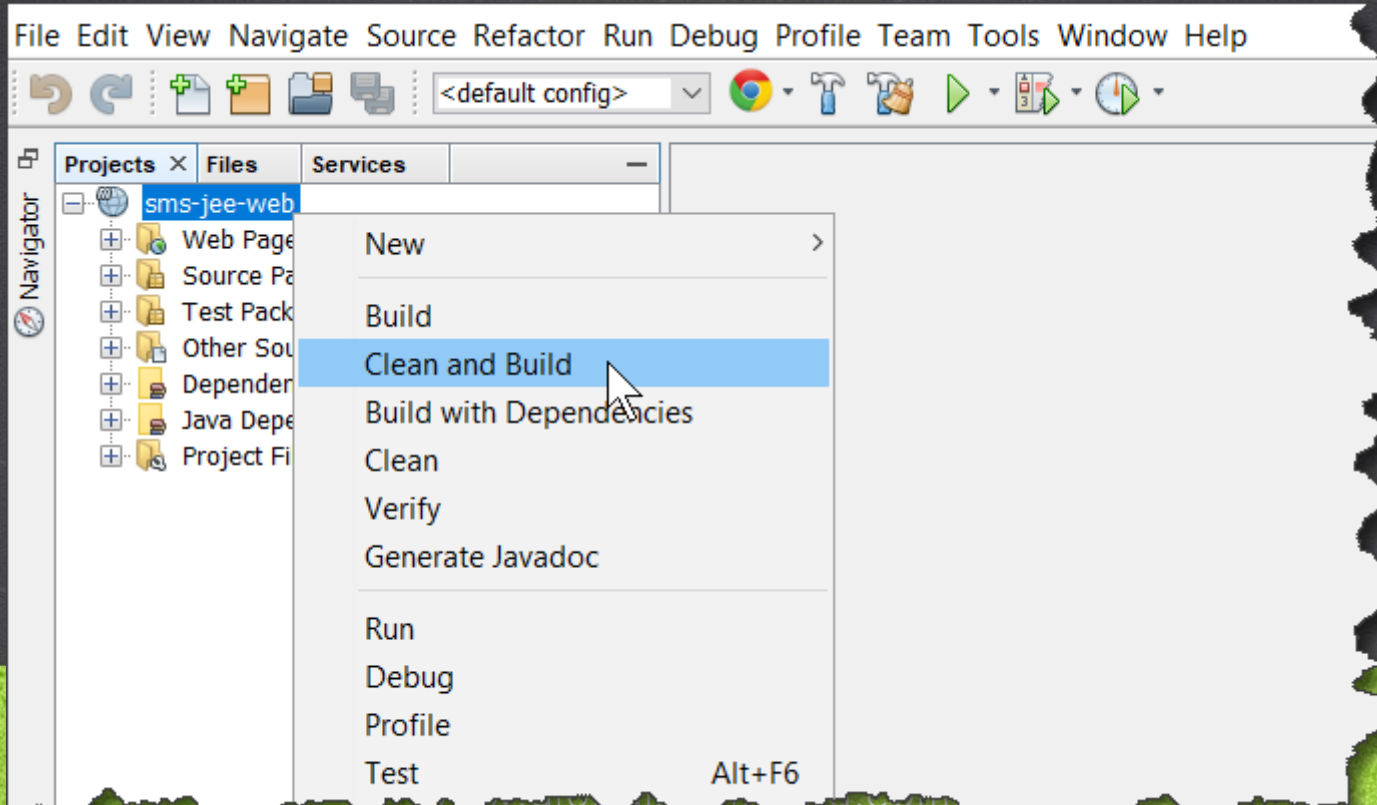
@FacesConfig(
    // Activates CDI build-in beans
    version = JSF_2_3
)
@ApplicationScoped
public class ConfigurationBean {
}
```

JAVA EE COURSE

www.globalmentoring.com.mx

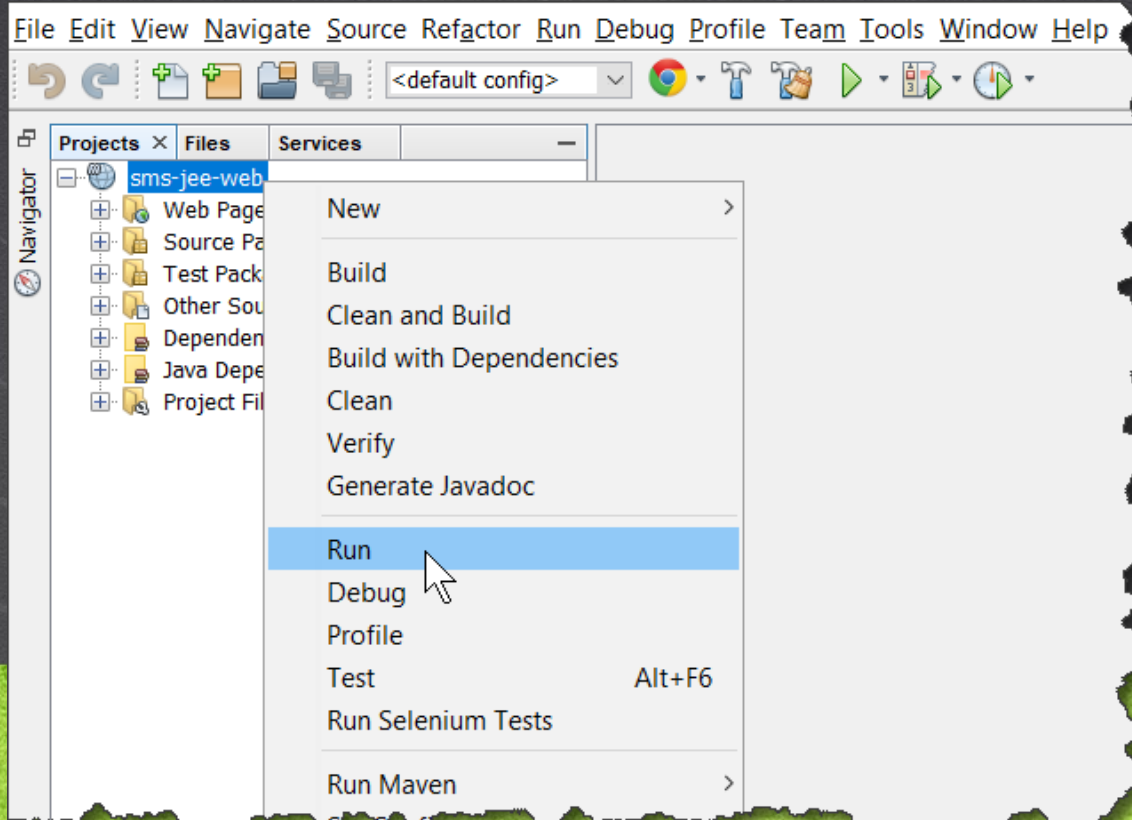
18. EXECUTE CLEAN & BUILD

We do a Clean & Build to have the latest version of each file:



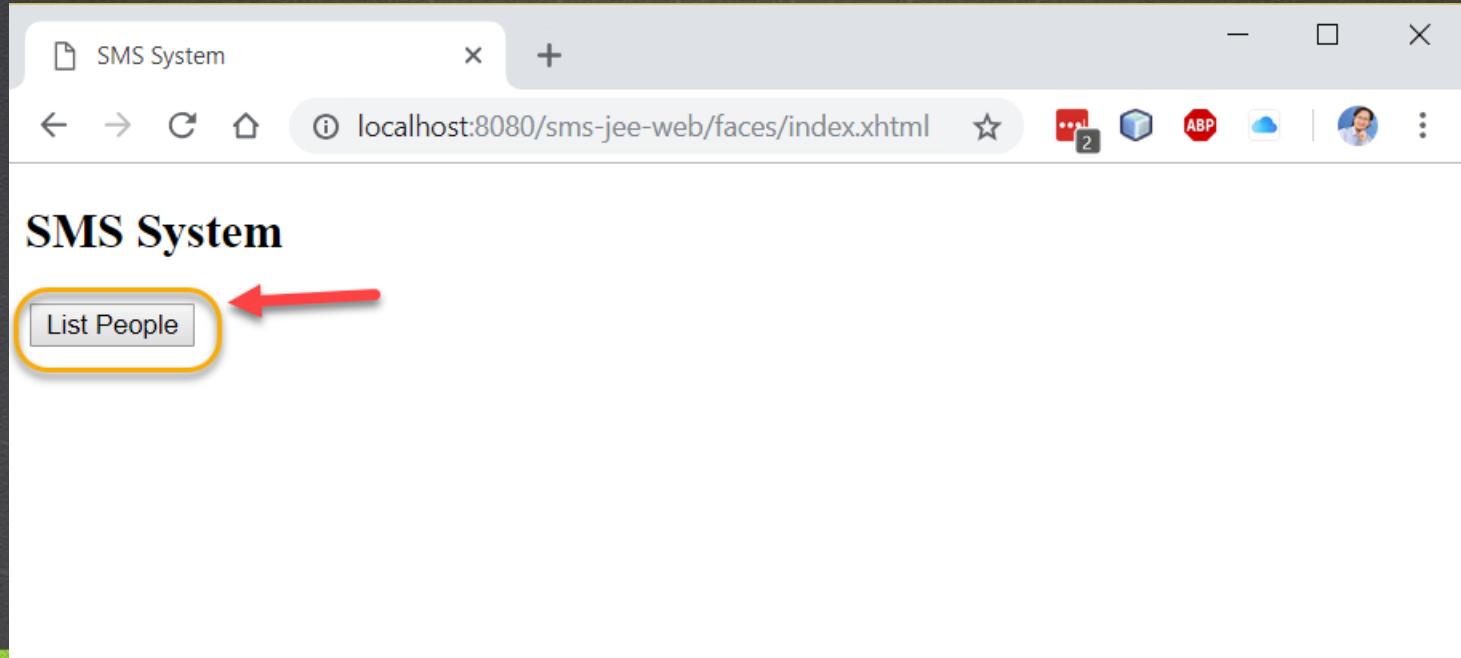
19. EXECUTE THE PROJECT

We execute the project, we give right click and run:



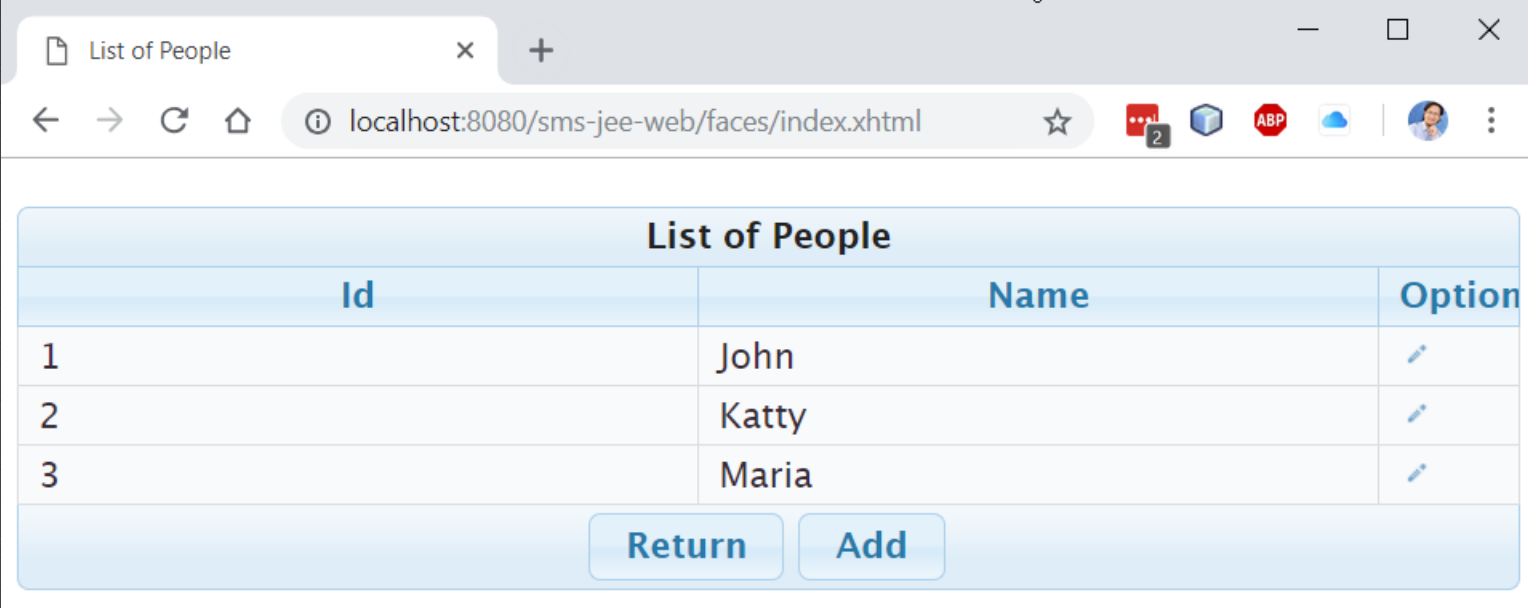
PASO 19. EJECUTAMOS EL PROYECTO

El resultado es como sigue:






19. EXECUTE THE PROJECT

The result is as follows:



The screenshot shows a web browser window with the title 'List of People'. The address bar displays 'localhost:8080/sms-jee-web/faces/index.xhtml'. The page content features a table titled 'List of People' with three columns: 'Id', 'Name', and 'Option'. The table contains three rows of data. Below the table, there are two buttons: 'Return' and 'Add'.

List of People		
Id	Name	Option
1	John	
2	Katty	
3	Maria	

[Return](#) [Add](#)

EXTRA TASKS

It is left as an exercise to create the cases of Add, Edit and Delete (right click) to Person.

Test with different values and verify the result of each case.



Experiencia y Conocimiento para tu vida

JAVA EE COURSE

www.globalmentoring.com.mx

EXTRA NOTES OF THE EXERCISE

- To download and / or try more CSS themes of primeFaces you can consult the following link:

<https://repository.primefaces.org/org/primefaces/themes/>



IN CASE OF ERRORS

- In case of errors, it is recommended to load the resolved project to download configuration problems, and in case it works, it is necessary to compare with the resolved project to correct any code problem.



JAVA EE COURSE

www.globalmentoring.com.mx

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 Java Enterprise application.

JSF technology, in combination with AJAX and extensions such as PrimeFaces, iceFaces or RichFaces allow us to quickly improve our Web and Business applications, as well as the configuration and integration with the EJB has been greatly simplified, to the extent of being able to eliminate the file faces-config.xml .

This project is already an excellent base for your own Java Web and Business projects using the JSF / AJAX / PrimeFaces + EJB + JPA technologies.

JAVA EE COURSE

www.globalmentoring.com.mx

ONLINE COURSE

JAVA EE JAKARTA EE

By: Eng. Ubaldo Acosta



JAVA EE COURSE

www.globalmentoring.com.mx