

JAVASERVER FACES COURSE

EXERCISE

INTERNATIONALIZATION WITH JSF

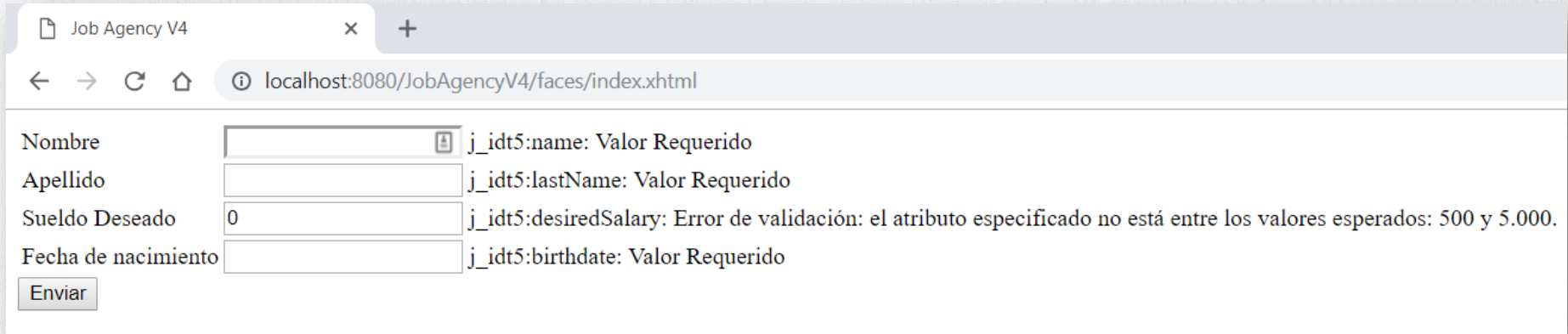


JAVASERVER FACES COURSE

www.globalmentoring.com.mx

EXERCISE OBJECTIVE

- The objective of the exercise is to modify our project to add the concept of internationalization in JSF. The result should be similar to the following:



A screenshot of a web browser window titled "Job Agency V4" showing a JSF form at the URL "localhost:8080/JobAgencyV4/faces/index.xhtml". The form contains four input fields: "Nombre", "Apellido", "Sueldo Deseado", and "Fecha de nacimiento". Each field has a corresponding validation error message displayed to its right. The "Sueldo Deseado" field contains the value "0" and has a validation error message: "Error de validación: el atributo especificado no está entre los valores esperados: 500 y 5.000." Below the form is an "Enviar" button.

Nombre	<input type="text"/>	j_idt5:name: Valor Requerido
Apellido	<input type="text"/>	j_idt5:lastName: Valor Requerido
Sueldo Deseado	<input type="text" value="0"/>	j_idt5:desiredSalary: Error de validación: el atributo especificado no está entre los valores esperados: 500 y 5.000.
Fecha de nacimiento	<input type="text"/>	j_idt5:birthdate: Valor Requerido

EXERCISE OBJECTIVE

The objective of this exercise is to put into practice the concept of Internationalization in JSF.

This will be achieved by creating a property file with a different language, based on the Java I18n specification.

In this way we can handle different languages in our JSF application.

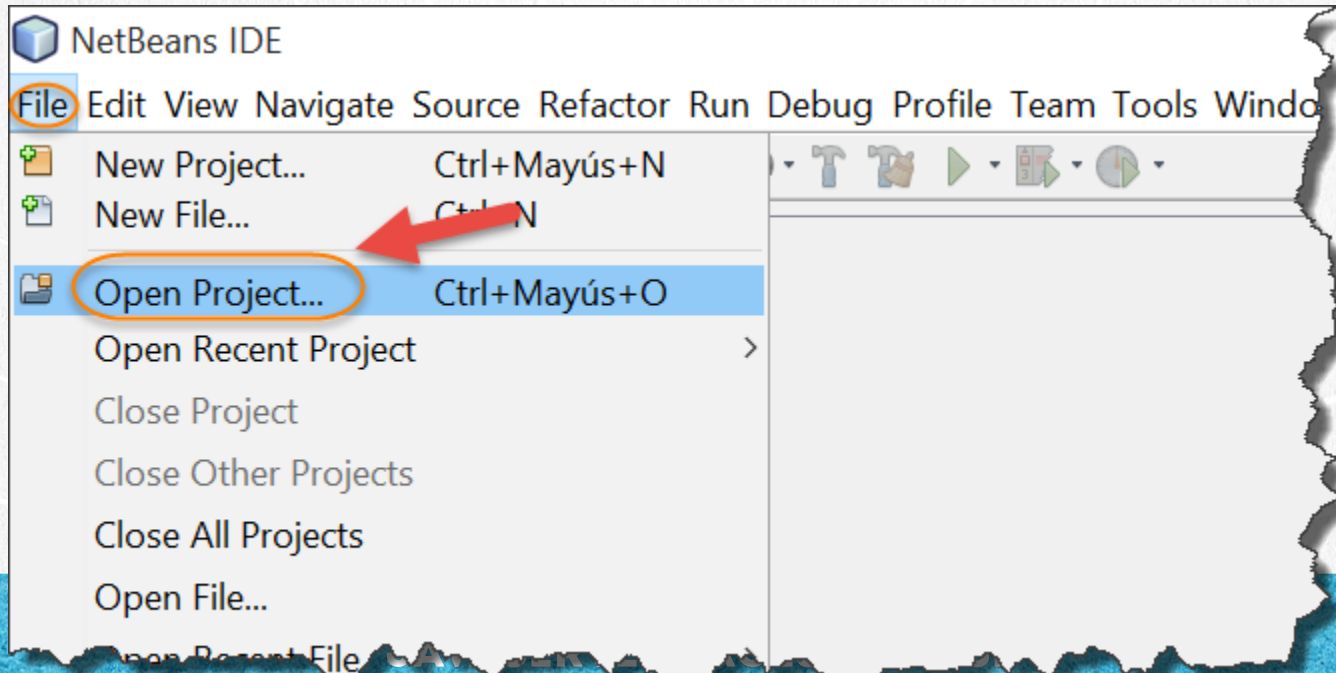


JAVASERVER FACES COURSE

www.globalmentoring.com.mx

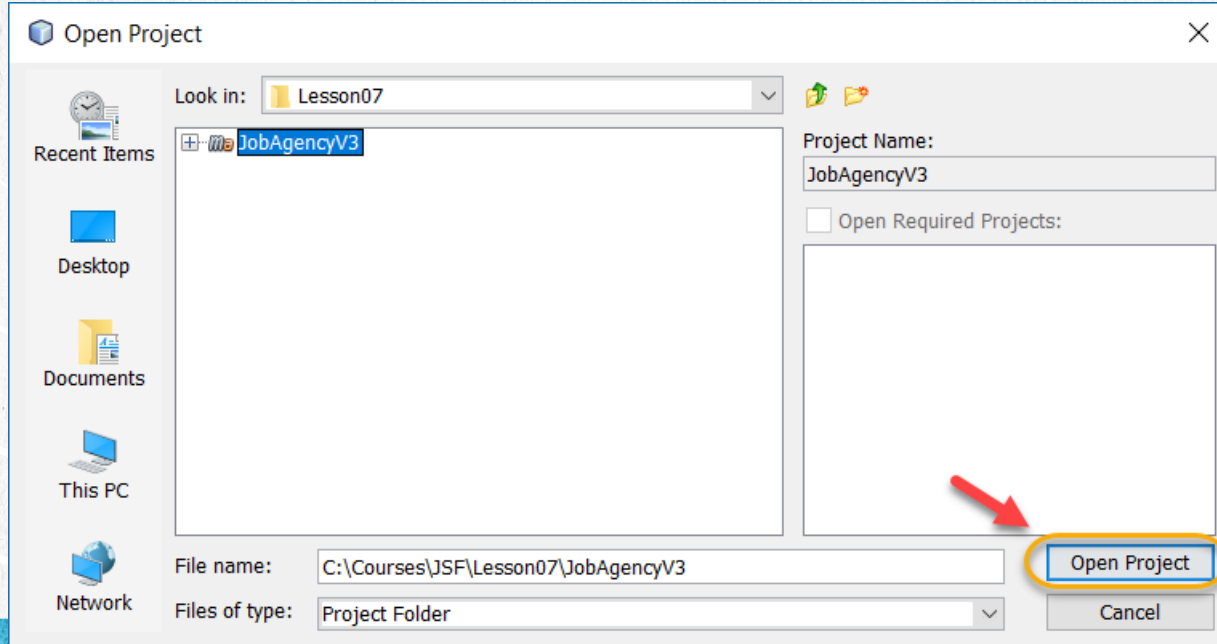
1. OPEN THE PROJECT

We open the JobAgencyV3 project, just in case we do not have it open:



1. OPEN THE PROJECT

We open the JobAgencyV3 project, just in case we do not have it open:

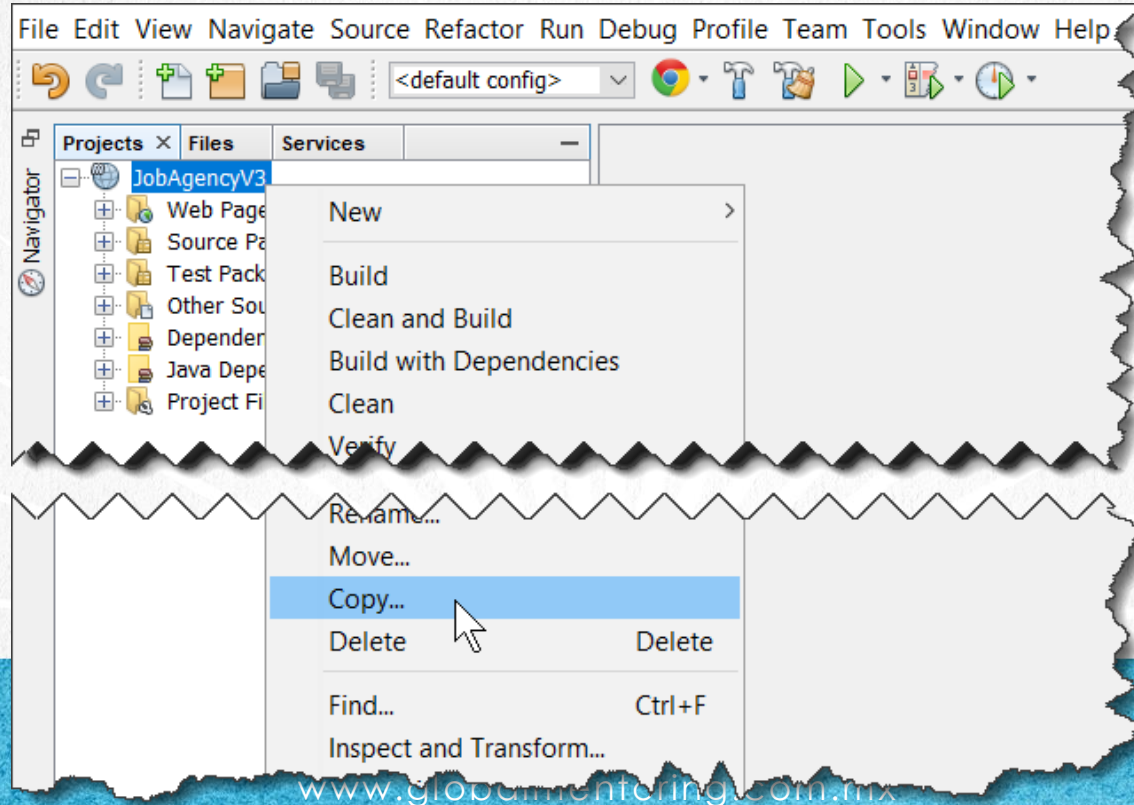


JAVASERVER FACES COURSE

www.globalmentoring.com.mx

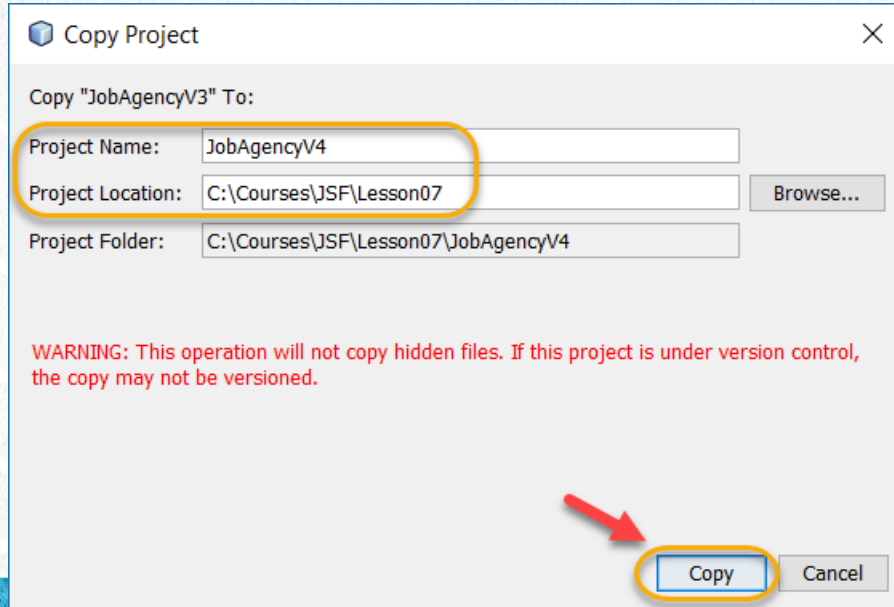
2. COPY THE PROJECT

Copy the project JobAgencyV3:



2. COPY THE PROJECT

We copy the JobAgencyV3 project, we rename it to JobAgencyV4 and deposit it in the new path shown:

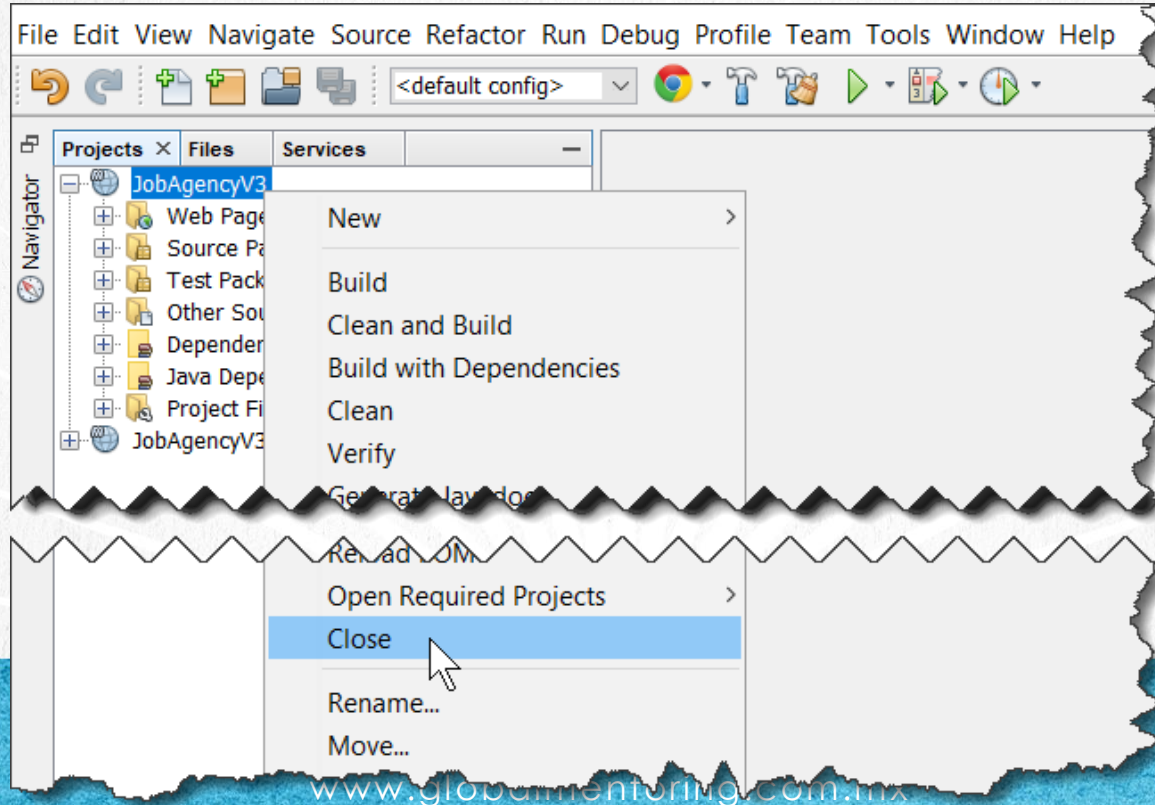


JAVASERVER FACES COURSE

www.globalmentoring.com.mx

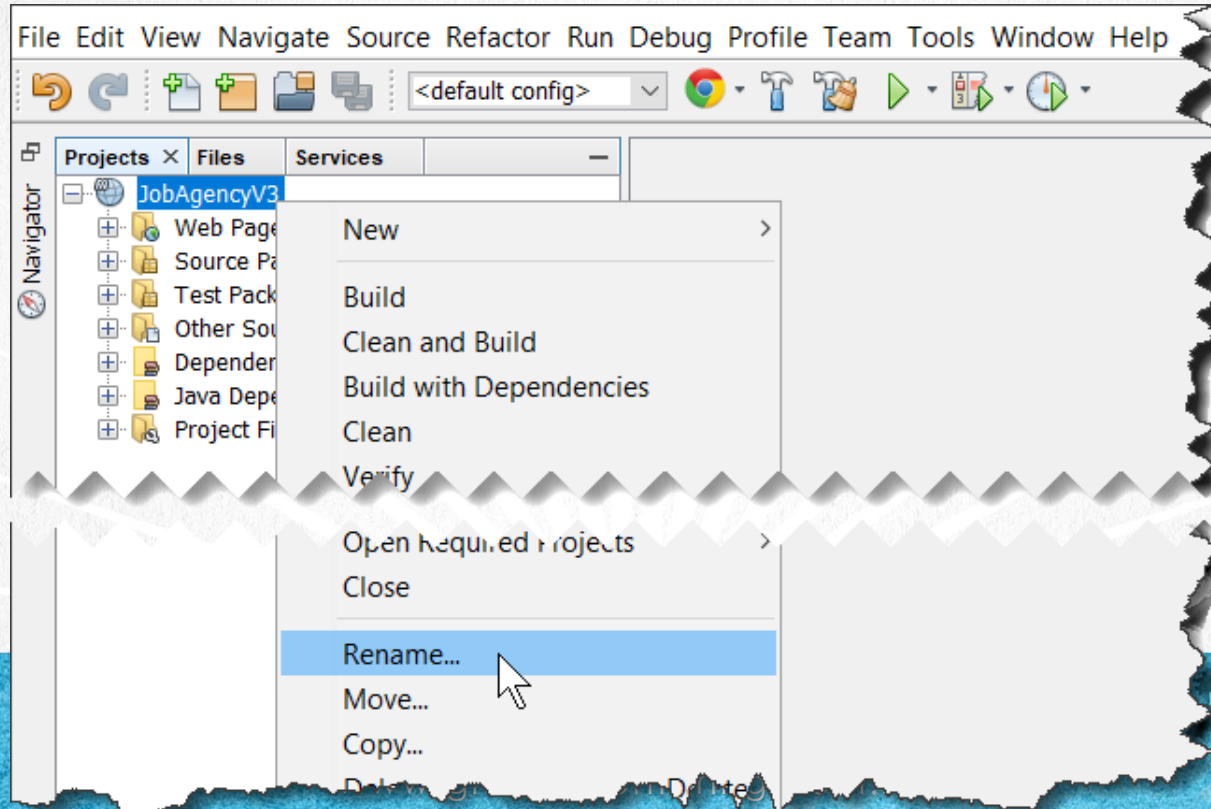
3. CLOSE THE PROJECT

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



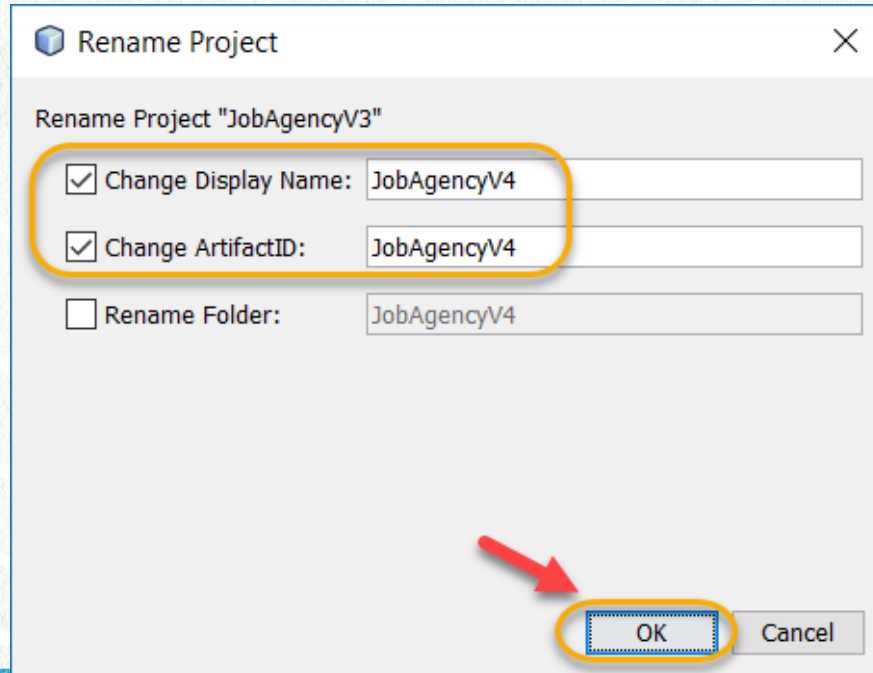
4. RENAME THE PROJECT

Change the name of the project to JobAgencyV4:



4. RENAME THE PROJECT

We changed the name of the project to JobAgencyV4:

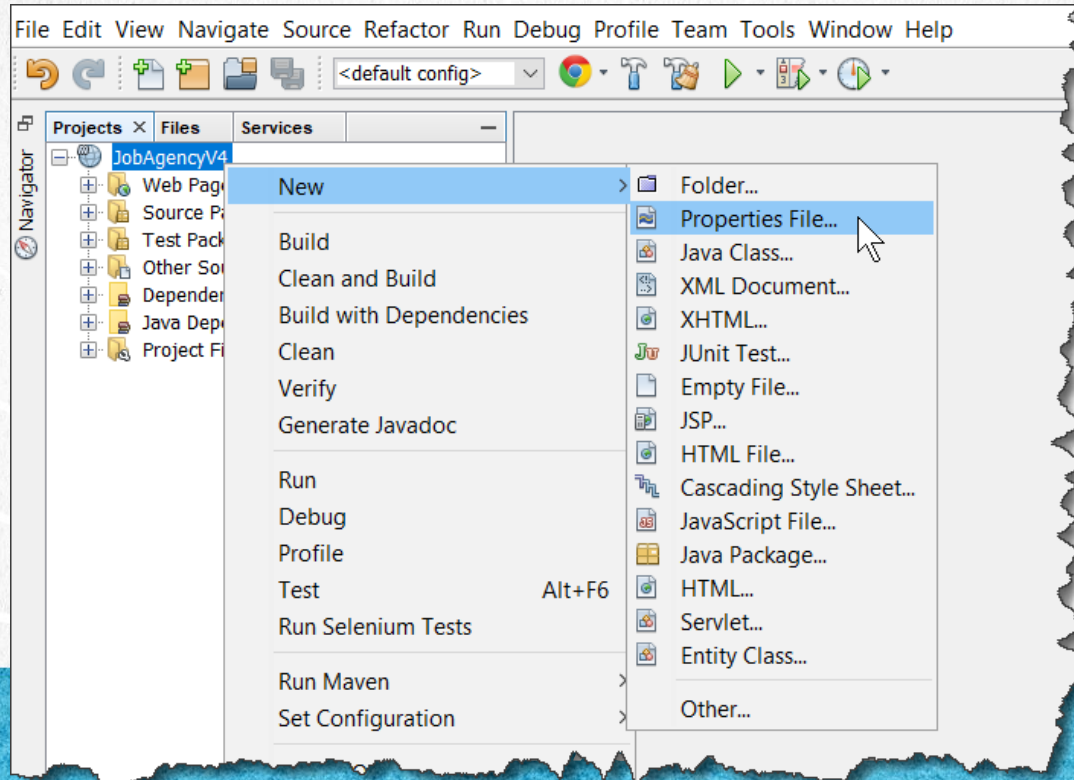


JAVASERVER FACES COURSE

www.globalmentoring.com.mx

5. CREATE A PROPERTIES FILE

We create a properties file named messages_es.properties:



5. CREATE A PROPERTIES FILE

We create a properties file named messages_es.properties:

New Properties File

Steps

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

Name and Location

File Name:

Project:

Folder:

Created File:

6. MODIFY THE CODE

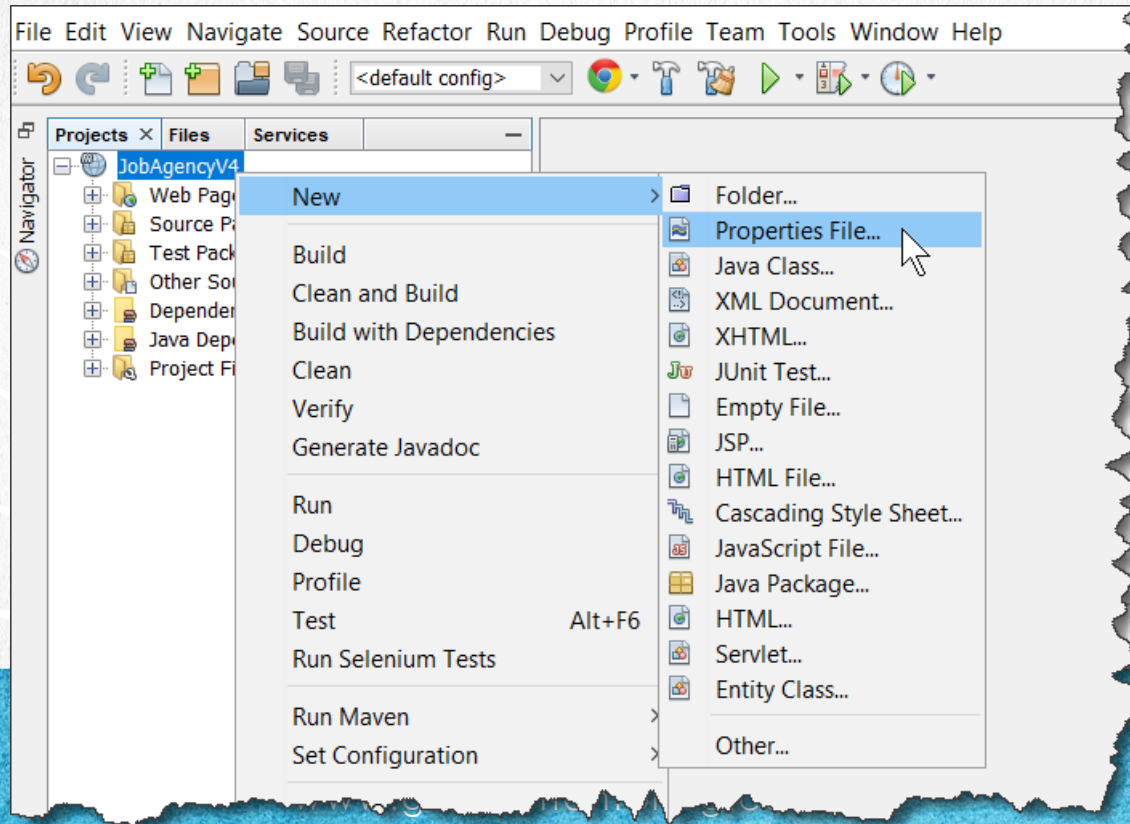
[messages_es.properties:](#)

Click to download

```
vacantForm.name=Nombre  
vacantForm.lastName=Apellido  
vacantForm.desiredSalary=Sueldo Deseado  
vacantForm.birthdate=Fecha de nacimiento  
vacantForm.send=Enviar
```

7. CREATE A PROPERTIES FILE

We create a properties file called jsf_es.properties:



7. CREATE A PROPERTIES FILE

We create a properties file called jsf_es.properties:

New Properties File

Steps

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

Name and Location

File Name:

Project:

Folder:

Created File:

8. MODIFY THE CODE

[jsf_es.properties:](#)

Click to download

```
javax.faces.component.UIInput.REQUIRED={0}: Valor Requerido
```


9. MODIFY THE XML FILE

We modified the faces-config.xml file. This is where we specify the default language we want. Locate the <default-locale> tag in the faces-config.xml file and change the default language from en (English) to es (Spanish) :

If we want to change it programmatically we can use the following code, specifying the desired language and region:

```
FacesContext.getCurrentInstance().getViewRoot().setLocale("es");
```

Let's see how the modified file is :

9. MODIFY THE CODE

[faces-config.xml:](#)

Click to download

```
<?xml version="1.0" encoding="UTF-8"?>
<faces-config
  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-
facesconfig_2_3.xsd"
  version="2.3">
  <lifecycle>
    <phase-listener>
      beans.lifecycle.DebugListener
    </phase-listener>
  </lifecycle>
  <application>
    <locale-config>
      <default-locale>es</default-locale>
    </locale-config>
    <!--form labels-->
    <resource-bundle>
      <!--No need to add the file extension-->
      <base-name>messages</base-name>
      <var>msgs</var>
    </resource-bundle>
    <!--change validator texts-->
    <message-bundle>jsf</message-bundle>
  </application>
</faces-config>
```


10. MODIFY THE XHTML FILE

It is not necessary to modify the index.xhtml file, because we have already used the tags in the messages.properties file, and just changing the language will now take the tags from the messages_en.properties file

This is precisely the benefit of using the l18n standard in the name of the property files that handle the tags.

From the index.xhtml file, we only modify the title so that now JobAgencyV4 appears.

10. MODIFY THE CODE

[index.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://xmlns.jcp.org/jsf/core">
  <h:head>
    <title>Job Agency V4</title>
  </h:head>
  <h:body>
    <h:form>
      <h:messages globalOnly="true" ></h:messages>
      <table>
        <tr>
          <td><h:outputLabel for="name" value="#{msgs['vacantForm.name']}" /></td>
          <td><h:inputText id="name" required="true" value="#{candidate.name}" /></td>
          <td><h:message for="name" /></td>
        </tr>
        <tr>
          <td><h:outputLabel for="lastName" value="#{msgs['vacantForm.lastName']}" /></td>
          <td><h:inputText id="lastName" required="true" value="#{candidate.lastName}" /></td>
          <td><h:message for="lastName" /></td>
        </tr>
      </table>
    </h:form>
  </h:body>
</html>
```


10. MODIFY THE CODE

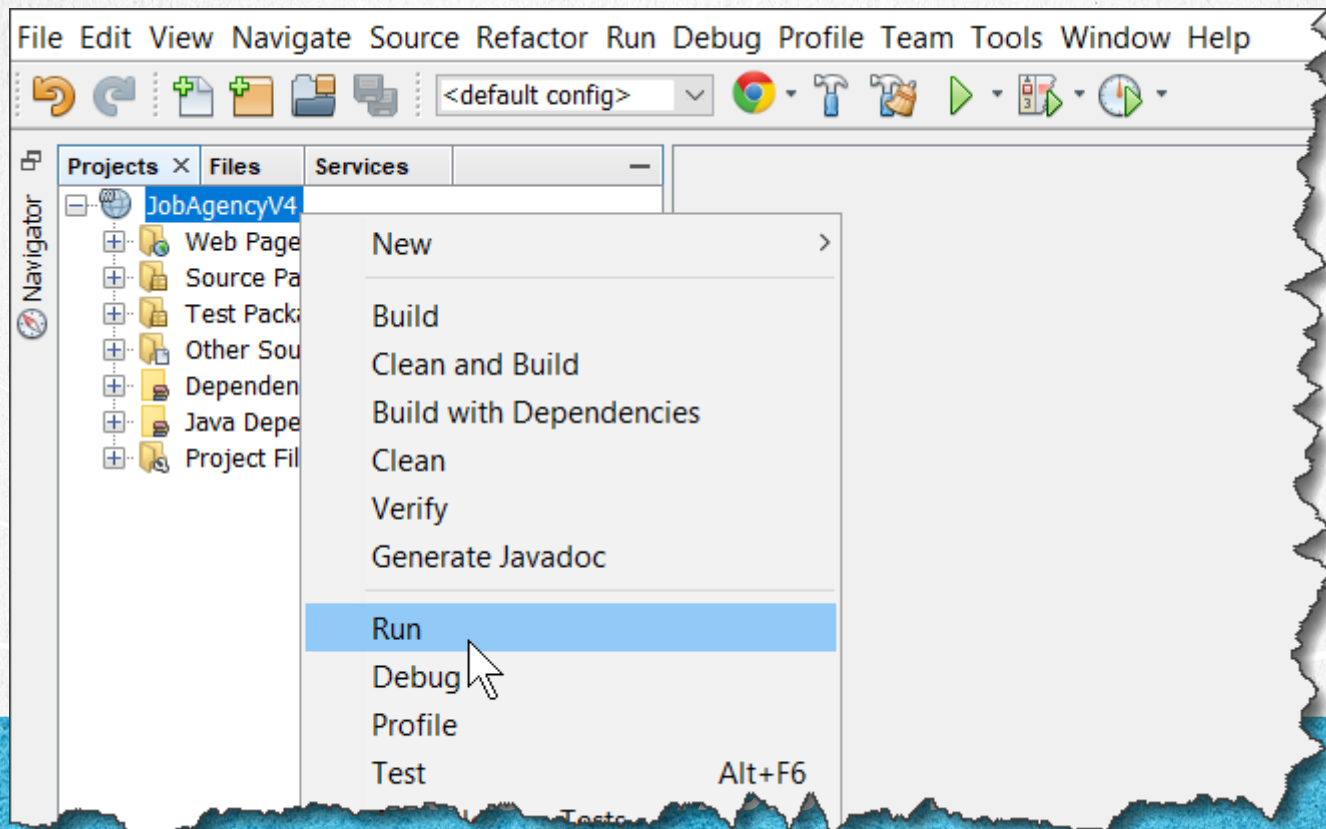
[index.xhtml:](#)

Click to download

```
<tr>
  <td><h:outputLabel for="desiredSalary" value="#{msgs['vacantForm.desiredSalary']}" /></td>
  <td><h:inputText id="desiredSalary" required="true"
    value="#{candidate.desiredSalary}">
    <f:validateLongRange minimum="500" maximum="5000"/>
  </h:inputText></td>
  <td><h:message for="desiredSalary" /></td>
</tr>
<tr>
  <td><h:outputLabel for="birthdate" value="#{msgs['vacantForm.birthdate']}" /></td>
  <td><h:inputText id="birthdate" required="true"
    value="#{candidate.birthdate}">
    <f:convertDateTime pattern="MM/dd/yyyy" />
  </h:inputText></td>
  <td><h:message for="birthdate" /></td>
</tr>
</table>
<h:commandButton action="#{vacantForm.send}" value="#{msgs['vacantForm.send']}" />
</h:form>
</h:body>
</html>
```

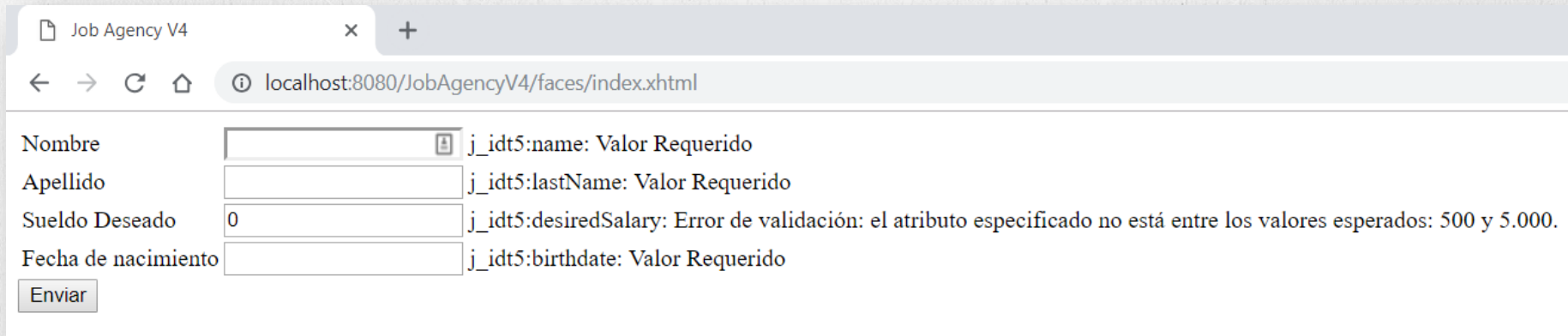
11. EXECUTE THE PROJECT

Execute the Project:



11. EXECUTE THE PROJECT

We execute the project. The result is similar to the following:



The screenshot shows a web browser window with the title 'Job Agency V4'. The address bar displays 'localhost:8080/JobAgencyV4/faces/index.xhtml'. The page contains a registration form with the following fields and validation messages:

Field Label	Field Value	Validation Message
Nombre		j_idt5:name: Valor Requerido
Apellido		j_idt5:lastName: Valor Requerido
Sueldo Deseado	0	j_idt5:desiredSalary: Error de validación: el atributo especificado no está entre los valores esperados: 500 y 5.000.
Fecha de nacimiento		j_idt5:birthdate: Valor Requerido

Below the form is an 'Enviar' button.

LAST STEP

Finally, we return the language to English in the faces-config.xml file:

```
<locale-config>  
    <default-locale>en</default-locale>  
</locale-config>
```



JAVASERVER FACES COURSE

www.globalmentoring.com.mx

EXERCISE CONCLUSION

With this exercise we put into practice the concept of Internationalization with JSF. In this way we can modify the language of our application without affecting the programming and only adding property files respecting the Java I18n standard.

With this we conclude the topic of message handling and internationalization with JSF.

For more information on the topic of internationalization and the I18n standard with Java, you can consult the following link:

<https://docs.oracle.com/javase/tutorial/i18n/intro/quick.html>

ONLINE COURSE

JAVASERVER FACES (JSF)

By: Eng. Ubaldo Acosta



JAVASERVER FACES COURSE

www.globalmentoring.com.mx