

**JAVASERVER FACES COURSE**

# **EXERCISE**

## **HANDLING OF VALUE CHANGE LISTENER IN JSF**



**JAVASERVER FACES COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

# EXERCISE OBJECTIVE

The objective of this exercise is to implement the Value Change Listener concept.

We will configure several extra fields, and we will associate an event when the zip code field changes, this will allow modifying the value of the City field automatically. The result is shown below:

Job Agency V5

localhost:8080/JobAgencyV5/faces/index.xhtml

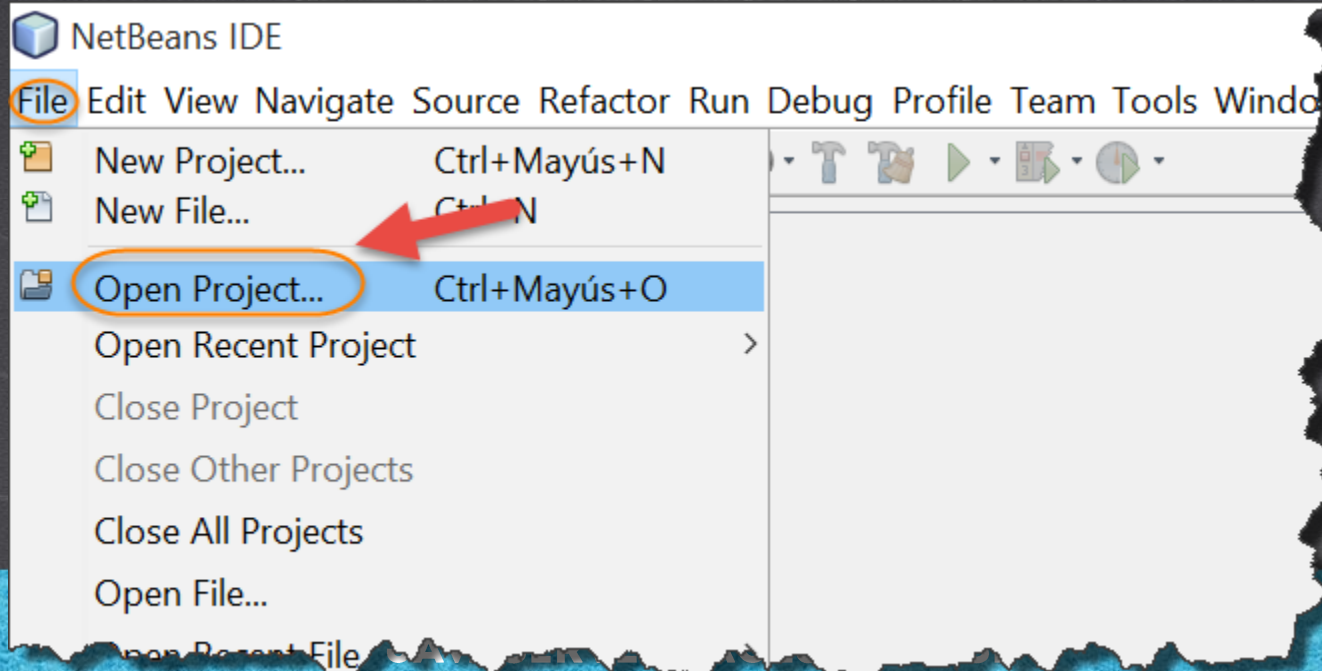
Name	<input type="text" value="Introduce your name"/>
Last Name	<input type="text"/>
Desired Salary	<input type="text" value="0"/>
Birthdate	<input type="text"/>
Zip Code	<input type="text" value="85001"/>
City	<input type="text" value="PHOENIX"/>

Send



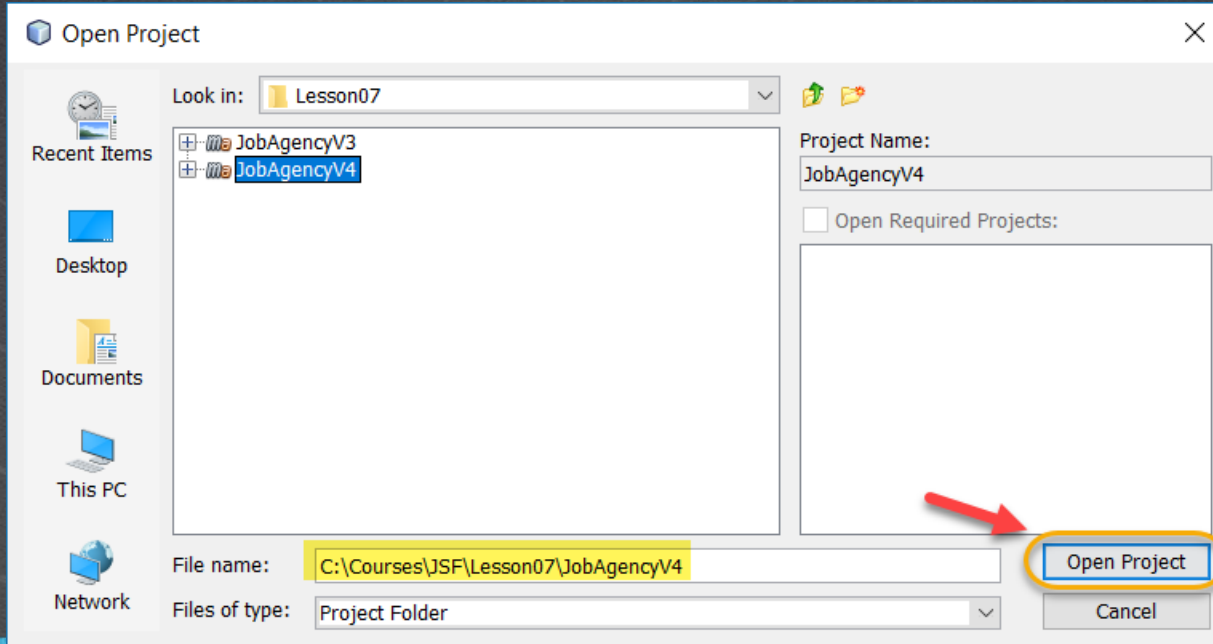
# 1. OPEN THE PROJECT

We open the JobAgencyV4 project, just in case we do not have it open:



# 1. OPEN THE PROJECT

We open the JobAgencyV4 project, just in case we do not have it open:

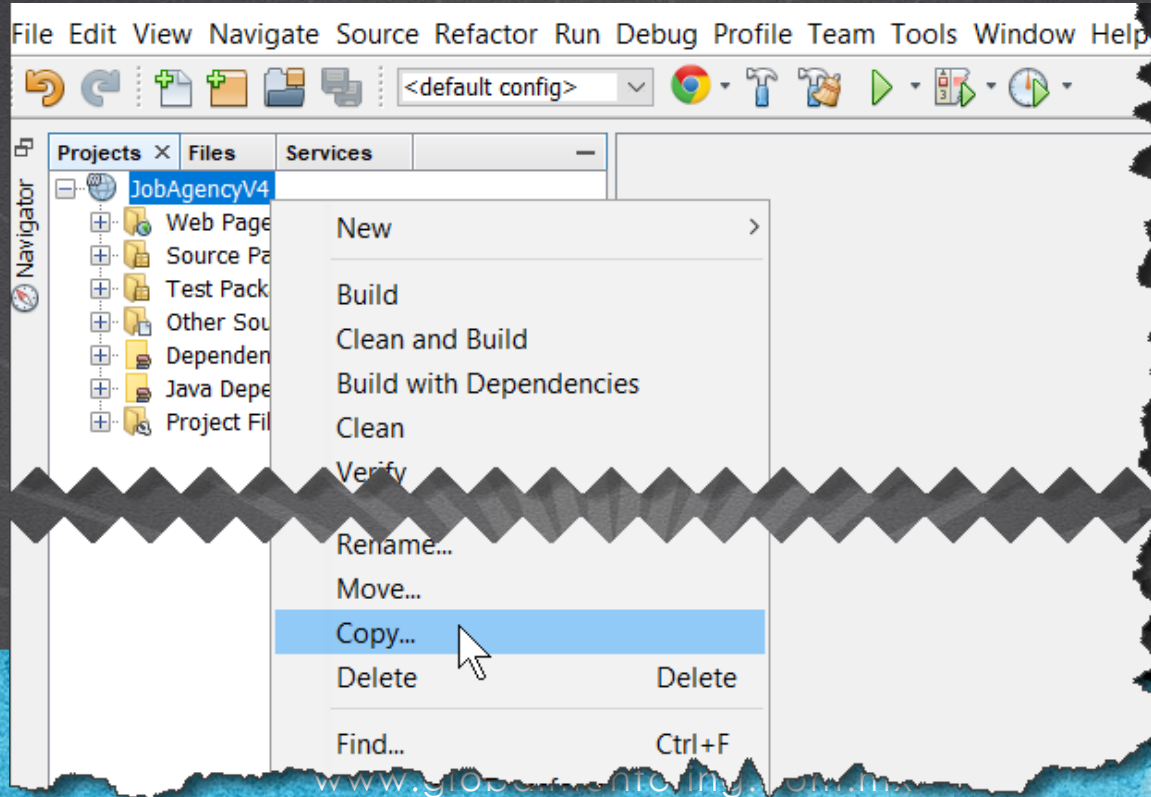


**JAVASERVER FACES COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

## 2. COPY THE PROJECT

We copy the JobAgencyV4 project:





## 2. COPY THE PROJECT

We copy the JobAgencyV4 project and rename it to JobAgencyV5 and deposit it in the new path shown:

Copy Project

Copy "JobAgencyV4" To:

Project Name: JobAgencyV5

Project Location: C:\Courses\JSF\Lesson08 Browse...

Project Folder: C:\Courses\JSF\Lesson08\JobAgencyV5

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

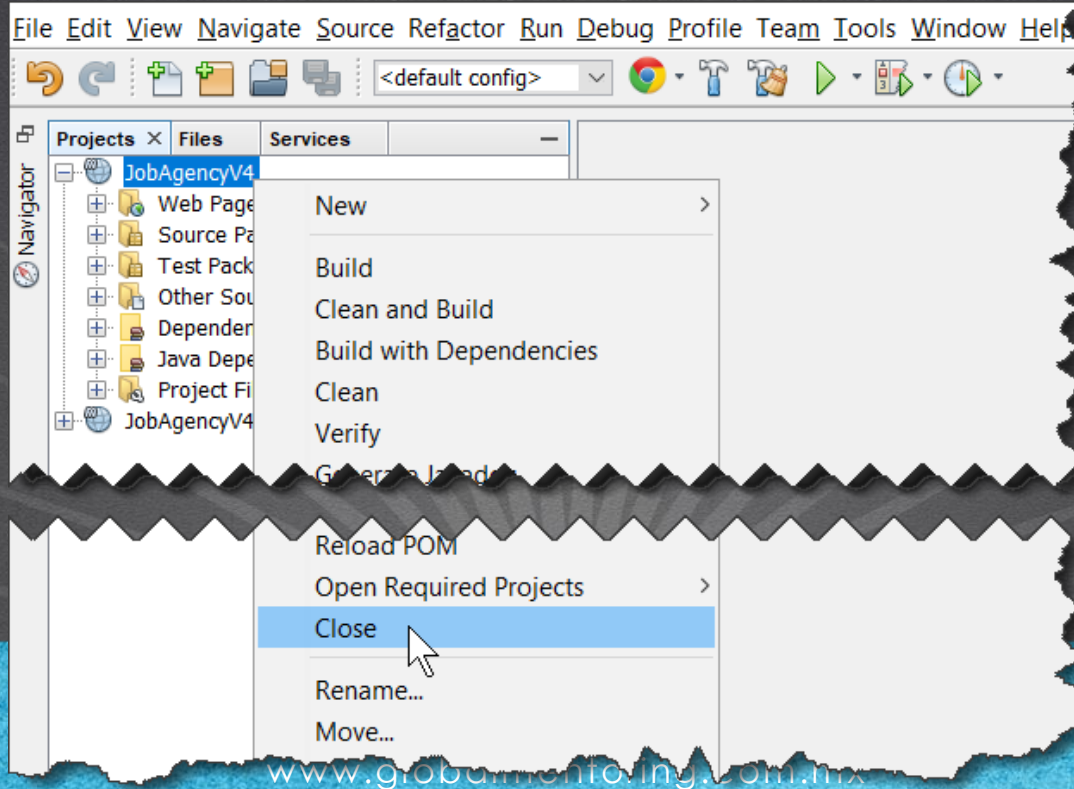
Copy Cancel

**JAVASERVER FACES COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

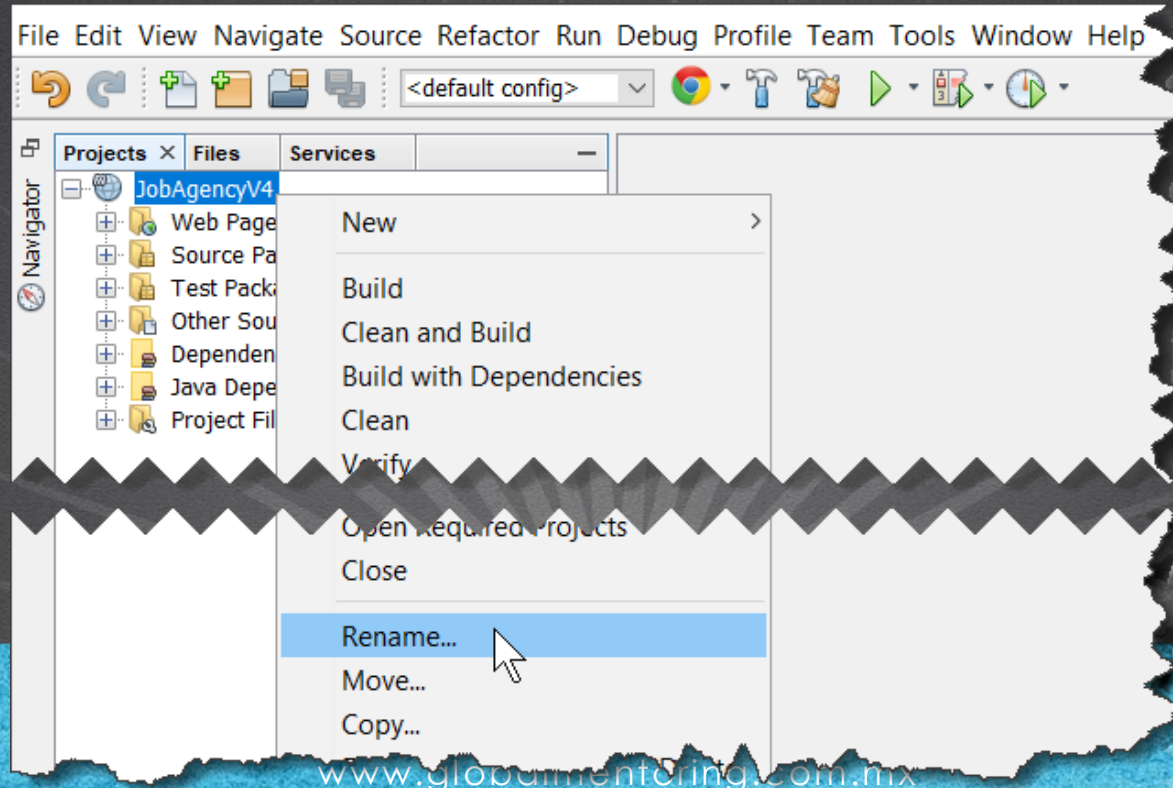
### 3. CLOSE THE PROJECT

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



## 4. RENAME THE PROJECT

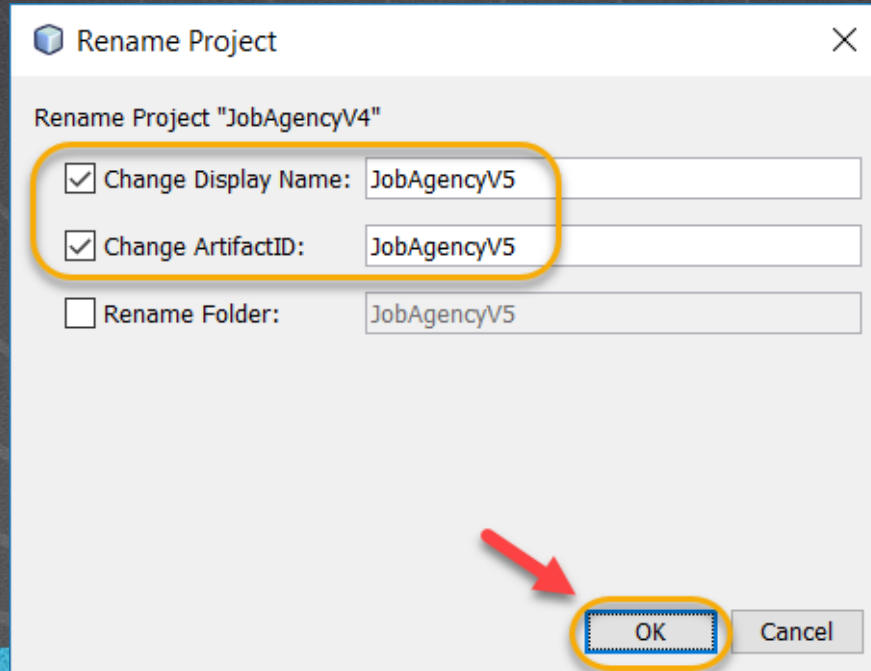
We change the name of the project to JobAgencyV5:





## 4. RENAME THE PROJECT

We change the name of the project to JobAgencyV5:



**JAVASERVER FACES COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

## 5. MODIFY AN XHTML FILE

We changed the index.xhtml file as follows:

Add to the tag h: form the attribute id = "vacantForm", the result should be like the following :

```
<h:form id="vacantForm">
```



## 5. MODIFY AN XHTML FILE

Add the attribute `zipCode` to the file `index.xhtml`. Add the following code, creating a new line:

```
<tr>
  <td><h:outputLabel for="zipCode" value="#{msgs['vacanteForm.zipCode']}"
  /></td>
  <td><h:inputText id="zipCode" immediate="true"
  onchange="this.form.submit()" required="true"
  value="#{candidate.zipCode}"
  valueChangeListener="#{vacantForm.zipCodeListener}" /></td>
  <td><h:message for="zipCode" /></td>
</tr>
```



## 5. MODIFY AN XHTML FILE

Add the city attribute to the index.xhtml file. Add the following code, creating a new line:

```
<tr>
  <td><h:outputLabel for="city" value="#{msgs['vacanteForm.city']}"
  /></td>
  <td><h:inputText id="city" required="true"
  value="#{candidate.city}" /></td>
  <td><h:message for="city" /></td>
</tr>
```

# 5. 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 V5</title>
  </h:head>
  <h:body>
    <h:form id="vacantForm">
      <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>
        <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>
      </table>
    </h:form>
  </h:body>
</html>
```

# 5. MODIFY THE CODE

[index.xhtml:](#)

Click to download

```
<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>
<tr>
  <td><h:outputLabel for="zipCode" value="#{msgs['vacantForm.zipCode']}" /></td>
  <td><h:inputText id="zipCode" immediate="true"
    onchange="this.form.submit()" required="true"
    value="#{candidate.zipCode}"
    valueChangeListener="#{vacantForm.zipCodeListener}" /></td>
  <td><h:message for="zipCode" /></td>
</tr>
<tr>
  <td><h:outputLabel for="city" value="#{msgs['vacantForm.city']}" /></td>
  <td><h:inputText id="city" required="true"
    value="#{candidate.city}" /></td>
  <td><h:message for="city" /></td>
</tr>
</table>
<h:commandButton action="#{vacantForm.send}" value="#{msgs['vacantForm.send']}" />
</h:form>
</h:body>
</html>
```



## 6. MODIFY THE JAVA CLASS

We changed the code of the class Candidate.java.

We add the following fields:

```
private String zipCode;  
private String city;
```

Also the getters / setters methods of each attribute. At the end the Candidate.java class should be as follows:

## 6. MODIFY THE CODE

### Candidate.java:

Click to download

```
package beans.model;

import java.util.Date;
import javax.inject.Named;
import javax.enterprise.context.RequestScoped;
import org.apache.logging.log4j.*;

@Named
@RequestScoped
public class Candidate {

    Logger log = LogManager.getRootLogger();

    private String name;
    private String lastName;
    private int desiredSalary;
    private Date birthdate;
    private String zipCode;
    private String city;

    public Candidate() {
        log.info("Creating the Candidate object");
        this.setName("Introduce your name");
    }

    public String getName() {
        return name;
    }
}
```

# 6. MODIFY THE CODE

Candidate.java:

Click to download

```
public void setName(String name) {
    this.name = name;
    log.info("Modifying the name property:" + this.name);
}

public String getLastName() {
    return lastName;
}

public void setLastName(String lastName) {
    this.lastName = lastName;
    log.info("Modifying the lastName property:" + this.name);
}

public int getDesiredSalary() {
    return desiredSalary;
}

public void setDesiredSalary(int desiredSalary) {
    this.desiredSalary = desiredSalary;
    log.info("Modifying the desiredSalary property:" + this.desiredSalary);
}

public Date getBirthdate() {
    return birthdate;
}
```



# 6. MODIFY THE CODE

Candidate.java:

Click to download

```
public void setBirthdate(Date birthdate) {
    this.birthdate = birthdate;
    log.info("Modifying the birthdate property:" + this.birthdate);
}

public String getZipCode() {
    return zipCode;
}

public void setZipCode(String zipCode) {
    this.zipCode = zipCode;
    log.info("Modifying the zipCode property:" + this.zipCode);
}

public String getCity() {
    return city;
}

public void setCity(String city) {
    this.city = city;
    log.info("Modifying the city property:" + this.city);
}
}
```

## 7. MODIFY THE JAVA CLASS

We changed the code of the VacantForm.java class. We add a method of type Value Change Listener, so that when we modify the value in the zipCode field, in automatic fill the field of City.

The VacantForm class should be as follows:



**JAVASERVER FACES COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

# 7. MODIFY THE CODE

## VacantForm.java:

Click to download

```
package beans.backing;

import beans.model.Candidate;
import javax.inject.*;
import javax.enterprise.context.RequestScoped;
import javax.faces.application.FacesMessage;
import javax.faces.component.*;
import javax.faces.context.FacesContext;
import javax.faces.event.ValueChangeEvent;
import org.apache.logging.log4j.*;

@Named
@RequestScoped
public class VacantForm {

    Logger log = LogManager.getRootLogger();

    @Inject
    private Candidate candidate;

    public VacantForm() {
        log.info("Creating VacantForm object");
    }

    public void setCandidate(Candidate candidate) {
        this.candidate = candidate;
    }
}
```



# 7. MODIFY THE CODE

VacantForm.java:

Click to download

```
public String send() {
    log.info("send() Name=" + this.candidate.getName());
    log.info("send() Desired Salary=" + this.candidate.getDesiredSalary());

    if (this.candidate.getName().equals("John")) {

        if (this.candidate.getLastName().equals("Smith")) {
            String msg = "Thanks, but John Smith is already working with us.";
            FacesMessage facesMessage = new FacesMessage(FacesMessage.SEVERITY_ERROR, msg, msg);
            FacesContext facesContext = FacesContext.getCurrentInstance();
            String componentId = null; //This is a global message
            facesContext.addMessage(componentId, facesMessage);
            return "index";
        }
        return "success";//success.xhtml
    } else {
        return "failure"; //failure.xhtml
    }
}
```

# 7. MODIFY THE CODE

VacantForm.java:

Click to download

```
//Method of type Value Change Listener
public void zipCodeListener(ValueChangeEvent valueChangeEvent) {
    FacesContext facesContext = FacesContext.getCurrentInstance();
    UIViewRoot uiViewRoot = facesContext.getViewRoot();
    String newZipCode = (String) valueChangeEvent.getNewValue();
    if ("85001".equals(newZipCode)) {
        log.info("Modify the city value dynamically with ValueChangeListener");
        //We use the name of the form of index.xhtml to find the component
        UIInput cityInputText = (UIInput) uiViewRoot.findComponent("vacantForm:city");
        String city = "PHOENIX";
        cityInputText.setValue(city);
        cityInputText.setSubmittedValue(city);
        //Return the response
        facesContext.renderResponse();
    }
}
```

## 8. MODIFY THE XML FILE

We change the faces-config.xml file to place the language again in English:

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



**JAVASERVER FACES COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)



## 8. 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>en</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>
```

## 9. MODIFY THE PROPERTIES FILE

We changed the messages.properties file and the messages\_es.properties file to add the new fields.

The files should be as follows:



**JAVASERVER FACES COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

## 9. MODIFY THE CODE

[messages.properties:](#)

Click to download

```
vacantForm.name=Name  
vacantForm.lastName=Last Name  
vacantForm.desiredSalary=Desired Salary  
vacantForm.birthdate=Birthdate  
vacantForm.zipCode=Zip Code  
vacantForm.city=City  
vacantForm.send=Send
```

**JAVASERVER FACES COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)



## 10. MODIFY THE FILE

[messages\\_es.properties:](#)

Click to download

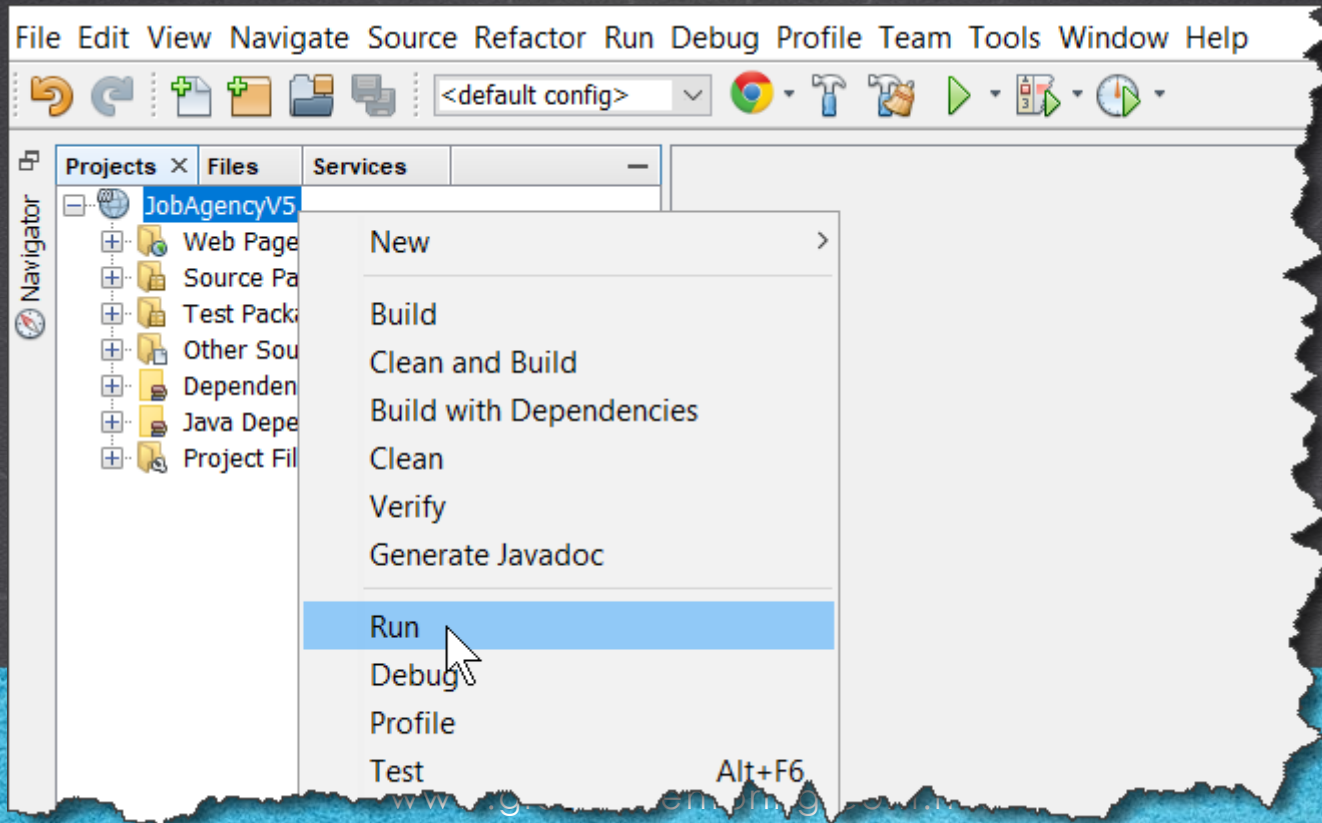
```
vacantForm.name=Nombre  
vacantForm.lastName=Apellido  
vacantForm.desiredSalary=Sueldo Deseado  
vacantForm.birthdate=Fecha de nacimiento  
vacantForm.zipCode=Codigo Postal  
vacantForm.city=Ciudad  
vacantForm.send=Enviar
```

**JAVASERVER FACES COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

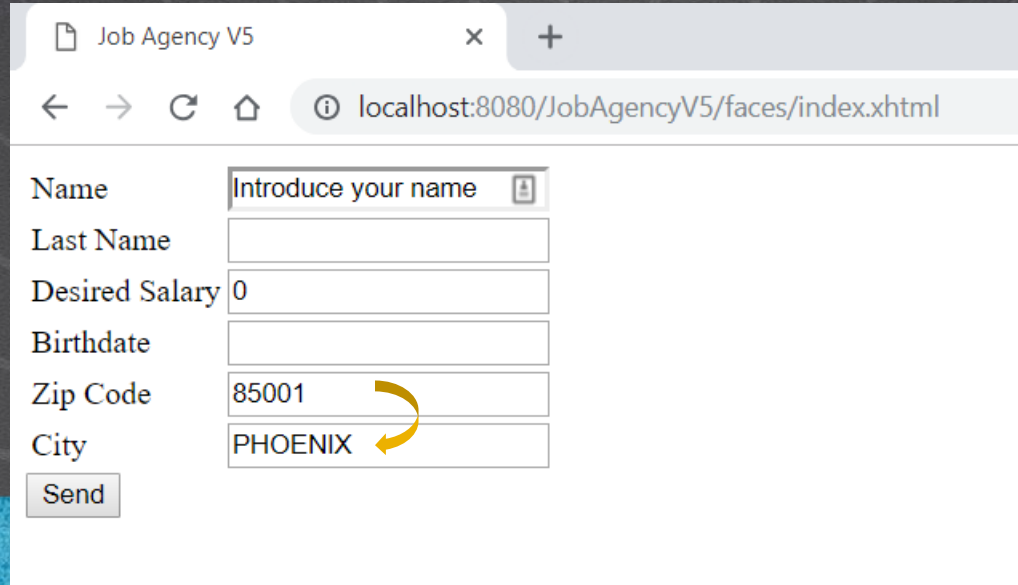
# 11. EXECUTE THE PROJECT

Execute the Project:



# 11. EXECUTE THE PROJECT

We provide the value of **85001** in the postal code value to observe that the colony and city fields are filled automatically. The result is as follows:



The screenshot shows a web browser window with the title 'Job Agency V5'. The address bar displays 'localhost:8080/JobAgencyV5/faces/index.xhtml'. The form contains the following fields:

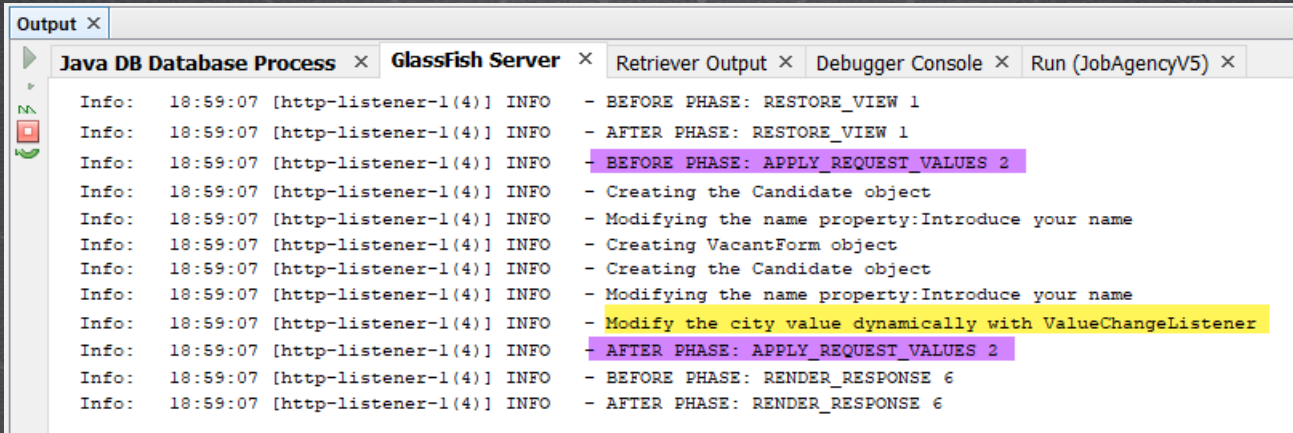
- Name:
- Last Name:
- Desired Salary:
- Birthdate:
- Zip Code:
- City:

A yellow curved arrow points from the Zip Code field to the City field, indicating the automatic population of the city based on the zip code. A 'Send' button is located at the bottom left of the form.



# EXTRA TASKS

- Perform the test with the postal code with the value **85001** and check in the server log the result of executing the value change listener code and verify the result according to the life cycle of JSF.



The screenshot shows the 'Output' window of an IDE, specifically the 'GlassFish Server' tab. It displays a series of log messages from the 'http-listener-1(4)' thread at 18:59:07. The logs track the JSF lifecycle phases and actions. Key entries include 'BEFORE PHASE: RESTORE\_VIEW 1', 'AFTER PHASE: RESTORE\_VIEW 1', 'BEFORE PHASE: APPLY\_REQUEST\_VALUES 2', and 'AFTER PHASE: APPLY\_REQUEST\_VALUES 2'. Between the 'APPLY\_REQUEST\_VALUES' phases, several actions are logged: 'Creating the Candidate object', 'Modifying the name property: Introduce your name', 'Creating VacantForm object', and 'Modify the city value dynamically with ValueChangeListener'. The log ends with 'BEFORE PHASE: RENDER\_RESPONSE 6' and 'AFTER PHASE: RENDER\_RESPONSE 6'.

```
Info: 18:59:07 [http-listener-1(4)] INFO - BEFORE PHASE: RESTORE_VIEW 1
Info: 18:59:07 [http-listener-1(4)] INFO - AFTER PHASE: RESTORE_VIEW 1
Info: 18:59:07 [http-listener-1(4)] INFO - BEFORE PHASE: APPLY_REQUEST_VALUES 2
Info: 18:59:07 [http-listener-1(4)] INFO - Creating the Candidate object
Info: 18:59:07 [http-listener-1(4)] INFO - Modifying the name property: Introduce your name
Info: 18:59:07 [http-listener-1(4)] INFO - Creating VacantForm object
Info: 18:59:07 [http-listener-1(4)] INFO - Creating the Candidate object
Info: 18:59:07 [http-listener-1(4)] INFO - Modifying the name property: Introduce your name
Info: 18:59:07 [http-listener-1(4)] INFO - Modify the city value dynamically with ValueChangeListener
Info: 18:59:07 [http-listener-1(4)] INFO - AFTER PHASE: APPLY_REQUEST_VALUES 2
Info: 18:59:07 [http-listener-1(4)] INFO - BEFORE PHASE: RENDER_RESPONSE 6
Info: 18:59:07 [http-listener-1(4)] INFO - AFTER PHASE: RENDER_RESPONSE 6
```

# EXERCISE CONCLUSION

- With this exercise we have added the concept of Value Change Listener with JSF.
- We made several changes to the index.xhtml form, however the most important change is that when changing the zip code value, a method is automatically called in the Managed Bean of VacanteForm.java, and this method dynamically modify the city value from the VacanteForm.java class.



**ONLINE COURSE**

# **JAVASERVER FACES (JSF)**

By: Eng. Ubaldo Acosta



**JAVASERVER FACES COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)