

**JAVASERVER FACES COURSE**

# **EXERCISE**

## **SELECT ITEM COMPONENT IN JSF**

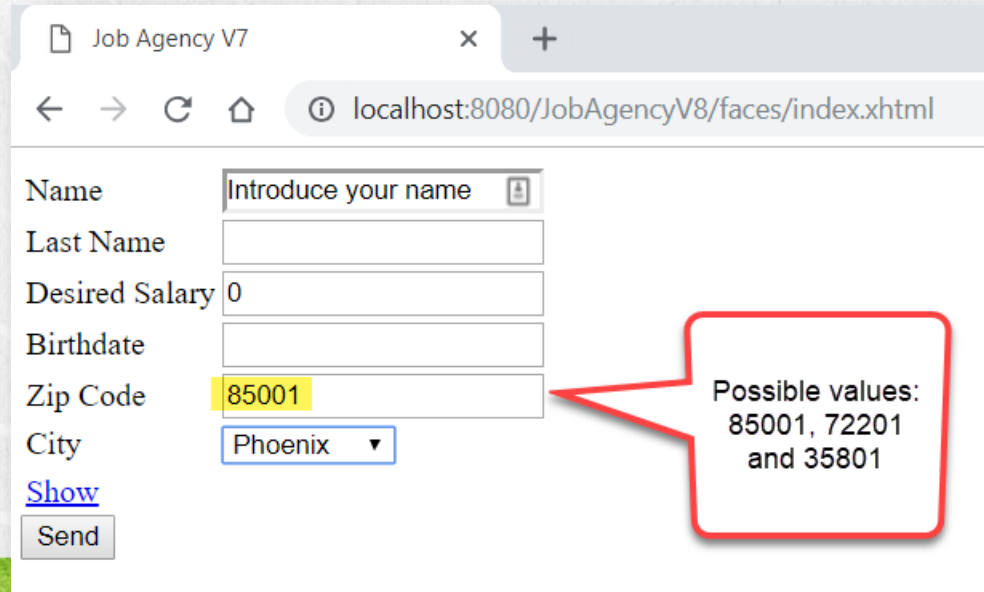


**JAVASERVER FACES COURSE**

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

# EXERCISE OBJECTIVE

The objective of this exercise is to implement the Select Items component of JSF. In addition, we will create the City entity to simulate a list that is obtained from a database. The result will be similar to the following figure:



The screenshot shows a web browser window with the title "Job Agency V7" and the address bar displaying "localhost:8080/JobAgencyV8/faces/index.xhtml". The form contains the following fields:

- Name:
- Last Name:
- Desired Salary:
- Birthdate:
- Zip Code:  (highlighted in yellow)
- City:

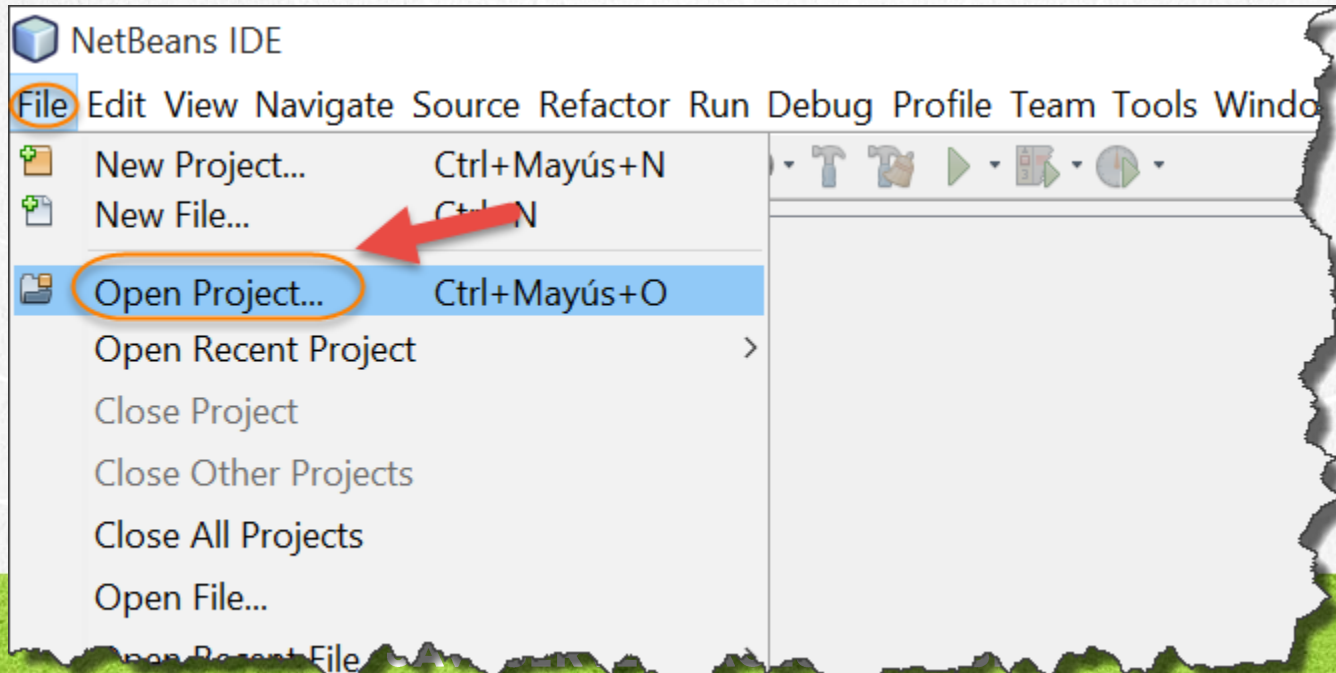
Below the form are a [Show](#) link and a  button. A red callout box points to the Zip Code field with the text: "Possible values: 85001, 72201 and 35801".

**JAVASERVER FACES COURSE**

www.globalmentoring.com.mx

# 1. OPEN THE PROJECT

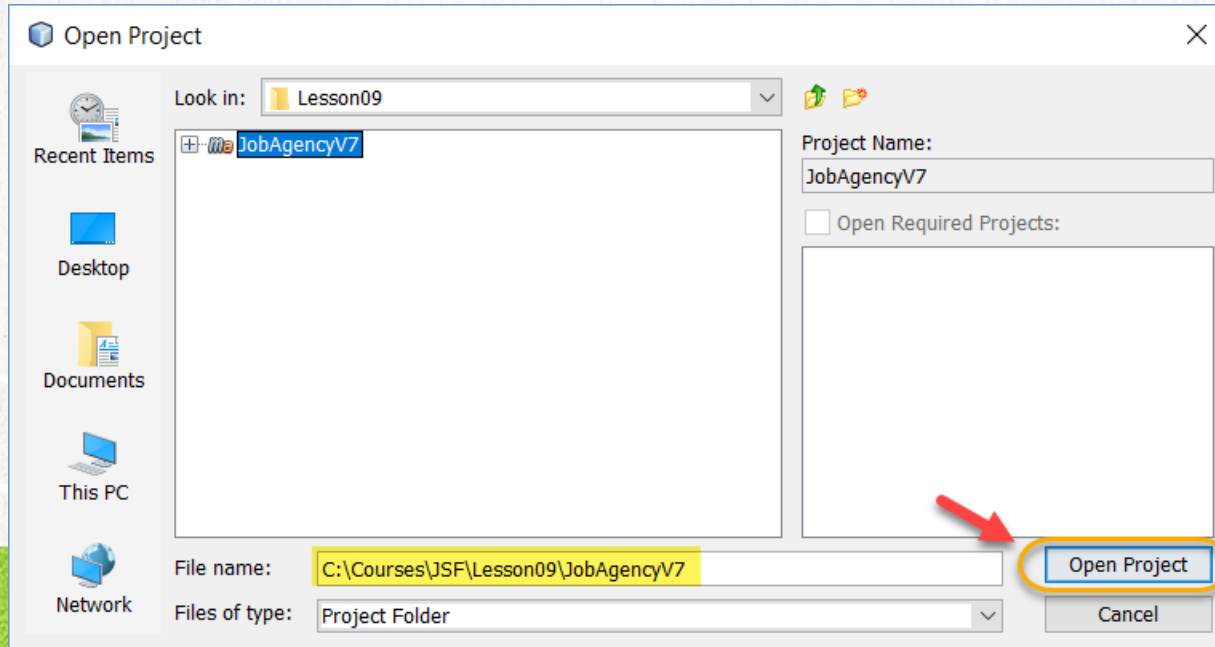
We open the JobAgencyV7 project, just in case we do not have it open:





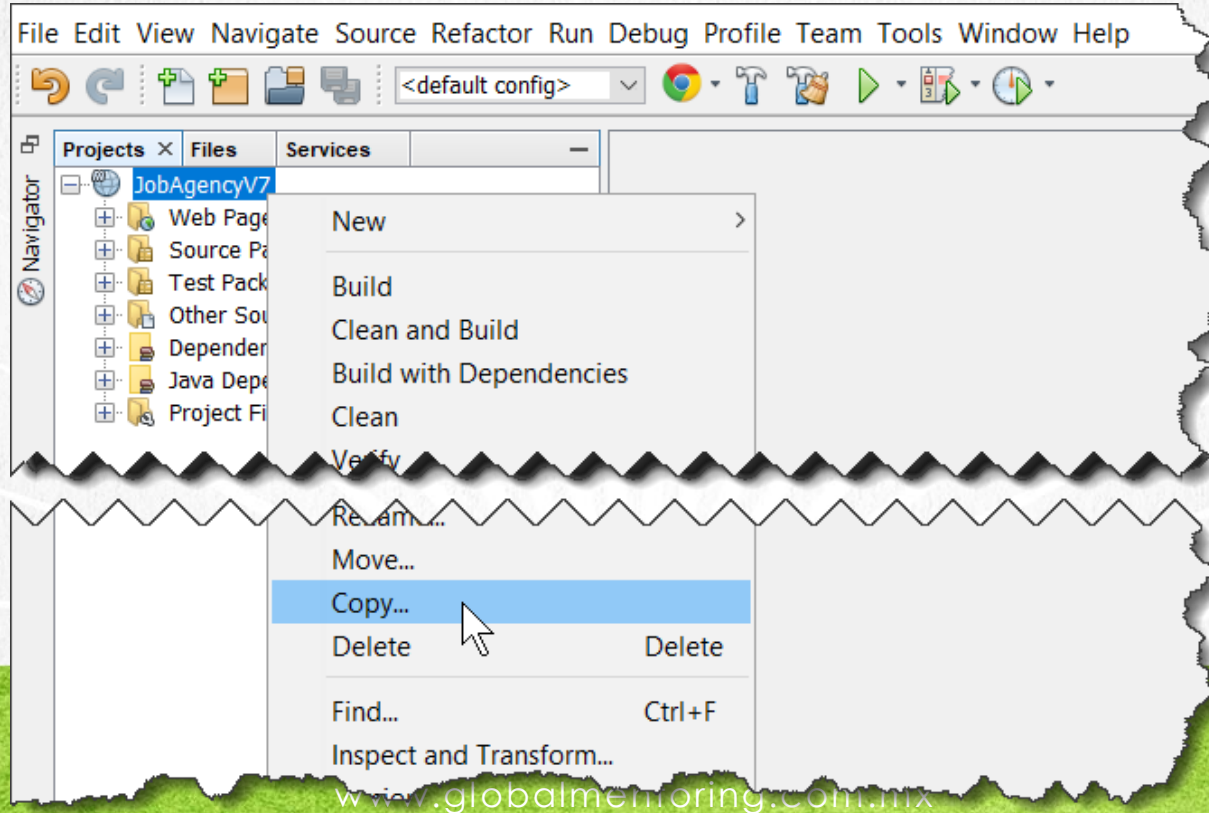
# 1. OPEN THE PROJECT

Abrimos el proyecto JobAgencyV7, sólo en caso de que no lo tengamos ya abierto:



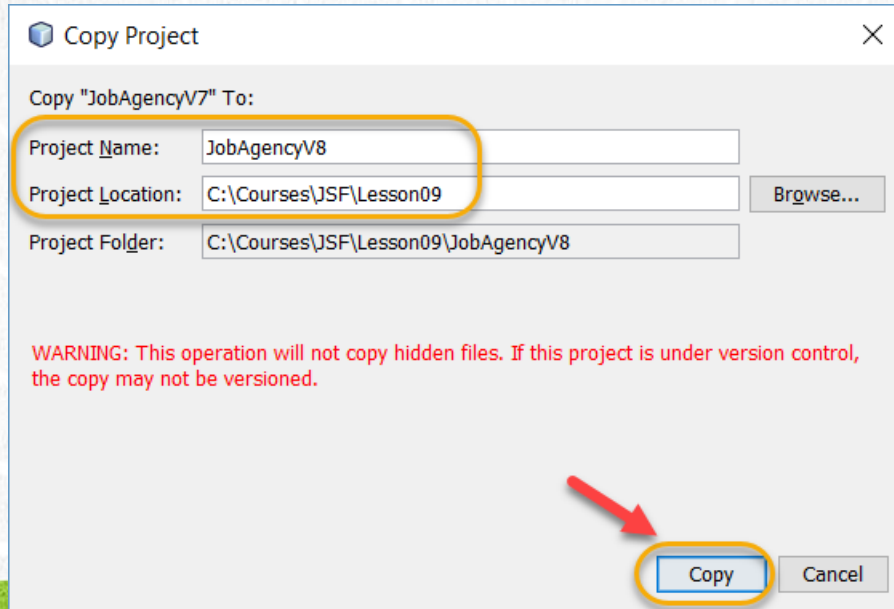
## 2. COPY THE PROJECT

We copy the JobAgencyV7 project:



## 2. COPY THE PROJECT

We copy the JobAgencyV7 project, we rename it to JobAgencyV8 and deposit it in the new path shown:



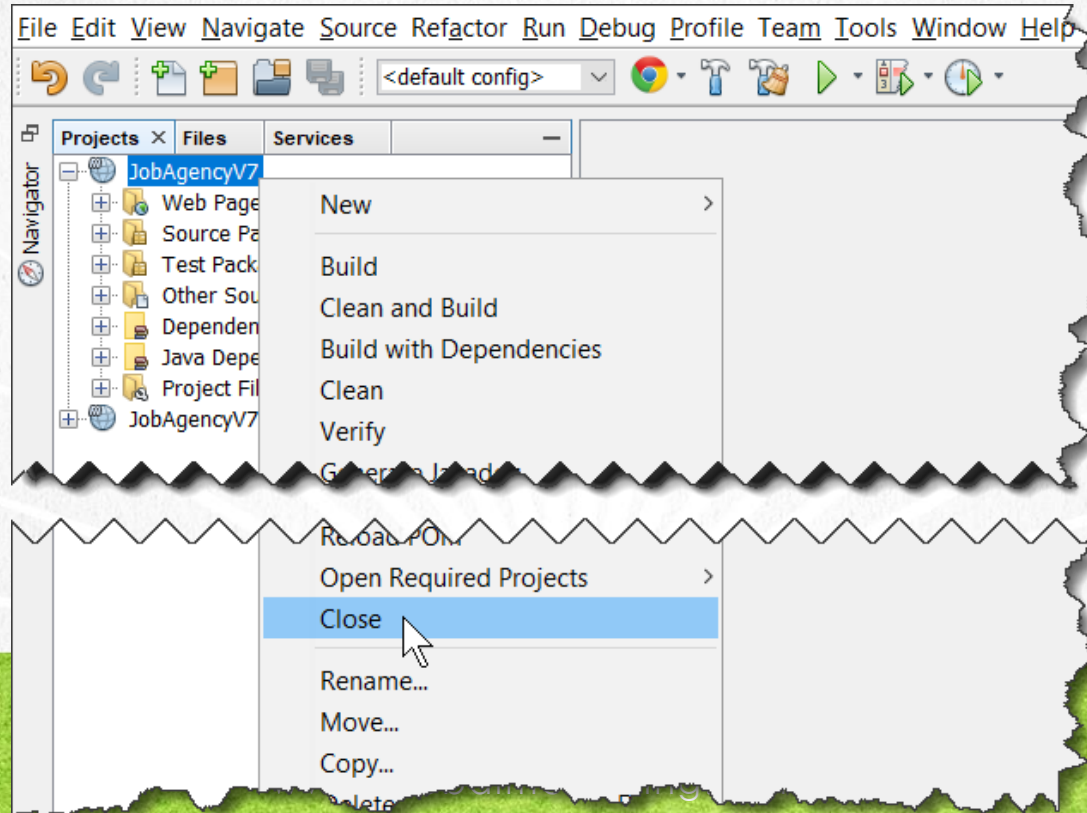
**JAVASERVER FACES COURSE**

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



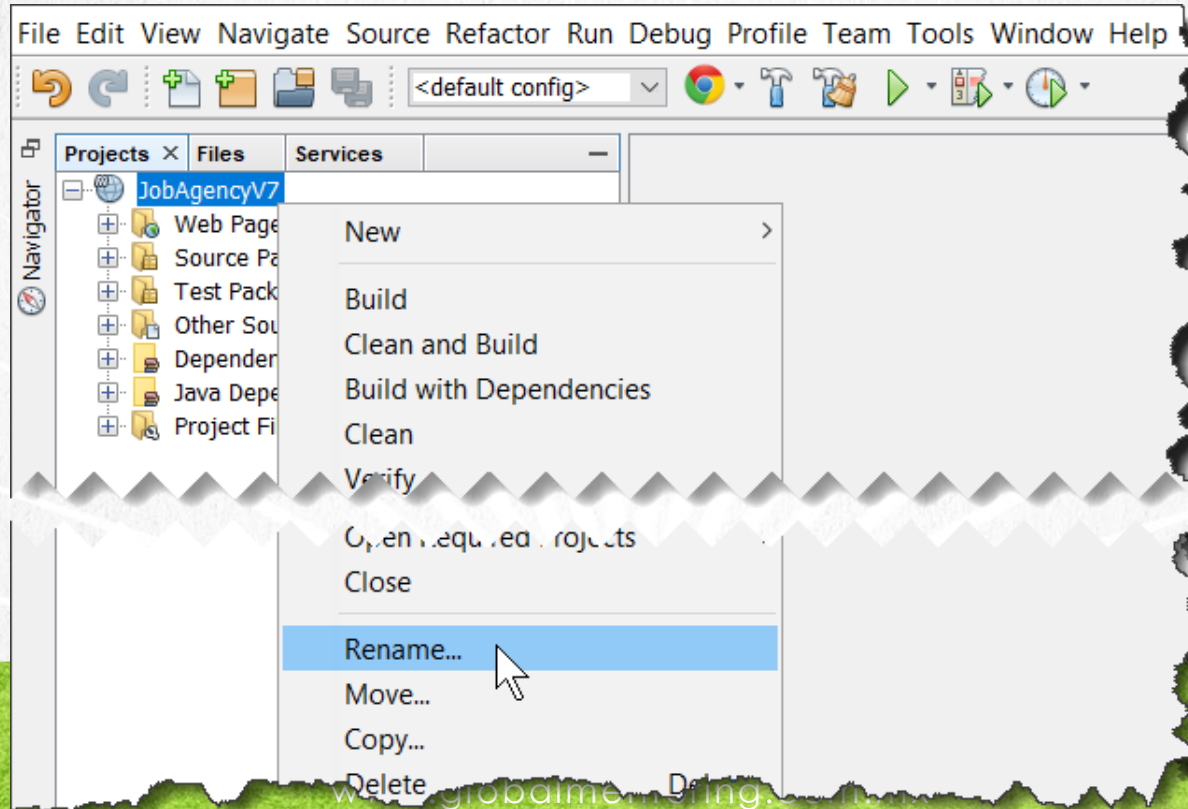
### 3. CLOSE THE PROJECT

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



## 4. RENAME THE PROJECT

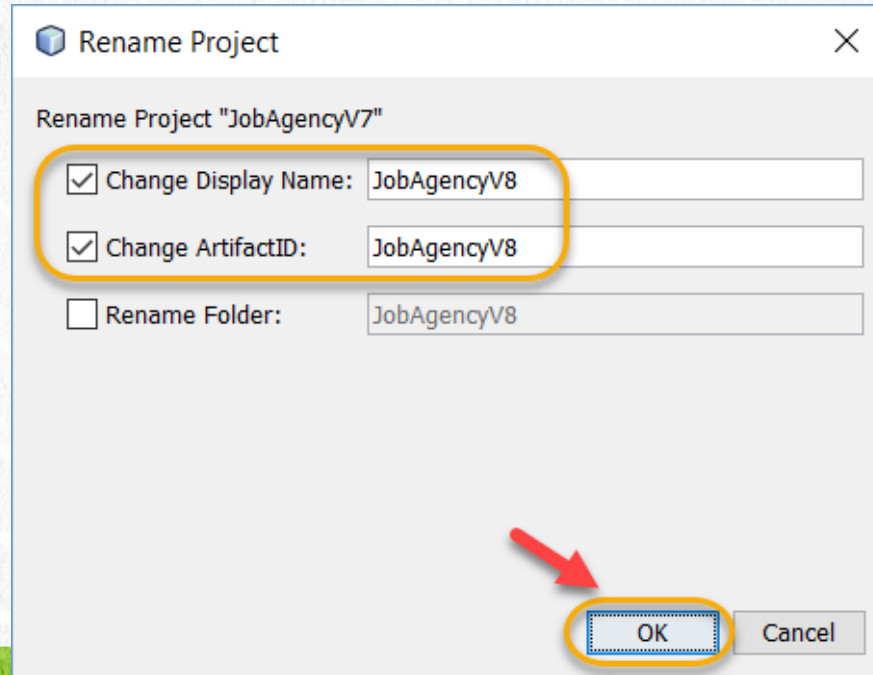
We changed the name of the project to JobAgencyV7:





## 4. RENAME THE PROJECT

We changed the name of the project to JobAgencyV8:

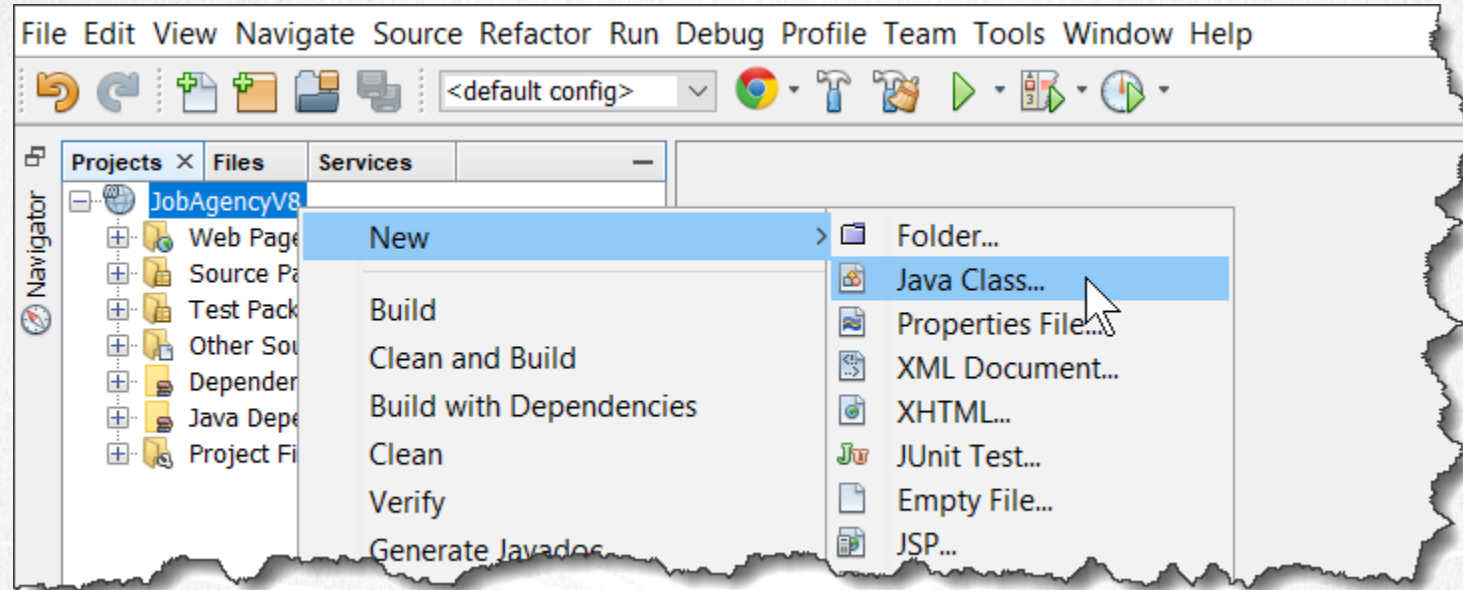


**JAVASERVER FACES COURSE**

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

## 5. CREATE A JAVA CLASS

We create the class City.java as follows:



## 5. CREATE A JAVA CLASS

We create the class City.java as follows:

**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

**JAVASERVER FACES COURSE**

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



## 6. MODIFY THE CODE

City.java:

[Click to download](#)

```
package beans.domain;

public class City {

    private long cityId;
    private String name;
    private long zipCode;

    public City(long cityId, String name, long zipCode) {
        this.cityId = cityId;
        this.name = name;
        this.zipCode = zipCode;
    }

    public long getCityId() {
        return cityId;
    }

    public void setCityId(long cityId) {
        this.cityId = cityId;
    }

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

**JAVASERVER FACES COURSE**

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

## 6. MODIFY THE CODE

City.java:

[Click to download](#)

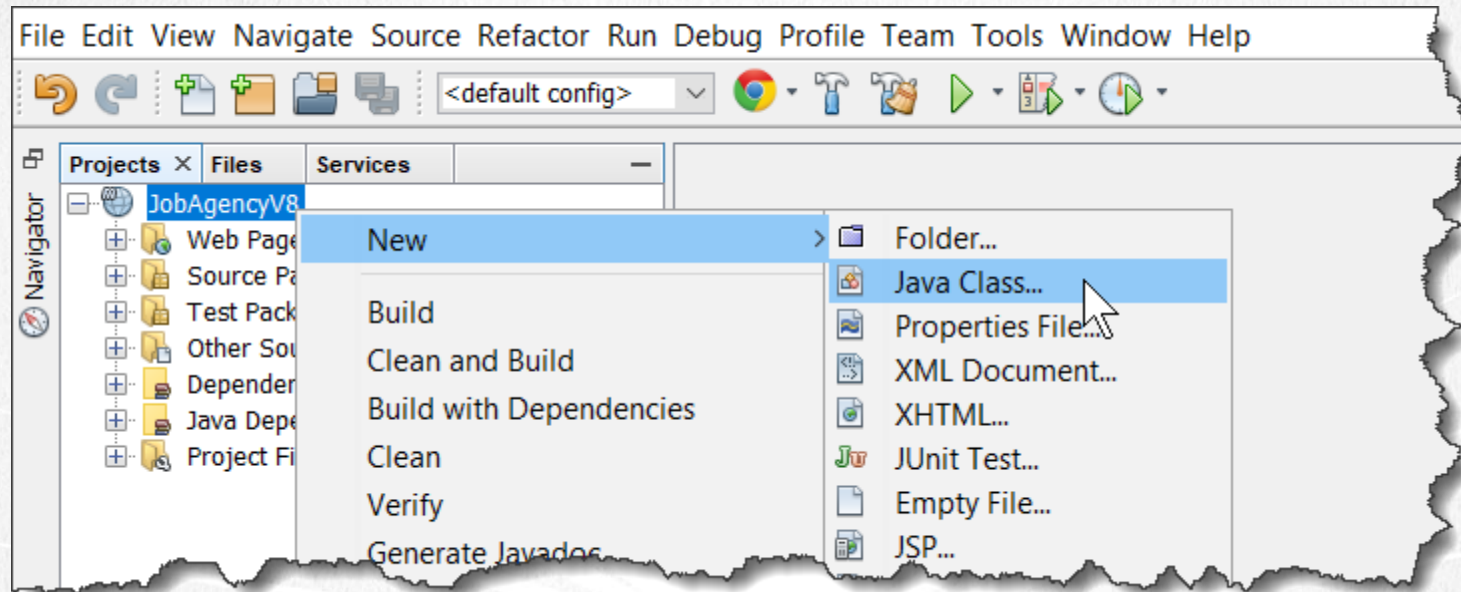
```
public void setName(String name) {  
    this.name = name;  
}  
  
public long getZipCode() {  
    return zipCode;  
}  
  
public void setZipCode(long zipCode) {  
    this.zipCode = zipCode;  
}  
}
```

**JAVASERVER FACES COURSE**

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

## 7. CREATE A JAVA CLASS

We create the CityHelper class as follows :





## 7. CREATE A JAVA CLASS

We create the CityHelper class as follows :

**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

## 8. MODIFY THE CODE

### CityHelper.java:

[Click to download](#)

```
package beans.helper;

import beans.domain.City;
import java.util.ArrayList;
import java.util.List;
import javax.inject.Named;
import javax.enterprise.context.RequestScoped;
import javax.faces.model.SelectItem;

@Named
@RequestScoped
public class CityHelper {

    public long getCityIdByName(String cityName) {
        long cityId = 0;
        List<City> cities = getCities();
        for (City city : cities) {
            if (city.getName().equals(cityName)) {
                cityId = city.getCityId();
                break;
            }
        }
        return cityId;
    }
}
```

**JAVASERVER FACES COURSE**

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

## 8. MODIFY THE CODE

[CityHelper.java:](#)

[Click to download](#)

```
public long getCityIdByZipCode(long zipCode) {
    long cityId = 0;
    List<City> citys = getCities();
    for (City city : citys) {
        if (city.getZipCode() == zipCode) {
            cityId = city.getCityId();
            break;
        }
    }
    return cityId;
}

public List<SelectedItem> getSelectItems() {
    List<SelectedItem> selectItems;
    selectItems = new ArrayList<>();
    List<City> cities = getCities();
    for (City city : cities) {
        SelectedItem selectItem = new SelectedItem(city.getCityId(),
            city.getName());
        selectItems.add(selectItem);
    }
    return selectItems;
}
```



## 8. MODIFY THE CODE

CityHelper.java:

[Click to download](#)

```
public List<City> getCities() {  
    List<City> citys = new ArrayList<>();  
    long cityId = 1;  
    City city = new City(cityId++, "Phoenix", 85001);  
    citys.add(city);  
    city = new City(cityId++, "Little Rock", 72201);  
    citys.add(city);  
    city = new City(cityId++, "Huntsville", 35801);  
    citys.add(city);  
    return citys;  
}
```

**JAVASERVER FACES COURSE**

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

## 9. MODIFY THE XHTML FILE

Modify the index.xhtml file to use the new classes. Search the following lines:

```
<h:inputText id="zipCode" immediate="true"  
             onchange="this.form.submit()" required="true"  
             value="#{candidate.zipCode}"  
             valueChangeListener="#{vacantForm.zipCodeListener}" />
```

Replace with the following code:

```
<h:inputText id="zipCode" immediate="true"  
             onchange="this.form.submit()" required="true"  
             value="#{candidate.zipCode}"  
             valueChangeListener="#{vacantForm.zipCodeListener}" >  
    <f:convertNumber integerOnly="true" pattern="00000"/>  
</h:inputText>
```

## 9. MODIFY THE JAVA CLASS

Also look for the following lines :

```
<h:selectOneMenu id="cityId" required="true" value="#{candidate.cityId}">
    <f:selectItem itemLabel="#{msgs['vacanteForm.select']}" itemValue="" />
    <f:selectItem itemLabel="Phoenix" itemValue="1" />
    <f:selectItem itemLabel="Little Rock" itemValue="2" />
    <f:selectItem itemLabel="Huntsville" itemValue="3" />
</h:selectOneMenu>
```

Replace with the following code :

```
<h:selectOneMenu id="cityId" required="true" value="#{candidate.cityId}">
    <f:selectItem itemLabel="#{msgs['vacanteForm.select']}" itemValue="" />
    <f:selectItems value="#{cityHelper.selectItems}" />
</h:selectOneMenu>
```



## 9. 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:f="http://xmlns.jcp.org/jsf/core">
  <h:head>
    <title>Job Agency V7</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>
      </table>
    </h:form>
  </h:body>
</html>
```

JAVASERVER FACES COURSE

www.globalmentoring.com.mx

## 9. MODIFY THE FILE

[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>
<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}" >
    <f:convertNumber integerOnly="true" pattern="00000"/>
  </h:inputText>
  </td>
  <td><h:message for="zipCode" /></td>
</tr>
```

## 9. MODIFY THE FILE

[index.xhtml:](#)

[Click to download](#)

```
<tr>
  <td><h:outputLabel for="cityId" value="#{msgs['vacantForm.cityId']}" /></td>
  <td>
    <h:selectOneMenu id="cityId" required="true" value="#{candidate.cityId}">
      <f:selectItem itemLabel="#{msgs['vacantForm.select']}" itemValue="" />
      <f:selectItems value="#{cityHelper.selectItems}" />
    </h:selectOneMenu>
  </td>
  <td><h:message for="cityId" /></td>
</tr>
</table>
<table>
  <tr>
    <td>
      <h:commandLink actionListener="#{vacantForm.hideComment}"
        immediate="true" rendered="#{!vacantForm.commentSent}"
        value="#{msgs['vacantForm.show']}" />
      <h:commandLink actionListener="#{vacantForm.hideComment}"
        immediate="true" rendered="#{vacantForm.commentSent}"
        value="#{msgs['vacantForm.hide']}" />
    </td>
  </tr>
</table>
```

**JAVASERVER FACES COURSE**

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

## 9. MODIFY THE FILE

[index.xhtml:](#)

[Click to download](#)

```
<h:panelGroup rendered="#{vacantForm.commentSent}">
  <table>
    <tr>
      <td><h:outputLabel for="comment" value="#{msgs['vacantForm.comment']}" /></td>
    </tr>
    <tr>
      <td><h:inputTextarea id="comment" cols="50" rows="10" value="#{candidate.comment}" /></td>
    </tr>
  </table>
</h:panelGroup>
<h:commandButton action="#{vacantForm.send}" value="#{msgs['vacantForm.send']}" />
</h:form>
</h:body>
</html>
```

**JAVASERVER FACES COURSE**

www.globalmentoring.com.mx



# 10. MODIFY THE JAVA CLASS

Modify the VacantForm.java class to inject the dependency of the CityHelper.java bean:

```
@Inject  
private CityHelper cityHelper;
```

You also include the respective import of the CityHelper.java class and add the getter / setter methods of the cityHelper attribute.



**JAVASERVER FACES COURSE**

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

# 10. MODIFY THE JAVA CLASS

In addition, in the VacanteForm.java class we replace the code of the zipCodeListener with the following:

```
public void zipCodeListener(ValueChangeEvent valueChangeEvent) {  
  
    FacesContext facesContext = FacesContext.getCurrentInstance();  
    UIViewRoot uiViewRoot = facesContext.getViewRoot();  
    Long newZipCode = (Long) valueChangeEvent.getNewValue();  
    log.info("New zipCode: " + newZipCode);  
    UIInput cityInputText = (UIInput) uiViewRoot.findComponent("vacantForm:cityId");  
    // We seek the city by id with the help of the cityHelper bean  
    Long cityId = this.cityHelper.getCityIdByZipCode(newZipCode);  
    cityInputText.setValue(cityId);  
    cityInputText.setSubmittedValue(cityId);  
    facesContext.renderResponse();  
}
```

# 10. MODIFY THE CODE

## VacantForm.java:

[Click to download](#)

```
package beans.backing;

import beans.helper.CityHelper;
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.*;
import org.apache.logging.log4j.*;

@Named
@RequestScoped
public class VacantForm {

    Logger log = LogManager.getRootLogger();

    @Inject
    private Candidate candidate;

    @Inject
    private CityHelper cityHelper;

    private boolean commentSent = false;

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



# 10. MODIFY THE CODE

VacantForm.java:

[Click to download](#)

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

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
    }
}
```

**JAVASERVER FACES COURSE**

www.globalmentoring.com.mx

# 10. MODIFY THE CODE

## VacantForm.java:

[Click to download](#)

```
public void zipCodeListener(ValueChangeEvent valueChangeEvent) {
    FacesContext facesContext = FacesContext.getCurrentInstance();
    UIViewRoot uiViewRoot = facesContext.getViewRoot();
    Long newZipCode = (Long) valueChangeEvent.getNewValue();
    log.info("New zipCode: " + newZipCode);
    UIInput cityInputText = (UIInput) uiViewRoot.findComponent("vacantForm:cityId");
    // We seek the cityId with the help of the cityHelper bean
    Long cityId = this.cityHelper.getCityIdByZipCode(newZipCode);
    cityInputText.setValue(cityId);
    cityInputText.setSubmittedValue(cityId);
    facesContext.renderResponse();
}

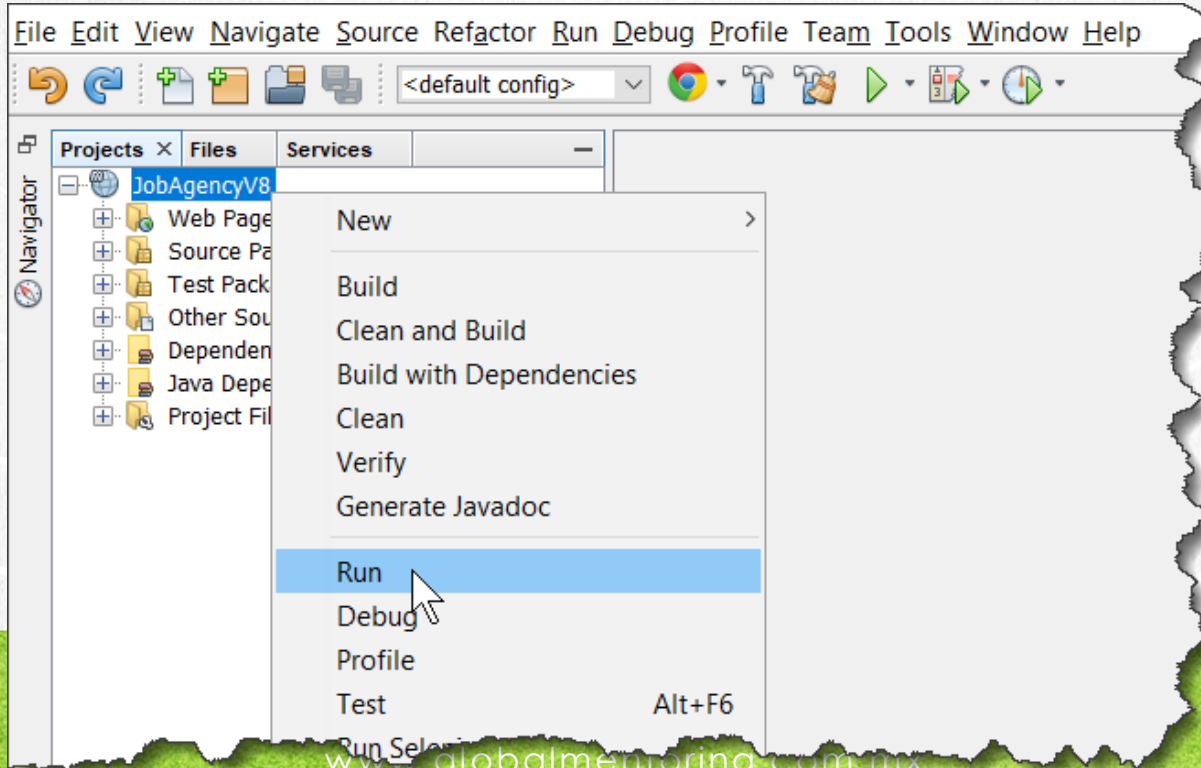
public void hideComment(ActionEvent actionEvent) {
    this.setCommentSent(!this.commentSent);
    log.info("Executing actionEvent hideComment");
}

public boolean isCommentSent() {
    return commentSent;
}

public void setCommentSent(boolean commentSent) {
    this.commentSent = commentSent;
}
}
```

# 11. EXECUTE THE PROJECT

We execute the project. Before running the project it is recommended to stop the Java server, do a clean & build of the project and run the application as follows:





# 11. EXECUTE THE PROJECT

We execute the project. The result is as follows:

Job Agency V7

localhost:8080/JobAgencyV8/faces/index.xhtml

Name

Last Name

Desired Salary

Birthdate

Zip Code

City

[Show](#)

Possible values:  
85001, 72201  
and 35801

# EXERCISE CONCLUSION

With this exercise we have laid the foundations of our project to recover the list of cities dynamically, which is already in a separate class, and the next step will be to obtain the list of a database.

We continue using the component SelectItem, and with the help of this component we now recover the collection of data from Java code, and no longer from the index.xhtml page.



**JAVASERVER FACES COURSE**

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

**ONLINE COURSE**

# **JAVASERVER FACES (JSF)**

---

By: Eng. Ubaldo Acosta



**JAVASERVER FACES COURSE**

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