

JAVASERVER FACES COURSE

EXERCISE

MANAGED BEANS IN JSF

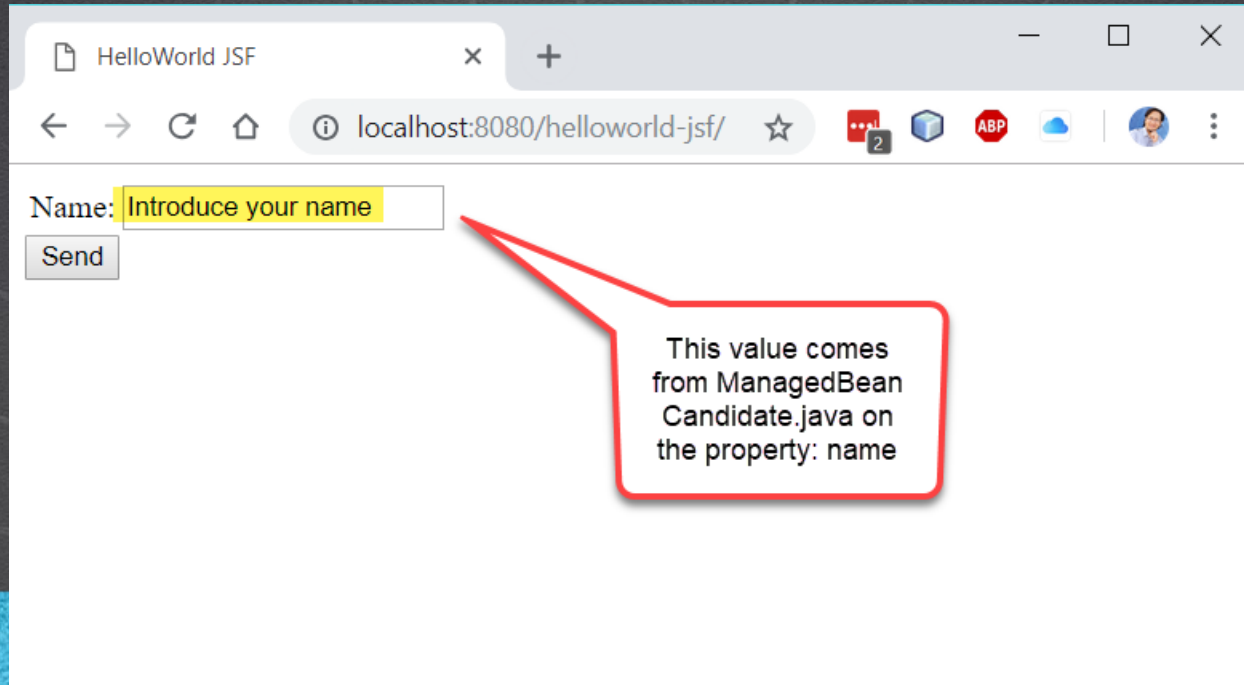


JAVASERVER FACES COURSE

www.globalmentoring.com.mx

EXERCISE OBJECTIVE

Modify the project of helloworld-jsf to apply the ManagedBean concept. The final result is as follows:



OBJECTIVE OF THE EXERCISE

We will introduce the concept of Model Managed Bean to our application.

We will use the concept of JSF annotations to declare the Managed Bean.

We will link the Model Bean to our JSF page to display the initial value of the "name" property in the view.

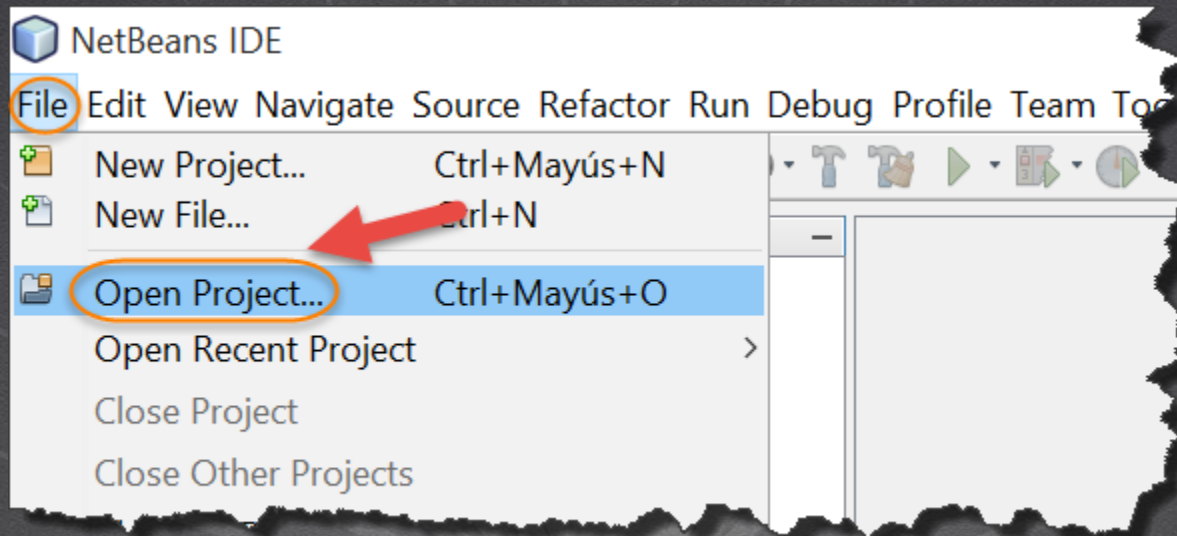


JAVASERVER FACES COURSE

www.globalmentoring.com.mx

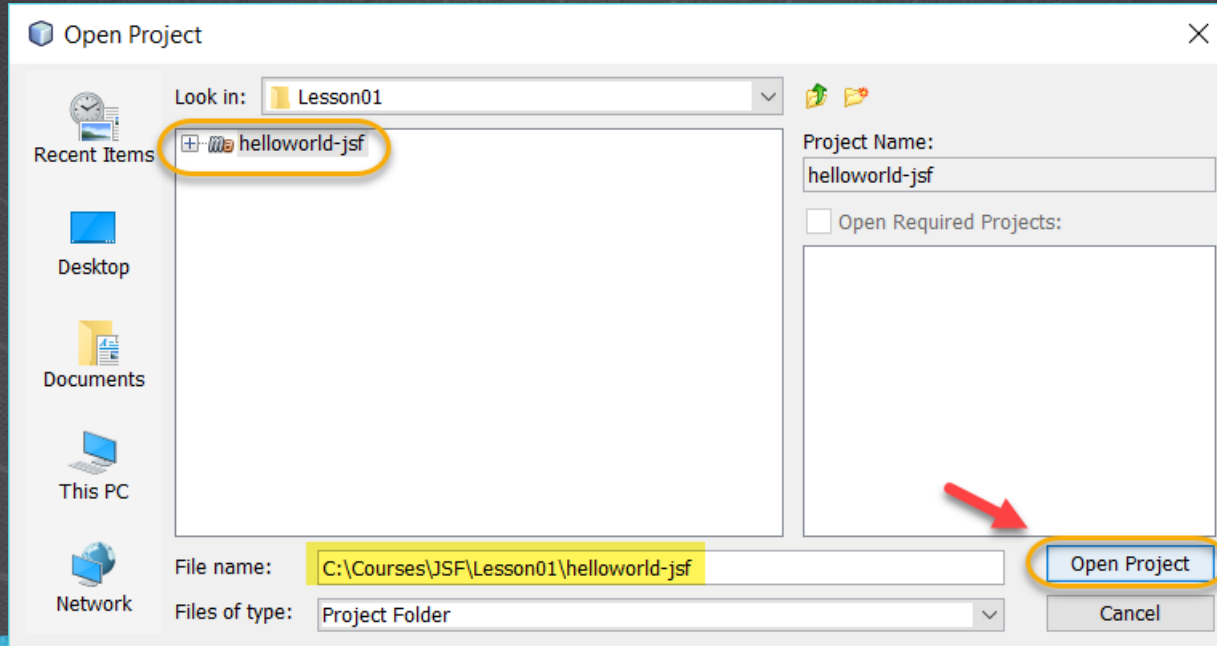
1. OPEN THE PROJECT

We open the project helloworld-jsf if it is not open:



1. OPEN THE PROJECT

We open the project helloworld-jsf if it is not open:

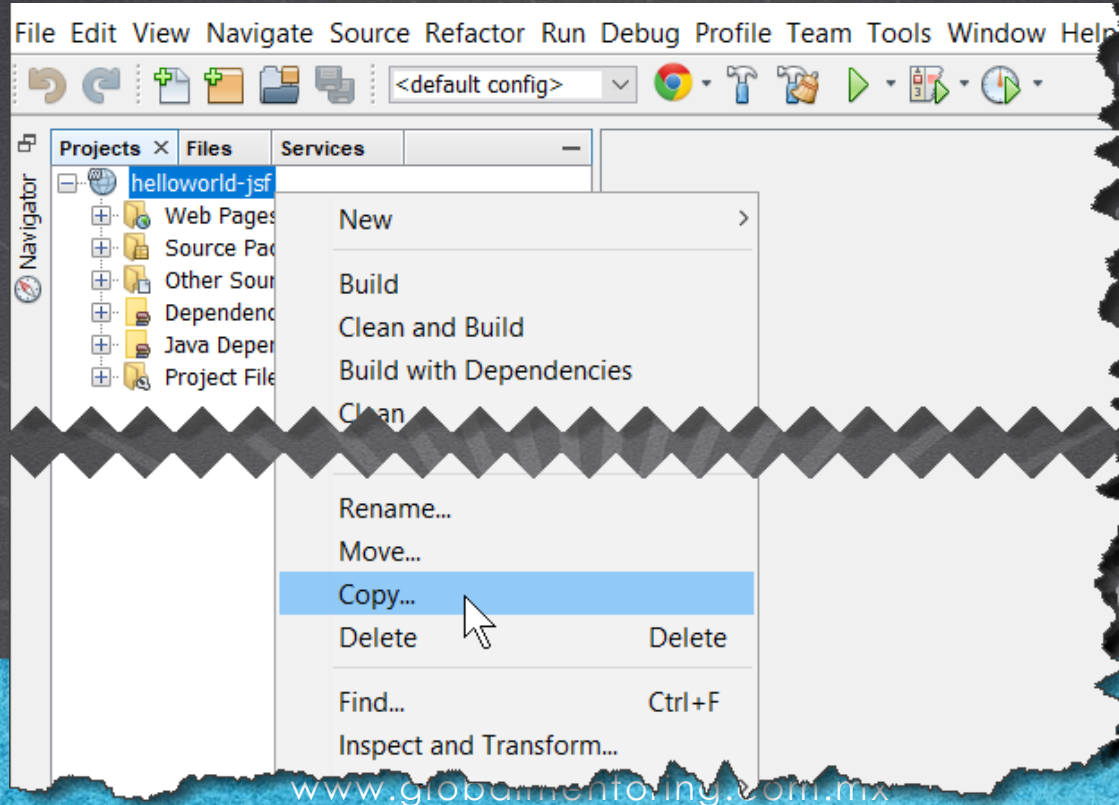


JAVASERVER FACES COURSE

www.globalmentoring.com.mx

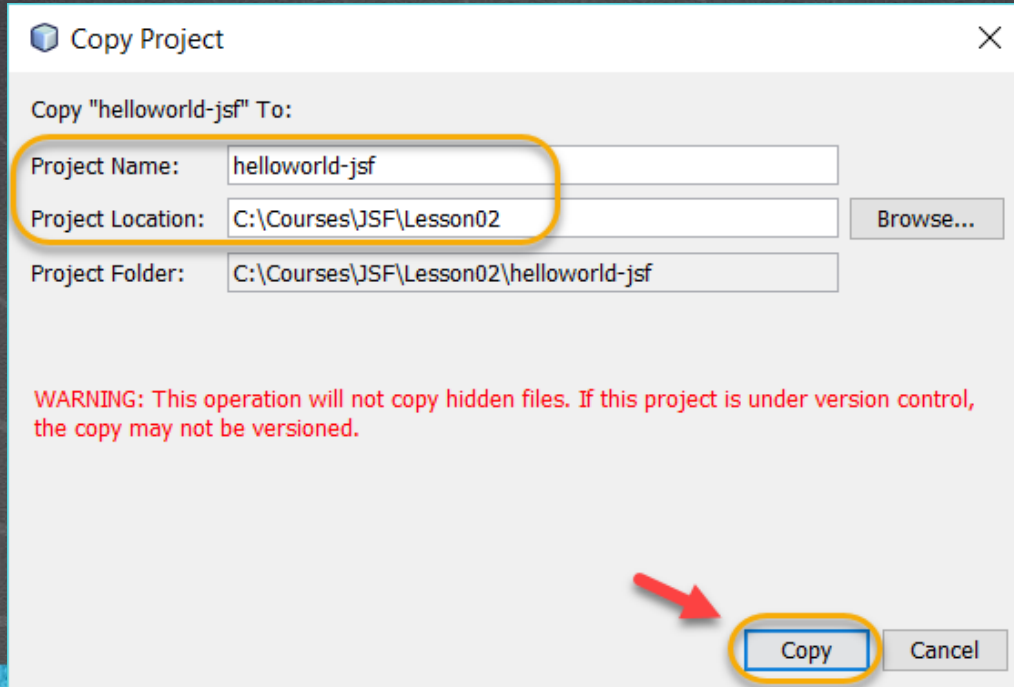
2. COPY THE PROJECT

We copy the project to deposit it in the Lesson2 folder:



2. COPY THE PROJECT

We copy the project to deposit it in the Lesson02 folder:

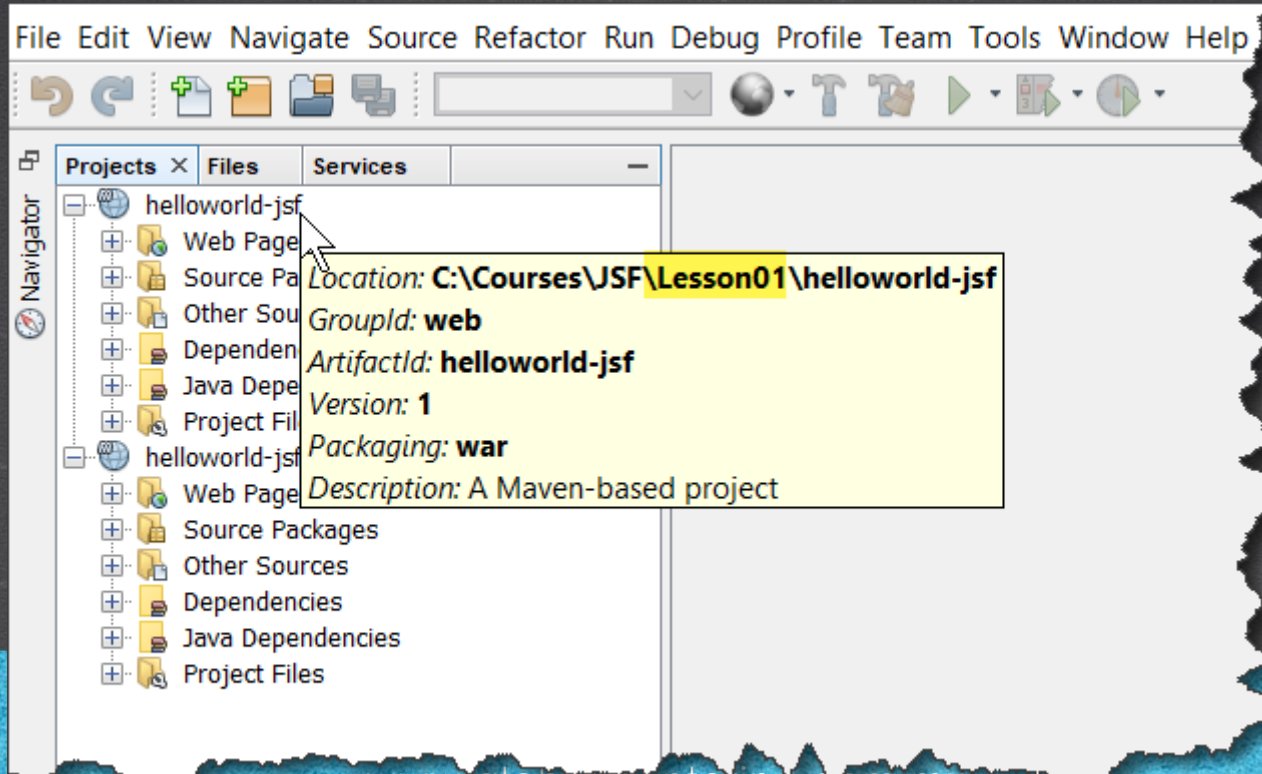


JAVASERVER FACES COURSE

www.globalmentoring.com.mx

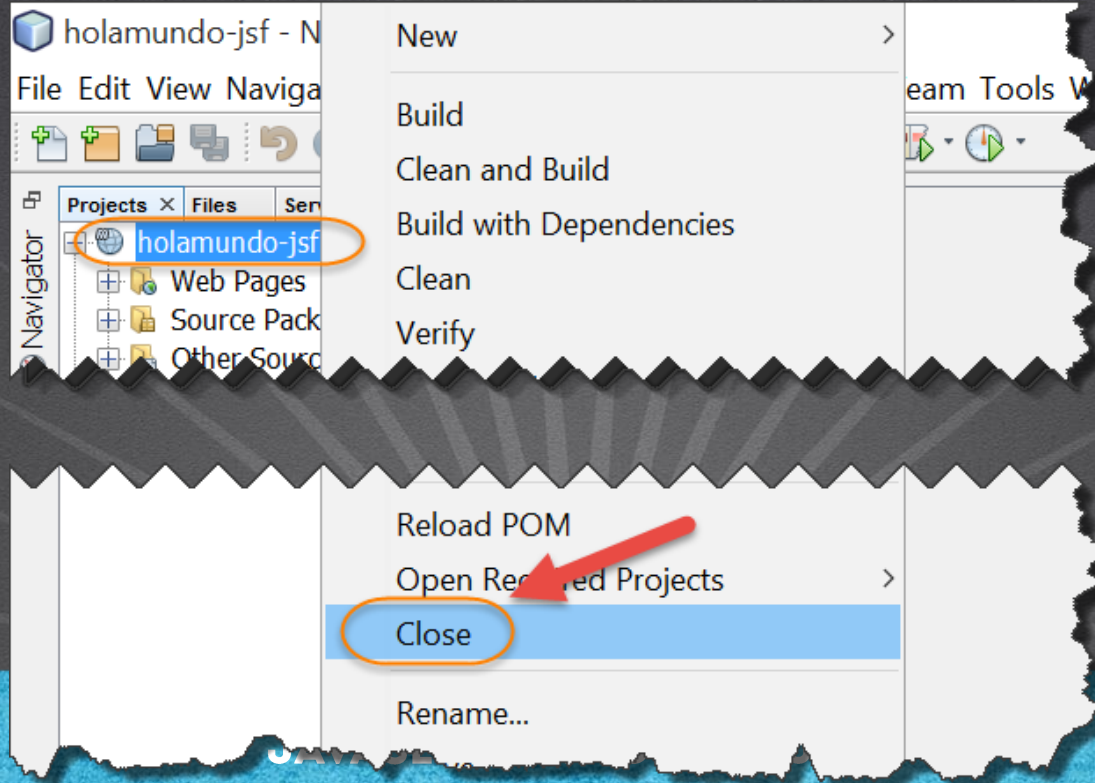
3. CLOSE THE PROJECT

We closed the previous project (The one in Lesson01):



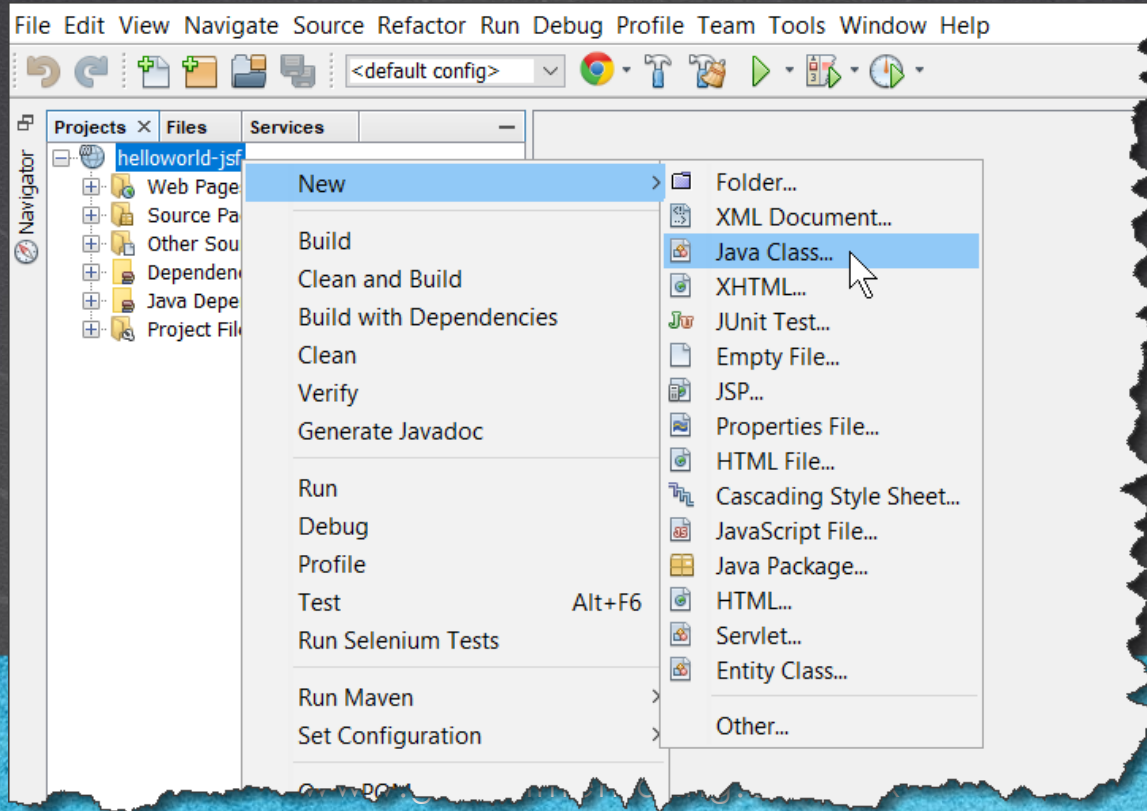
3. CLOSE THE PROJECT

We closed the previous project (The one in Lesson01):



4. CREATE A NEW JAVA CLASS

We create a Java class that will be a ManagedBean of JSF, called Candidate.java:



4. CREATE A NEW CLASS

We create a Java class that will be a ManagedBean of JSF, called Candidate.java:

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

5. MODIFY THE CODE

[Candidate.java:](#)

Click to download

```
package beans.model;

import javax.inject.Named;
import javax.enterprise.context.RequestScoped;

@Named
@RequestScoped
public class Candidate {

    private String name = "Introduce your name";

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

}
```

JAVASERVER FACES COURSE

www.globalmentoring.com.mx

6. RELATE THE BEAN TO THE PAGE XHTML

Locate the content of the index.xhtml page:

```
<h:inputText id="name" />
```

- Replace it with the following code:

```
<h:inputText id="name" value="#{candidate.name}" />
```

6. MODIFY THE CODE

[index.xhtml:](#)

Click to download

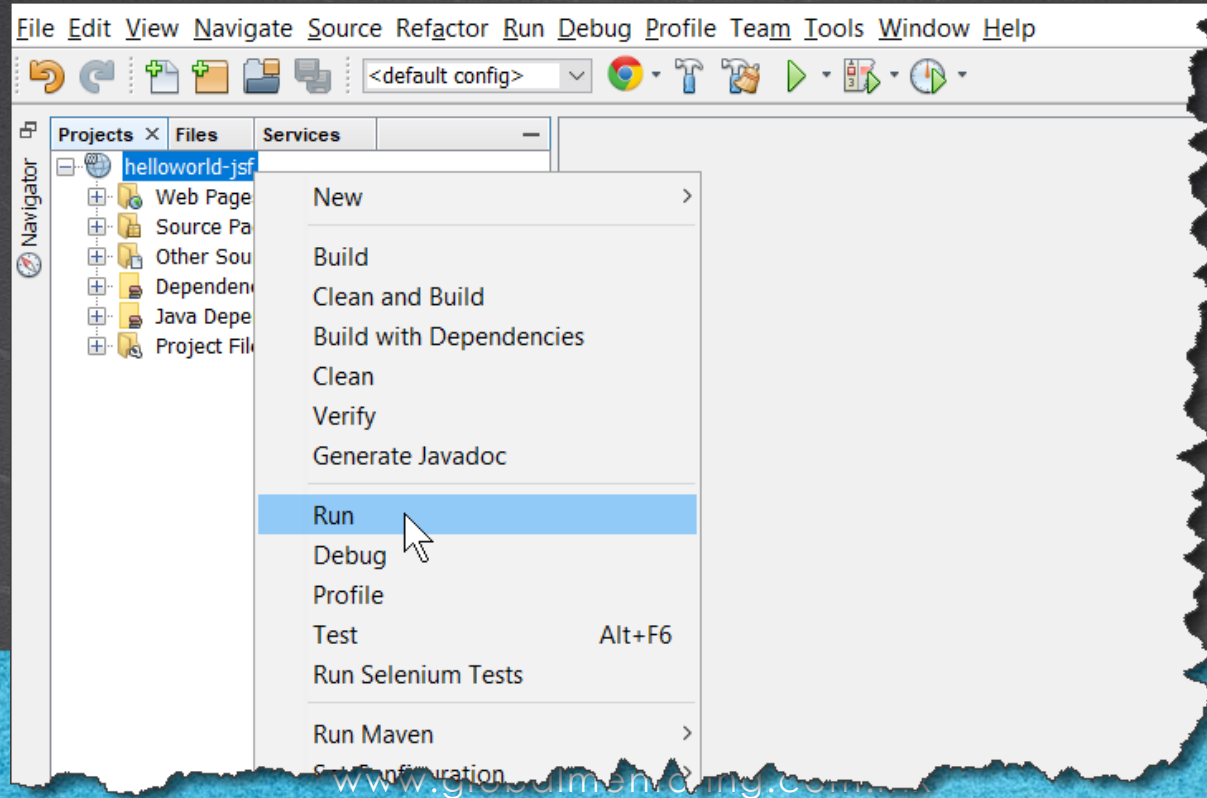
```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html">
  <h:head>
    <title>HelloWorld JSF</title>
  </h:head>
  <h:body>
    <h:form>
      <table>
        <tr>
          <td><h:outputLabel for="name" value="Name:" /></td>
          <td><h:inputText id="name" value="#{candidate.name}" /></td>
          <td><h:message for="name" /></td>
        </tr>
      </table>
      <h:commandButton value="Send" />
    </h:form>
  </h:body>
</html>
```

JAVASERVER FACES COURSE

www.globalmentoring.com.mx

7. EXECUTE THE PROJECT

Execute the Project:



7. EXECUTE THE PROJECT

We execute the project. The result is as follows:



EXERCISE CONCLUSION

- With this exercise we have added the JSF ManagedBean concept.
- We create the class Candidate.java, which has a property called name, and we use this property from the index.xhtml page.
- We must remember that the Candidate class to be used must be placed in some context, be it Application, Session or Request or another of JSF, and this is how the JSF page index.xhtml can access this object and its properties.

ONLINE COURSE

JAVASERVER FACES (JSF)

By: Ing. Ubaldo Acosta



JAVASERVER FACES COURSE

www.globalmentoring.com.mx