

# HIBERNATE & JPA COURSE

## EXERCISE

# WEB STUDENTS PERSISTENCE LAB



**HIBERNATE & JPA COURSE**

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

# EXERCISE OBJECTIVE

Create a Web project application to put the life cycle of entity objects with Hibernate and JPA. At the end we should observe the following :

The screenshot shows a web browser window with the title 'List of Students'. The address bar displays 'localhost:8080/WebStudentsPersistence/ServletController'. The page content includes a title 'List of Students', a link 'Add', and a table with the following data:

Student ID	Name	Street name	Street number	Country
<a href="#">1</a>	Jhon Smith	new street	555	Mexico
<a href="#">2</a>	Charly	Merside	419	England

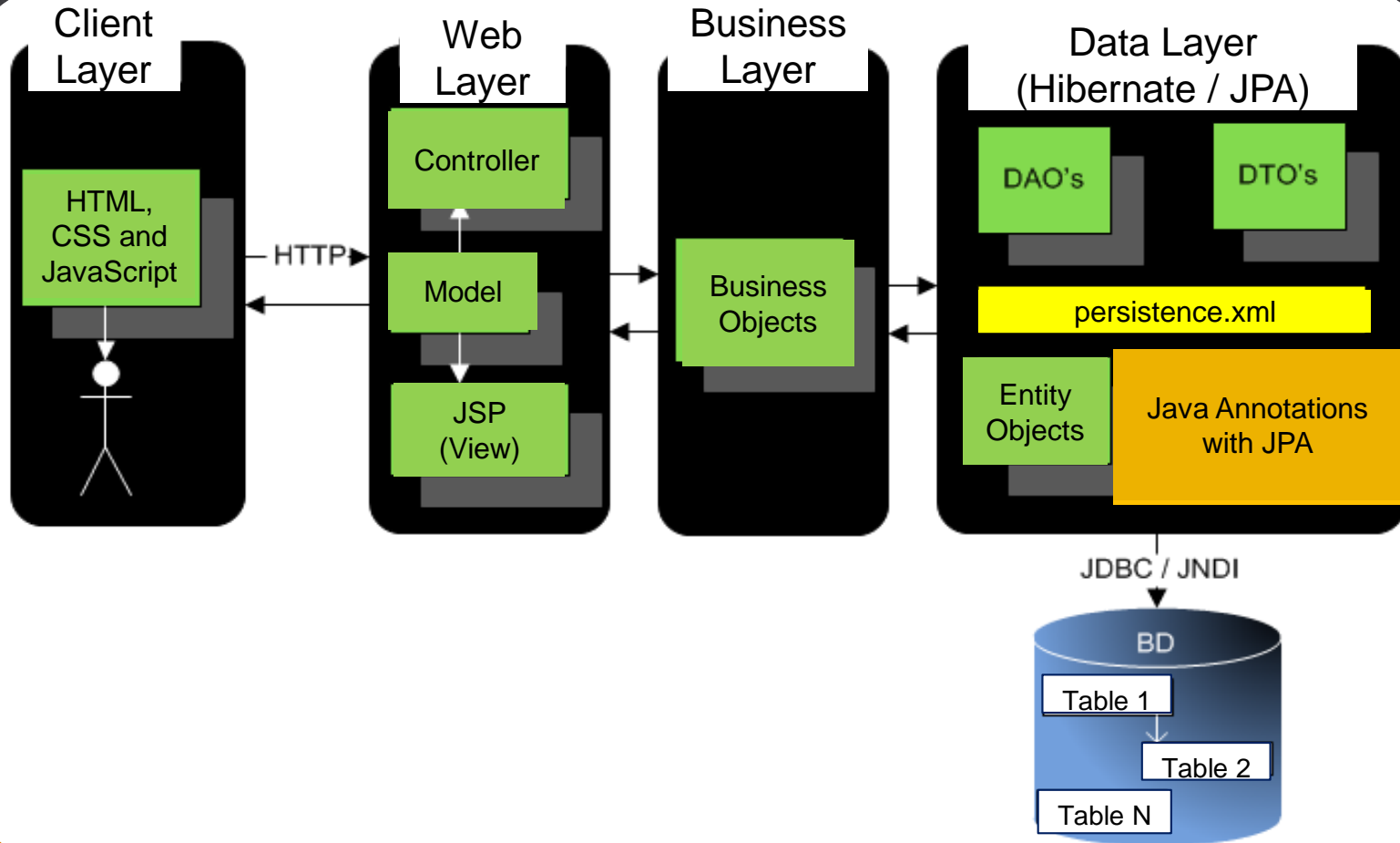


# CONTENT OF THE LAB

Create a Web application that contains the following :

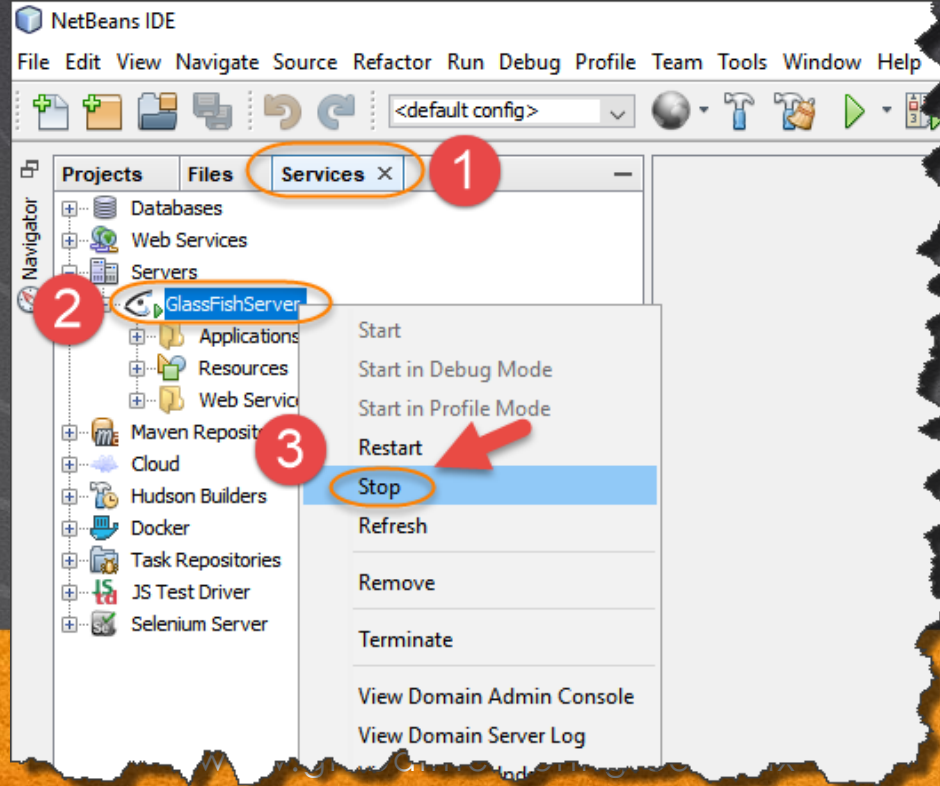
- ✓ List the contents of the Students table.
- ✓ Create the options to add, edit and delete a new student.
- ✓ In the add and edit options, a screen should be created that allows capturing the Student's values along with their address information, such as street Name, street Number and Country, using the concept of Cascade persistence.
- ✓ Use the CascadePersistence project previously created to use the Entity objects and DAO objects already created in this project.

# WEB PROJECT ARCHITECTURE



# STOP GLASSFISH IF IT'S RUNNING

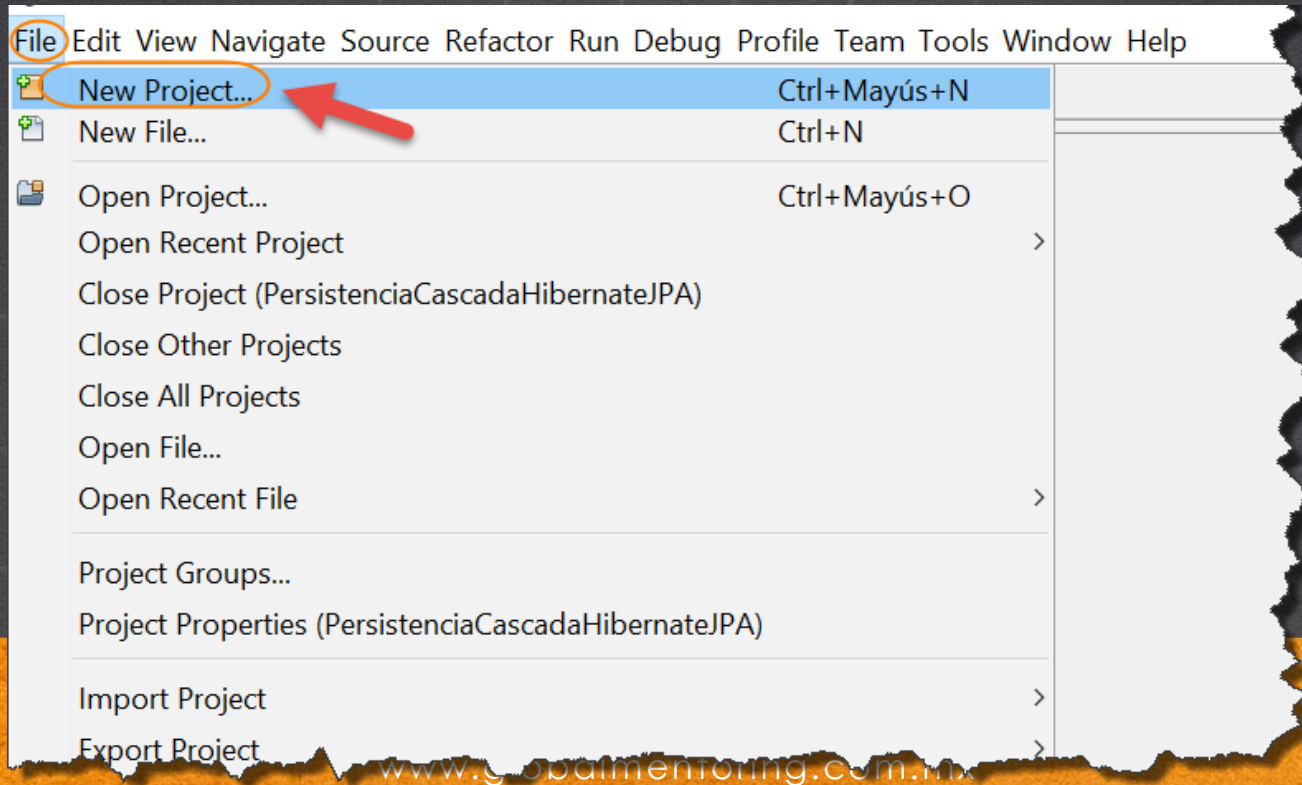
We stopped Glassfish if it was running :





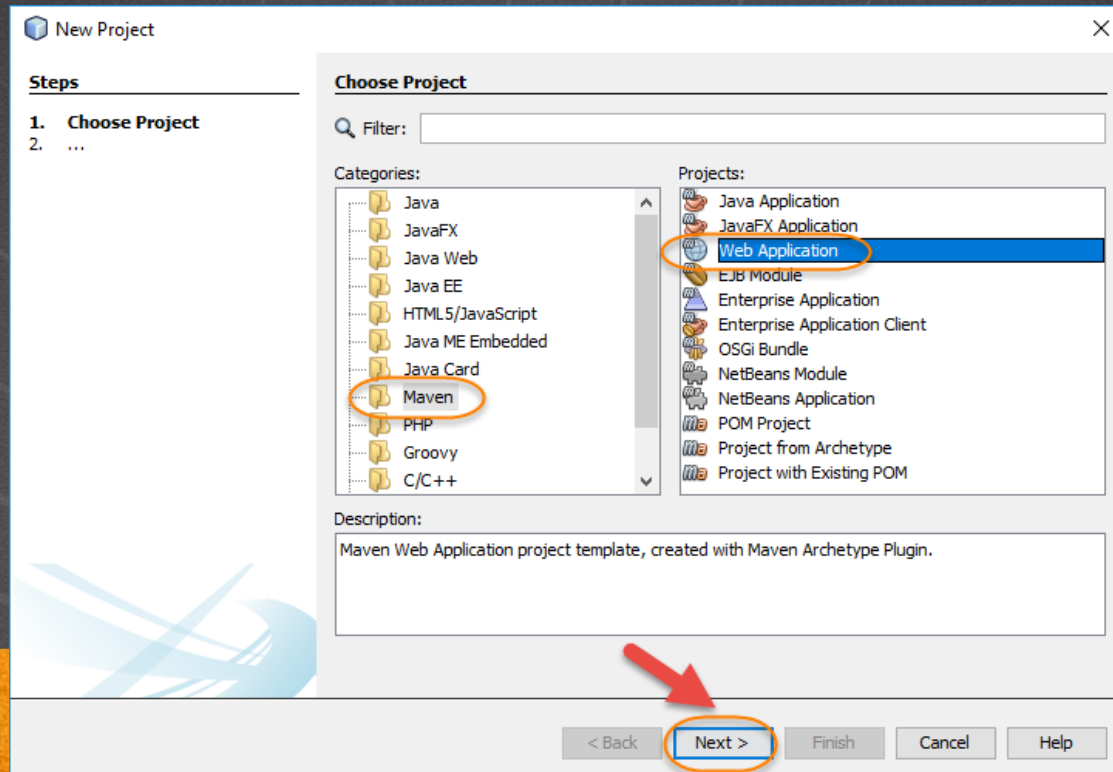
# 1. CREATE A NEW PROJECT

Create a new Web project :



# 1. CREATE A NEW PROJECT

Create a new Web project:



# 1. CREATE A NEW PROJECT

Create a new Web project :

**New Web Application**

**Steps**

1. Choose Project
- 2. Name and Location**
3. Settings

**Name and Location**

Project Name: WebStudentsPersistence

Project Location: C:\Courses\Hibernate\Lesson07 Browse...

Project Folder: \Courses\Hibernate\Lesson07\WebStudentsPersistence

Artifact Id: WebStudentsPersistence

Group Id: web

Version: 1

Package: (Optional)

< Back **Next >** Finish Cancel Help

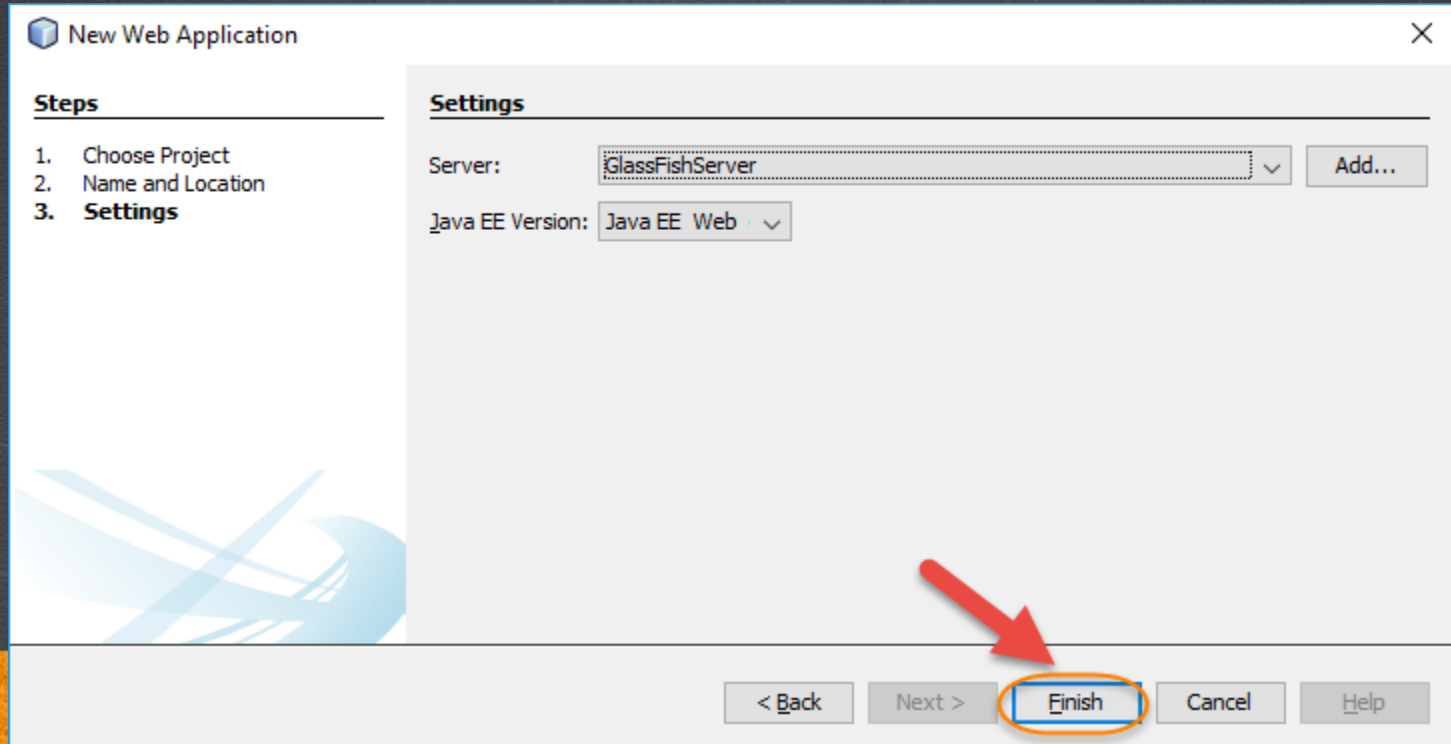
**HIBERNATE & JPA COURSE**

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



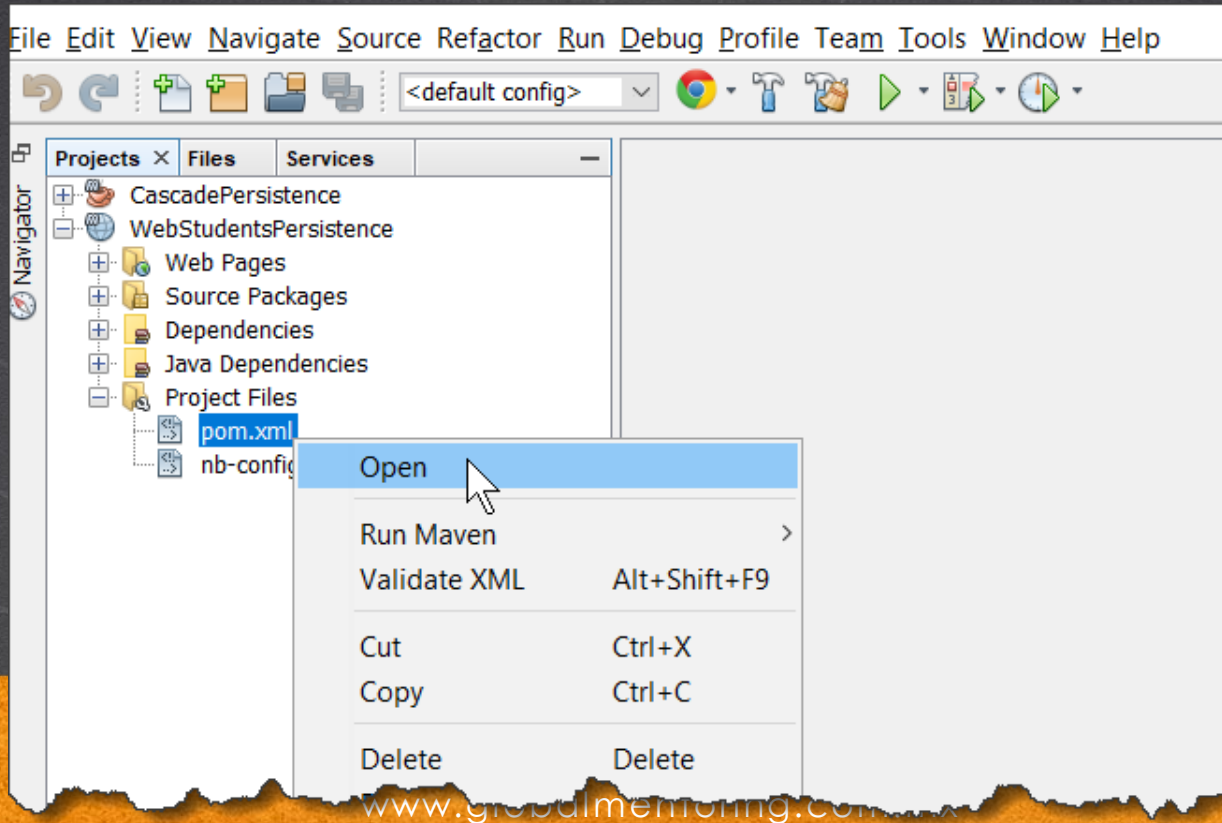
# 1. CREATE A NEW PROJECT

Create a new Web project :



## 2. OPEN MAVEN'S POM.XML FILE

- The maven pom.xml file manages the Java libraries we will use:



# 3. MODIFY THE FILE

[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>web</groupId>
  <artifactId>WebStudentsPersistence</artifactId>
  <version>1</version>
  <packaging>war</packaging>

  <name>WebStudentsPersistence</name>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>
  <dependencies>
    <dependency>
      <groupId>javax</groupId>
      <artifactId>javaee-web-api</artifactId>
      <version>8.0</version>
      <scope>provided</scope>
    </dependency>
    <dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-core</artifactId>
      <version>5.3.6.Final</version>
    </dependency>
```



### 3. MODIFY THE FILE

[pom.xml:](#)

Click to download

```
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>5.1.46</version>
</dependency>
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-api</artifactId>
  <version>2.11.1</version>
</dependency>
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-core</artifactId>
  <version>2.11.1</version>
</dependency>
</dependencies>
```

**CURSO DE JAVA CON JDBC**

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

# 3. MODIFY THE FILE

[pom.xml:](#)

[Click to download](#)

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-war-plugin</artifactId>
      <version>2.3</version>
      <configuration>
        <failOnMissingWebXml>false</failOnMissingWebXml>
      </configuration>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>3.7.0</version>
      <configuration>
        <source>1.8</source>
        <target>1.8</target>
      </configuration>
    </plugin>
  </plugins>
</build>

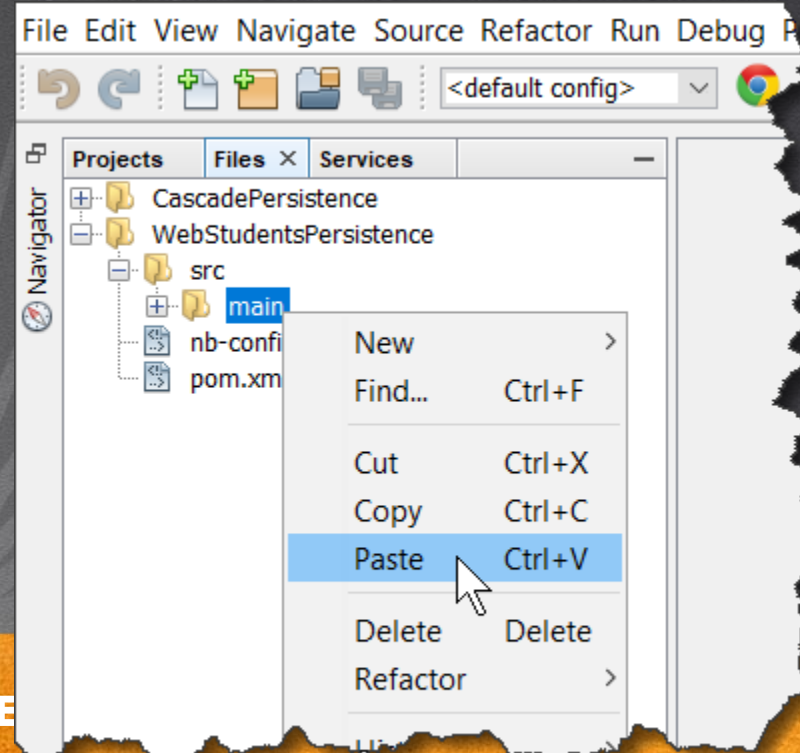
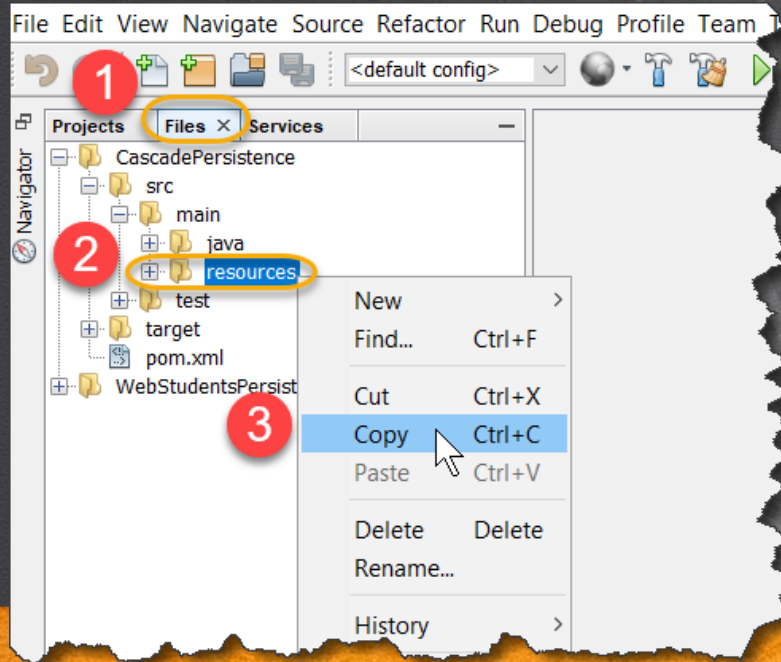
</project>
```

**CURSO DE JAVA CON JDBC**

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

## 4. COPY CLASSES AND FILES

Copy the file log4j2.xml and META-INF\persistence.xml



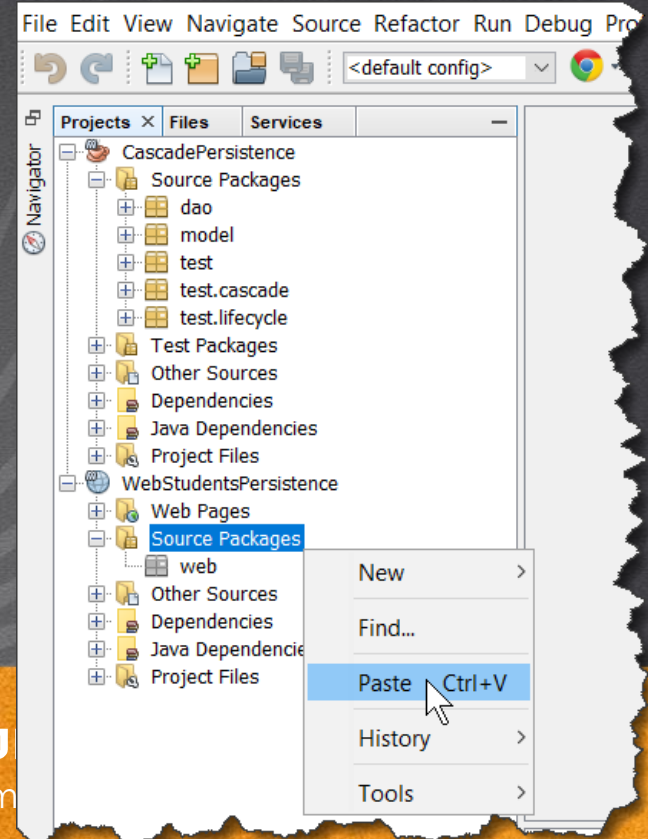
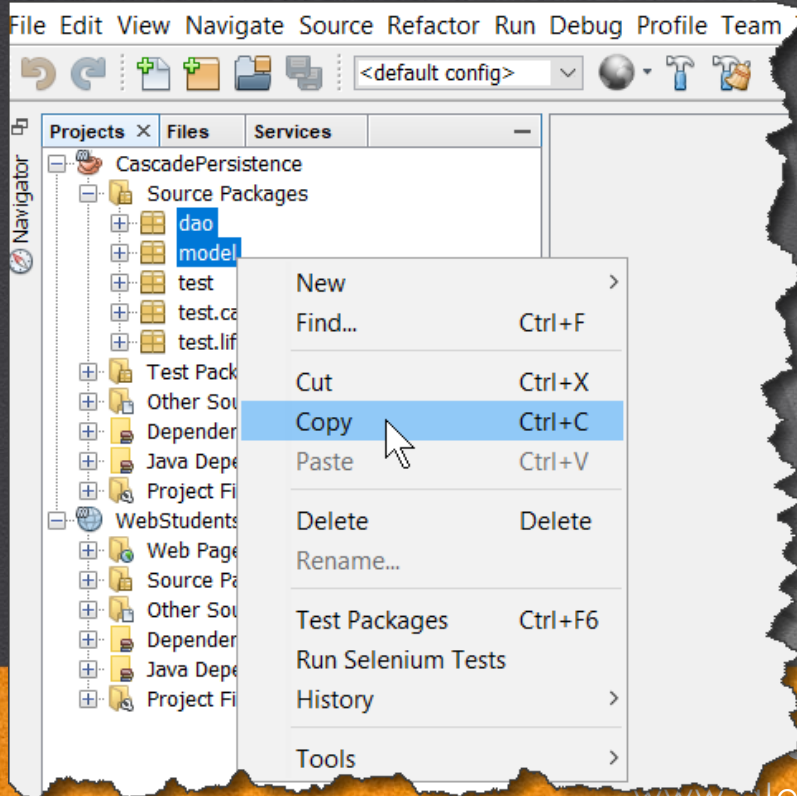
**CURSO HIBERNATE**

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



# 5. COPY THE MODEL CLASSES

We copy the classes of the model and dao package :

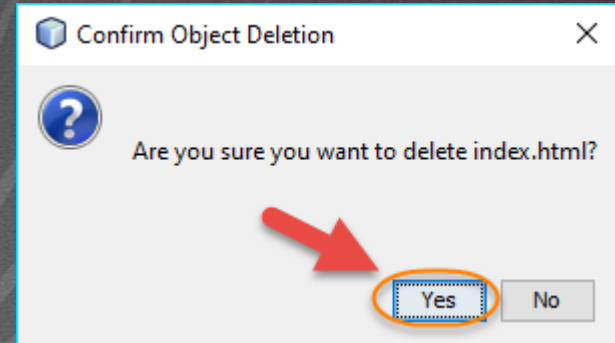
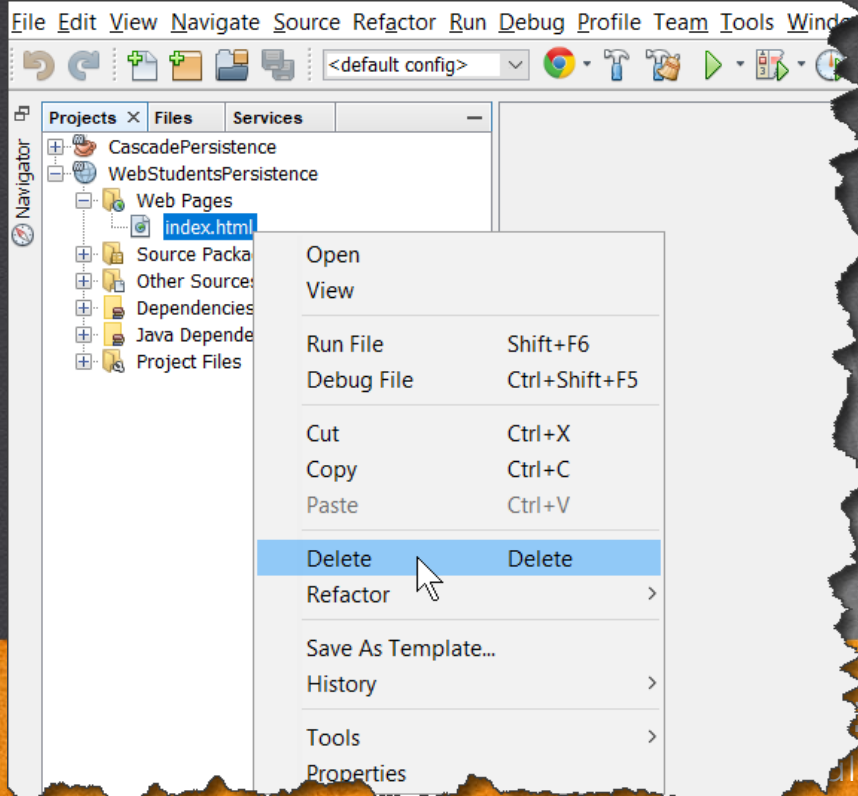


HIBERNATE Y J

www.globalmentoring.com

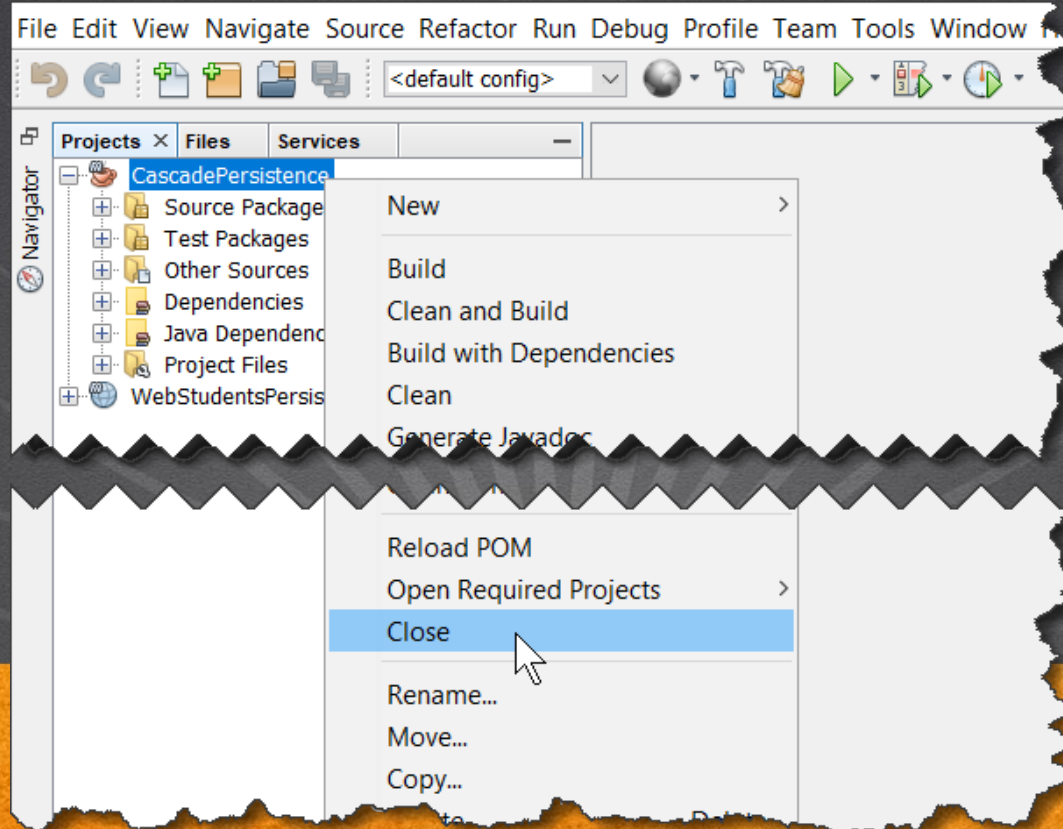
## 6. DELETE THE INDEX.HTML FILE

Delete the index.html file if it exists:



# 7. CLOSE THE PROJECT THAT WE DO NOT USE

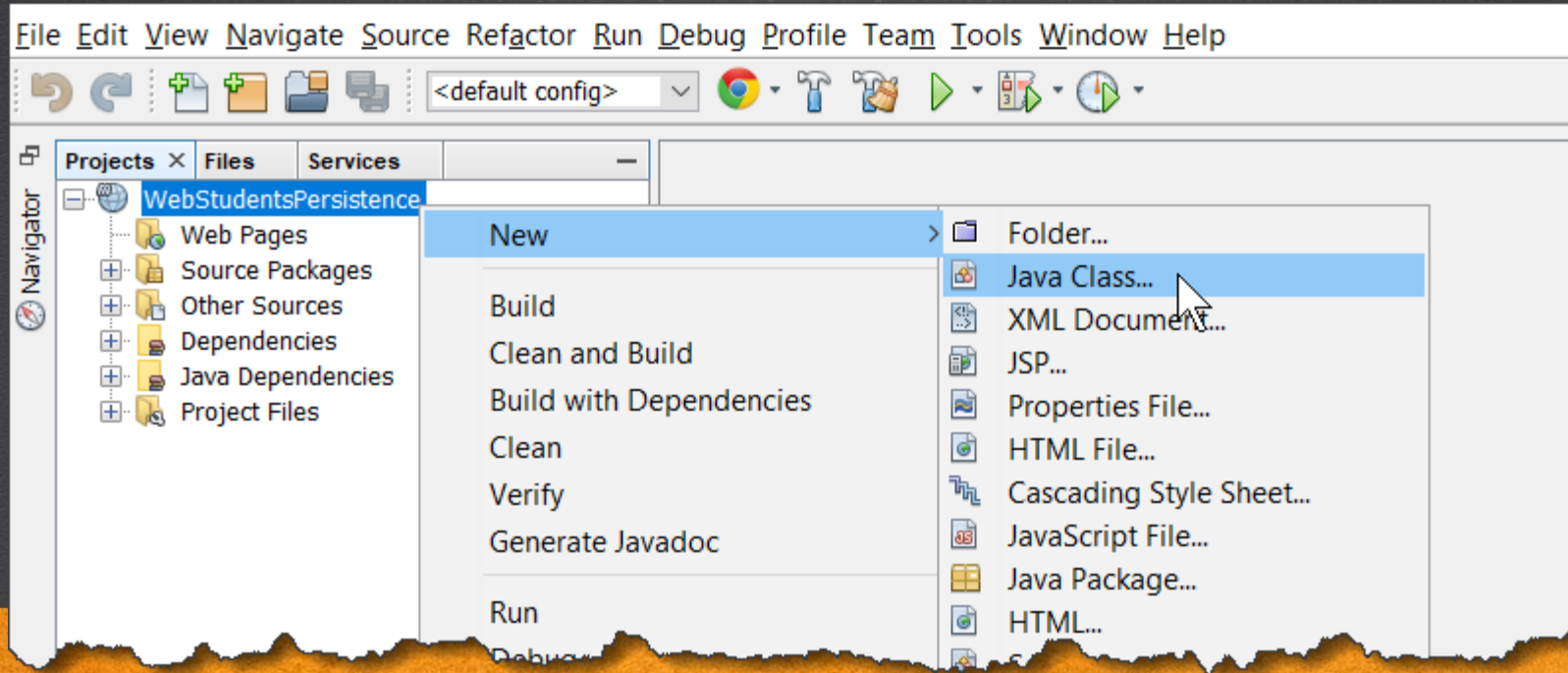
We close the project that we no longer use:





## 8. CREATE A NEW CLASS

We create the class StudentService.java



## 8. CREATE A NEW CLASS

We create the class StudentService.java

New Java Class

**Steps**

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

**Name and Location**

Class Name: StudentService

Project: WebStudentsPersistence

Location: Source Packages

Package: service

Created File: C:\Courses\Hibernate\Lesson07\WebStudentsPersistence\src\main\java\service\StudentService.java

< Back Next > **Finish** Cancel Help

**HIBERNATE & JPA COURSE**

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

## 9. MODIFY THE CODE

### StudentService.java:

[Click to download](#)

```
package service;

import dao.StudentDAO;
import model.Student;
import java.util.List;

public class StudentService {

    StudentDAO studentDao = new StudentDAO();

    public List<Student> listStudents() {
        return studentDao.list();
    }
}
```

**CURSO HIBERNATE Y JPA**

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



## 9. MODIFY THE CODE

StudentService.java:

Click to download

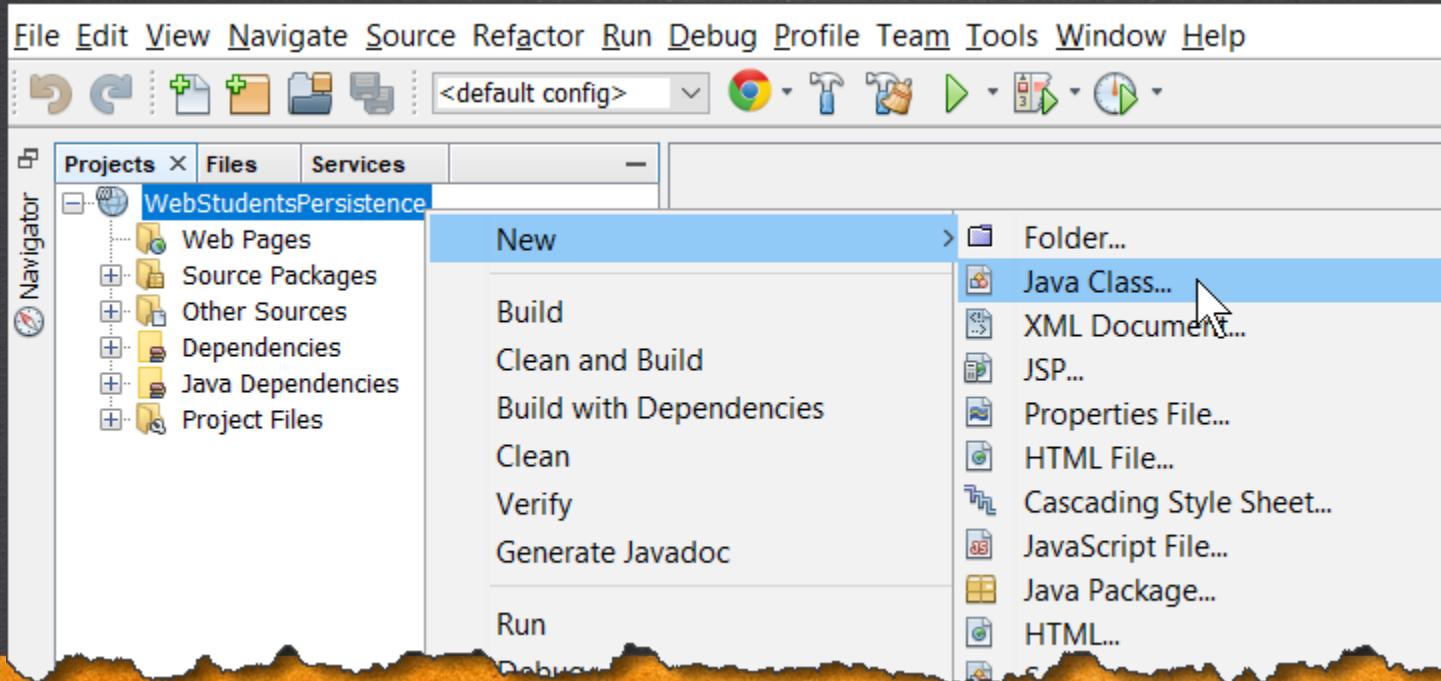
```
public boolean saveStudent(Student student) {
    if(student != null && student.getIdStudent() == null){
        studentDao.insert(student);
    }
    else{
        studentDao.update(student);
    }
    return true;// If nothing fails, we return true
}

public boolean deleteStudent(Integer idStudent) {
    studentDao.delete(new Student(idStudent));
    return true;// If nothing fails, we return true
}

public Student findStudent(Integer idStudent) {
    return studentDao.findById(new Student(idStudent));
}
}
```

# 10. CREATE A NEW CLASS

Create the ServletController.java class:



**CURSO HIBERNATE Y JPA**

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

# 10. CREATE A NEW CLASS

Create the ServletController.java class:

**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



# 11. MODIFY THE CODE

## ServletController.java:

Click to download

```
package web;

import java.io.IOException;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
import model.Student;
import service.StudentService;

@WebServlet("/ServletController")
public class ServletController extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        //1. We communicate with the service layer
        StudentService studentService = new StudentService();

        //2. We recover all the students
        List<Student> students = studentService.listStudents();

        //3. We share the information (Model) with the view
        request.setAttribute("students", students);

        //4. We select the view to show the information of students
        request.getRequestDispatcher("/WEB-INF/listStudents.jsp").forward(request, response);
    }
}
```

# 11. MODIFY THE CODE

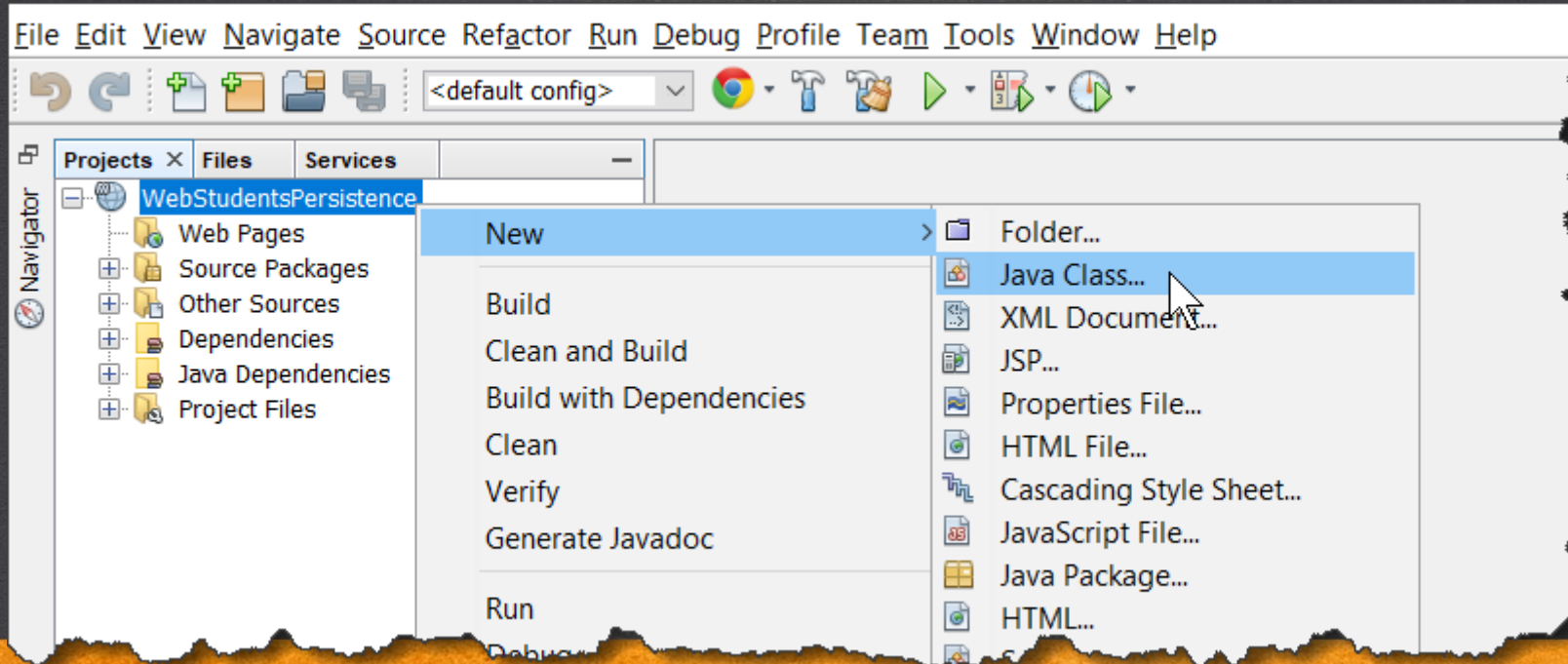
[ServletController.java:](#)

Click to download

```
@Override
public void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    this.doGet(request, response);
}
}
```

# 12. CREATE A NEW CLASS

Create a new ServletAdd.java class:



**CURSO HIBERNATE Y JPA**

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



## 12. CREATE A NEW CLASS

Create a new ServletAdd.java class:

**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

**CURSO HIBERNATE Y JPA**

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

# 13. MODIFY THE CODE

## ServletAdd.java:

[Click to download](#)

```
package web;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
import model.*;
import service.StudentService;

@WebServlet("/ServletAdd")
public class ServletAdd extends HttpServlet {

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        //1. We retrieve the parameters of the form
        String name = request.getParameter("name");
        String streetName = request.getParameter("streetName");
        String streetNumber = request.getParameter("streetNumber");
        String country = request.getParameter("country");
```

**CURSO HIBERNATE Y JPA**

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

## 13. MODIFY THE CODE

ServletAdd.java:

[Click to download](#)

```
//2. We fill the Student's DTO object to insert
//We create the Address object
Address address = new Address();
address.setStreetName(streetName);
address.setStreetNumber(streetNumber);
address.setCountry(country);

Student student = new Student();
student.setName(name);
//We inject the dependency of the Address object
student.setAddress(address);

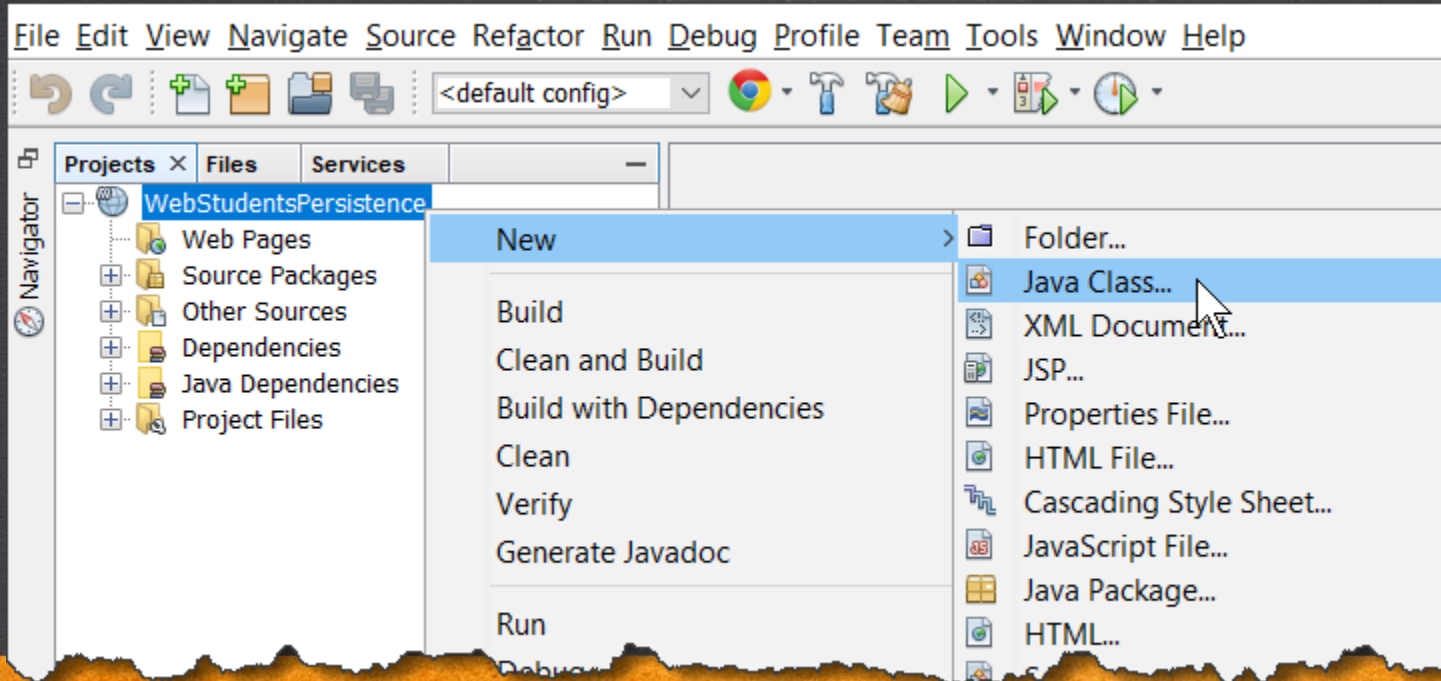
//3. We rely on the service layer
StudentService studentService = new StudentService();
studentService.saveStudent(student);

//4. Redirect to the home page
request.getRequestDispatcher("/index.jsp").forward(request, response);
}
```



# 14. CREATE A NEW CLASS

Create the ServletModify.java class:



**CURSO HIBERNATE Y JPA**

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

# 14. CREATE A NEW CLASS

Create the ServletModify.java class:

**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

**CURSO HIBERNATE Y JPA**

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

# 15. MODIFY THE CODE

## ServletModify.java:

[Click to download](#)

```
package web;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
import model.Student;
import service.StudentService;

@WebServlet("/ServletModify")
public class ServletModify extends HttpServlet {

    @Override
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        // This get method is going to take charge of recovering
        // the object to modify to show it in the form
        //1. We retrieve the idStudent of the request
        String idStudent = request.getParameter("idStudent");

        //2. We rely on the service class to recover the object to be modified
        StudentService studentService = new StudentService();
        Student student = studentService.findStudent(Integer.parseInt(idStudent));
```



# 15. MODIFY THE CODE

## ServletModify.java:

Click to download

```
//3. We share the found object with the view
//Let's share it in the session
HttpSession session = request.getSession();
session.setAttribute("student", student);

//4. Redirect in view of modifying
request.getRequestDispatcher("/WEB-INF/modifyStudent.jsp").forward(request, response);
}

@Override
public void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    //This post method will be responsible for modifying
    //the values that are sent from the form
    //We check if the button to modify or delete is pressed
    String action = request.getParameter("modify");
    if ("modify".equals(action)) {

        //1. We retrieve the parameters
        String name = request.getParameter("name");
        String streetName = request.getParameter("streetName");
        String streetNumber = request.getParameter("streetNumber");
        String country = request.getParameter("country");
```

# 15. MODIFY THE CODE

## ServletModify.java:

[Click to download](#)

```
//2. We recover the object of the session
HttpSession session = request.getSession();
Student student = (Student) session.getAttribute("student");

//3. We update the values of the modification of the student
student.setName(name);
//We update the values of the embedded object of Address
student.getAddress().setStreetName(streetName);
student.getAddress().setStreetNumber(streetNumber);
student.getAddress().setCountry(country);

//4. We communicate with the service layer to save the changes
StudentService studentService = new StudentService();
studentService.saveStudent(student);

//5. We eliminated the modified object from the session
//since we only used it while modifying the student
session.removeAttribute("student");

} //else if("delete".equals( accion )){
```

**CURSO HIBERNATE Y JPA**

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

# 15. MODIFY THE CODE

ServletModify.java:

Click to download

```
else { //delete was selected
    //We eliminate the student object

    //1. We retrieve the id of the student to be eliminated
    String idStudent = request.getParameter("idStudent");

    //2. We rely on the service layer to eliminate the Student object
    StudentService studentService = new StudentService();
    studentService.deleteStudent(Integer.parseInt(idStudent));
}

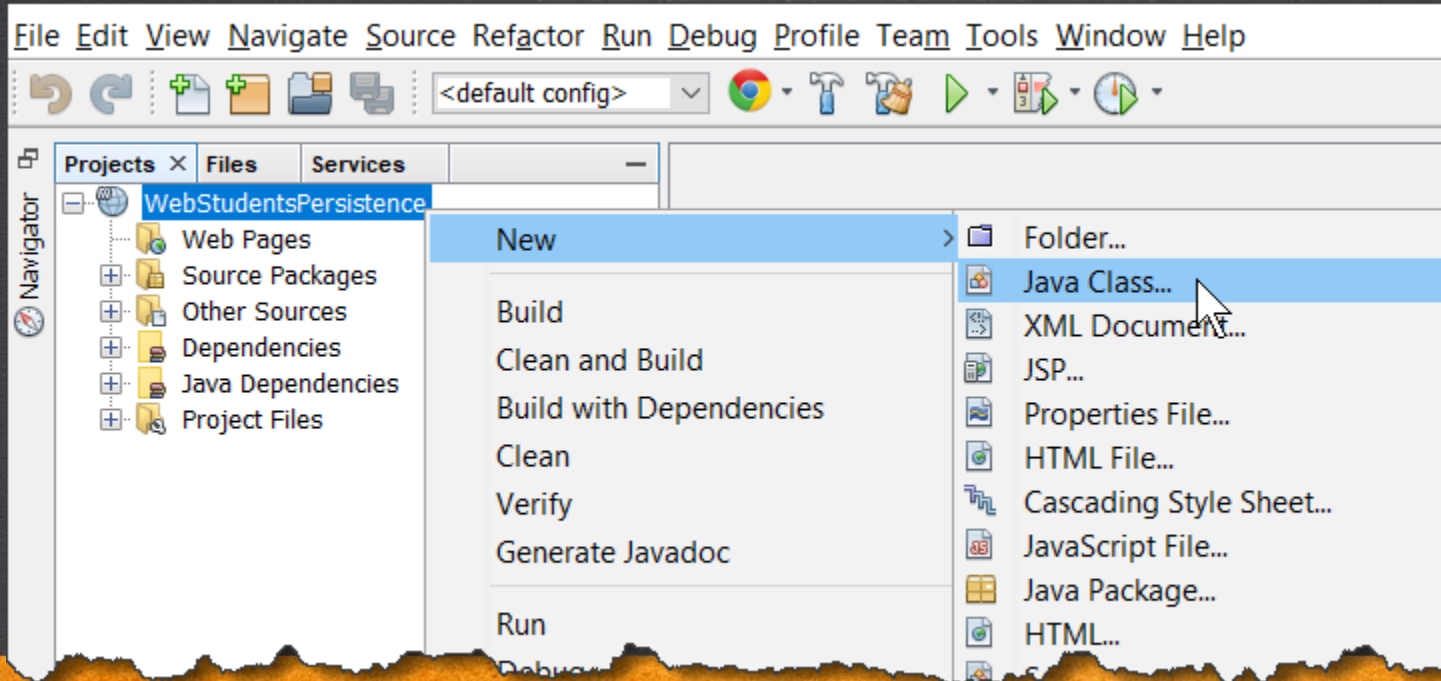
//6.Redirect to the index page
request.getRequestDispatcher("/index.jsp").forward(request, response);

}
```



# 16. CREATE A NEW CLASS

Create the ServletRedirect.java class:



**CURSO HIBERNATE Y JPA**

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

# 16. CREATE A NEW CLASS

Create the ServletRedirect.java class:

**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

# 17. MODIFY THE CODE

## ServletRedirect.java:

Click to download

```
package web;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;

@WebServlet("/ServletRedirect")
public class ServletRedirect extends HttpServlet {

    @Override
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        //Redirect to the Add Student page
        request.getRequestDispatcher("WEB-INF/addStudent.jsp").forward(request, response);
    }
}
```

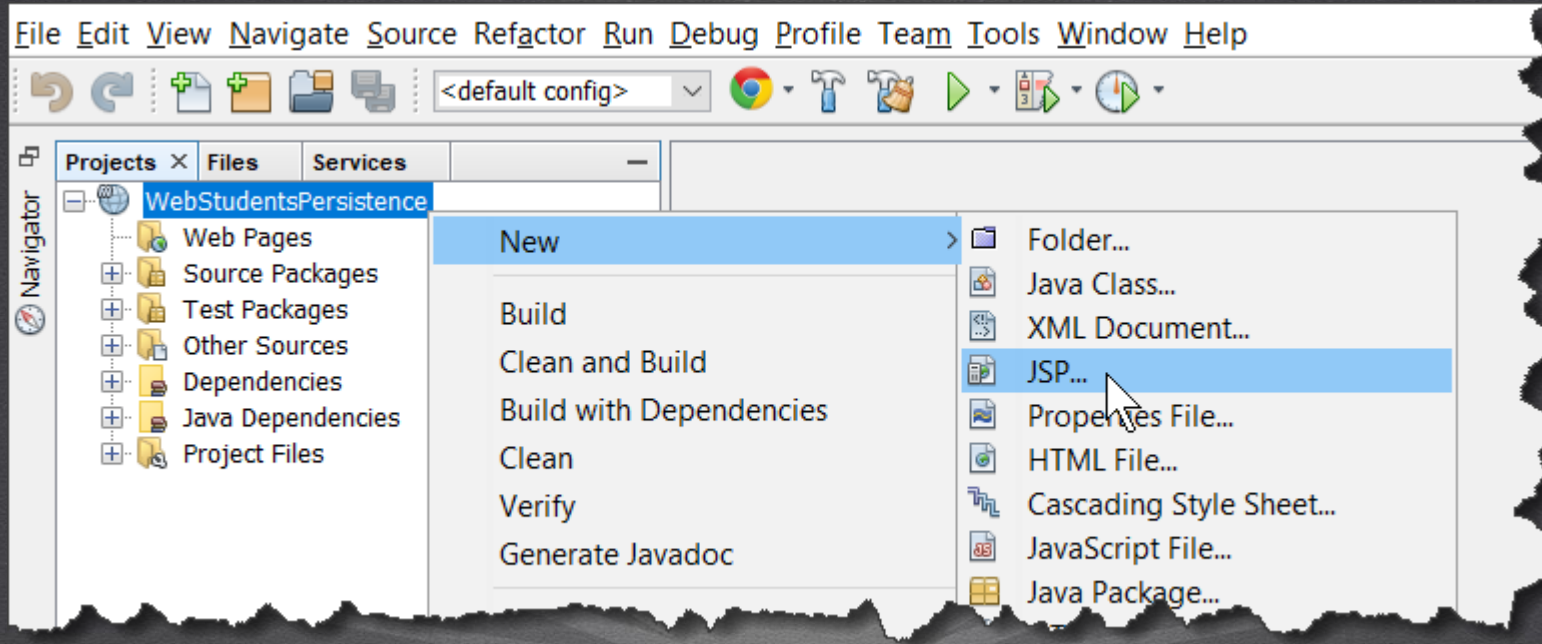
**CURSO HIBERNATE Y JPA**

www.globalmentoring.com.mx



# 18. CREATE A NEW JSP

Create the index.jsp file:



**CURSO HIBERNATE Y JPA**

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



# 18. CREATE A NEW JSP

Create the index.jsp file:

**New JSP**

**Steps**

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

**Name and Location**

File Name:

Project:

Location:

Folder:

Browse...

Created File:

Options:

☒ JSP File (Standard Syntax) ☐ Create as a JSP Segment

☐ JSP Document (XML Syntax)

Description:

< Back Next > **Finish** Cancel Help

# 19. MODIFY THE CODE

[index.jsp:](#)

Click to download

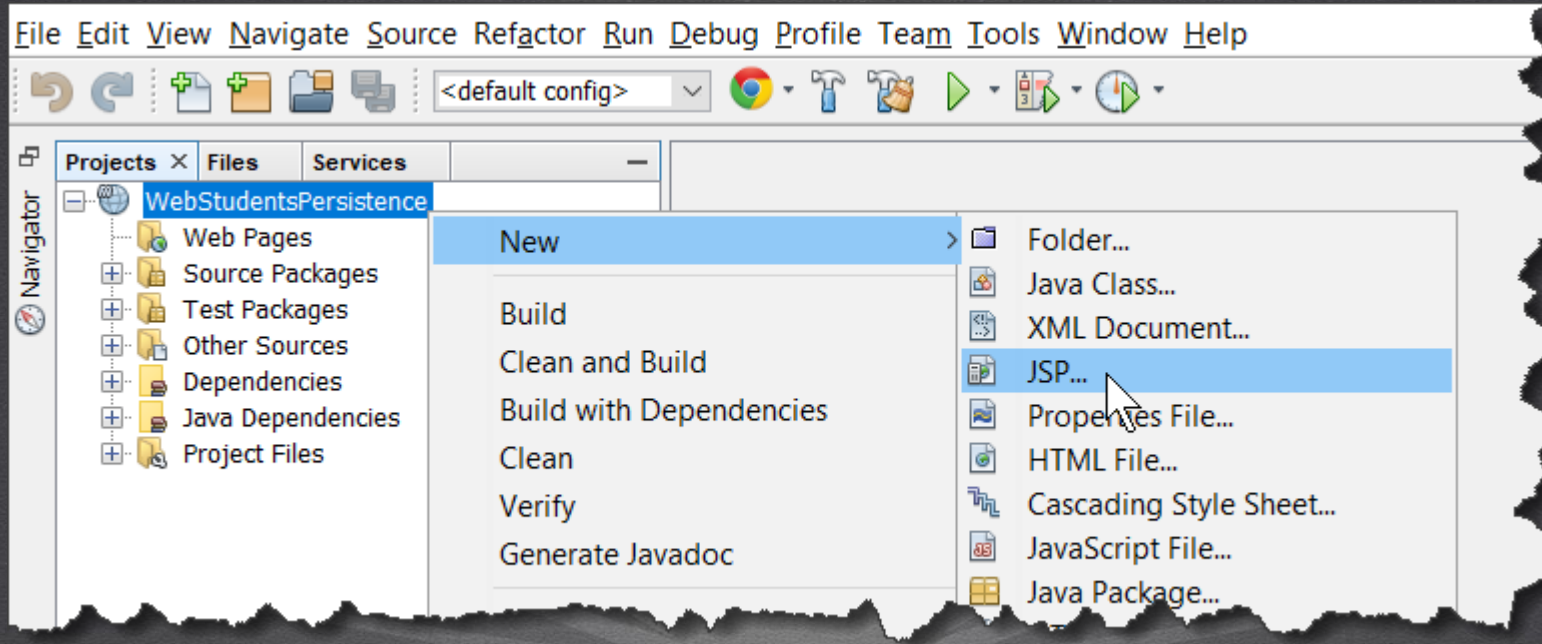
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Index</title>
  </head>
  <body>
    <a href="${pageContext.request.contextPath}/ServletController">
      Catalog of Students
    </a>
  </body>
</html>
```

**CURSO HIBERNATE Y JPA**

www.globalmentoring.com.mx

## 20. CREATE A NEW JSP

Create the listStudents.jsp:



**CURSO HIBERNATE Y JPA**

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

## 20. CREATE A JSP FILE

Create the listStudents.jsp:

**New JSP**

**Steps**

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

**Name and Location**

File Name: listStudents

Project: WebStudentsPersistence

Location: Web Pages

Folder: WEB-INF Browse...

Created File: C:\Courses\Hibernate\Lesson07\WebStudentsPersistence\src\main\webapp\WEB-INF\listStudents.jsp

Options:

☒ JSP File (Standard Syntax) ☐ Create as a JSP Segment

☐ JSP Document (XML Syntax)

Description:

A JSP file using JSP standard syntax.

< Back Next > **Finish** Cancel Help



# 21. MODIFY THE CODE

## listStudents.jsp:

Click to download

```
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>List of Students</title>
  </head>
  <body>
    List of Students
    <br/>
    <a href="${pageContext.request.contextPath}/ServletRedirect">
      Add
    </a>
    <br/>
    <table border="1">
      <tr>
        <th>Student ID</th>
        <th>Name</th>
        <th>Street name</th>
        <th>Street number</th>
        <th>Country</th>
      </tr>
```

## 21. MODIFY THE CODE

[listStudents.jsp:](#)

Click to download

```
<c:forEach var="student" items="${students}">
    <tr>
        <td>

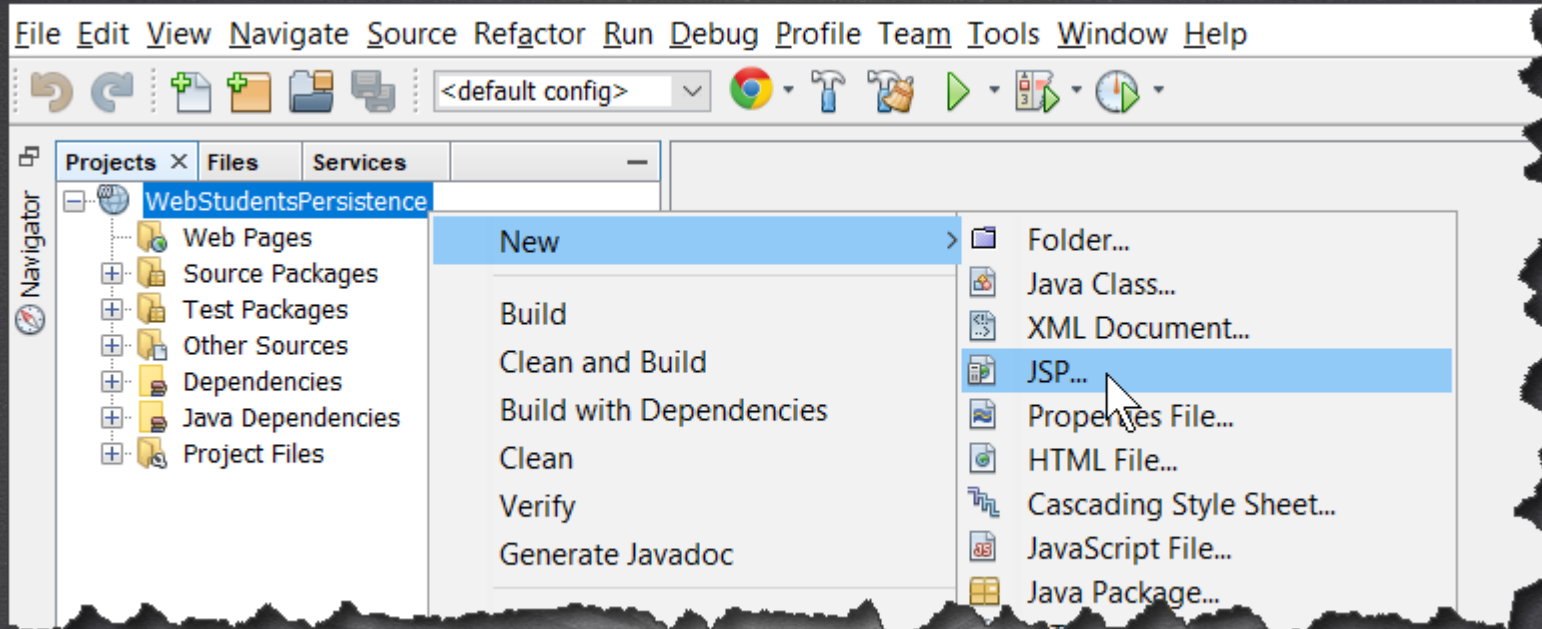
<a href="${pageContext.request.contextPath}/ServletModify?idStudent=${student.idStudent}">
            ${student.idStudent}
        </a>
    </td>
    <td>${student.name}</td>
    <td>${student.address.streetName }</td>
    <td>${student.address.streetNumber }</td>
    <td>${student.address.country }</td>
    </tr>
</c:forEach>
</table>
</body>
</html>
```

**CURSO HIBERNATE Y JPA**

www.globalmentoring.com.mx

## 22. CREATE A NEW JSP

Create a addStudent.jsp:



**CURSO HIBERNATE Y JPA**

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

## 22. CREATE A NEW JSP

Create a addStudent.jsp:

**New JSP**

**Steps**

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

**Name and Location**

File Name:

Project:

Location:

Folder:

Created File:

**Options:**

☒ JSP File (Standard Syntax) ☐ Create as a JSP Segment

☐ JSP Document (XML Syntax)

**Description:**



## 23. MODIFY THE FILE

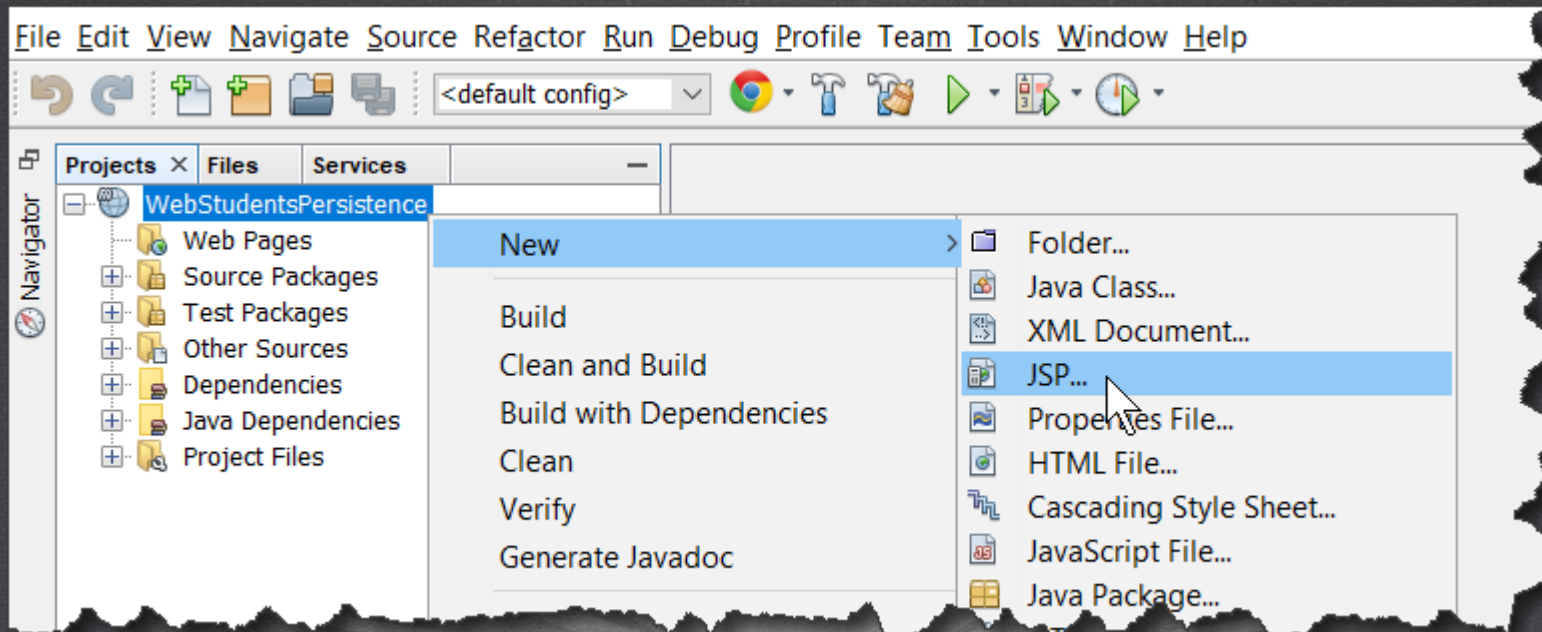
[addStudent.jsp:](#)

Click to download

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Add Student</title>
  </head>
  <body>
    Add Student
    <br/>
    <form name="form1" action="${pageContext.request.contextPath}/ServletAdd" method="post">
      Name: <input type="text" name="name" >
      <br/>
      <br/>
      Address:
      <br/>
      Street Name: <input type="text" name="streetName">
      <br/>
      Street Number: <input type="text" name="streetNumber">
      <br/>
      Country: <input type="text" name="country">
      <br/>
      <input type="submit" name="add" value="Add">
    </form>
  </body>
</html>
```

## 24. CREATE A NEW JSP

Create the modifyStudent.jsp:



**CURSO HIBERNATE Y JPA**

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

## 24. CREATE A NEW JSP

Create the modifyStudent.jsp:

**New JSP**

**Steps**

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

**Name and Location**

File Name:

Project:

Location:

Folder:

Created File:

**Options:**

☒ JSP File (Standard Syntax) ☐ Create as a JSP Segment

☐ JSP Document (XML Syntax)

**Description:**

# 25. MODIFY THE FILE

[modifyStudent.jsp:](#)

Click to download

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Modify Student</title>
  </head>
  <body>
    Modify Student
    <br/>
    <form name="form1" action="${pageContext.request.contextPath}/ServletModify" method="post">
      Id Student: <input type="text" name="idStudent" value="${student.idStudent}" readonly="readonly">
      <br/>
      Name: <input type="text" name="name" value="${student.name}">
      <br/>
      <br/>
      Address:
      <br/>
      Street Name: <input type="text" name="streetName" value="${student.address.streetName}">
      <br/>
      Street Number: <input type="text" name="streetNumber" value="${student.address.streetNumber}">
      <br/>
      Country: <input type="text" name="country" value="${student.address.country}">
      <br/>
```



## 25. MODIFY THE FILE

[modifyStudent.jsp:](#)

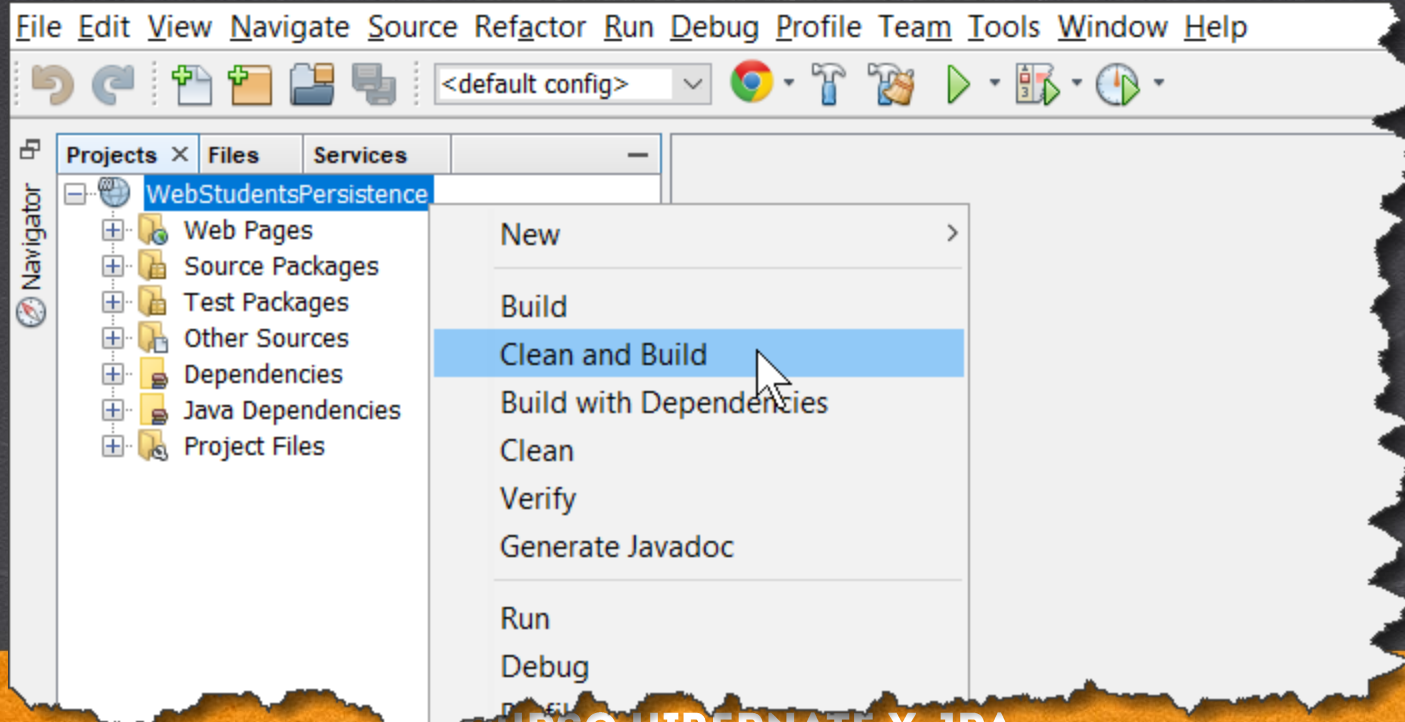
Click to download

```
<input type="submit" name="modify" value="modify">
<input type="submit" name="delete" value="delete">

<!-- We do not need to handle hidden values since we will
      rely on the object in session for editing -->
</form>
</body>
</html>
```

# 26. EXECUTE CLEAN & BUILD

Execute Clean & Build:

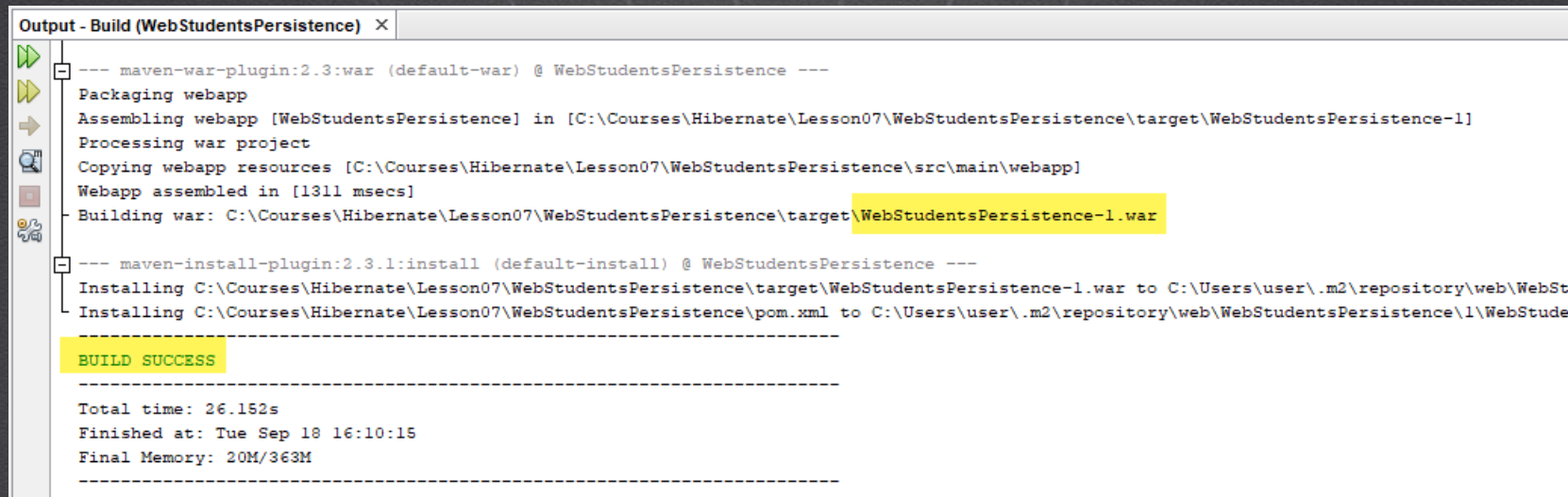


**CURSO HIBERNATE Y JPA**

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

# 26. EXECUTE CLEAN & BUILD

## Execute Clean & Build:



The screenshot shows the 'Output - Build (WebStudentsPersistence)' window in an IDE. The output text is as follows:

```
--- maven-war-plugin:2.3:war (default-war) @ WebStudentsPersistence ---
Packaging webapp
Assembling webapp [WebStudentsPersistence] in [C:\Courses\Hibernate\Lesson07\WebStudentsPersistence\target\WebStudentsPersistence-1]
Processing war project
Copying webapp resources [C:\Courses\Hibernate\Lesson07\WebStudentsPersistence\src\main\webapp]
Webapp assembled in [1311 msecs]
Building war: C:\Courses\Hibernate\Lesson07\WebStudentsPersistence\target\WebStudentsPersistence-1.war

--- maven-install-plugin:2.3.1:install (default-install) @ WebStudentsPersistence ---
Installing C:\Courses\Hibernate\Lesson07\WebStudentsPersistence\target\WebStudentsPersistence-1.war to C:\Users\user\.m2\repository\web\WebSt
Installing C:\Courses\Hibernate\Lesson07\WebStudentsPersistence\pom.xml to C:\Users\user\.m2\repository\web\WebStudentsPersistence\1\WebStude

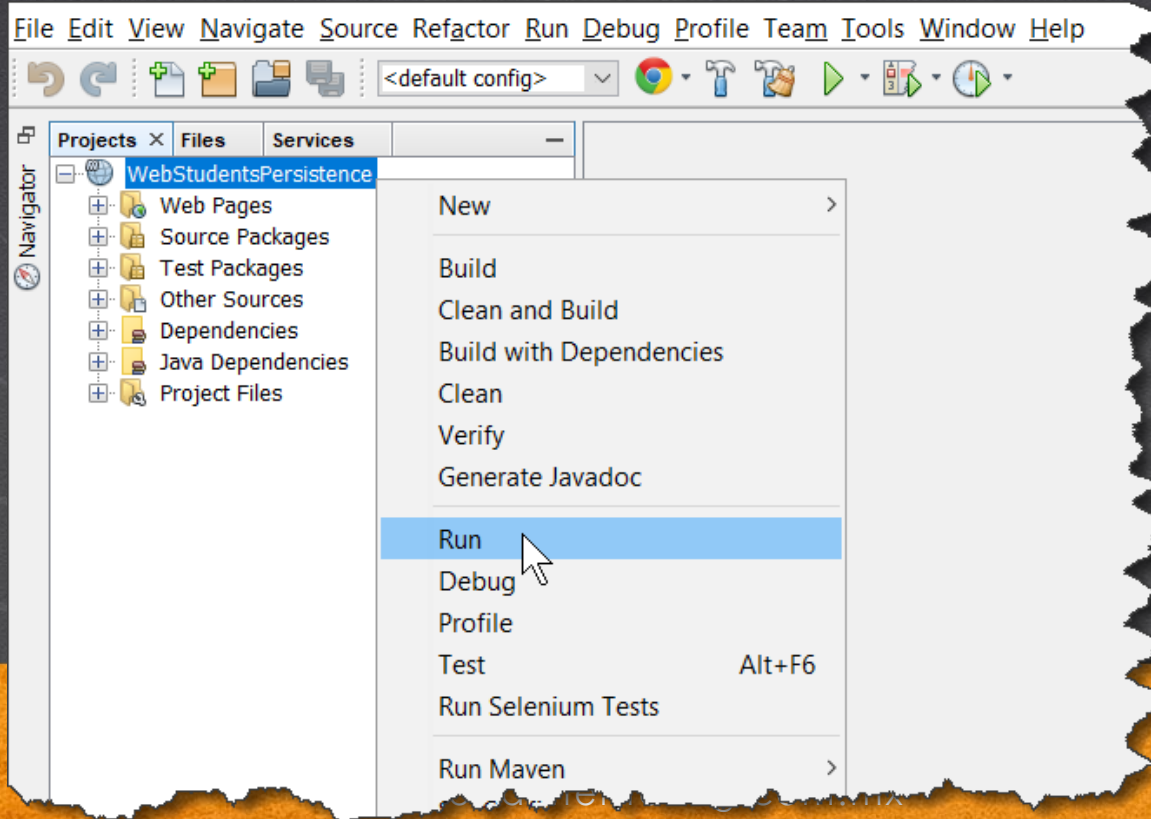
BUILD SUCCESS

Total time: 26.152s
Finished at: Tue Sep 18 16:10:15
Final Memory: 20M/363M
```



# 27. EXECUTE THE PROJECT

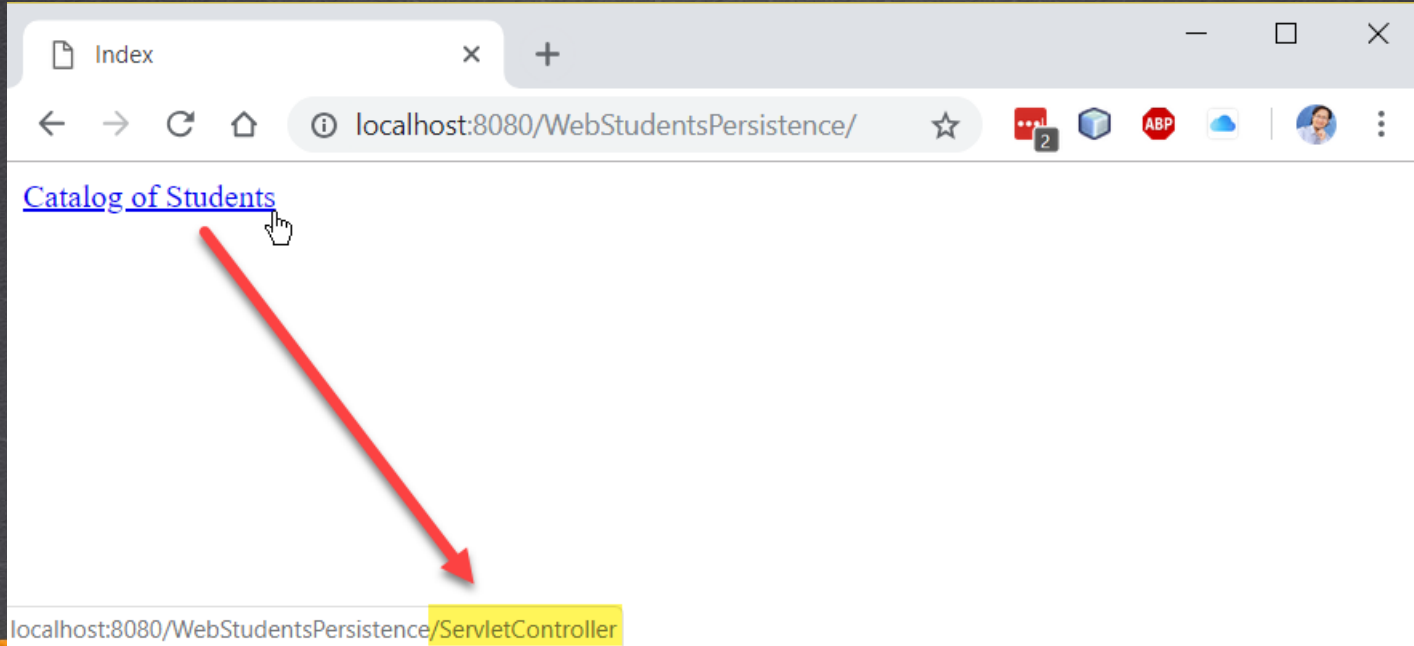
Execute the project:





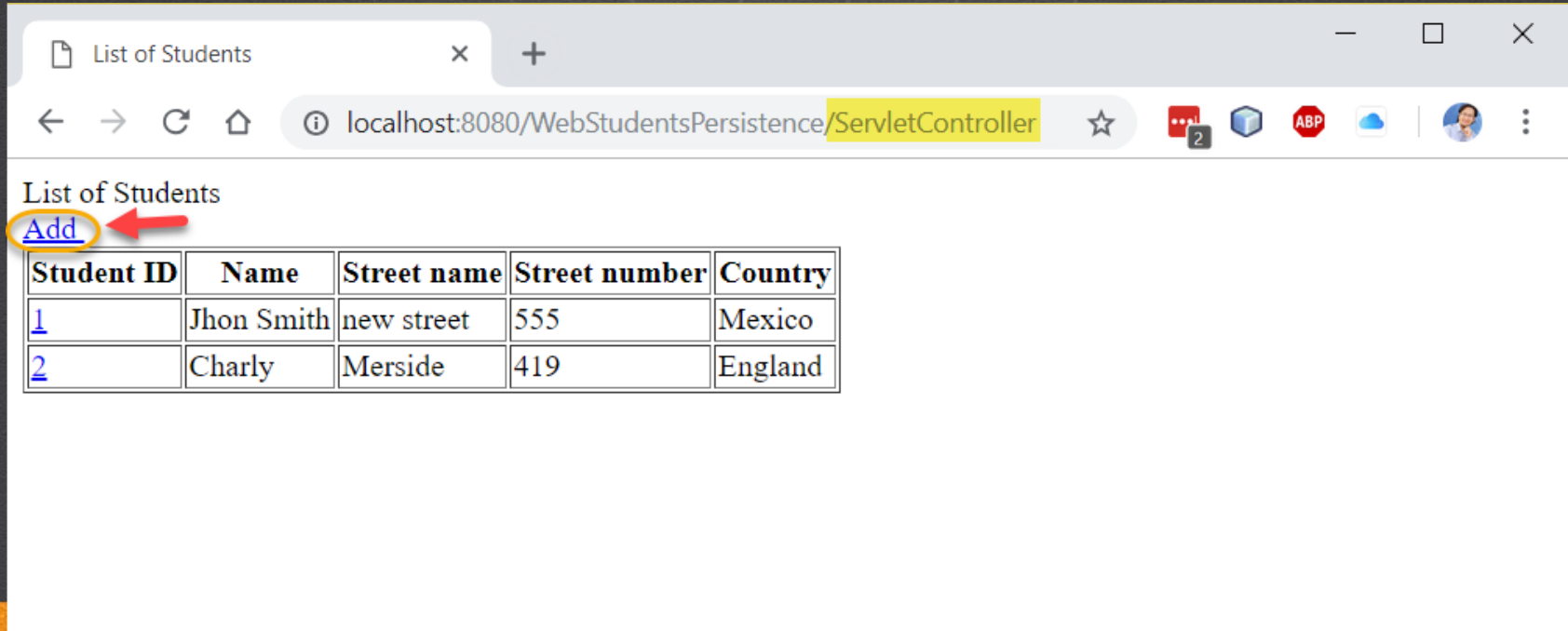
## 27. EXECUTE THE PROJECT

Enter to the student's catalog:



# 27. EXECUTE THE PROJECT

Add an Student:



List of Students

[Add](#)

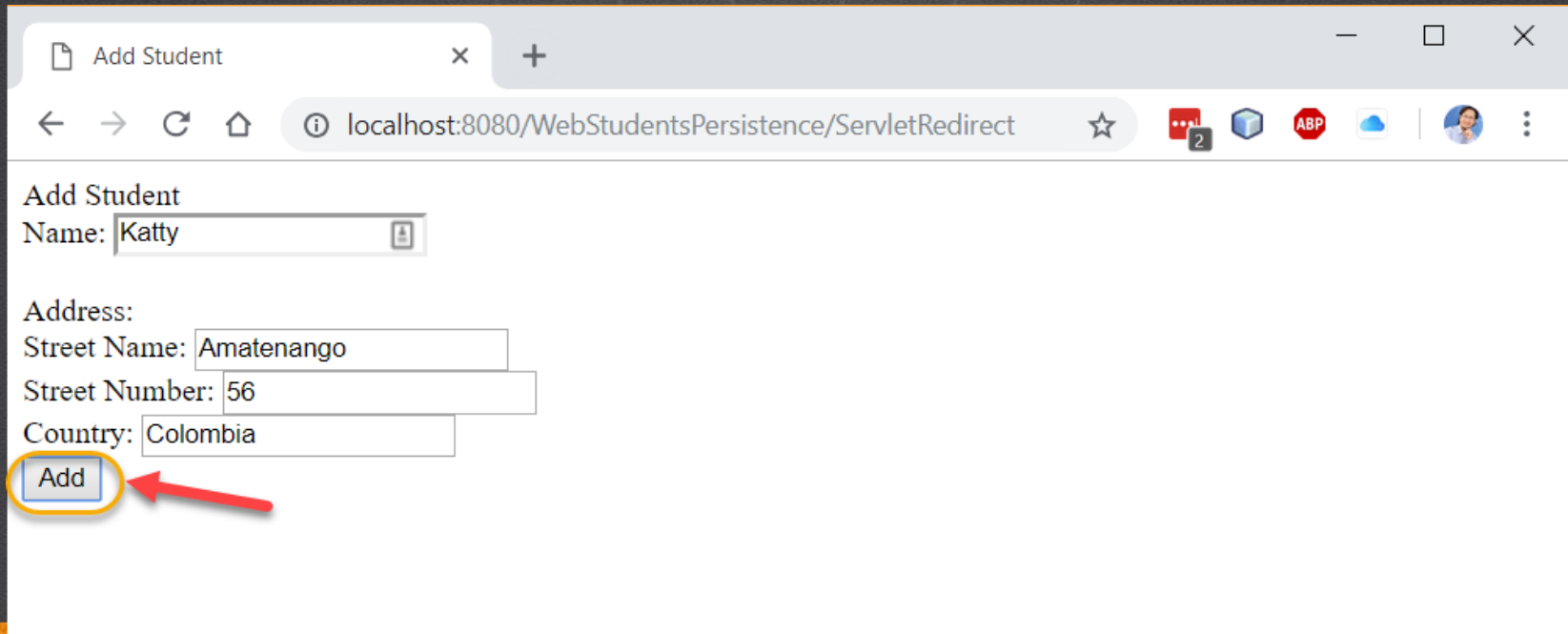
Student ID	Name	Street name	Street number	Country
<a href="#">1</a>	Jhon Smith	new street	555	Mexico
<a href="#">2</a>	Charly	Merside	419	England

**CURSO HIBERNATE Y JPA**

www.globalmentoring.com.mx

# 27. EXECUTE THE PROJECT

Add a Student:

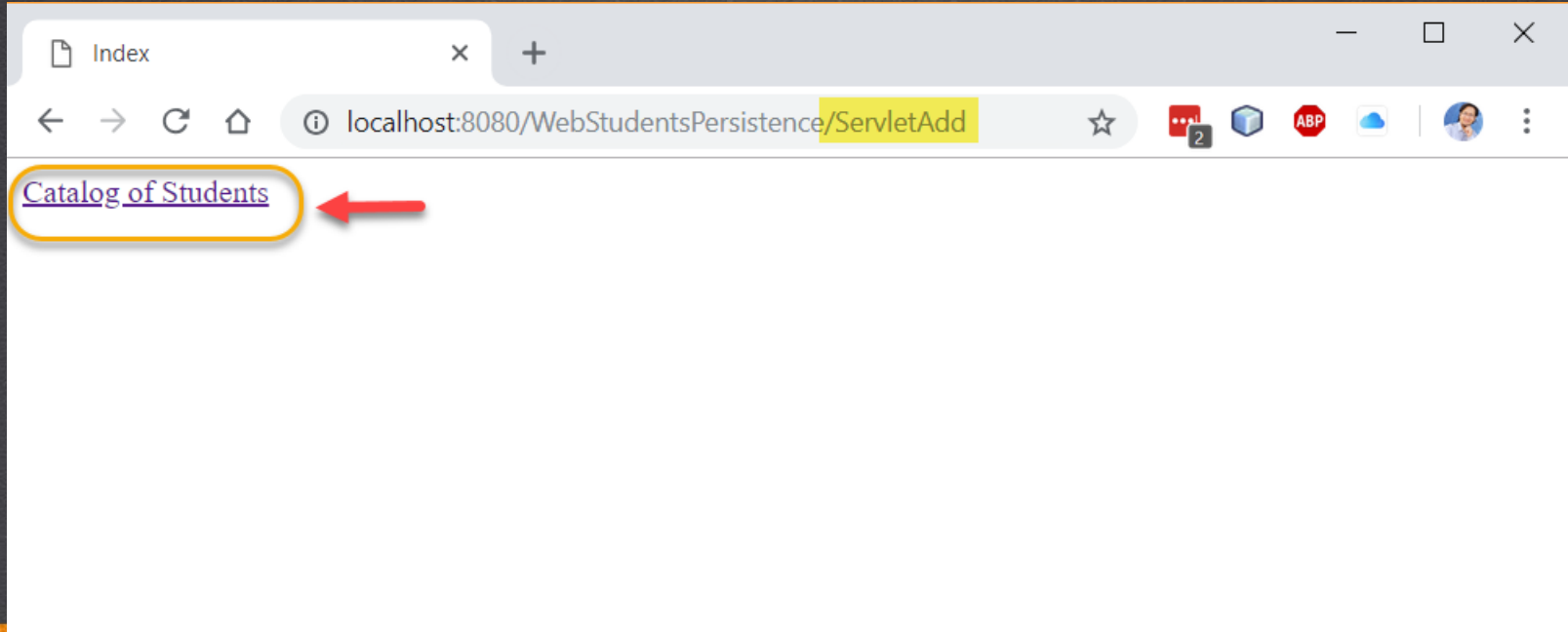


The screenshot shows a web browser window with the title 'Add Student'. The address bar displays 'localhost:8080/WebStudentsPersistence/ServletRedirect'. The form contains the following fields and values:

- Name: Katty
- Address:
  - Street Name: Amatenango
  - Street Number: 56
  - Country: Colombia
- Add (button, highlighted with a red circle and a red arrow pointing to it)

## 27. EXECUTE THE PROJECT

Add a Student:





# 27. EXECUTE THE PROJECT

We review the added student, and edit it:

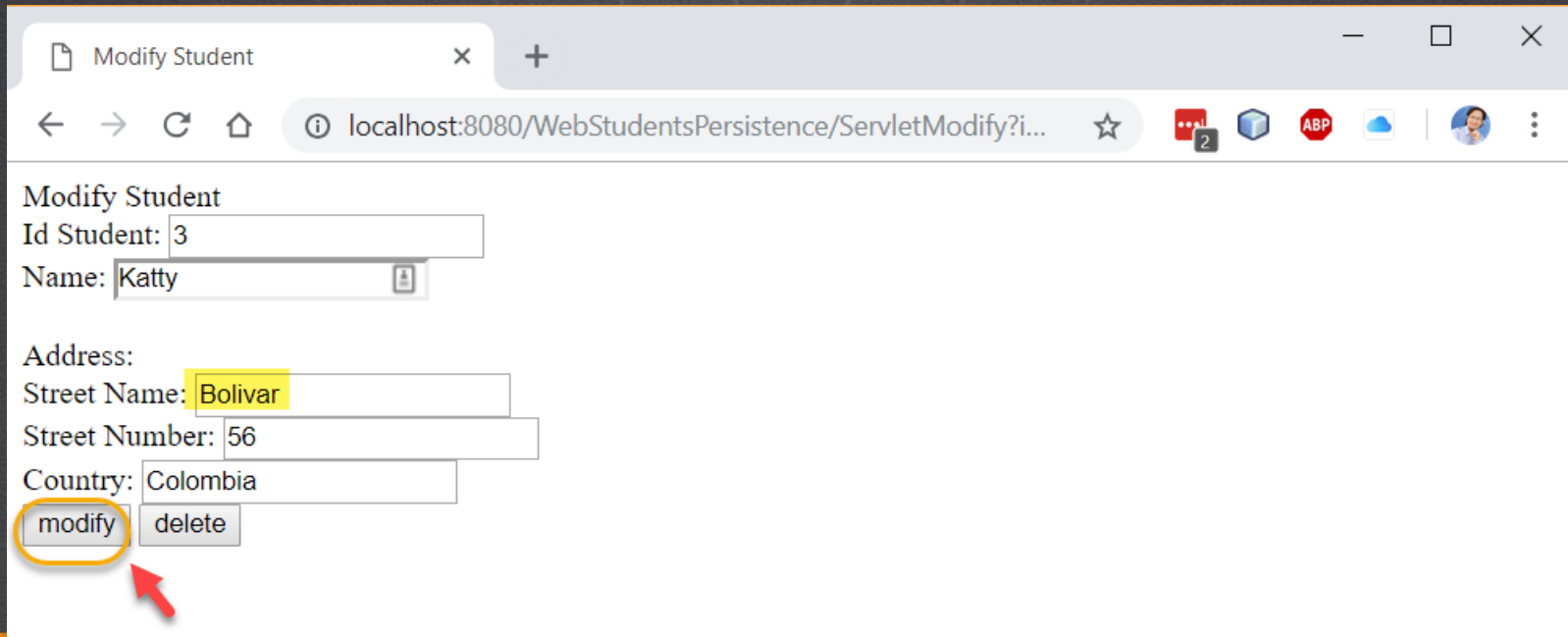
List of Students

[Add](#)

Student ID	Name	Street name	Street number	Country
<a href="#">1</a>	Jhon Smith	new street	555	Mexico
<a href="#">2</a>	Charly	Merside	419	England
<a href="#">3</a>	Katty	Amatenango	56	Colombia

## 27. EXECUTE THE PROJECT

Modify the selected Student:



Modify Student

Id Student:

Name:

Address:

Street Name:

Street Number:

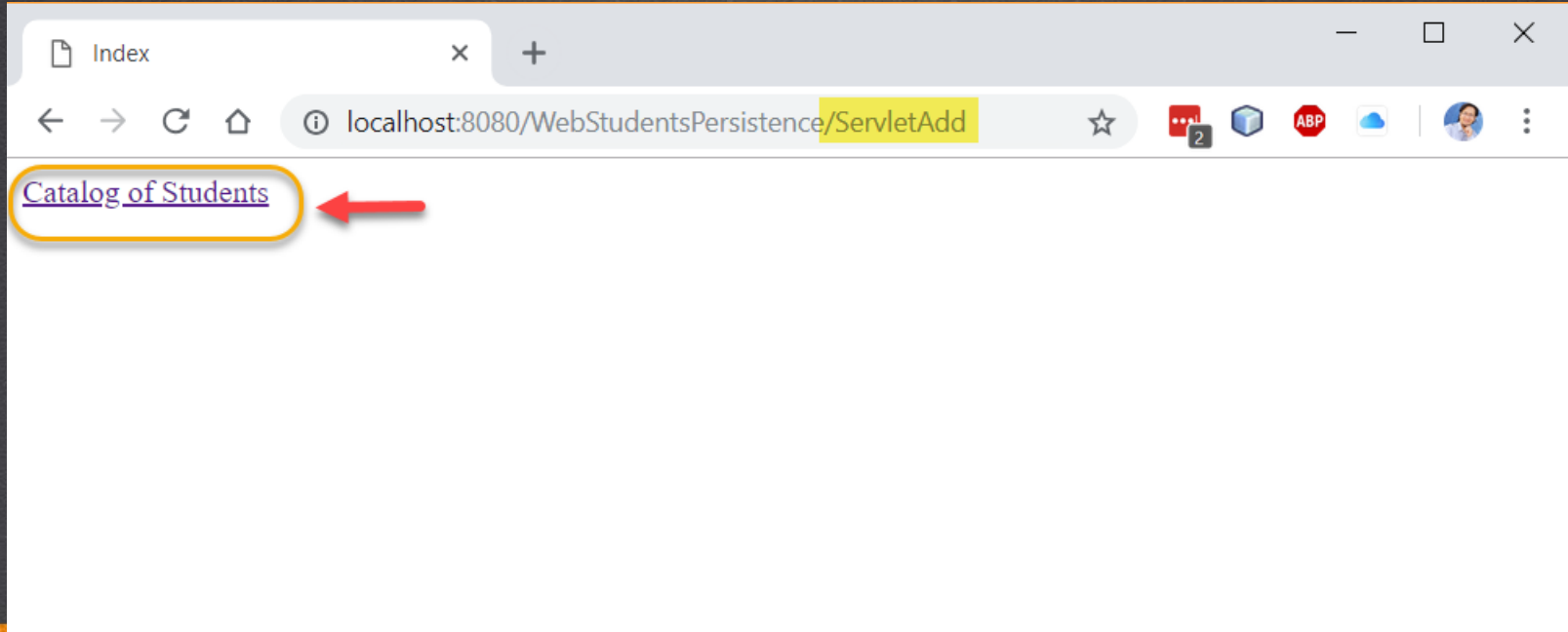
Country:

**CURSO HIBERNATE Y JPA**

www.globalmentoring.com.mx

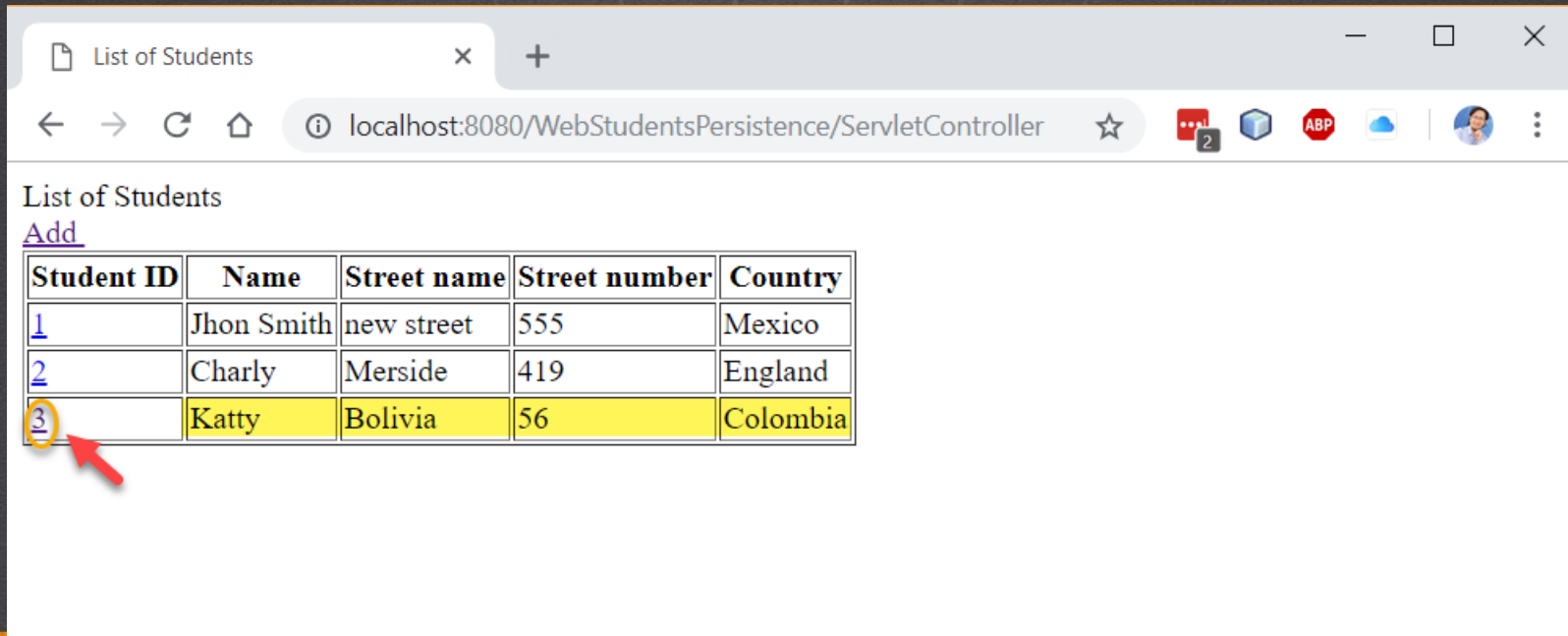
## 27. EXECUTE THE PROJECT

Delete an Student:



## 27. EXECUTE THE PROJECT

We see the modified students and we select it to eliminate it:



List of Students

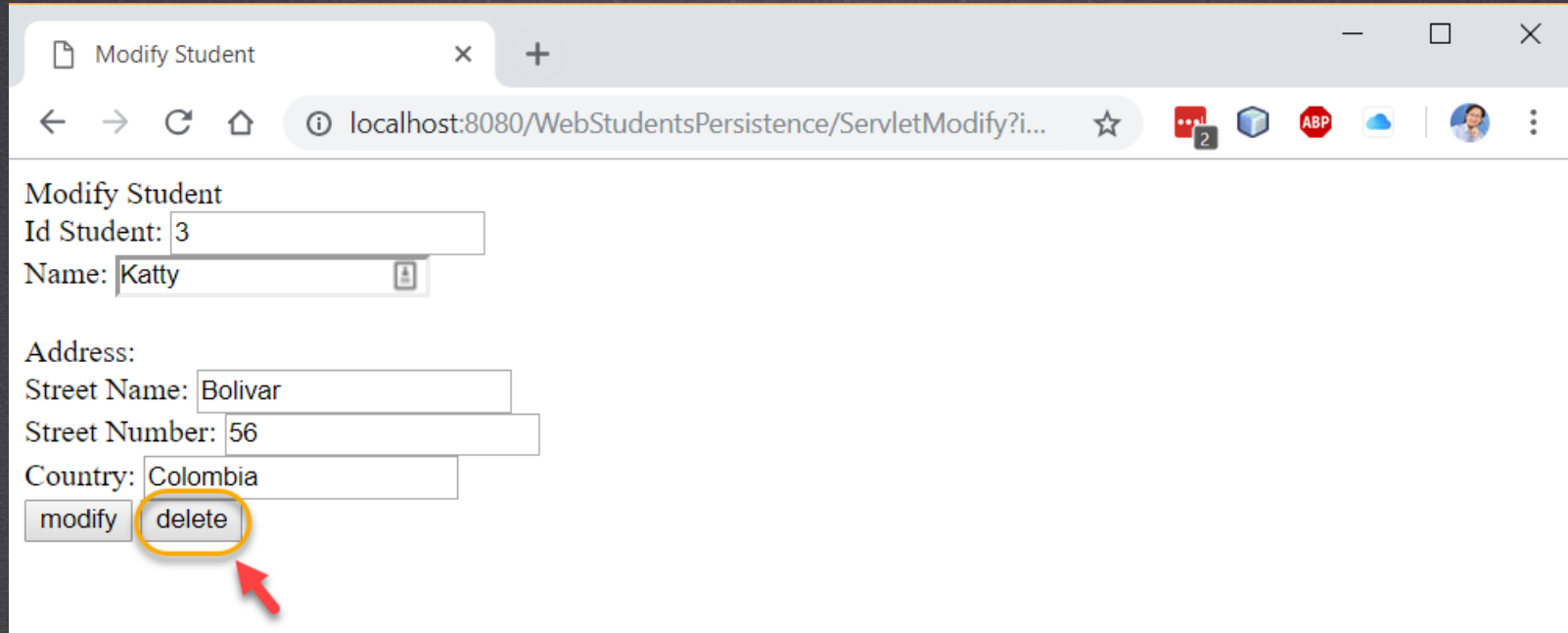
[Add](#)

Student ID	Name	Street name	Street number	Country
<a href="#">1</a>	Jhon Smith	new street	555	Mexico
<a href="#">2</a>	Charly	Merside	419	England
<a href="#">3</a>	Katty	Bolivia	56	Colombia



## 27. EXECUTE THE PROJECT

We eliminate the selected student:



Modify Student

Id Student: 3

Name: Katty

Address:

Street Name: Bolivar

Street Number: 56

Country: Colombia

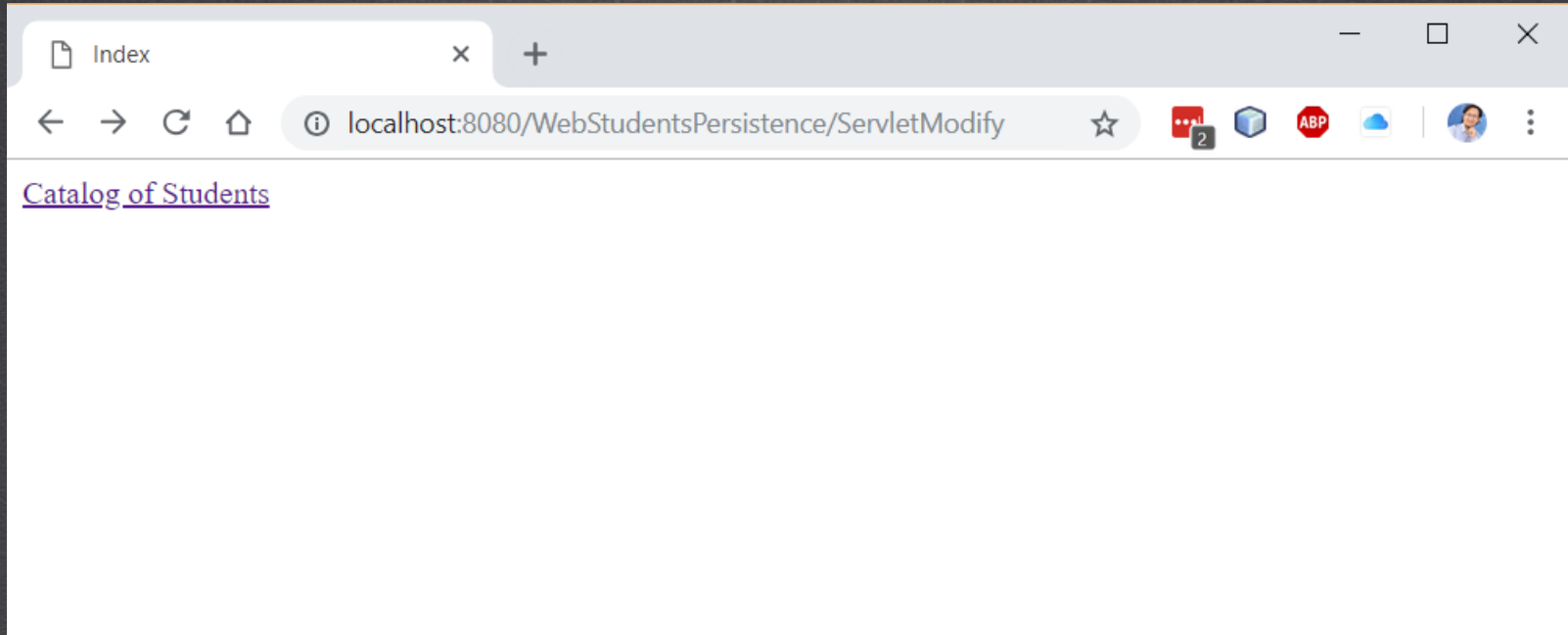
modify delete

**CURSO HIBERNATE Y JPA**

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

## 27. EXECUTE THE PROJECT

We eliminate the selected student:



## 27. EXECUTE THE PROJECT

We see that the student has already been eliminated:

List of Students

[Add](#)

Student ID	Name	Street name	Street number	Country
<a href="#">1</a>	Jhon Smith	new street	555	Mexico
<a href="#">2</a>	Charly	Merside	419	England



# EXERCISE CONCLUSION

- With this exercise we have integrated a Web application with Hibernate and JPA using Glassfish as an application server.
- We apply several design patterns such as MVC, DAO and DTO, as a whole we can deploy the respective actions, all this with the help of the persistence layer supported by Hibernate and JPA.
- We add the actions to list, add, modify and eliminate the list of Students.
- We also take advantage of applying the concept of cascade persistence, both when adding, modifying or deleting an object of the Student type along with the relationships marked as Cascade, which is the object of Address.



**ONLINE COURSE**

# **HIBERNATE & JPA**

By: Eng. Ubaldo Acosta



**HIBERNATE & JPA COURSE**

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