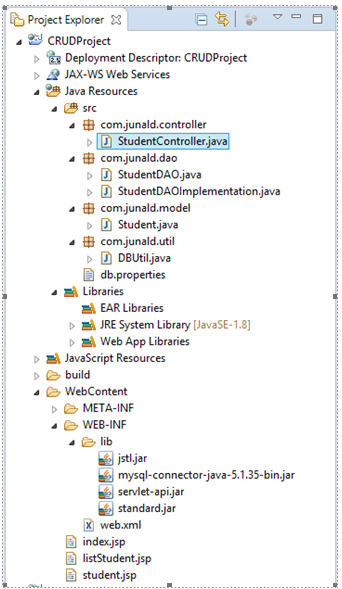
**Un exemple: Comment créer un CRUD application en utilisant Java Servlet**

Créer une simple application CRUD pour un auditeur en utilisant Servlet/JSP, JDBC et MYSQL.

On a besoin :

1. Eclipse IDE pour Java EE Developpers, j’ai utilisé version luna .
2. MYSQL Commumity Server.
3. Apache Tomcat 8.0
4. MySQL Connector Jar
5. Servlet api jar
6. Jstl jar

Au-dessous le chemin du répertoire de mon projet :



J’ai créé « a new dynamic web project » dans eclipse IDE avec select pour l’option «generate web.xml », puis quatres packages au-dessous du répertoire «src ».

1. **com.junald.controller,** contenant les servlets
2. **com.junald.dao,** contenant les classes ou les interfaces
3. **com.junald.model,** contenant les POJO
4. **com.junald.util,** contenant les classes utiles pour l’application

D’abord j’ai créé les tables dans mon database :

**CREATE TABLE `student` (**

**`studentId` int(5) NOT NULL AUTO\_INCREMENT,**

**`firstName` varchar(25) DEFAULT NULL,**

**`lastName` varchar(25) DEFAULT NULL,**

**`course` varchar(15) DEFAULT NULL,**

**`year` int(2) DEFAULT NULL,**

**PRIMARY KEY (`studentId`));**

Puis un nouveau classe nommé “**DBUtil**” est placé sous **com.junald.util**package pour connecté et se déconnecté a la base de donnée.

**package com.junald.util;**

**import java.io.IOException;**

**import java.io.InputStream;**

**import java.sql.Connection;**

**import java.sql.DriverManager;**

**import java.sql.SQLException;**

**import java.util.Properties;**

**public class DBUtil {**

**private static Connection conn;**

**public static Connection getConnection() {**

**if( conn != null )**

**return conn;**

**InputStream inputStream = DBUtil.class.getClassLoader().getResourceAsStream( "/db.properties" );**

**Properties properties = new Properties();**

**try {**

**properties.load( inputStream );**

**String driver = properties.getProperty( "driver" );**

**String url = properties.getProperty( "url" );**

**String user = properties.getProperty( "user" );**

**String password = properties.getProperty( "password" );**

**Class.forName( driver );**

**conn = DriverManager.getConnection( url, user, password );**

**} catch (IOException e) {**

**e.printStackTrace();**

**} catch (ClassNotFoundException e) {**

**e.printStackTrace();**

**} catch (SQLException e) {**

**e.printStackTrace();**

**}**

**return conn;**

**}**

**public static void closeConnection( Connection toBeClosed ) {**

**if( toBeClosed == null )**

**return;**

**try {**

**toBeClosed.close();**

**} catch (SQLException e) {**

**e.printStackTrace();**

**}**

**}**

**}**

Puis le fichier “**db.properties**” placé sous **src folder** comme nom de la base de donné, username et password.

**driver=com.mysql.jdbc.Driver**

**url=jdbc:**[**mysql://localhost:8080/UserDB**](mysql://localhost:3306/UserDB)

**user=Chantale**

**password=P@ssw0rd**

Un nouveau class appelé “**Student**” :

**package com.junald.model;**

**public class Student {**

**private int studentId;**

**private String firstName;**

**private String lastName;**

**private String course;**

**private int year;**

**public int getStudentId() {**

**return studentId;**

**}**

**public void setStudentId(int studentId) {**

**this.studentId = studentId;**

**}**

**public String getFirstName() {**

**return firstName;**

**}**

**public void setFirstName(String firstName) {**

**this.firstName = firstName;**

**}**

**public String getLastName() {**

**return lastName;**

**}**

**public void setLastName(String lastName) {**

**this.lastName = lastName;**

**}**

**public String getCourse() {**

**return course;**

**}**

**public void setCourse(String course) {**

**this.course = course;**

**}**

**public int getYear() {**

**return year;**

**}**

**public void setYear(int year) {**

**this.year = year;**

**}**

**@Override**

**public String toString() {**

**return "Student [studentId=" + studentId + ", firstName=" + firstName**

**+ ", lastName=" + lastName + ", course=" + course + ", year="**

**+ year + "]";**

**}**

**}**

Un interface“**StudentDAO**” sous **com.junald.dao** package.

**package com.junald.dao;**

**import java.util.List;**

**import com.junald.model.Student;**

**public interface StudentDAO {**

**public void addStudent( Student student );**

**public void deleteStudent( int studentId );**

**public void updateStudent( Student student );**

**public List<Student> getAllStudents();**

**public Student getStudentById( int studentId );**

**}**

Un nouveau class“**StudentDAOImplementation**” sous **com.junald.dao** package pour les opérations CRUD dans la base de données.

package com.junald.dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.List;

import com.junald.model.Student;

import com.junald.util.DBUtil;

public class StudentDAOImplementation implements StudentDAO {

    private Connection conn;

    public StudentDAOImplementation() {

        conn = DBUtil.getConnection();

    }

    @Override

    public void addStudent( Student student ) {

        try {

            String query = "insert into student (firstName, lastName, course, year) values (?,?,?,?)";

            PreparedStatement preparedStatement = conn.prepareStatement( query );

            preparedStatement.setString( 1, student.getFirstName() );

            preparedStatement.setString( 2, student.getLastName() );

            preparedStatement.setString( 3, student.getCourse() );

            preparedStatement.setInt( 4, student.getYear() );

            preparedStatement.executeUpdate();

            preparedStatement.close();

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }

    @Override

    public void deleteStudent( int studentId ) {

        try {

            String query = "delete from student where studentId=?";

            PreparedStatement preparedStatement = conn.prepareStatement(query);

            preparedStatement.setInt(1, studentId);

            preparedStatement.executeUpdate();

            preparedStatement.close();

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }

    @Override

    public void updateStudent( Student student ) {

        try {

            String query = "update student set firstName=?, lastName=?, course=?, year=? where studentId=?";

            PreparedStatement preparedStatement = conn.prepareStatement( query );

            preparedStatement.setString( 1, student.getFirstName() );

            preparedStatement.setString( 2, student.getLastName() );

            preparedStatement.setString( 3, student.getCourse() );

            preparedStatement.setInt( 4, student.getYear() );

            preparedStatement.setInt(5, student.getStudentId());

            preparedStatement.executeUpdate();

            preparedStatement.close();

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }

    @Override

    public List<Student> getAllStudents() {

        List<Student> students = new ArrayList<Student>();

        try {

            Statement statement = conn.createStatement();

            ResultSet resultSet = statement.executeQuery( "select \* from student" );

            while( resultSet.next() ) {

                Student student = new Student();

                student.setStudentId( resultSet.getInt( "studentId" ) );

                student.setFirstName( resultSet.getString( "firstName" ) );

                student.setLastName( resultSet.getString( "lastName" ) );

                student.setCourse( resultSet.getString( "course" ) );

                student.setYear( resultSet.getInt( "year" ) );

                students.add(student);

            }

            resultSet.close();

            statement.close();

        } catch (SQLException e) {

            e.printStackTrace();

        }

        return students;

    }

    @Override

    public Student getStudentById(int studentId) {

        Student student = new Student();

        try {

            String query = "select \* from student where studentId=?";

            PreparedStatement preparedStatement = conn.prepareStatement( query );

            preparedStatement.setInt(1, studentId);

            ResultSet resultSet = preparedStatement.executeQuery();

            while( resultSet.next() ) {

                student.setStudentId( resultSet.getInt( "studentId" ) );

                student.setFirstName( resultSet.getString( "firstName" ) );

                student.setLastName( resultSet.getString( "LastName" ) );

                student.setCourse( resultSet.getString( "course" ) );

                student.setYear( resultSet.getInt( "year" ) );

            }

            resultSet.close();

            preparedStatement.close();

        } catch (SQLException e) {

            e.printStackTrace();

        }

        return student;

    }

}

Maintenant créant le SERVLET de l’application “**StudentController**” sous **com.junald.controller** package

package com.junald.controller;

import java.io.IOException;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.junald.dao.StudentDAO;

import com.junald.dao.StudentDAOImplementation;

import com.junald.model.Student;

@WebServlet("/StudentController")

public class StudentController extends HttpServlet {

    private StudentDAO dao;

    private static final long serialVersionUID = 1L;

    public static final String lIST\_STUDENT = "/listStudent.jsp";

    public static final String INSERT\_OR\_EDIT = "/student.jsp";

    public StudentController() {

        dao = new StudentDAOImplementation();

    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        String forward = "";

        String action = request.getParameter( "action" );

        if( action.equalsIgnoreCase( "delete" ) ) {

            forward = lIST\_STUDENT;

            int studentId = Integer.parseInt( request.getParameter("studentId") );

            dao.deleteStudent(studentId);

            request.setAttribute("students", dao.getAllStudents() );

        }

        else if( action.equalsIgnoreCase( "edit" ) ) {

            forward = INSERT\_OR\_EDIT;

            int studentId = Integer.parseInt( request.getParameter("studentId") );

            Student student = dao.getStudentById(studentId);

            request.setAttribute("student", student);

        }

        else if( action.equalsIgnoreCase( "insert" ) ) {

            forward = INSERT\_OR\_EDIT;

        }

        else {

            forward = lIST\_STUDENT;

            request.setAttribute("students", dao.getAllStudents() );

        }

        RequestDispatcher view = request.getRequestDispatcher( forward );

        view.forward(request, response);

    }

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

        Student student = new Student();

        student.setFirstName( request.getParameter( "firstName" ) );

        student.setLastName( request.getParameter( "lastName" ) );

        student.setCourse( request.getParameter( "course" ) );

        student.setYear( Integer.parseInt( request.getParameter( "year" ) ) );

        String studentId = request.getParameter("studentId");

        if( studentId == null || studentId.isEmpty() )

            dao.addStudent(student);

        else {

            student.setStudentId( Integer.parseInt(studentId) );

            dao.updateStudent(student);

        }

        RequestDispatcher view = request.getRequestDispatcher( lIST\_STUDENT );

        request.setAttribute("students", dao.getAllStudents());

        view.forward(request, response);

    }

}

On a besoin d’un forme pour inserrer les auditeurs pour cela on crée un nouveau JSP nommé “**student.jsp**“.

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

    pageEncoding="ISO-8859-1"%>

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

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Add New Student</title>

</head>

<body>

    <form action="StudentController.do" method="post">

        <fieldset>

            <div>

                <label for="studentId">Student ID</label> <input type="text"

                    name="studentId" value="<c:out value="${student.studentId}" />"

                    readonly="readonly" placeholder="Student ID" />

            </div>

            <div>

                <label for="firstName">First Name</label> <input type="text"

                    name="firstName" value="<c:out value="${student.firstName}" />"

                    placeholder="First Name" />

            </div>

            <div>

                <label for="lastName">Last Name</label> <input type="text"

                    name="lastName" value="<c:out value="${student.lastName}" />"

                    placeholder="Last Name" />

            </div>

            <div>

                <label for="course">Course</label> <input type="text" name="course"

                    value="<c:out value="${student.course}" />" placeholder="Course" />

            </div>

            <div>

                <label for="year">Year</label> <input type="text" name="year"

                    value="<c:out value="${student.year}" />" placeholder="Year" />

            </div>

            <div>

                <input type="submit" value="Submit" />

            </div>

        </fieldset>

    </form>

</body>

</html>

Et pour la liste des auditeurs“**listStudent.jsp**“.

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

    pageEncoding="ISO-8859-1"%>

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

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "<http://www.w3.org/TR/html4/loose.dtd>">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Show All Students</title>

</head>

<body>

    <table>

        <thead>

            <tr>

                <th>Student ID</th>

                <th>First Name</th>

                <th>Last Name</th>

                <th>Course</th>

                <th>Year</th>

                <th colspan="2">Action</th>

            </tr>

        </thead>

        <tbody>

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

                <tr>

                    <td><c:out value="${student.studentId}" /></td>

                    <td><c:out value="${student.firstName}" /></td>

                    <td><c:out value="${student.lastName}" /></td>

                    <td><c:out value="${student.course}" /></td>

                    <td><c:out value="${student.year}" /></td>

                    <td><a

                        href="StudentController.do?action=edit&studentId=<c:out value="${student.studentId }"/>">Update</a></td>

                    <td><a

                        href="StudentController.do?action=delete&studentId=<c:out value="${student.studentId }"/>">Delete</a></td>

                </tr>

            </c:forEach>

        </tbody>

    </table>

    <p>

        <a href="StudentController.do?action=insert">Add Student</a>

    </p>

</body>

</html>

Et **index.jsp** comme page default pour l’application :

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

    pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "<http://www.w3.org/TR/html4/loose.dtd>">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Welcome</title>

</head>

<body>

    <jsp:forward page="/StudentController?action=listStudent"></jsp:forward>

</body>

</html>

Donc notre application ca marche enfin :

