Eclipse : oxygen, java 1.8, servlet 3.1.0, Tomcat 9.0.10

1.include jar files inside web-inf / lib folder

2. Put JSTL , servlet and Mysql jar files inside lib folder

3.Model class Book.java

package com.model;

public class Book {

protected int id;

protected String title;

protected String author;

protected float price;

public Book() {

}

public Book(int id) {

this.id = id;

}

public Book(int id, String title, String author, float price) {

this(title, author, price);

this.id = id;

}

public Book(String title, String author, float price) {

this.title = title;

this.author = author;

this.price = price;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getAuthor() {

return author;

}

public void setAuthor(String author) {

this.author = author;

}

public float getPrice() {

return price;

}

public void setPrice(float price) {

this.price = price;

}

}

4.DAO class

package com.dao;

import java.sql.Connection;

import java.sql.DriverManager;

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.model.\*;

public class BookDAO {

private String jdbcURL;

private String jdbcUsername;

private String jdbcPassword;

private Connection jdbcConnection;

public BookDAO(String jdbcURL, String jdbcUsername, String jdbcPassword) {

this.jdbcURL = jdbcURL;

this.jdbcUsername = jdbcUsername;

this.jdbcPassword = jdbcPassword;

}

protected void connect() throws SQLException {

if (jdbcConnection == null || jdbcConnection.isClosed()) {

try {

Class.forName("com.mysql.jdbc.Driver");

} catch (ClassNotFoundException e) {

throw new SQLException(e);

}

jdbcConnection = DriverManager.getConnection(

jdbcURL, jdbcUsername, jdbcPassword);

}

}

protected void disconnect() throws SQLException {

if (jdbcConnection != null && !jdbcConnection.isClosed()) {

jdbcConnection.close();

}

}

public boolean insertBook(Book book) throws SQLException {

String sql = "INSERT INTO book (title, author, price) VALUES (?, ?, ?)";

connect();

PreparedStatement statement = jdbcConnection.prepareStatement(sql);

statement.setString(1, book.getTitle());

statement.setString(2, book.getAuthor());

statement.setFloat(3, book.getPrice());

boolean rowInserted = statement.executeUpdate() > 0;

statement.close();

disconnect();

return rowInserted;

}

public List<Book> listAllBooks() throws SQLException {

List<Book> listBook = new ArrayList<>();

String sql = "SELECT \* FROM book";

connect();

Statement statement = jdbcConnection.createStatement();

ResultSet resultSet = statement.executeQuery(sql);

while (resultSet.next()) {

int id = resultSet.getInt("book\_id");

String title = resultSet.getString("title");

String author = resultSet.getString("author");

float price = resultSet.getFloat("price");

Book book = new Book(id, title, author, price);

listBook.add(book);

}

resultSet.close();

statement.close();

disconnect();

return listBook;

}

public boolean deleteBook(Book book) throws SQLException {

String sql = "DELETE FROM book where book\_id = ?";

connect();

PreparedStatement statement = jdbcConnection.prepareStatement(sql);

statement.setInt(1, book.getId());

boolean rowDeleted = statement.executeUpdate() > 0;

statement.close();

disconnect();

return rowDeleted;

}

public boolean updateBook(Book book) throws SQLException {

String sql = "UPDATE book SET title = ?, author = ?, price = ?";

sql += " WHERE book\_id = ?";

connect();

PreparedStatement statement = jdbcConnection.prepareStatement(sql);

statement.setString(1, book.getTitle());

statement.setString(2, book.getAuthor());

statement.setFloat(3, book.getPrice());

statement.setInt(4, book.getId());

boolean rowUpdated = statement.executeUpdate() > 0;

statement.close();

disconnect();

return rowUpdated;

}

public Book getBook(int id) throws SQLException {

Book book = null;

String sql = "SELECT \* FROM book WHERE book\_id = ?";

connect();

PreparedStatement statement = jdbcConnection.prepareStatement(sql);

statement.setInt(1, id);

ResultSet resultSet = statement.executeQuery();

if (resultSet.next()) {

String title = resultSet.getString("title");

String author = resultSet.getString("author");

float price = resultSet.getFloat("price");

book = new Book(id, title, author, price);

}

resultSet.close();

statement.close();

return book;

}

}

**5.Controller class**

package com.controller;

import java.io.IOException;

import java.sql.SQLException;

import java.util.List;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.dao.BookDAO;

import com.model.Book;

public class ControllerServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

private BookDAO bookDAO;

public void init() {

String jdbcURL = getServletContext().getInitParameter("jdbcURL");

String jdbcUsername = getServletContext().getInitParameter("jdbcUsername");

String jdbcPassword = getServletContext().getInitParameter("jdbcPassword");

bookDAO = new BookDAO(jdbcURL, jdbcUsername, jdbcPassword);

}

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

doGet(request, response);

}

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

String action = request.getServletPath();

try {

switch (action) {

case "/new":

showNewForm(request, response);

break;

case "/insert":

insertBook(request, response);

break;

case "/delete":

deleteBook(request, response);

break;

case "/edit":

showEditForm(request, response);

break;

case "/update":

updateBook(request, response);

break;

default:

listBook(request, response);

break;

}

} catch (SQLException ex) {

throw new ServletException(ex);

}

}

private void listBook(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException, ServletException {

List<Book> listBook = bookDAO.listAllBooks();

request.setAttribute("listBook", listBook);

RequestDispatcher dispatcher = request.getRequestDispatcher("BookList.jsp");

dispatcher.forward(request, response);

}

private void showNewForm(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

RequestDispatcher dispatcher = request.getRequestDispatcher("BookForm.jsp");

dispatcher.forward(request, response);

}

private void showEditForm(HttpServletRequest request, HttpServletResponse response)

throws SQLException, ServletException, IOException {

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

Book existingBook = bookDAO.getBook(id);

RequestDispatcher dispatcher = request.getRequestDispatcher("BookForm.jsp");

request.setAttribute("book", existingBook);

dispatcher.forward(request, response);

}

private void insertBook(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

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

float price = Float.parseFloat(request.getParameter("price"));

Book newBook = new Book(title, author, price);

bookDAO.insertBook(newBook);

response.sendRedirect("list");

}

private void updateBook(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

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

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

float price = Float.parseFloat(request.getParameter("price"));

Book book = new Book(id, title, author, price);

bookDAO.updateBook(book);

response.sendRedirect("list");

}

private void deleteBook(HttpServletRequest request, HttpServletResponse response)

throws SQLException, IOException {

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

Book book = new Book(id);

bookDAO.deleteBook(book);

response.sendRedirect("list");

}

}

6.Views BookForm.jsp

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

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

<html>

<head>

<title>Books Store Application</title>

</head>

<body>

<center>

<h1>Books Management</h1>

<h2>

<a href="/crud/new">Add New Book</a>

&nbsp;&nbsp;&nbsp;

<a href="/crud/list">List All Books</a>

</h2>

</center>

<div align="center">

<c:if test="${book != null}">

<form action="update" method="post">

</c:if>

<c:if test="${book == null}">

<form action="insert" method="post">

</c:if>

<table border="1" cellpadding="5">

<caption>

<h2>

<c:if test="${book != null}">

Edit Book

</c:if>

<c:if test="${book == null}">

Add New Book

</c:if>

</h2>

</caption>

<c:if test="${book != null}">

<input type="hidden" name="id" value="<c:out value='${book.id}' />" />

</c:if>

<tr>

<th>Title: </th>

<td>

<input type="text" name="title" size="45"

value="<c:out value='${book.title}' />"

/>

</td>

</tr>

<tr>

<th>Author: </th>

<td>

<input type="text" name="author" size="45"

value="<c:out value='${book.author}' />"

/>

</td>

</tr>

<tr>

<th>Price: </th>

<td>

<input type="text" name="price" size="5"

value="<c:out value='${book.price}' />"

/>

</td>

</tr>

<tr>

<td colspan="2" align="center">

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

</td>

</tr>

</table>

</form>

</div>

</body>

</html>

7.Views BookList.jsp

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

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

<html>

<head>

<title>Books Store Application</title>

</head>

<body>

<center>

<h1>Books Management</h1>

<h2>

<a href="/crud/new">Add New Book</a>

&nbsp;&nbsp;&nbsp;

<a href="/crud/list">List All Books</a>

</h2>

</center>

<div align="center">

<table border="1" cellpadding="5">

<caption><h2>List of Books</h2></caption>

<tr>

<th>ID</th>

<th>Title</th>

<th>Author</th>

<th>Price</th>

<th>Actions</th>

</tr>

<c:forEach var="book" items="${listBook}">

<tr>

<td><c:out value="${book.id}" /></td>

<td><c:out value="${book.title}" /></td>

<td><c:out value="${book.author}" /></td>

<td><c:out value="${book.price}" /></td>

<td>

<a href="/crud/edit?id=<c:out value='${book.id}' />">Edit</a>

&nbsp;&nbsp;&nbsp;&nbsp;

<a href="/crud/delete?id=<c:out value='${book.id}' />">Delete</a>

</td>

</tr>

</c:forEach>

</table>

</div>

</body>

</html>

**8.Error.jsp**

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8" isErrorPage="true" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<title>Error</title>

</head>

<body>

<center>

<h1>Error</h1>

<h2><%=exception.getMessage() %><br/> </h2>

</center>

</body>

</html>

**9.web.xml**

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns="http://xmlns.jcp.org/xml/ns/javaee"

xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee

http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"

id="WebApp\_ID" version="3.1">

<display-name>Books Management Web Application</display-name>

<context-param>

<param-name>jdbcURL</param-name>

<param-value>jdbc:mysql://localhost:3306/Bookstore</param-value>

</context-param>

<context-param>

<param-name>jdbcUsername</param-name>

<param-value>root</param-value>

</context-param>

<context-param>

<param-name>jdbcPassword</param-name>

<param-value>root</param-value>

</context-param>

<servlet>

<servlet-name>ControllerServlet</servlet-name>

<servlet-class>com.controller.ControllerServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>ControllerServlet</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

<!-- any exception will forward to error.jsp-->

**<error-page>**

**<exception-type>java.lang.Exception</exception-type>**

**<location>/Error.jsp</location>**

**</error-page>**

</web-app>

10. Database setup

CREATE DATABASE 'Bookstore';

USE Bookstore;

CREATE TABLE `book` (

`book\_id` int(11) NOT NULL AUTO\_INCREMENT,

`title` varchar(128) NOT NULL,

`author` varchar(45) NOT NULL,

`price` float NOT NULL,

PRIMARY KEY (`book\_id`),

UNIQUE KEY `book\_id\_UNIQUE` (`book\_id`),

UNIQUE KEY `title\_UNIQUE` (`title`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=latin1

Source : http://www.codejava.net/coding/jsp-servlet-jdbc-mysql-create-read-update-delete-crud-example