MVC CRUD USING JSP, SERVLET AND MYSQL

You can view and download the complete source code of this tutorial from my github account.
In this tutorial, we will create a simple CRUD (Create Read Update Delete) User Management Web Application using Jsp, Servlet and MySQL.
For this tutorial, we will need the following tools: (The older or newer version should also works). Moreover, basic Java knowledge is assumed.
1. Eclipse IDE for Java EE Developers (Indigo – ver. 3.7)
2. Apache Tomcat ver 7.0
3. MySQL Community Server and MySQL Workbench (GUI Tool)
4. MySQL Connector for Java
5. jstl.jar and standard.jar. You can get these jars from your Tomcat. Check in this directory : (your tomcat directory)—>apache-tomcat-7.0.26-windows-x86—>apache-tomcat-7.0.26—>webapps—>examples—>WEB-INF—>lib
I will tell you where you should put these jars later.
6. jQuery for javascript capability. In this case, we only use it for the datepicker component

---

First, lets create the database and table for User using the following SQL scripts:

```
create database UserDB;
use UserDB;
grant all on UserDB.* to 'admin'@'localhost' identified by 'test';

CREATE TABLE UserDB.`users` (
  `userid` int(11) NOT NULL AUTO_INCREMENT,
  `firstname` varchar(45) DEFAULT NULL,
  `lastname` varchar(45) DEFAULT NULL,
  `dob` date DEFAULT NULL,
  `email` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`userid`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8
```

Go to eclipse. Before we create a new project for our application, we need to setup the server. Select File—>New—>Other. From the tree, Select Server.
Choose Apache—>Tomcat v7.0 Server and set the runtime environment.
Next, create a new project. Select File—>New—>Dynamic Web Project.
Enter “SimpleJspServletDB” as the project name. Select target runtime to Apache Tomcat v7.0 which we already setup before. Click Finish.
Please refer to this project directory in case you miss something along the way

Copy the standard.jar, mysql-connector jar and jstl jar to WEB-INF—>lib folder.
Create four packages in the src folder.

- **com.daniel.controller**: contains the servlets
- **com.daniel.dao**: contains the logic for database operation
- **com.daniel.model**: contains the POJO (Plain Old Java Object). Each class in this package represents the database table. For this tutorial, however, we only have one table.
- **com.daniel.util** : contains the class for initiating database connection

Next, create a new Java class. in **com.daniel.model** folder. Name it “**User.java**” and insert these following codes. Each of the variables in this class represents the field in USERS table in our database.

```
package com.daniel.model;

import java.util.Date;

public class User {

    private int userid;
    private String firstName;
    private String lastName;
    private Date dob;
    private String email;
    public int getUserid() {
        return userid;
    }
    public void setUserid(int userid) {
        this.userid = userid;
    }
    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 Date getDob() {
        return dob;
    }
    public void setDob(Date dob) {
        this.dob = dob;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    @Override
    public String toString() {
        return "User [userid=" + userid + ", firstName=" + firstName
                + ", lastName=" + lastName + ", dob=" + dob + ", email="
                + email + "]";
    }
}
```

Create a new class in **com.daniel.util** package and name it **DbUtil.java**. This class handles the database connection to our MySQL server. In this class, we read a .properties file which contains the information necessary for the connection.

```
package com.daniel.util;

import java.io.FileNotFoundException;
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 connection = null;

    public static Connection getConnection() {
        if (connection != null)
            return connection;
        else {
            try {
                Properties prop = new Properties();
                InputStream              inputStream                =
DbUtil.class.getClassLoader().getResourceAsStream("/db.properties");
                prop.load(inputStream);
                String driver = prop.getProperty("driver");
                String url = prop.getProperty("url");
                String user = prop.getProperty("user");
                String password = prop.getProperty("password");
                Class.forName(driver);
                connection = DriverManager.getConnection(url, user,
password);
            } catch (ClassNotFoundException e) {
                e.printStackTrace();
            } catch (SQLException e) {
                e.printStackTrace();
            } catch (FileNotFoundException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            }
            return connection;
        }

    }
}
```

Create the properties file directly under the **src** folder. Create a new file, name it**db.properties.** Put the following information inside.

```
driver=com.mysql.jdbc.Driver
url=jdbc:mysql://localhost:3306/UserDB
user=admin
password=test
```

Next, create a new class in **com.daniel.dao** package, name it **UserDao.java.** Dao stands for Data Access Object. It contains the logic for database operation.

```
package com.daniel.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.daniel.model.User;
import com.daniel.util.DbUtil;

public class UserDao {

    private Connection connection;

    public UserDao() {
        connection = DbUtil.getConnection();
    }

    public void addUser(User user) {
        try {
            PreparedStatement preparedStatement = connection
                    .prepareStatement("insert                              into
users(firstname,lastname,dob,email) values (?, ?, ?, ? )");
            // Parameters start with 1
            preparedStatement.setString(1, user.getFirstName());
            preparedStatement.setString(2, user.getLastName());
            preparedStatement.setDate(3,                                  new
java.sql.Date(user.getDob().getTime()));
            preparedStatement.setString(4, user.getEmail());
            preparedStatement.executeUpdate();

        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public void deleteUser(int userId) {
        try {
            PreparedStatement preparedStatement = connection
                    .prepareStatement("delete from users where userid=?");
            // Parameters start with 1
            preparedStatement.setInt(1, userId);
            preparedStatement.executeUpdate();

        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public void updateUser(User user) {
        try {
            PreparedStatement preparedStatement = connection
                    .prepareStatement("update users set firstname=?, lastname=?,
dob=?, email=?" +
                            "where userid=?");
            // Parameters start with 1
            preparedStatement.setString(1, user.getFirstName());
            preparedStatement.setString(2, user.getLastName());
            preparedStatement.setDate(3,                         new
java.sql.Date(user.getDob().getTime()));
            preparedStatement.setString(4, user.getEmail());
            preparedStatement.setInt(5, user.getUserid());
            preparedStatement.executeUpdate();

        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public List<User> getAllUsers() {
        List<User> users = new ArrayList<User>();
        try {
            Statement statement = connection.createStatement();
            ResultSet rs = statement.executeQuery("select * from users");
            while (rs.next()) {
                User user = new User();
                user.setUserid(rs.getInt("userid"));
                user.setFirstName(rs.getString("firstname"));
                user.setLastName(rs.getString("lastname"));
                user.setDob(rs.getDate("dob"));
                user.setEmail(rs.getString("email"));
                users.add(user);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }

        return users;
    }

    public User getUserById(int userId) {
        User user = new User();
        try {
            PreparedStatement preparedStatement = connection.
                    prepareStatement("select * from users where userid=?");
            preparedStatement.setInt(1, userId);
            ResultSet rs = preparedStatement.executeQuery();

            if (rs.next()) {
                user.setUserid(rs.getInt("userid"));
                user.setFirstName(rs.getString("firstname"));
                user.setLastName(rs.getString("lastname"));
                user.setDob(rs.getDate("dob"));
                user.setEmail(rs.getString("email"));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }

        return user;
    }
```

```
}
```

Finally, create a new **Servlet** inside the **com.daniel.controller** package and name it **UserController.java**

```
package com.daniel.controller;

import java.io.IOException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;

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.daniel.dao.UserDao;
import com.daniel.model.User;

public class UserController extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private static String INSERT_OR_EDIT = "/user.jsp";
    private static String LIST_USER = "/listUser.jsp";
    private UserDao dao;

    public UserController() {
        super();
        dao = new UserDao();
    }

    protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException
{
        String forward="";
        String action = request.getParameter("action");

        if (action.equalsIgnoreCase("delete")){
            int userId =
Integer.parseInt(request.getParameter("userId"));
            dao.deleteUser(userId);
            forward = LIST_USER;
            request.setAttribute("users", dao.getAllUsers());
        } else if (action.equalsIgnoreCase("edit")){
            forward = INSERT_OR_EDIT;
            int userId =
Integer.parseInt(request.getParameter("userId"));
            User user = dao.getUserById(userId);
            request.setAttribute("user", user);
        } else if (action.equalsIgnoreCase("listUser")){
            forward = LIST_USER;
            request.setAttribute("users", dao.getAllUsers());
        } else {
            forward = INSERT_OR_EDIT;
        }
        RequestDispatcher view =
request.getRequestDispatcher(forward);
        view.forward(request, response);
    }

    protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException
{
        User user = new User();
        user.setFirstName(request.getParameter("firstName"));
        user.setLastName(request.getParameter("lastName"));
        try {
            Date dob = new
SimpleDateFormat("MM/dd/yyyy").parse(request.getParameter("dob"));
            user.setDob(dob);
        } catch (ParseException e) {
            e.printStackTrace();
        }
        user.setEmail(request.getParameter("email"));
        String userid = request.getParameter("userid");
        if(userid == null || userid.isEmpty())
        {
            dao.addUser(user);
        }
        else
        {
            user.setUserid(Integer.parseInt(userid));
            dao.updateUser(user);
        }
        RequestDispatcher view =
request.getRequestDispatcher(LIST_USER);
        request.setAttribute("users", dao.getAllUsers());
        view.forward(request, response);
    }
}
```

Now, it’s time for us to create the jsp, the view for our application. Under the **WebContent**folder, create a jsp file, name it **index.jsp**

```
<%@ page language="java" contentType="text/html; charset=EUC-KR" pageEncoding="EUC-
KR"%>
<!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=EUC-KR">
<title>Insert title here</title>
</head>
<body>
<jsp:forward page="/UserController?action=listUser" />
</body>
</html>
```

This jsp serves as the entry point for our application. In this case, it will redirect the request to our servlet to list all the users in the database.

Next, create the jsp to list all the users in the **WebContent** folder. Name it **listUser.jsp**

```
<%@ page language="java" contentType="text/html; charset=EUC-KR" pageEncoding="EUC-
KR"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>
<!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=EUC-KR">
<title>Show All Users</title>
</head>
<body>
    <table border=1>
        <thead>
            <tr>
                <th>User Id</th>
                <th>First Name</th>
                <th>Last Name</th>
                <th>DOB</th>
                <th>Email</th>
                <th colspan=2>Action</th>
            </tr>
        </thead>
        <tbody>
            <c:forEach items="${users}" var="user">
                <tr>
                    <td><c:out value="${user.userid}" /></td>
                    <td><c:out value="${user.firstName}" /></td>
                    <td><c:out value="${user.lastName}" /></td>
                    <td><fmt:formatDate pattern="yyyy-MMM-dd"
value="${user.dob}" /></td>
                    <td><c:out value="${user.email}" /></td>
                    <td><a href="UserController?action=edit&userId=<c:out
value="${user.userid}"/>">Update</a></td>
                    <td><a href="UserController?action=delete&userId=<c:out
value="${user.userid}"/>">Delete</a></td>
                </tr>
            </c:forEach>
        </tbody>
    </table>
    <p><a href="UserController?action=insert">Add User</a></p>
</body>
</html>
```

In this jsp, we use JSTL to connect between the jsp and the servlet. We should refrain from using scriplet inside the jsp because it will make the jsp more difficult to maintain. Not to mention it will make the jsp looks ugly.

Next, create a new jsp in **WebContent** folder and name it **user.jsp**

```
<%@ page language="java" contentType="text/html; charset=EUC-KR" pageEncoding="EUC-
KR"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>
```

```
<!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=EUC-KR">
<link type="text/css"
    href="css/ui-lightness/jquery-ui-1.8.18.custom.css" rel="stylesheet" />
<script type="text/javascript" src="js/jquery-1.7.1.min.js"></script>
<script type="text/javascript" src="js/jquery-ui-1.8.18.custom.min.js"></script>
<title>Add new user</title>
</head>
<body>
    <script>
        $(function() {
            $('input[name=dob]').datepicker();
        });
    </script>

    <form method="POST" action='UserController' name="frmAddUser">
        User ID : <input type="text" readonly="readonly" name="userid"
            value="<c:out value="${user.userid}" />" /> <br />
        First Name : <input
            type="text" name="firstName"
            value="<c:out value="${user.firstName}" />" /> <br />
        Last Name : <input
            type="text" name="lastName"
            value="<c:out value="${user.lastName}" />" /> <br />
        DOB : <input
            type="text" name="dob"
            value="<fmt:formatDate pattern="MM/dd/yyyy" value="${user.dob}" />"
/> <br />
        Email : <input type="text" name="email"
            value="<c:out value="${user.email}" />" /> <br /> <input
            type="submit" value="Submit" />
    </form>
</body>
</html>
```

Lastly, check the web.xml file located in WebContent—>WEB-INF folder in your project structure. Make sure it looks like this

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/javaee/web-
app_2_5.xsd" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" id="WebApp_ID" version="2.5">
  <display-name>SimpleJspServletDB</display-name>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
  <servlet>
    <description></description>
    <display-name>UserController</display-name>
    <servlet-name>UserController</servlet-name>
    <servlet-class>com.daniel.controller.UserController</servlet-class>
  </servlet>
```

```
    <servlet-mapping>
        <servlet-name>UserController</servlet-name>
        <url-pattern>/UserController</url-pattern>
    </servlet-mapping>
</web-app>
```

That is it. Right click the project name and run it using Run As–>Run on server option.