

课程名称: JavaEE 技术	年级: XXXX	实验日期:
姓名: XXX	学号: XXXXX	班级: XXXXX
实验名称: Servlet 编程基础	实验序号: 实验二	成员人数: 1

一、实验目的及要求

实验目的:

- 1、了解 servlet 的生命周期;
- 2、掌握 servlet 如何接收和相应 http 请求;
- 3、servlet 如何处理页面跳转时数据的传送;

实验原理:

- 1、servlet 独立于服务器和用户平台,使得 servlet 可以移植并可以重用。
- 2、servlet 是实现 javax.servlet.Servlet 接口类的一个实例。
- 3、servlet 会驻留在服务器内存中,可以为某种目的委托服务器执行特定的功能。

二、实验环境(仪器与材料)

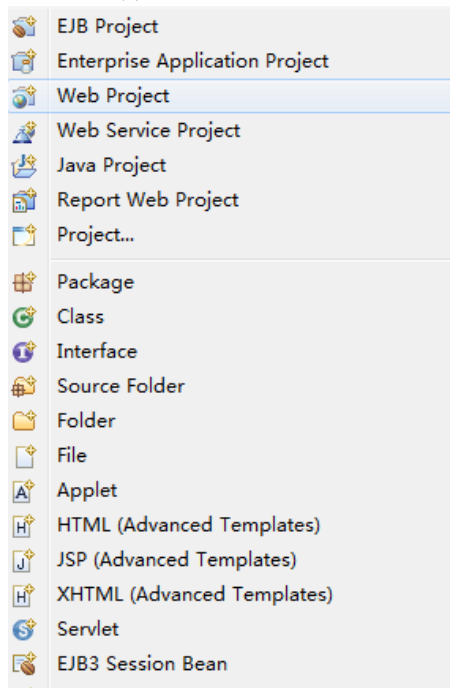
Windows 7, Microsoft SQL Server 2008, MyEclipse, TomCat

三、实验内容及完成情况

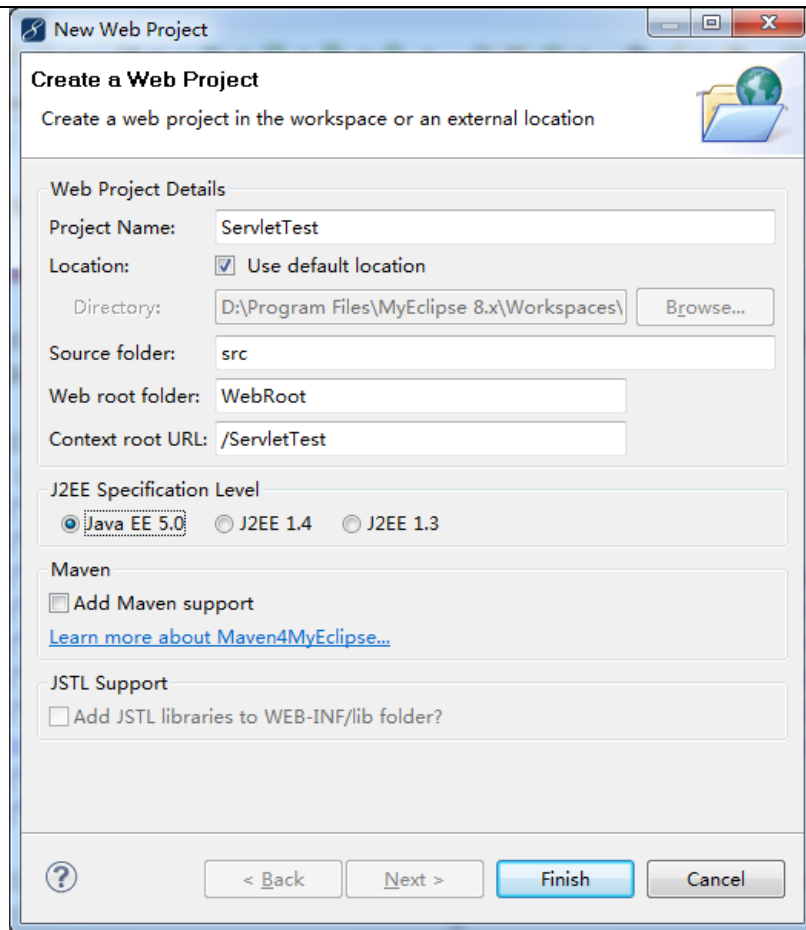
- 1、基础数据库为实验一所用数据库
- 2、编写 servlet 程序,验证用户是否为合法的用户,若是非法用户提示出错,重新登录,合法用户,则登录到主页面,并显示当前登录用户名。

步骤如下:

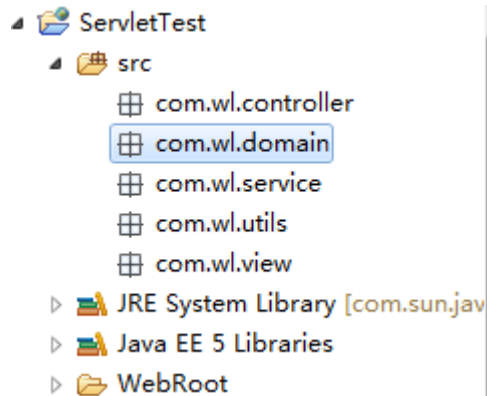
- (1) file->new-> Web Project 点击文件,新建 web 项目



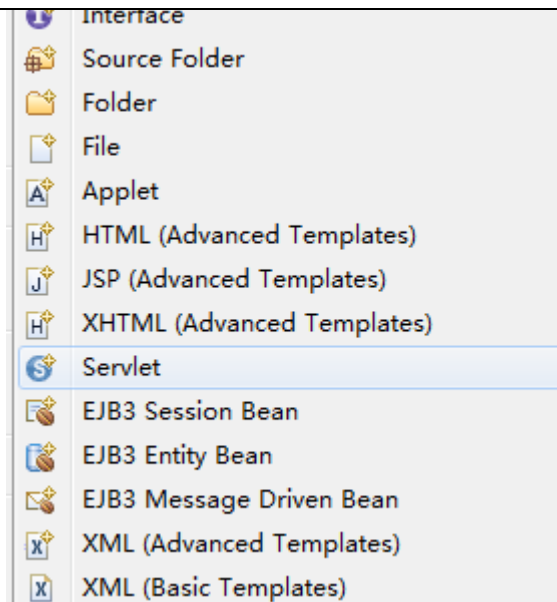
- (2) 2、命名: ServletTest



(3) 3、新建如下几个包



(4) 在 com.wl.view 包下新建 servlet



(5) 代码如下:

```
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class Login extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    response.setContentType("text/html;charset=utf-8");
    PrintWriter out = response.getWriter();
    out.println("<center>");
    out.println("<h1>用户登录</h1>");
    out.println("<form action='/ServletTest/LoginServlet'
method='post'>");
    out.println("<table border='1' cellspacing='0'
bordercolor='green'>");
    out.println("<tr><td>用户名: </td><td><input type='text'
name='username' /></td></tr>");
    out.println("<tr><td>密 码: </td><td><input
type='password' name='password' /></td></tr>");
    out.println("<tr><td colspan='2'><center><input
type='submit' value='登录' /> +
        "&nbsp;&nbsp;&nbsp;<input type='reset' value='清空
' /></center></td></tr>");
    out.println("</table>");
    out.println("</form>");

    out.println("</center>");
}
public void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
    this.doGet(request, response);
}
}
```

(6) 项目部署、发布

(7) 在浏览器的地址栏中输入: <http://localhost:8080/ServletTest/Login>

(8) 观察运行结果:

????

???	<input type="text"/>
???	<input type="text"/>
<input type="button" value="??"/> <input type="button" value="??"/>	

(9) 解决乱码问题

```
response.setContentType("text/html;charset=utf-8");
```

(10) 重新运行结果:

用户登录

用户名:	<input type="text"/>
密 码:	<input type="text"/>
<input type="button" value="登录"/> <input type="button" value="清空"/>	

(11) 在工具包下放入工具类, 将配置文件放入 src 文件夹下面; 将 sql 数据库的驱动放置在 WebRoot/WEB-INF/lib 文件夹下面 (非常重要)

(12) 在 com.wl.view 下面创建 servlet, 命名为 Main
代码如下:

```
package com.wl.view;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class Main extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=utf-8");
        PrintWriter out = response.getWriter();
        out.println("<h1>登录成功, 主页面! </h1>");
        String username=(String) request.getAttribute("userinfo");
        out.println("欢迎"+username+"登录! ");
    }
    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        this.doGet(request, response);
    }
}
```

(13) 编写 UserService 服务类

```
package com.wl.service;
import java.sql.ResultSet;
import java.sql.SQLException;
import com.wl.utils.SQLHelper;
```

```

public class UserService {
public boolean checkUser(String username,String password){
    boolean b=false;
    String sql="select * from TUserLogin where username=? and
password=?";
    String []parameters={username,password};
    ResultSet rs=SQLHelper.executeQuery(sql, parameters);
    try {
        if (rs.next()){
            b=true;
        }else{
            b=false;
        }
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }finally{
        SQLHelper.close(rs, SQLHelper.getPs(),
SQLHelper.getCt());
    }
    return b;
}
}

```

(14) 编写登录的控制器 LoginServlet

```

package com.wl.controller;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.wl.service.UserService;
public class LoginServlet extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    response.setContentType("text/html;charset=utf-8");
    PrintWriter out = response.getWriter();
    //获取登录页面传递过来的参数
    String username=request.getParameter("username");
    String password=request.getParameter("password");
    //创建用户服务类对象
    UserService userService=new UserService();
    if(userService.checkUser(username, password)){
        //判断条件为真，则页面跳转到主页面，登录成功
        //跳转前，把用户姓名放置在attribute域中
        request.setAttribute("userinfo", username);
        request.getRequestDispatcher("/Main").forward(request,
response);
    }else{
        //跳转前，把错误信息放置在attribute域中
        request.setAttribute("error", "错误的用户名或密码");
        request.getRequestDispatcher("/Login").forward(request,

```

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

(15) 运行结果

输入一个不存在的用户名或者密码

用户登录

用户名:	<input type="text" value="aaa"/>
密 码:	<input type="password" value="..."/>
<input type="button" value="登录"/> <input type="button" value="清空"/>	

点击登录

用户登录

用户名:	<input type="text"/>
密 码:	<input type="password"/>
<input type="button" value="登录"/> <input type="button" value="清空"/>	

错误的用户名或密码

页面重新定位到登录界面，并显示出错信息

输入正确的用户名和密码，wl1, 123456，则跳转到主页面，并显示当前登录用户名

登录成功，主页面！

欢迎wl1登录！

四、 出现的问题及解决方案

- 1、没有引入 sqljdbc4.jar 驱动，没有把配置文件放置在正确的路径下。
- 2、中文乱码问题要注意。
- 3、页面跳转可以使用重定向和请求转发两种方式，建议使用:请求转发方式，可以把需要带过去的信息放置在 request 的 Attribute 域中。
- 4、注意用户业务逻辑服务类的写法。数据库中的每一张表都要对应一个服务类。负责处理该表的所有业务逻辑。
- 5、没有使用 JavaBean，可以进一步改进。