

天津理工大学实验报告

学院（系）名称：计算机科学与工程学院

姓名	王帆	学号	20152180	专业	计算机科学与技术
班级	2015 级 1 班	实验项目	JSP、Servlet 及 JDBC 应用开发		
课程名称		Web 应用程序设计与开发		课程代码	0668026
实验时间		2018 年 5 月 22 日 中午 5、6 节		实验地点	7-215
考核标准	实验准备（实验目的/工具熟悉情况）10 分	实验过程（实验方案可行性及步骤完整性）40 分	实验报告（实验内容丰富度与格式清晰度）30 分	实验结果（结论正确性以及分析合理性）20 分	成绩
考核内容	评价实验目的是否明确，实验工具是否清晰了解以及熟悉情况	<input type="radio"/> 可行，完整 <input type="radio"/> 可行，不完整 <input type="radio"/> 不可行，不完整	<input type="radio"/> 丰富，清晰 <input type="radio"/> 较丰富，较清晰 <input type="radio"/> 丰富，不清晰 <input type="radio"/> 不丰富，不清晰	<input type="radio"/> 结论正确，分析合理 <input type="radio"/> 结论正确，分析不充分 <input type="radio"/> 结论不正确，分析不合理	教师签字：

一、实验目的

掌握使用 JDBC 访问数据库，学会使用 JSP 的内置对象的使用，学会 Servlet 和 JavaBean 等技术的应用。

二、实验环境

Windows 操作系统，Tomcat，MyEclipse，HBuilder，记事本。

三、实验要求

1. 实现系统的用户登录，要求：

- （1）创建登录表，并利用此表实现系统的登录功能；
- （2）登录成功要求能够保存或维护用户信息，以便能够被其他界面使用用户信息。

2. 实现对单表的信息的添加功能，要求：

(1) 建立数据库的表结构，至少包含 4 个以上字段；

(2) 利用表单的提交功能。

(3) 利用 Servlet 处理，充分考虑编码及封装。

3. 对数据库的表的信息查询功能，要求：

(1) 利用 JavaBean 实现，并将表中的信息按行进行显示。

(2) 建立数据库的表结构，至少包含 4 个以上字段；表的信息可以是学生信息、图书信息、商品信息等。

(3) 实现时充分考虑代码的封装。

四、实验过程记录（源程序、测试用例、测试结果及心得体会等）

1. 实现系统的用户登录

(1) 创建登录表，并利用此表实现系统的登录功能；。

表结构如下：

名称	数据类型
userid	INT
username	VARCHAR
password	VARCHAR

(2) 登录成功要求能够保存或维护用户信息，以便能够被其他界面使用用户信息。

在 Servlet 中使用 Session 对象对登录状态进行存储，以便在其他界面调用验证
后端代码：

```
response.setContentType("text/html;charset=utf-8");
String username = WebUtil.getParameter(request,"username");
String password = WebUtil.getParameter(request,"password");
PrintWriter out = response.getWriter();

HashMap<String, String> hashMap = new HashMap<String, String>();
hashMap.put("username", username);
hashMap.put("password", password);

if(DBUtil.getDataCount("user",hashMap)==1) {
    HttpSession session = request.getSession();
    session.setAttribute("username",username); //用户名
    session.setAttribute("loginState","1"); //登录状态
    response.sendRedirect("./loginCheck.jsp");
}
else {
    HttpSession session = request.getSession();
```

```
session.setAttribute("loginState","0"); //登录状态
response.sendRedirect("./loginCheck.jsp");
```

}

演示：



图 1-1 登录前提示并自动跳转

2. 实现对单表的信息的添加功能

(1) 建立数据库的表结构

表结构如下：

名称	数据类型
sno	INT
sname	VARCHAR
age	VARCHAR
politicalstate	VARCHAR
birthday	VARCHAR
address	VARCHAR
phone	VARCHAR
institute	VARCHAR
demo	VARCHAR

(2) 利用表单的提交功能。

3. 对数据库的表的信息查询功能，要求：

(1) 利用 JavaBean 实现，并将表中的信息按行进行显示。

(2) 建立数据库的表结构，至少包含 4 个以上字段；表的信息可以是学生信息、图书信息、商品信息等。

(3) 实现时充分考虑代码的封装。

前端代码：

```
<form action="${pageContext.request.contextPath}/FormServlet" method="post">
```

```
    <table style="width: auto;" class=".table-bordered table table-striped table-  
hover table-bordered table table-condensed">
```

```
        <tr>
```

```
            <th>Key</th>
```

```
            <th>Value</th>
```

```
        </tr>
```

```
        <tr>
```

```
            <td>sno</td>
```

```
            <td>
```

```
                <input type="text" name="sno" id="sno">
```

```
            </td>
```

```
        </tr>
```

```
        <tr>
```

```
            <td>sname</td>
```

```
            <td>
```

```
                <input type="text" name="sname" id="sname">
```

```
            </td>
```

```
        </tr>
```

```
        <tr>
```

```
            <td>age</td>
```

```
            <td>
```

```
                <input type="text" name="age" id="age">
```

```
            </td>
```

```
        </tr>
```

```
        <tr>
```

```
            <td>politicalstate</td>
```

```
            <td>
```

```
                <input type="text" name="politicalstate" id="politicalstate">
```

```
            </td>
```

```
        </tr>
```

```
        <tr>
```

```
            <td>birthday</td>
```

```
            <td>
```

```
                <input type="text" name="birthday" id="birthday">
```

```
            </td>
```

```
        </tr>
```

```
        <tr>
```

```
            <td>address</td>
```

```
            <td>
```

```
                <input type="text" name="address" id="address">
```

```

        </td>
    </tr>
    <tr>
        <td>phone</td>
        <td>
            <input type="text" name="phone" id="phone">
        </td>
    </tr>
    <tr>
        <td>institute</td>
        <td>
            <input type="text" name="institute" id="institute">
        </td>
    </tr>
    <tr>
        <td>demo</td>
        <td>
            <input type="text" name="demo" id="demo">
        </td>
    </tr>
</table>
<div class="form-actions">
    <button type="submit" class="btn btn-primary">提交</button>
    <button type="button" class="btn">重置</button>
</div>
</form>

```

(3) 利用 Servlet 处理，充分考虑编码及封装。

后端代码：

```

request.setCharacterEncoding("UTF-8");
HttpSession session = request.getSession();
String op = request.getParameter("op");
System.out.print(op);
switch (op) {
    case "2Bean":
        response.setContentType("text/html;charset=utf-8");

        Student student_bean = new Student();
        student_bean.setAddress(request.getParameter("address"));
        student_bean.setAge(request.getParameter("age"));
        student_bean.setBirthday(request.getParameter("birthday"));
        student_bean.setDemo(request.getParameter("demo"));
        student_bean.setInstitute(request.getParameter("institute"));
        student_bean.setPhone(request.getParameter("phone"));

```

```
student_bean.setName(request.getParameter("sname"));
student_bean.setSno(Integer.parseInt(request.getParameter("sno")));
student_bean.setPoliticalstate(request.getParameter("politicalstate"));
```

```
session.setAttribute("studentBean", student_bean);
response.sendRedirect("./FormJavabean.jsp");
break;
```

case "2DB":

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

```
Student student_db = new Student();
System.out.println("OK2");
student_db.setAddress(request.getParameter("address"));
student_db.setAge(request.getParameter("age"));
student_db.setBirthday(request.getParameter("birthday"));
student_db.setDemo(request.getParameter("demo"));
student_db.setInstitute(request.getParameter("institute"));
student_db.setPhone(request.getParameter("phone"));
student_db.setName(request.getParameter("sname"));
student_db.setSno(Integer.parseInt(request.getParameter("sno")));
student_db.setPoliticalstate(request.getParameter("politicalstate"));
```

```
String sql = "INSERT INTO
student(sno,sname,age,politicalstate,birthday,address,phone,institute,demo) VALUES('";
sql+=student_db.getSno()+"', '"+student_db.getName()+"', '"+student_db.getAge()+"', '"+
student_db.getPoliticalstate()+"', '"+student_db.getBirthday()+"', '"+student_db.getAddre
ss()+"', '"+student_db.getPhone()+"', '"+student_db.getInstitute()+"', '"+student_db.getDem
o()+"'";
```

```
if(DBUtil.executeBatch(sql)){
    PrintWriter outPrintWriter = response.getWriter();
    outPrintWriter.println("表更新成功! ");
    System.out.println("表更新成功! ");
}
else{
    PrintWriter outPrintWriter = response.getWriter();
    outPrintWriter.println("表更新失败! ");
    System.out.println("表更新失败! ");
}
```

```
break;
```

default:

```
session.setAttribute("error", "参数错误");
```

```

        response.sendRedirect("./errorHandle.jsp");
        break;
    }
}

private void insertStudentInfo(HttpServletRequest response, String sno) {
    try {
        String string = "";
        JSONObject jsonObject = new JSONObject();
        ArrayList<String> List = new ArrayList<>();
        String delete = "INSERT INTO student VALUES(?,?,?,?,?,?,?,?,?)";
        List.add(delete);
        boolean result = DBUtil.executeBatch(List);
        PrintWriter out = response.getWriter();
        response.setCharacterEncoding("utf-8");
        if(result){
            jsonObject.put("ret", "1");
        }
        else {
            jsonObject.put("ret", "0");
            jsonObject.put("reason", "数据库操作失败");
        }
        out.print(jsonObject);
        out.close();
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}
}

```

演示:

Key	Value
sno	20152155
sname	王航w
age	21
politicalstate	共青团员
birthday	1997-04-18
address	天津理工大学
phone	13137127027
institute	计算机科学与工程学院
demo	学习委员

图 2-1 表单录入

Key	Value
sno	20152100
sname	王明d
age	21
politicalstate	共青团员
birthday	1997-04-18
address	天津理工大学
phone	13137127027
institute	计算机科学与工程学院
demo	学习委员

图 2-2 表单确认

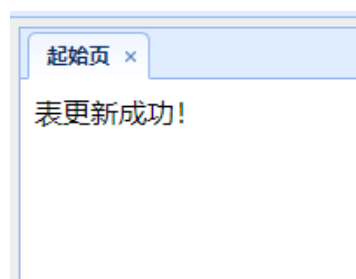


图 2-3 表单更新成功提醒

心得体会:

通过本次实验，我加深了对 JSP/Servlet 以及 JDBC 开发的理解。JSP 全名为 Java Server Pages，中文名叫 java 服务器页面，其根本是一个简化的 Servlet 设计。Servlet (Server Applet) 是 Java Servlet 的简称，称为小服务程序或服务连接器，用 Java 编写的服务器端程序，主要功能在于交互式地浏览和修改数据，生成动态 Web 内容。JDBC (Java DataBase Connectivity, java 数据库连接) 是一种用于执行 SQL 语句的 Java API，可以为多种关系数据库提供统一访问，它由一组用 Java 语言编写的类和接口组成。通过对以上技术的运用与实践，我可以初步做成基本的 WEB 前后端交互页面，并实现了与数据库的连接。在今后的学习中，我将继续对 WEB 应用程序进行了解与分析，实现更多功能。