



《JavaEE 实验手册》

Java 教研室

版本 1.3

文档提供：Java 教研室 孙丽萍

修 改 记 录

修改时间	修改人	修改内容
2009. 8. 29	孟双英	文档创建
2012. 8. 27	张立飞	修改
2016. 3. 7	孙丽萍	修改

目录

一、内容概述.....	4
二、实验一 利用 jsp 访问数据库实现用户登录功能	4
2.1 实验目的.....	4
2.2 准备	4
2.3 实验步骤.....	4
2.4 实验结论.....	7
三、实验二 体验用 JavaBean 和不用的区别.....	8
3.1 实验目的.....	8
3.2 准备	8
3.3 实验步骤.....	8
3.4 实验结论.....	11

第九章 JSP 进阶

一、内容概述

本章的教学内容是介绍 JSP 与 Servlet 具体的转换过程，JSP 的 9 个内置对象 application、session、request、response、out、page、pageContext、config、exception，JSP 的 4 个作用域范围由大到小为 application 范围、session 范围、request 范围、page 范围，同时介绍 JavaBean 以及 JavaBean 在 JSP、Servlet 中的应用

通过本章实验应该达到的目的：理解 JSP 中常用内置对象的功能以及应用场景，掌握 JSP 的 4 个作用域范围，掌握 JavaBean 在 JSP 中的应用，能利用 JSP 相关知识实现基本的网站登录，数据库查询功能。

二、实验一 利用 jsp 访问数据库实现用户登录功能

2.1 实验目的

用 Eclipse 工具，开发 JSP 应用程序

实现功能：利用 jsp 访问数据库实现用户登录功能

2.2 准备

在进行此实验前首先保证本地机器上配置好了正确的 Java 开发环境(JDK 环境)，已安装 Web 服务器 Tomcat，如果以上设置不正确，请参考《Tomcat 安装手册》进行正确设置

2.3 实验步骤

步骤一：

分析要实现用户登录首先数据库中要存在 Users 用户表，在 Mysql 数据库中新建数据库“03_mysql”，（参看《创建 mysql 数据库手册》）在“03_mysql”数据库中新建表 users，users 表字段及类型如下图所示

Field Name	Field Type	Size	Precision	Not Null	Unsigned	Default	Comment
id	INTEGER	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Null	
username	VARCHAR	20	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Null	
password	VARCHAR	20	0	<input type="checkbox"/>	<input type="checkbox"/>	Null	

图 2-1

向 users 表中插入以下数据

<i>id</i>	<i>username</i>	<i>password</i>
1	admin	123456
2	mengsy	123456
3	sun	123456

步骤二:

打开 Eclipse 工具, 新建 Dynamic project “03_mysql” 工程, 将 mysql 驱动包拷贝到 “03_mysql/WebContent/WEB-INF/lib” 目录下

在 “03_mysql/WebContent” 目录下新建 index.jsp, index.jsp 为显示输入用户名、密码页面, 效果图如下:

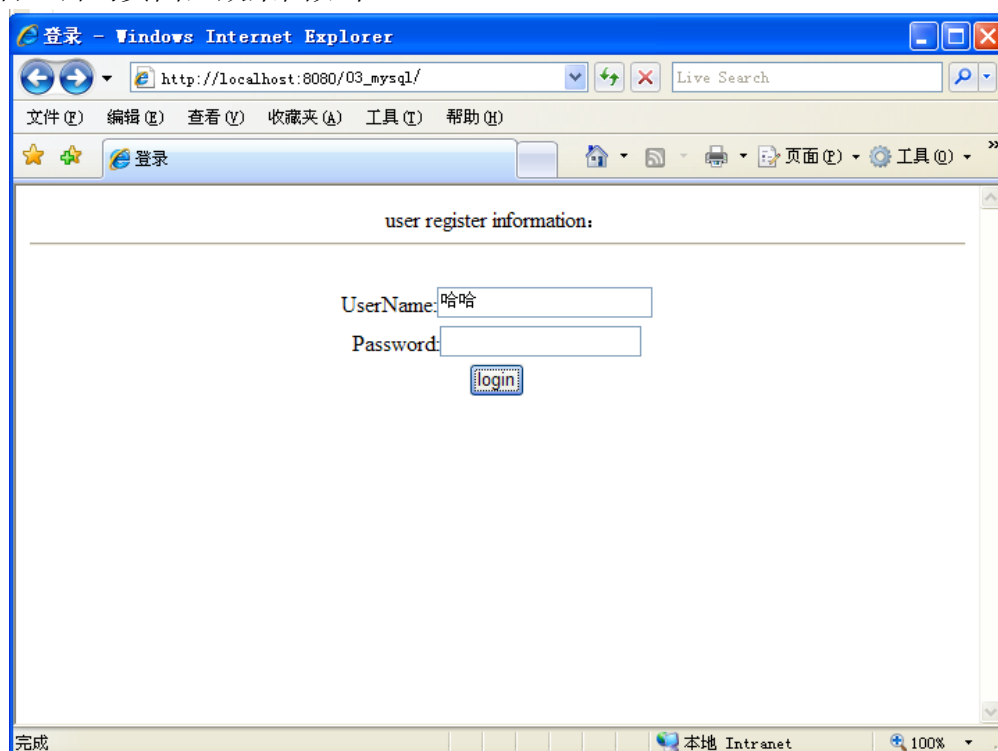


图 2-2

登录页面的<form>表单提交到 login.jsp 页面

步骤四:

在 “03_mysql/WebContent” 目录下新建 login.jsp, 编写 login.jsp 页面, 首先链接数据库, 将 index 页面提交过来的表单信息 (用户名、密码) 与数据库中的 users 表中数据进行比对, 比对成功跳转到 loginsuccess.jsp, 比对失败返回到 index.jsp 重新登录

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@page import="java.sql.*"%>
<!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=UTF-8">
<title>登录处理页面</title>
</head>
<body>
<%!String nextPage = "";%>
<%
    Connection con = null;
    PreparedStatement pre = null;
    ResultSet res = null;
    try {
        Class.forName("com.mysql.jdbc.Driver");
        con = DriverManager
            .getConnection("jdbc:mysql://localhost:3306/07-03?user=root&password=&characterEncoding=utf8");
        String sql = "select * from users where username=? and password=?";
        pre = con.prepareStatement(sql);
        pre.setString(1, request.getParameter("userName"));
        pre.setString(2, request.getParameter("password"));
        res = pre.executeQuery();
        if (res.next()) {
            session.setAttribute("user", request.getParameter("userName"));

            nextPage="loginsuccess.jsp";
        } else {
            nextPage="index.jsp";
        }
    } catch (Exception e) {
        out.println(e);
    } finally {
        try {
            if (res != null)
                res.close();
            if (pre != null)
                pre.close();
            if (con != null)
                con.close();
        } catch (Exception e) {
            out.println(e);
        }
    }
%>
<jsp:forward page="<%=nextPage %>"></jsp:forward>
</body>
</html>

```

图 2-3

步骤五:

编写 loginsuccess.jsp 页面, loginsuccess 要从数据库中查询出所有用户信息并显示, 效果如下图



图 2-4

步骤六:

如果不登录的情况下，直接在浏览器中输入

http://localhost:8080/03_mysql/loginsuccess.jsp，效果如图

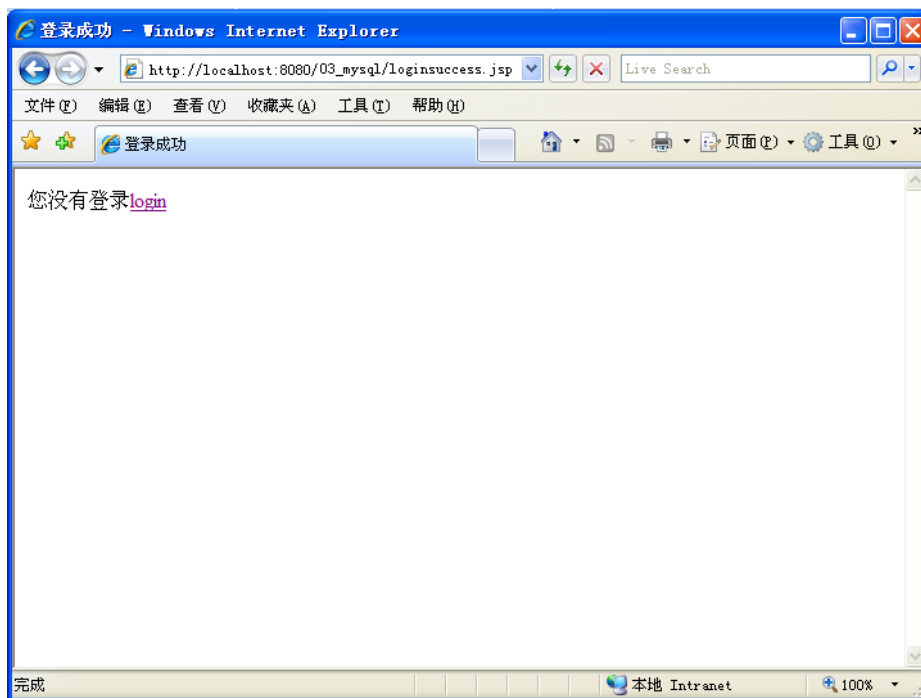


图 2-5

上述功能关键实现代码如下

```
<%  
    if (session.getAttribute("user") == null) {  
        您没有登录<a href="login.jsp">login</a>  
    } else {  
    }  
%>
```

图 2-6

2.4 实验结论

通过实验得出利用 JSP 进行数据库访问，需要在 JSP 页面中嵌入大量的 java 代码，才能完成逻辑操作，同时在 login.jsp 中设置的 session.setAttribute(“user”, request.getParameter()) 作用域范围为 session 范围，实验最后一步得到了验证

三、实验二 体验用 JavaBean 和不用的区别

3.1 实验目的

在实验一基础上在 JSP 利用 javaBean 实现和实验一相同的功能，体验用 JavaBean 和不用有什么区别

3.2 准备

在进行此实验前首先保证本地机器上配置好了正确的 Java 开发环境(JDK 环境)，已安装 Web 服务器 Tomcat，如果以上设置不正确，请参考《Tomcat 安装手册》进行正确设置

3.3 实验步骤

步骤一：

在实验一的基础上开始，在“03_mysql/src”目录下新建 com/Users.java 文件(com 为包名)，此 java 文件为封装了数据库中用户信息的 JavaBean，Users.java 成员属性为

```
package com;

public class Users {
    private int id;
    private String userName;
    private String password;
```

步骤二：

在“03_mysql/src”目录下新建 com.DbConnection.java 文件，此文件为封装获取数据库连接以及查询方法的 JavaBean，成员方法如下

```
public Connection getConnection() {
    Connection con = null;
    try {
        Class.forName("com.mysql.jdbc.Driver");
        con = DriverManager
            .getConnection("jdbc:mysql://localhost:3306/07-03?user=root&password=&characterEncoding=utf8");
    } catch (ClassNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return con;
}
```



```

public Users login(Users user) {
    Connection con = getConnection();
    PreparedStatement pre = null;
    ResultSet res = null;
    Users u = null;
    String sql = "select * from users where username=? and password=?";
    try {
        pre = con.prepareStatement(sql);
        pre.setString(1, user.getUserName());
        pre.setString(2, user.getPassword());
        res = pre.executeQuery();
        if (res.next()) {
            u = new Users();
            u.setId(res.getInt("id"));
            u.setUserName(res.getString("username"));
            u.setPassword(res.getString("password"));
        }
    } catch (Exception e) {
        System.out.println(e);
    } finally {
        try {
            if (res != null)
                res.close();
            if (pre != null)
                pre.close();
            if (con != null)
                con.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
    return u;
}

public ArrayList<Users> select() {
    Connection con = getConnection();
    PreparedStatement pre = null;
    ResultSet res = null;
    ArrayList<Users> list = new ArrayList<Users>();
    String sql = "select * from users";
    try {
        pre = con.prepareStatement(sql);
        res = pre.executeQuery();
        while (res.next()) {
            Users u = new Users();
            u.setId(res.getInt("id"));
            u.setUserName(res.getString("username"));
            u.setPassword(res.getString("password"));
            list.add(u);
        }
    } catch (Exception e) {
        System.out.println(e);
    } finally {
        try {
            if (res != null)
                res.close();
            if (pre != null)
                pre.close();
            if (con != null)
                con.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
    return list;
}

```

图 3-1

登录页面的<form>表单提交到 login.jsp 页面

步骤三:

login.jsp 页面首先利用<jsp:useBean>标签将 index 页面提交过来的表单信息（用户名、密码）封装成 Users 的一个对象 user，作用域范围为 Session（会话）范围，利用 JSP <%@page import=" " %> 引入 DbContention 对象

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@page import="java.sql.*"%>
<%@page import="com.DbConnection"%>
<!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=UTF-8">
<title>登录处理页面</title>
</head>
<body>
<jsp:useBean id="user" class="com.Users" scope="session" />
<jsp:setProperty property="*" name="user" />
<%!String nextPage = "";%>
<%
    DbConnection db = new DbConnection();
    if (db.login(user) == null) {
        nextPage = "index.jsp";
    } else {
        nextPage = "loginsuccess.jsp";
    }
%>
<jsp:forward page="<%=nextPage %>"></jsp:forward>
</body>
</html>

```

图 3-2

步骤五:

编写 loginsuccess.jsp 页面，同样使用 login.jsp 页面定义好的 User，DbContentention 对象，代码如下

```

<table bgcolor="yellow" width="80%">
    <tr>
        <td>userID</td>
        <td>userName</td>
        <td>password</td>
    </tr>
    <%
        DbConnection db = new DbConnection();
        ArrayList<Users> list = db.select();
        for (int i = 0; i < list.size(); i++) {
    %>
    <tr>
        <td><%=list.get(i).getId()%></td>
        <td><%=list.get(i).getUserName()%></td>
        <td><%=list.get(i).getPassword()%></td>
    </tr>
    <%}%>
</table>
</center>
<%}%>
</body>
</html>

```

图 3-3

步骤六:

如果不登录的情况下，直接在浏览器中输入

http://localhost:8080/03_mysql/loginsuccess.jsp，效果如图

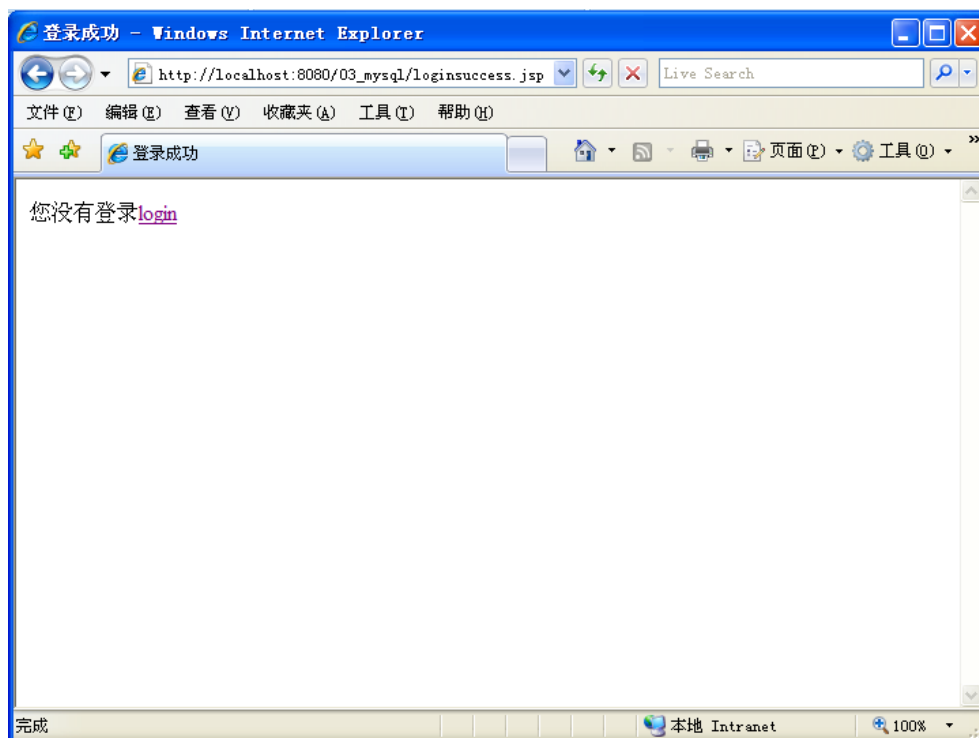


图 3-4

上述功能关键实现代码如下

```
<%  
    if (session.getAttribute("user") == null) {  
        您没有登录<a href="login.jsp">login</a>  
    } else {  
    }  
%>
```

图 3-5

3.4 实验结论

通过实验得出 JSP 中使用 JavaBean 会大大缩减 JSP 中 java 代码量,增强代码的复用,同时练习使用 JSP 中 `<jsp:useBean>` 标签将表单数据封装成 JavaBean 对象非常简单,设置作用域的范围也非常简单