**实验三 用户后台登录退出**

## **实验目标**

本实验任务实现“Q-ITOffer”锐聘网站后台管理系统的用户登陆和退出功能：

* 【实验3-1】 用户后台登陆。
* 【实验3-2】 用户后台退出。
* 【实验3-3】 登陆验证码生成和验证。

## **【实验3-1】用户登录**

### 知识点回顾

* 会话跟踪技术
* HttpSession对象的创建和使用
* 隐藏表单域的使用

### 功能描述

实现“Q-ITOffer”锐聘网站后台管理系统的用户登陆功能。

### 实验分析

1. 创建用户登录表单页面login.html，页面运行效果如图3- 2所示。
2. 创建处理用户登陆请求的Servlet文件UserServlet.java对用户登陆表单提交的请求数据进行获取，然后通过用户信息的数据库操作对象UserDAO对登陆数据进行查询，UserSerlet根据查询结果判断登录数据的正确性，若校验成功，将用户信息存入会话域属性，从而实现对该用户的会话跟踪；若校验失败，返回登陆页面进行错误提示。
3. 创建对用户登陆信息进行验证的数据库操作DAO类UserDAO.java。
4. 上述各资源组件间的关系如图3- 1所示。



图3- 1 用户登陆功能组件关系图

### 实验步骤

1. 在Eclipse工具中创建Java Web项目“Q\_ITOffer\_Experiment03”，导入【实验二】中的所有资源，使每个后续实验都能够保证前期实验的所有功能的实现。
2. 在项目工程目录WebContent下，创建用户登陆页面login.html，页面实现代码如下所示，页面运行效果如图3- 2所示。

**【代码3- 1】login.html**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>用户登录</title>

<script type="text/javascript">

**// 表单提交请求验证**

**function validate() {**

**if(document.getElementById("userLogname").value == ""){**

**alert("用户登录名不能为空！");**

**document.getElementById("userLogname").focus();**

**return false;**

**}**

**if(document.getElementById("userPwd").value == ""){**

**alert("登录密码不能为空！");**

**document.getElementById("userPwd").focus();**

**return false;**

**}**

**return true;**

**}**

</script>

</head>

<body>

<div class="login\_content">

<div class="login\_name">

<h3>用户登录</h3>

**<form action="UserServlet?type=login" method="post"**

**onsubmit="return validate();">**

<div>

用户名：**<input name="userLogname" id="userLogname" type="text"**

**class="login\_input" />**

</div>

<div>

密&nbsp;&nbsp;&nbsp;&nbsp;码：**<input name="userPwd"**

**id="userPwd" type="password" class="login\_input" />**

</div>

<div align="center">

**<input type="submit" class="login\_button"**

**value="登&nbsp;&nbsp;录"></input>**

</div>

</form>

</div>

</div>

</body>

</html>



图3- 2 用户登陆页面运行效果

1. 在项目工程目录src🡪com.qst.itofferbacker.servlet包下创建处理用户登陆请求的Servlet文件UserServlet.java，其实现代码如下所示。

**【代码3- 2】UserServlet.java**

package com.qst.itofferbacker.servlet;

import com.qst.itofferbacker.dao.UserDAO;

import com.qst.itofferbacker.javabean.User;

/\*\*

\* 后台系统用户信息操作Servlet

\* @公司 青软实训

\* @作者 fengjj

\*/

@WebServlet("/UserServlet")

public class UserServlet extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

**this.doPost(request, response);**

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// 设置请求字符编码

request.setCharacterEncoding("UTF-8");

// 设置响应字符编码

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

// 获取响应字符输出流

PrintWriter out = response.getWriter();

// 获取操作类型

**String type = request.getParameter("type");**

**if("login".equals(type)){**

// 获取用户登录信息

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

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

// 用户登录判断

UserDAO dao = new UserDAO();

User user = dao.login(userLogname,userPwd);

if(user != null){

**// 登陆成功，使用会话域属性记录用户信息，进入管理主界面**

**request.getSession().setAttribute("SESSION\_USER", user);**

**response.sendRedirect("manage/main.html");**

}else{

// 登录失败，错误信息提示，返回登录页面

out.print("<script type='text/javascript'>");

out.print("alert('用户名或密码错误，请重新输入！');");

out.print("window.location='login.html';");

out.print("</script>");

}

**}**

}

}

上述代码中通过判断UserDAO对象的login()方法的返回值User对象是否存在确定用户登陆信息的正确性。

1. 在项目工程目录src🡪com.qst.itofferbacker.dao包下创建对用户登陆信息进行数据库查询的操作类UserDAO.java，其实现代码如下所示。

**【代码3- 3】UserDAO.java**

package com.qst.itofferbacker.dao;

import com.qst.itofferbacker.javabean.User;

import com.qst.itofferbacker.util.DBUtil;

/\*\*

\* 后台管理员用户信息数据库操作类

\* @公司 青软实训

\* @作者 fengjj

\*/

public class UserDAO {

/\*\*

\* 用户登录

\* @param userLogname

\* @param userPwd

\* @return

\*/

public User login(String userLogname, String userPwd) {

Connection conn = DBUtil.getConnection();

PreparedStatement pstmt = null;

User u = null;

String sql = "SELECT user\_id,user\_realname,user\_email,user\_role,

user\_state FROM tb\_users WHERE user\_logname=? and user\_pwd=?";

try {

pstmt = conn.prepareStatement(sql);

pstmt.setString(1, userLogname);

pstmt.setString(2, userPwd);

ResultSet rs = pstmt.executeQuery();

if(rs.next()){

u = new User();

u.setUserId(rs.getInt(1));

u.setUserRealname(rs.getString(2));

u.setUserEmail(rs.getString(3));

u.setUserRole(rs.getInt(4));

u.setUserState(rs.getInt(5));

u.setUserLogname(userLogname);

}

} catch (SQLException e) {

e.printStackTrace();

}

return u;

}

## **【实验3-2】用户退出**

### 知识点回顾

* 会话跟踪技术
* Session会话跟踪技术
* 隐藏表单域技术

### 功能描述

实现“Q-ITOffer”锐聘网站后台管理系统的用户退出功能。

### 实验分析

1. 在后台管理主界面中设置用户退出功能入口。
2. 在处理用户模块请求的Servlet文件UserServlet.java中，增加对用户退出请求的处理。退出成功后，返回用户登录页面。
3. 上述各资源组件间的关系如图3- 3所示。



图3- 3 用户退出功能组件关系图

### 实验步骤

1. 在项目“Q\_ITOffer\_Experiment03”中选择用户登陆成功后的后台主页面main.jsp，在其包含的头文件top.html中编写用户的退出请求。top.html的代码实现如下所示。main.jsp的运行效果如图3- 4所示。

**【代码3- 4】top.html**

<li>**<a href="UserServlet?type=logout" target="\_parent">退出</a>**</li>



图3- 4 main.jsp运行效果

1. 在UserServlet.java中增加对用户退出请求的处理，实现代码如下所示。

**【代码3- 5】UserServlet.java**

@WebServlet("/UserServlet")

public class UserServlet extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

**this.doPost(request, response);**

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

......

**// 获取操作类型**

**String type = request.getParameter("type");**

if("login".equals(type)){

......

}else **if("logout".equals(type)){ // 用户退出**

**// 用户本次会话失效**

**request.getSession().invalidate();**

**// 返回登陆页面**

**out.print("<script type='text/javascript'>");**

**out.print("window.location='login.html';");**

**out.print("</script>");**

**}**

}

}

## **【实验3-3】登录验证码生成和验证**

### 知识点回顾

* 会话跟踪技术
* Session会话技术
* ServletOutputStream对象输出二进制数据

### 功能描述

实现“Q-ITOffer”锐聘网站后台管理系统的登录验证码的生成和验证功能。

### 实验分析

1. 在用户登录表单页面login.html中增加验证码的显示和输入功能，页面运行效果如图3- 6所示。
2. 创建负责输出验证码图像的Servlet文件ValidateCodeServlet.java。
3. 在负责处理登陆请求的Servlet文件UserServlet.java中增加对验证码输入正确性的验证，如果验证错误重新返回登陆页面，如果验证正确进入管理主界面。
4. 上述各资源组件间的关系如图3- 5所示。



图3- 5 验证码功能组件关系图

### 实验步骤

1. 修改Q\_ITOffer\_Experiment03项目的用户登录页面login.html，增加验证码显示、更换、输入功能，实现代码如下所示，页面运行效果如图3- 6所示。

**【代码3- 6】login.html**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>用户登录</title>

<script type="text/javascript">

**// 验证码的更换**

**function changeValidateCode() {**

**document.getElementById("validateCodeImg").src =**

**"ValidateCodeServlet?rand="+ Math.random();**

**}**

// 表单提交请求验证

function validate() {

if(document.getElementById("userLogname").value == ""){

alert("用户登录名不能为空！");

document.getElementById("userLogname").focus();

return false;

}

if(document.getElementById("userPwd").value == ""){

alert("登录密码不能为空！");

document.getElementById("userPwd").focus();

return false;

}

**if(document.getElementById("validateCode").value == ""){**

**alert("验证码不能为空！");**

**document.getElementById("validateCode").focus();**

**return false;**

**}**

return true;

}

</script>

</head>

<body>

<div class="login\_content">

<div class="login\_name">

<h3>用户登录</h3>

<form action="UserServlet" method="post" onsubmit="return validate();">

<div>

用户名：<input name="userLogname" id="userLogname" type="text"

class="login\_input" />

</div>

<div>

密&nbsp;&nbsp;&nbsp;&nbsp;码：<input name="userPwd"

id="userPwd" type="password" class="login\_input" />

</div>

**<div>**

**验证码：<input name="validateCode" id="validateCode"**

**type="text" class="yzm\_input" />**

**<img src="ValidateCodeServlet" id="validateCodeImg"**

**title="点击换一换" onclick="changeValidateCode()" />**

**<a href="javascript:changeValidateCode();"**

**style="color: blue;">看不清？</a>**

**</div>**

<div align="center">

<input type="submit" class="login\_button"

value="登&nbsp;&nbsp;录"></input>

</div>

<input type="hidden" name="type" value="login" />

</form>

</div>

</div>

</body>

</html>



图3- 6 登录页面运行效果

1. 创建输出验证码功能的Servlet文件ValidateCodeServlet.java，生成包含随机数的图像，并将随机数保存到会话对象中，然后进行输出，具体实现代码如下所示。（注意：本类可以不要求学生自己编写实现，能够看懂、理解即可。）

**【代码3- 7】ValidateCodeServlet.java**

package com.qst.itofferbacker.servlet;

import java.awt.Color;

import java.awt.Font;

import java.awt.Graphics2D;

import java.awt.image.BufferedImage;

import java.io.IOException;

import java.util.Random;

import javax.imageio.ImageIO;

import javax.servlet.ServletException;

import javax.servlet.ServletOutputStream;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/\*\*

\* 验证码图像生成

\* @公司 青软实训

\* @作者 fengjj

\*/

@WebServlet("/ValidateCodeServlet")

public class ValidateCodeServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

public ValidateCodeServlet() {

super();

}

protected void doGet(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException {

// 设置响应头 Content-type类型

response.setContentType("image/jpeg");

// 获取二进制数据输出流对象

ServletOutputStream out = response.getOutputStream();

// 创建缓冲图象

int width = 60;

int height = 20;

BufferedImage imgbuf = new BufferedImage(width, height,

BufferedImage.TYPE\_INT\_RGB);

Graphics2D g = imgbuf.createGraphics();

// 设定背景色

g.setColor(getRandColor(200, 250));

// 设定图像形状及宽高

g.fillRect(0, 0, width, height);

// 随机产生100条干扰线，使图象中的认证码不易被其它程序探测到

Random r = new Random();

g.setColor(getRandColor(160, 200));

for (int i = 0; i < 100; i++) {

int x = r.nextInt(width);

int y = r.nextInt(height);

int xl = r.nextInt(12);

int yl = r.nextInt(12);

g.drawLine(x, y, x + xl, y + yl);

}

// 随机产生100个干扰点，使图像中的验证码不易被其他分析程序探测到

g.setColor(getRandColor(120, 240));

for (int i = 0; i < 100; i++) {

int x = r.nextInt(width);

int y = r.nextInt(height);

g.drawOval(x, y, 0, 0);

}

// 随机产生0-9之间的4位数字验证码

g.setFont(new Font("Times New Roman", Font.PLAIN, 18));

String code = "";

for (int i = 0; i < 4; i++) {

String rand = String.valueOf(r.nextInt(10));

code += rand;

g.setColor(new Color(20 + r.nextInt(110), 20 + r.nextInt(110),

20 + r.nextInt(110)));

g.drawString(rand, 13 \* i + 6, 16);

}

**// 将验证码保存到session中**

**request.getSession().setAttribute("SESSION\_VALIDATECODE", code);**

// 输出图像

ImageIO.write(imgbuf, "JPEG", out);

out.close();

}

protected void doPost(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException {

}

// 获取指定范围的随机颜色

private Color getRandColor(int fc, int bc) {

Random random = new Random();

if (fc > 255)

fc = 255;

if (fc < 0)

fc = 0;

if (bc > 255)

bc = 255;

if (bc < 0)

bc = 0;

int r = fc + random.nextInt(bc - fc);

int g = fc + random.nextInt(bc - fc);

int b = fc + random.nextInt(bc - fc);

return new Color(r, g, b);

}

}

1. 在UserServlet.java中增加对验证码输入正确性的校验，具体实现代码如下所示。

**【代码3- 8】UserServlet.java**

@WebServlet("/UserServlet")

public class UserServlet extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// 设置请求字符编码

request.setCharacterEncoding("UTF-8");

// 设置响应字符编码

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

// 获取响应字符输出流

PrintWriter out = response.getWriter();

// 获取操作类型

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

if("login".equals(type)){

**// 获取用户提交的验证码**

**String validateCode = request.getParameter("validateCode");**

**// 获取本次请求会话中保存的验证码**

**String sessionValidateCode = (String)request.getSession()**

**.getAttribute("SESSION\_VALIDATECODE");**

**// 判断若验证码不一致，提示错误，返回登陆页面**

**if(!sessionValidateCode.equals(validateCode)){**

**out.print("<script type='text/javascript'>");**

**out.print("alert('验证码输入错误！');");**

**out.print("window.location='login.html';");**

**out.print("</script>");**

**return ;**

**}**

......

}

}