# 实验8 JSP内置对象（二）

**一、实验目的**

1、掌握session的基本用法。

2、理解各类域对象的区别。

**二、实验报告要求**

1、实验报告提交word文档，命名为“学号+姓名+实验8”。

2、本次实验共1题。本题应提供**完整代码**、以及**运行截图**。

运行截图应至少包含：使用**本人的姓名**作为用户名、**学号**作为密码登录的界面；以及登录成功的欢迎界面。

3、附上调试程序的过程中，出现的问题、以及解决方法。【可选】

**三、实验内容**

实现基于验证码的用户登录。



【实验步骤】

（1）新建一个名为”web8”的项目（Web应用程序）。在项目中，新建JSP，命名为“login”。

<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<html>

<head>

<title>login</title>

<script>

function \_change() {

//获取<img>元素

var imgEle = document.getElementById("vCode");

imgEle.src = "<%=request.getContextPath()%>/checkCodeServlet?"+new Date().getTime();

//因为某些浏览器会对src缓存，所以需要添加不同的参数，保证每次的URL参数都不同

}

</script>

<style>

div{

color: red;

}

</style>

</head>

<body>

<form action="<%= request.getContextPath()%>/login" method="post">

<table>

<tr>

<td>用户名</td><td><input type="text" name="username"></td></tr>

<tr>

<td>密码</td><td><input type="password" name="userpass"></td>

<td><div><%=request.getAttribute("login\_error") == null ? "" : request.getAttribute("login\_error")%></div></td></tr>

<tr>

<td>验证码</td><td><input type="text" name="checkCode"></td>

<td><div><%= request.getAttribute("code\_error") == null ? "" : request.getAttribute("code\_error")%></div></td></tr>

<tr>

<td colspan="2">

<img id="vCode" src="<%=request.getContextPath()%>/checkCodeServlet">

<a href="javascript:\_change()">看不清，换一张</a>

</td>

</tr>

<tr>

<td colspan="2">

<input type="submit" value="登录">

<input type="reset" value="重置">

</td>

</tr>

</table>

</form>

</body>

</html>

（2）新建java包“it.servlet”。在该包中新建java类，命名为“CheckCodeServlet”。

CheckCodeServlet.java的主要功能：生成验证码，将验证码显示在浏览器中，并将验证码存储于session对象。

package it.servlet;

import java.awt.Color;

import java.awt.Font;

import java.awt.Graphics;

import java.awt.image.BufferedImage;

import java.io.IOException;

import java.util.Random;

import javax.imageio.ImageIO;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.\*;

@WebServlet("/checkCodeServlet")

public class CheckCodeServlet extends HttpServlet {

protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

int width = 100;

int height = 40;

//创建验证码图片对象

BufferedImage image = new BufferedImage(width, height, BufferedImage.TYPE\_INT\_RGB);

//美化图片

//1.填充背景色

Graphics g = image.getGraphics();

g.setColor(Color.blue);

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

//2.画边框

g.setColor(Color.CYAN);

g.drawRect(0, 0, width - 1, height - 1);

//3.写验证码

//验证码的字符

String str = "ABCDEFGHIGKLMNOPQRSTUVWXYZabcdefghigklmnopqrstuvwxyz0123456789";

//生成随机字符串

Random ran = new Random();

StringBuilder sb = new StringBuilder();

for (int i = 1; i <= 4; i++) {

int index = ran.nextInt(str.length());

char ch = str.charAt(index);//随机字符

sb.append(ch);

g.setFont(new Font("TimesRoman", Font.PLAIN, 16));

g.drawString(ch + "", width / 5 \* i, height / 2);

}

String checkCode\_session = sb.toString();

//【代码1】将验证码存入session，属性名为checkCode\_session

req.getSession().setAttribute("checkCode\_session", checkCode\_session);

//4.画干扰线

g.setColor(Color.green);

//随机生成坐标点

for (int i = 1; i <= 10; i++) {

int x1 = ran.nextInt(width);

int x2 = ran.nextInt(width);

int y1 = ran.nextInt(height);

int y2 = ran.nextInt(height);

g.drawLine(x1, y1, x2, y2);

}

//将图片输出到页面展示

ImageIO.write(image, "jpg", resp.getOutputStream()/\*【代码2】\*/);

}

}

（3）在“it.servlet”包中新建名为“LoginServlet”的Java类。

LoginServlet.java用于检查用户名、密码以及验证码是否正确：

登录成功，跳转到welcome.jsp；

登录失败，跳转回login.jsp，并显示相应的错误提示。

package it.servlet;

import java.io.IOException;

import java.util.Objects;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.\*;

@WebServlet("/login")

public class LoginServlet extends HttpServlet {

protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

//【代码3】获取用户输入的用户名、密码以及验证码

req.setCharacterEncoding("UTF-8");

String username = req.getParameter("username");

String userpass = req.getParameter("userpass");

String checkCode = req.getParameter("checkCode");

//【代码4】获取生成的验证码

HttpSession session = req.getSession();

String checkCode\_session = (String) session.getAttribute("checkCode\_session");

//先判断验证码是否正确，忽略大小写比较

if (checkCode\_session != null && checkCode\_session.equalsIgnoreCase(checkCode)) {

if (/\*【代码5】进行用户名和密码的比较。登录成功的用户名为本人姓名，密码为本人学号\*/

Objects.equals("江炜", username) && Objects.equals("226709033", userpass)) {

//【代码6】登录成功，存储用户名到session，并重定向到welcome.jsp

session.setAttribute("username",username);

resp.sendRedirect(req.getContextPath() + "/welcome.jsp");

} else {

//【代码7】登录失败，存储提示信息“用户名或密码错误”到request，

req.setAttribute("login\_error", "用户或密码错误");

// 并转发到login.jsp

req.getRequestDispatcher("/login.jsp").forward(req, resp);

}

} else {

//验证码不一致，存储提示信息到request

req.setAttribute("code\_error", "验证码错误");

//转发到登录页面

req.getRequestDispatcher("/login.jsp").forward(req, resp);

}

}

}

（4）新建JSP文件，命名为“welcome“。

welcome.jsp检查用户是否登录成功；如果没有登录，则跳转回login.jsp；否则显示欢迎信息“\*\*\*，欢迎您”，效果如下：



<%@ page import="java.util.Objects" %>

<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<html>

<head>

<title>Title</title>

<style>

span {

color: red;

font-weight: bold

}

</style>

</head>

<body>

<%--【代码8】--%>

<%

String username = (String) session.getAttribute("username");

if (username == null || Objects.equals(username, "")) {

response.sendRedirect(request.getContextPath() + "/login.jsp");

return;

}

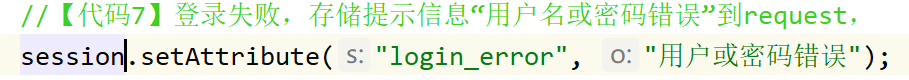
%>

<span><%=username%></span>,欢迎您！

</body>

**调试过程：**

1. 在登录失败页面不应该使用session.setAttribute()方法，而是req.setAttribute()方法，否则前端页面无法接收到登录失败的提示。



1. Username登录成功后未显示红色，是因为没有使用span标签选中，应该使用span标签把username包起来。



**实验结果：**

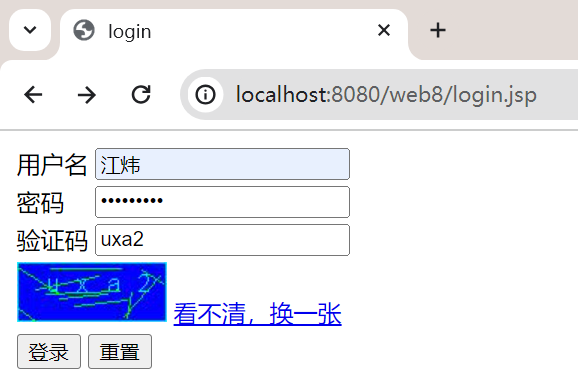


图 1登录页面

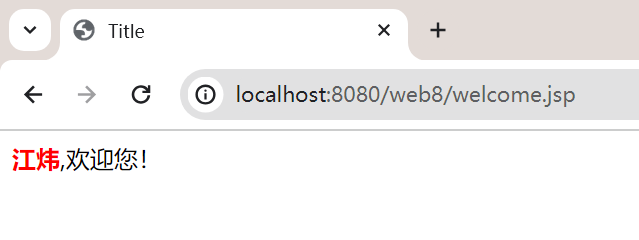


图 2登录成功页面



图 3验证码输入错误

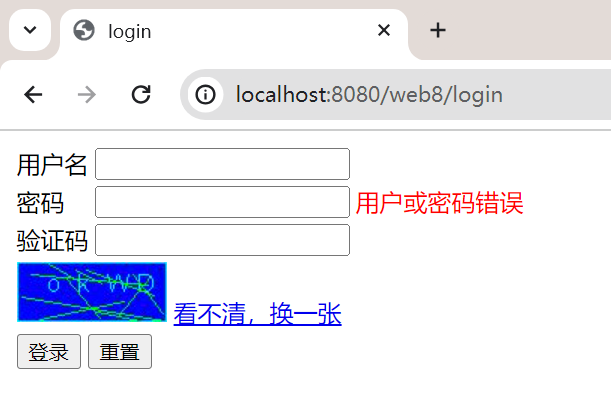


图 4用户名或密码错误