# 实验7 JSP内置对象（一）

**一、实验目的**

1、掌握response与request的基本用法。

1. 理解转发和重定向的区别。
2. 理解客户端路径和服务器端路径的区别。

**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7”。

2、本次实验共1题，报告中应提供**完整代码**、以及**运行截图**。

1. 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**

设计如图7-1所示的用户信息管理页面，并实现其中的“删除”与“编辑”功能。



图7-1 用户信息管理

【实验步骤】

（1）新建一个名为”web7”的项目（Web应用程序）。在项目中，新建“it.entity“包，在包中新建名为”User”的java类。

该部分与实验6相同。

**package** it.entity;  
**public class** User {  
 **private** String **sno**;  
 **private** String **name**;  
 **private** Integer **age**;  
 **private** String **gender**;  
 **public** User(String sno, String name, Integer age, String gender) {  
 **this**.**sno** = sno;  
 **this**.**name** = name;  
 **this**.**age** = age;  
 **this**.**gender** = gender;  
 }  
 **public** String getSno() {  
 **return sno**;  
 }  
  
 **public void** setSno(String sno) {  
 **this**.**sno** = sno;  
 }  
  
 **public** String getName() {  
 **return name**;  
 }  
  
 **public void** setName(String name) {  
 **this**.**name** = name;  
 }  
  
 **public** Integer getAge() {  
 **return age**;  
 }  
  
 **public void** setAge(Integer age) {  
 **this**.**age** = age;  
 }  
  
 **public** String getGender() {  
 **return gender**;  
 }  
  
 **public void** setGender(String gender) {  
 **this**.**gender** = gender;  
 }  
  
}

（2）新建JSP，命名为“users”。User.jsp文件用于展示用户的基本信息。

与实验6不同的地方是，为每一行增加了“编辑”超链接。

<%@ page import="it.entity.User" %>

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

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

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

<html>

<head>

<title>Title</title>

</head>

<body>

<div align="center">

<h1>用户信息管理</h1>

<%

//application为ServletContext对象，即代表当前web应用

List<User> users = (ArrayList<User>) application.getAttribute("users");

//当网站中没有users全局变量时，创建users对象，包含初始的用户信息

if (users == null) {

users = new ArrayList<>();

//【代码1】往users中添加一个代表本人信息的User对象

users.add(new User("226709033", "江炜", 20, "男")); // 代码1

users.add(new User("990001", "张三", 12, "女"));

users.add(new User("990002", "李四", 12, "男"));

users.add(new User("990003", "王五", 12, "女"));

users.add(new User("990004", "赵六", 12, "男"));

users.add(new User("990005", "小七", 12, "男"));

application.setAttribute("users", users);

}

%>

<table border="1px" width="90%" cellspacing="0">

<tr><th width="15%">编号</th>

<th width="15%">学号</th>

<th width="15%">姓名</th>

<th width="15%">年龄</th>

<th width="15%">性别</th>

<th width="25%">操作</th>

</tr>

<%

int rowNum = 0;//行编号

for (User user : users) {

rowNum++;

String sno = user.getSno();

String name = user.getName();

Integer age = user.getAge();

String gender = user.getGender();

String deleteLink = "delete?index=" + (rowNum - 1);

//【代码2】设置editLink变量

String editLink = "editPage.jsp?index=" + (rowNum - 1); // 代码2

%>

<tr style="text-align: center">

<td><%= rowNum%></td>

<td><%= sno%></td>

<td><%= name%></td>

<td><%= age%></td>

<td><%= gender%></td>

<!--点击“删除”链接，使用javascript脚本弹出确认对话框，单击“确定”后，才执行删除操作-->

<td>

<a href="<%=deleteLink%>" onClick=" return confirm('您确定要删除吗？请确认')">删除</a>

<%--【代码3】设置“编辑”的超链接--%>

<a href="<%=editLink%>">编辑</a> <!-- 代码3 -->

</td>

</tr>

<%

}

%>

</table>

</div>

</body>

</html>

（3）新建“it.servlet”包，在包中新建名为“DeleteServlet”的Java类。DeleteServlet.java文件用于删除一条用户信息。

该部分在实验6的基础上，增加了防盗链处理。

package it.servlet;

import it.entity.User;

import java.io.IOException;

import java.util.\*;

import javax.servlet.\*;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.\*;

@WebServlet("/delete")

public class DeleteServlet extends HttpServlet {

@Override

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

/\*【代码4】当用户不是通过点击users.jsp的超链接来到该页面时，跳转到users.jsp \*/

if (!Objects.equals("http://localhost:8080/web7/users.jsp", req.getHeader("referer"))) {

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

return;

}

/\* String referer = req.getHeader("referer");

if (referer==null||!referer.endsWith("user.jsp")) {

//Referer是什么：

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

return;

}\*/

int index = Integer.parseInt(req.getParameter("index"));

List<User> users = (ArrayList<User>) getServletContext().getAttribute("users");

users.remove(index);

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

}

}

（4）新建JSP，命名为“editPage”。editPage.jsp为编辑用户信息的页面。

<%@page import="java.util.\*"%>

<%@page import="it.entity.User"%>

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

<html>

<head>

<title>编辑</title>

</head>

<body>

<h1>编辑用户信息</h1>

<%

//从application(ServletContext对象)中获取全局变量users的值

List<User> users = (ArrayList<User>) application.getAttribute("users");

//获取被编辑的用户的下标值

int index = Integer.parseInt(request.getParameter("index"));

//根据下标获取待编辑的用户

User user = users.get(index);

//【代码5】读取待编辑用户的基本信息，为sno、name、age、gender变量赋值

String sno = user.getSno();

String name = user.getName();

Integer age = user.getAge();

String gender = user.getGender();

%>

<form action="edit?index=<%=index%>" method = "post">

<!--通过?在form的action后面传递参数时，需要指定form的method="post",不然无法获得?后面的参数-->

学号 <input type="text" name="sno" readonly="true" value="<%=sno%>"><br/>

姓名 <input type="text" name="name" value="<%=name%>"><br/>

年龄 <input type="text" name="age" value="<%=age%>"><br/>

性别 <input type="text" name="gender" value="<%=gender%>"><br/>

<br/>

<input type="submit" value="提交"></br>

</form>

</body>

</html>

（5）在“it.servlet”包中新建名为“EditServlet”的Java类。EditServlet.java文件完成用户信息的修改。

<%@page import="java.util.\*"%>

<%@page import="it.entity.User"%>

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

<html>

<head>

<title>编辑</title>

</head>

<body>

<h1>编辑用户信息</h1>

<%

//从application(ServletContext对象)中获取全局变量users的值

List<User> users = (ArrayList<User>) application.getAttribute("users");

//获取被编辑的用户的下标值

int index = Integer.parseInt(request.getParameter("index"));

//根据下标获取待编辑的用户

User user = users.get(index);

//【代码5】读取待编辑用户的基本信息，为sno、name、age、gender变量赋值

String sno = user.getSno();

String name = user.getName();

Integer age = user.getAge();

String gender = user.getGender();

%>

<form action="edit?index=<%=index%>" method = "post">

<!--通过?在form的action后面传递参数时，需要指定form的method="post",不然无法获得?后面的参数-->

学号 <input type="text" name="sno" readonly="true" value="<%=sno%>"><br/>

姓名 <input type="text" name="name" value="<%=name%>"><br/>

年龄 <input type="text" name="age" value="<%=age%>"><br/>

性别 <input type="text" name="gender" value="<%=gender%>"><br/>

<br/>

<input type="submit" value="提交"></br>

</form>

</body>

</html>

实验结果：













