**实验二 HTTP请求对象**

一、实验目的

1. 掌握如何获得请求参数；

2. 掌握如何检索请求头；

3. 了解如何通过请求对象获得客户信息；

4. 重点掌握使用请求对象存储数据和请求转发。

二、实验原理

对客户的请求，容器根据请求调用Servlet的相应doXxx()方法，该方法带有两个参数，一个是HttpServletRequest，一个是HttpServletResponse，它们分别表示请求对象和响应对象，它们是容器自动创建的两个对象。可以使用这两个接口定义的若干方法完成有关操作。

三、实验内容与步骤

**1、通过查询串传递请求参数，并在Servlet中检索请求参数**

【步骤1】编写HelloUserServlet类，实现显示Hello XXX。XXX表示由查询串传递的参数。



图2-1 HelloUserServlet程序的运行效果

**此处的XXX是由查询串传递得参数**

**（1）代码：【10分】**

package com.demo;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class HelloUserServlet extends HttpServlet{

public void service(HttpServletRequest request,HttpServletResponse response)

throws ServletException, IOException{

**String userName = request.getParameter("userName");【4分】**

PrintWriter pw = response.getWriter();

pw.println("<html>");

pw.println("<head>");

pw.println("</head>");

pw.println("<body>");

**pw.println("<h3>Hello, " + userName + "</h3>");【3分】**

pw.println("</body>");

pw.println("</html>");

}

}

【步骤2】编译该Servlet，然后修改web.xml文件。

**（2）代码**

**<servlet>【5分】**

**<servlet-name>helloUserServlet</servlet-name>**

**<servlet-class>com.demo.HelloUserServlet </servlet-class>**

**</servlet>**

**<servlet-mapping>【5分】**

**<servlet-name>helloUser</servlet-name>**

**<url-pattern>/hellouser</url-pattern>**

**</servlet-mapping>**

【步骤3】使用URL访问。

http://localhost:8080/XXXX/hellouser**?userName=John**

此处的XXXX为项目名称

**（3）运行截图：**

将userName的值改为你自己的名字，看输出结果。若不给出查询字符串，结果如何？参数值使用汉字如何？

方法一：request.setCharacterEncoding("UTF-8");

方法二：String name = new String(request.getParameter("userName").getBytes("ISO8859-1"),"UTF-8")

**（4）运行截图：**

**2、通过表单传递请求参数，使用HttpServletRequest对象获得请求参数**

【步骤1】首先编写一个input.html文件，存放在helloapp目录中。该页面在屏幕上显示一个表单，其中包括姓名和年龄域，点击“确定”按钮后，将表单参数传递给ParamServlet并显示出来。



图2-2 input.html程序的运行效果

**（1）input.html代码：【5分】**

<html>

<head>

<title>参数传递</title>

<meta http-equiv="Content-Type" content="text/html;charset=gb2312">

</head>

<body>

<center>

<h3>从表单向Servlet传递参数</h3>

**<form method="GET" action="paramtest" >【3分】**

姓名：<input type="text" name="name"><br>

年龄：<input type="text" name="age"><br>

<input type="submit" value="确定" >

<input type="reset" value="重置" >

</form>

<center>

</body>

</html>

【步骤2】编写ParamServlet.java，实现将从表单向Servlet传递的参数显示在浏览器内。

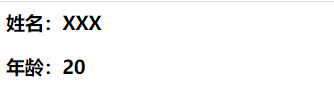


图2-3 ParamServlet程序的运行效果

**（2）代码：【10分】**

package com.demo;

import java.io.\*;

import java.util.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class ParamServlet extends HttpServlet {

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException{

**request.setCharacterEncoding("UTF-8");【2分】**

**response.setContentType("text/html;charset=UTF-8");【2分】**

**String name = request.getParameter("name");【2分】**

name = new String(name.getBytes("iso-8859-1"),"gb2312");

**String age = request.getParameter("age");【2分】**

age = new String(age.getBytes("iso-8859-1"),"gb2312");

PrintWriter pw = response.getWriter();

pw.println("<html>");

pw.println("<head>");

pw.println("</head>");

pw.println("<body>");

pw.println("**<h3>姓名："+name+"</h3><p>**");**【1分】**

pw.println("**<h3>年龄："+age+"</h3><p>**");**【1分】**

pw.println("</body>");

pw.println("</html>");

}

public void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException{

doGet(request, response);

}

}

【步骤3】修改web.xml文件，将下面代码添加到其中：

**（3）代码：**

**<servlet>【5分】**

**<servlet-name>ParamServlet</servlet-name>**

**<servlet-class>com.demo.ParamServlet</servlet-class>**

**</servlet>**

**<servlet-mapping>【5分】**

**<servlet-name>ParamServlet</servlet-name>**

**<url-pattern>/paramtest</url-pattern>**

**</servlet-mapping>**

【步骤4】测试参数传递。在浏览器的地址栏中输入：http://localhost:8080/XXXX/input.html。输入数据，点击“确定”按钮，体会表单参数的传递。此处的XXXX为项目名称

**（4）运行截图：**

问题1：如何通过超链接的锚标签（<a href>）传递请求参数？

**（5）代码：【5分】**

**<a href=“servlet地址?参数1=参数值1&参数2=参数值2”>**

**3、编写一个Servlet，显示所有请求头**

【步骤1】ShowHeadersServlet.java的Servlet能够检索所有的请求头。

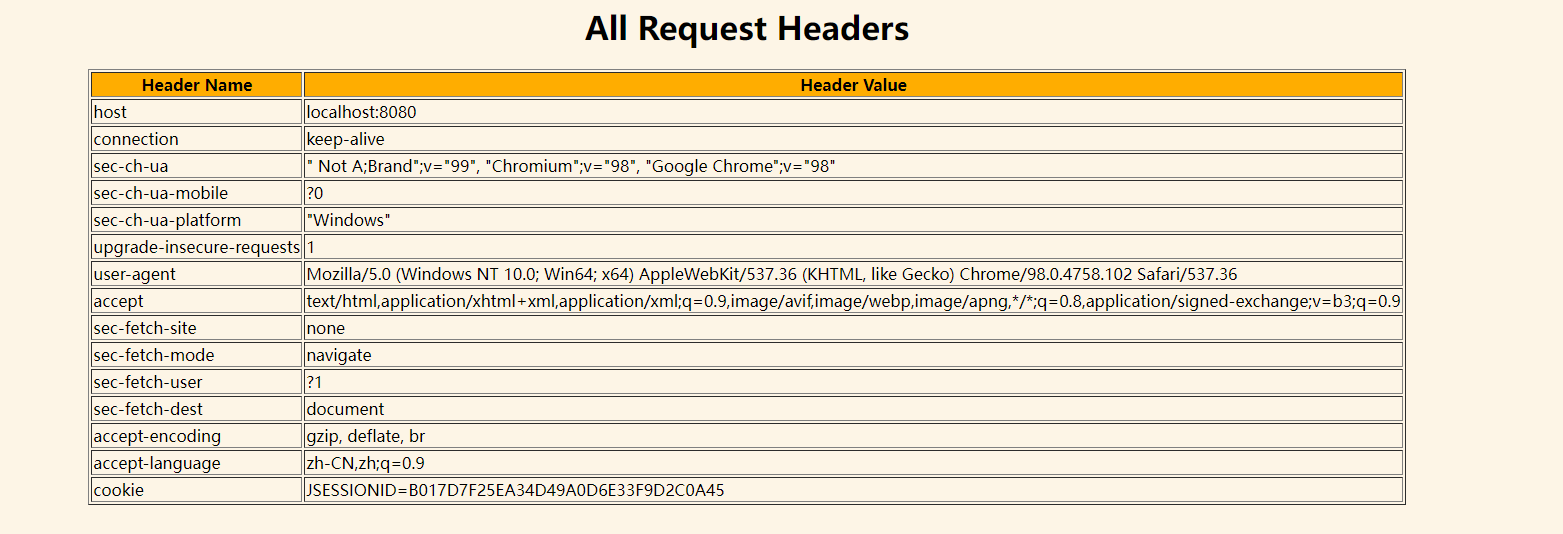


图2-4 ShowHeadersServlet程序的运行效果

**（1）代码**

package com.demo;

import java.io.\*;

import java.util.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class ShowHeadersServlet extends HttpServlet {

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException{

**response.setContentType("text/html;charset=gb2312");【2分】**

PrintWriter out = response.getWriter();

out.println("<html>");

out.println("<head><title>All Headers</title>");

out.println("</head>");

out.println("<body bgcolor=\"#fdf5e6\">");

out.println("<h1 align=\"center\">All Request Headers</h1>");

**out.println("<table border=1 align=\"center\">\n"+**

**"<tr bgcolor=\"#ffad00\">\n"+**

**"<th>Header Name<th>Header Value");【3分】**

**Enumeration headerNames = request.getHeaderNames();【2分】**

**while(headerNames.hasMoreElements()){【2分】**

**String headerName = (String)headerNames.nextElement();【2分】**

**out.println("<tr><td>"+headerName); 【2分】**

**out.println("<td>"+request.getHeader(headerName)); 【2分】**

**}**

out.println("</table>\n</body></html>");

}

public void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException{

doGet(request, response);

}

}

【步骤2】修改web.xml，加入下面内容：

<servlet>

<servlet-name>ShowHeadersServlet</servlet-name>

<servlet-class>\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>ShowHeadersServlet</servlet-name>

<url-pattern>/showheaders</url-pattern>

</servlet-mapping>

**（2）代码：**

**com.demo.ShowHeadersServlet 【2分】**

【步骤3】在浏览器地址栏中输入下面URL访问该Servlet

**（3）链接：**http://localhost:8080/XXX/showheaders

**（4）运行截图：**

**4、简单的综合应用**

【步骤1】创建一个名为input.html的HTML页面，其中包括一个表单，表单中包含两个文本域，分别供用户输入学号和姓名，该页面也包含提交和重置按钮。

【步骤2】定义一个名为com.demo.Student类，其中包括学号sno和姓名name两个private的成员变量，定义访问和修改sno和name的方法。

【步骤3】编写名为FirstServlet的Servlet，要求当用户在input.html中输入信息后点击“提交”按钮，请求FirstServlet对其处理。在FirstServlet中使用表单传递的参数（学号和姓名）创建一个Student对象并将其作为属性存储在请求对象中，然后获得通过请求对象的getRequestDispatcher()方法获得RequestDispatcher()对象，将请求转发到SecondServlet。

【步骤4】在SecondServlet中取出请求对象上存储的Student对象，并显示输出该学生的学号和姓名。在SecondServlet的输出中应该包含一个超链接，点击该连接可以返回input.html页面。



图2-5 input.html运行效果

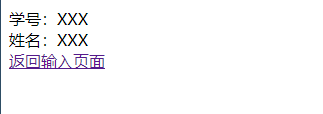


图2-6 运行结果图

**代码：**

**（1）input.html【5分】**

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

pageEncoding="UTF-8"%>

<!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>输入学生信息</title>

</head>

<body>

**<form action="FirstServlet" method="post">【3分】**

学号<input type="text" name="sno" size="15" /><br>

姓名<input type="text" name="sname" size="15" /><br>

<input type="submit" value="登录" />

<input type="reset" value="取消" />

</form>

</body>

</html>

**（2）Student.java【5分】**

package com.demo;

public class Student {

private String sno;

private String name;

public Student(String sno, String name) {

this.sno = sno;

this.name = name;

}

// 省略属性的setter方法和getter方法

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;

}

}

**（3）FirstServlet.java【15分】**

@WebServlet("/FirstServlet")

public class FirstServlet extends HttpServlet {

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

**String sno = request.getParameter("sno");【2分】**

**String name = request.getParameter("sname");【2分】**

**Student student = new Student(sno, name); 【2分】**

**request.setAttribute("student", student); 【3分】**

**RequestDispatcher rd = request.getRequestDispatcher("/SecondServlet");【4分】**

**rd.forward(request, response); 【2分】**

}

}

**（4）SecondServlet.java【10分】**

@WebServlet("/SecondServlet")

public class SecondServlet extends HttpServlet {

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

**Student student = (Student) request.getAttribute("student");【4分】**

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

PrintWriter out = response.getWriter();

out.println("<title>学生信息</title>");

**out.println("学号：" + student.getSno() + "<br>");【2分】**

**out.println("姓名：" + new String(student.getName().getBytes("iso-8859-1"), "UTF-8") + "<br>");【2分】**

**out.println("<a href='input.jsp'>返回输入页面</a>");【2分】**

}

}

**运行截图：**