## 实验三 Servlet基础

### 一、实验目的

1　了解MyEclipse 开发环境的构成

2　掌握MyEclipse 下JAVA EE开发环境的设置

3 了解和掌握Servlet和JSP页面的编写和运行。

4 比较Servlet与JSP的区别和联系，体会动态页面的含义。

5　掌握Servlet开发步骤

二、实验原理

Servlet是运行在服务器端的Java程序，它主要是为了扩展服务器的功能。编写和运行Servlet一般要经过4个步骤：（1）编辑源文件，它是继承HttpServlet的Java类；（2）编译源文件；（3）在服务器中部署该Servlet，即将类文件存放到服务器的适当目录中，并修改web.xml部署描述文件；（4）使用浏览器访问该Servlet程序。

JSP是一种服务器端的动态页面技术，它的实质也是Servlet。它的编写较Servlet容易。二者的主要区别是：Servlet通常是实现业务逻辑，而JSP通常实现表示逻辑。

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。

对Servlet的请求目的是要获得响应消息。Servlet要向用户发送响应数据首先要获得输出流对象。另外，通过响应对象还可向客户端发送状态码，也可以设置有关响应头，还可以对响应重定向。

**三、实验内容**

1．编写和运行一个简单的Servlet

【步骤1】在Tomcat安装目录的webapps下建立一个helloapp的目录，在其中再建立一个WEB-INF目录，在WEB-INF目录中建立一个classes目录。这样就建立了一个名为helloapp的Web应用程序。

【步骤2】编写下面Servlet类，编译并将类文件(HelloServlet.class)放入classes目录中

package com.demo;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class HelloServlet extends HttpServlet{

public void service(HttpServletRequest request,

HttpServletResponse response)

throws ServletException,IOException{

PrintWriter out = response.getWriter();

out.println("<html>");

out.println("<head>");

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

out.println("<body>");

out.println("<h3>Hello World!</h3>");

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

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

}

}

【步骤3】在WEB-INF目录中建立一个名为web.xml的文件，该文件为Web应用程序的部署描述文件，简称DD。具体内容如下：

<?xml version="1.0" encoding="ISO-8859-1"?>

<web-app xmlns="http://java.sun.com/xml/ns/j2ee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app\_2\_5.xsd"

version="2.5">

<servlet>

<servlet-name>helloServlet</servlet-name>

<servlet-class>com.demo.HelloServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>helloServlet</servlet-name>

<url-pattern>**/hello**</url-pattern>

</servlet-mapping>

</web-app>

【步骤4】在浏览器的地址栏中输入下列地址访问该Servlet。

http://localhost:8080/helloapp**/hello**

在浏览器的窗口中应该显示Hello, World!内容。

2．编写一个Servlet可显示系统当前时间。请在MyEclipse集成环境下编译并部署该Servlet，然后使用浏览器访问。

package com.demo;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class HelloWorldServlet extends HttpServlet{

public void service( HttpServvletRequest request, HttpServvletResponse response )

throws ServletException,IOException{

PrintWriter out = response.getWriter() ;

out.println("<html>");

out.println("<head>");

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

out.println("<body>");

out.println(" <font color = '#0000ff'>");

out.println("<h3>Hello,World!</h3>");

out.println("The time now is:"+new java.util.Date());

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

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

}

}

3．编写一个Servlet，可以通过查询串传递请求参数，并在Servlet中检索请求参数

【步骤1】编写下面的Servlet。

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");**

PrintWriter pw = response.getWriter();

pw.println("<html>");

pw.println("<head>");

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

pw.println("<body>");

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

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

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

}

}

【步骤2】编译该Servlet，然后修改web.xml文件，将下面代码添加到其中：

<servlet>

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

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

</servlet>

<servlet-mapping>

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

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

</servlet-mapping>

【步骤3】使用下面的URL访问之。

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

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？

4．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。

input.html代码如下：

<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" >

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，其代码如下：

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{

response.setCharacterEncoding("gb2312");

response.setContentType("text/html;charset=gb2312");

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

name = new String(name.getBytes("iso8859-1"),"gb2312");

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

age = new String(age.getBytes("iso8859-1"),"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>");

pw.println("<h3>年龄："+age+"</h3><p>");

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：

<servlet>

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

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

</servlet>

<servlet-mapping>

<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/helloapp/input.html。输入数据，点击“确定”按钮，体会表单参数的传递。

5．存储转发与响应重定向

【步骤1】建立下面的login.html登录页面

<html>

<head> <title>Login Page</title> </head>

<body>

<**form action="login.do" method="post"**>

<table width="298" border="0" align="center"

cellpading="2" cellspacing="1">

<tr>

<td align="right">用户名：</td>

<td align="left"><input type="text" **name="username"** size="15"></td>

</tr>

<tr>

<td align="right">密&nbsp;&nbsp;码：</td>

<td><input type="password" **name="password"** size="15"></td>

</tr>

<tr>

<td colspan="2" align="center"><input type="submit" value="登录">

&nbsp; <input type="reset" value="取消"></td>

</tr>

</form>

</body>

</html>

【步骤2】编写一个名为RedirectServlet的Servlet，在其中检索用户名和口令参数值，然后判断其是否正确（若用户名和口令都为admin即认为正确），如果正确将响应重定向到welcome.html页面，否则将响应重定向到error.html页面，这两个页面请你自己编写，存放在应用程序根目录。下面是Servlet的部分代码。

package com.demo;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class RedirectServlet extends HttpServlet{

public void doPost(HttpServletRequest request,

HttpServletResponse response)

throws IOException,ServletException{

// 此处添加若干代码

if( ){

**response.sendRedirect( );**

}else{

**response.sendRedirect( );**

}

}

}

思考题

1. 如何理解Servlet的生命周期？

2. 如何理解请求转发与响应重定向？

将RedirectServlet中语句：

response.sendRedirect("\_\_\_\_\_\_\_");

替换成：

RequestDispatcher rd=request.getRequestDispatcher("\_\_\_\_\_\_\_");

rd.forward(request, response);

结果怎样，为什么？

### 四、知识扩展

（1）信息系统中，均含有一个模块――系统用户管理，请分析该模块所需的功能。同时结合“面向对象课程设计”中所承担的项目，对该项目进行分析。

（2）设计名为Login 的UI界面，提供信息系统中的用户功能。UI界面上除了用户名与口令外，应该有“验证码”、“下次免登录”、“忘记密码”等功能。如下图所示。

