* HttpServletRequest的详解:

代表的是客户端的请求.

当web服务器得到该请求后,会把请求的信息封装在HttpServletRequest对象

Request.getqueryString()

该函数可以获取请求部分的数据,such as

<http://lenovo-pc:8080/servletPro2/GetInfoServlet?username=wsx&password=123456>

会获取到红色部分

URL和URI的区别



//得到URL

String url=request.getRequestURL().toString();

System.*out*.println("URL="+url);

//得到URI

String uri=request.getRequestURI();

System.*out*.println("uri="+uri);

//

String queryString=request.getQueryString();

System.*out*.println(queryString);

//getRemoteAddr()函数 ,可以得到远程主机的ip地址

String ip=request.getRemoteAddr();

System.*out*.println("ip="+ip);

//getRemoteHost(),可以得到对方主机名,如果该客户机在dns没注册过,则返回ip地址

//如果注册过,则返回主机的全名

String host=request.getRemoteHost();

System.*out*.println("host="+host);

//获取用户机使用的端口号

**int** port=request.getRemotePort();

**int** serverport=request.getLocalPort();

System.*out*.println("客户机"+ip+"客户机端口号="+port+"服务器端口号="+serverport);

* 通过表单提交信息

界面部分:

request.setCharacterEncoding("utf-8");

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

PrintWriter out = response.getWriter();

out.println("<form action='/servletPro2/RegisterServlet' method='post' /> ");

out.println("用户名:<input type='text' name='username' />"+"</br>");

out.println("密　码:<input type='password' name='pwd'/>"+"</br>");

out.println("性　别:<input type='radio' name='sex' value='男'/>男 <input type='radio' name='sex' value='女'/>女"+"</br>");

out.print("爱　好:<input type='checkbox' name='hobby' value='音乐' />音乐 <input type='checkbox' name='hobby' value='体育' />体育 <input type='checkbox' name='hobby' value='旅游' />旅游"+"</br>");

out.print("所在城市:<select name='city'><option value='北京'>北京</option><option value='南京'>南京</option><option value='天津'>天津</option></select>"+"</br>");

out.print("你的介绍:"+"</br>");

out.println("<textarea cols='20' rows='10' name='introduce'>请输入信息..</textarea>"+"</br>");

out.print("照片:<input type='file' name='photo'>"+"</br>");

//什么时候使用隐藏控件传递信息(隐藏控件的位置无所谓,随意放可以)1.不希望用户看到的信息2.要用到的逻辑信息,但是不希望影响界面的信息

out.print("<input type='hidden' name='hidden1' value='解密方法'>");

out.println("<input type='submit' value='提交信息' />"+"</br>");

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

}

接收信息的部分

request.setCharacterEncoding("utf-8");

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

PrintWriter out = response.getWriter();

String u=request.getParameter("username");

String p= request.getParameter("pwd");

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

//如果接受复选框的内容,要用getParameterValues()来接收内容

String hobbies[]=request.getParameterValues("hobby");

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

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

String hidden1=request.getParameter("hidden1");

//System.out.println(u);

//System.out.println(p);

out.print("用户名="+u+"</br>");

out.print("密　码="+p+"</br>");

out.print("性　别="+sex+"</br>");

**if**(hobbies!=**null**){

**for**(**int** i=0;i<hobbies.length;i++){

out.print("爱好:"+hobbies[i]+"</br>");

}

}

**else**{

out.print("你没有选择爱好"+"</br>");

}

out.print("所在城市:"+city+"</br>");

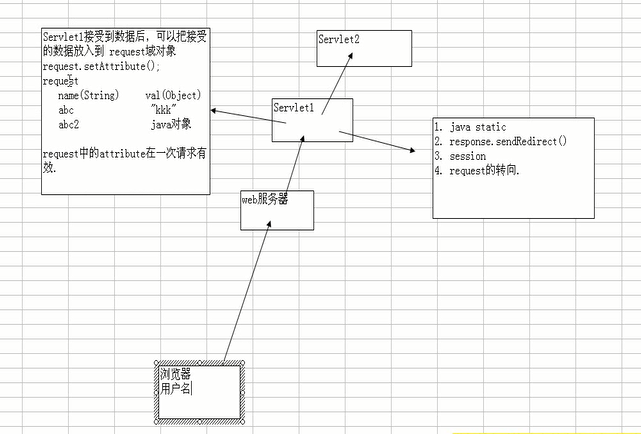
out.print("介绍:"+introduce);

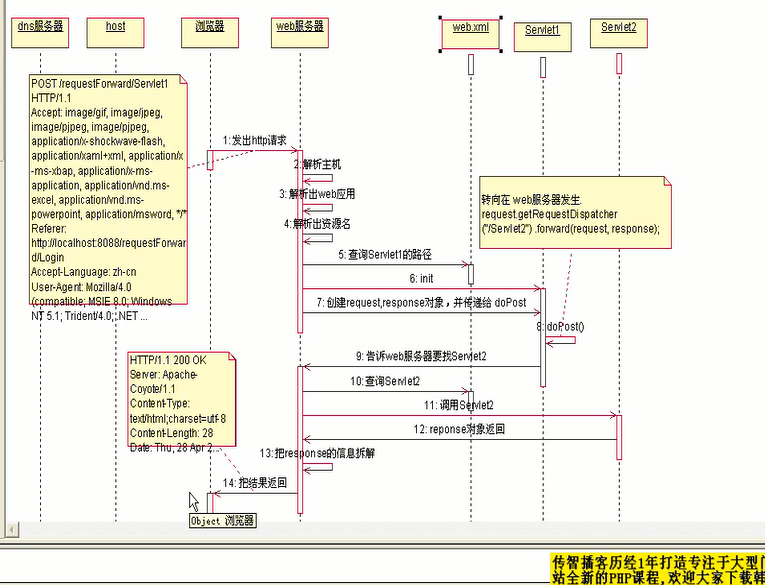
out.print("</br>隐藏控件的内容:"+hidden1);

请求转发

我们现在使用请求转发的方法来实现上一次我们使用response.sendRedirect()方法实现的效果

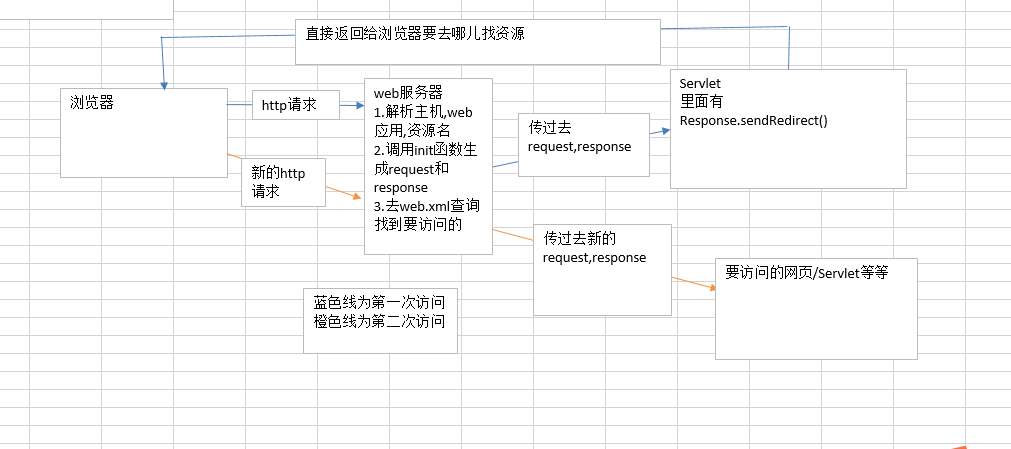
使用request的方法





转向Forword知识点

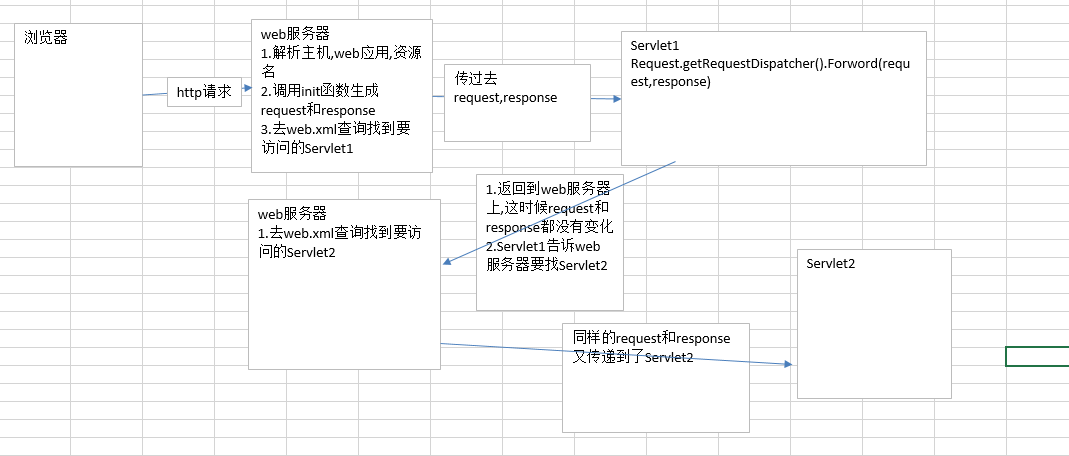
1. 使用forword不能转发到该web应用外的URL
2. 因为forword是发生在web服务器的,所以Servlet1和Servlet2是使用的同一个request和response
3. Request,response都是域对象,浏览器每访问一次web服务器待web服务器解析完成后都会重新创建request和response对象
4. Response.sendRedirect()方法是浏览器访问一次web服务器创建完一次request和response对象后访问到了Servlet,然后Servlet告诉他要去哪儿找资源,然后浏览器再去重新访问web服务器去找那个资源.这个过程是



所以若在第一次访问中设置的域对象request和response的属性(setAttribute)内的值,

到了要访问的资源页在调用getAttribute方法不会得到set的值,因为这完全是两次访问,第二次已经调用了init函数生成了新的域对象(request和response)所以不会传递过来

1. Request.getRequestDispatcher().Forword(request,response)方法是只有一次浏览器访问web服务器,,创建完request和response访问到了要访问的Servlet, 然后Request.getRequestDispatcher().Forword(request,response)这个方法会把request和response的值传递到第二个Servlet上,整个过程是:



所以若在Servlet1中调用setAttribute()设置了值,在Servlet2中就可以调用到设置的值,因为是同一组域对象(request,response),并且因为Request.getRequestDispatcher().Forword(request,response)这个方法两次访问都是在web服务期内发生的,所以这个方法在填写要访问的资源名是不用填写web应用名,只需填写要访问的Servlet名字就可以,系统会自动填写应用名

面试题:请问sendRedirect和forword有什么区别?

1. 叫法不一样sendRedirect叫请求重定向, forword叫请求转发
2. 实际发生位置不一样

sendRedirect发生在浏览器

forword发生在web服务器

1. 用法不一样

Request.getRequestDispatcher(“/资源名”).forword(request,response)

Response.sendRedirect(“/web应用名/资源名”)

1. 能够跳转的范围不一样

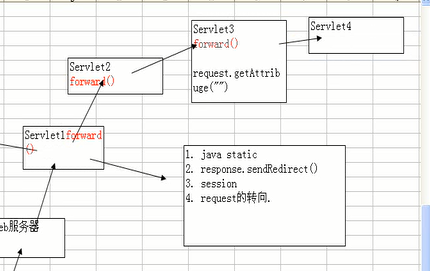
sendRedirect可以去任何一个URL

forword只能去当前的web应用资源

* 什么是一次http请求

只要没有停止,也没有回到浏览器重定向就算是一次http请求

Forword可以形成请求链



这个算一次请求,每次getAttribute()都能get到值

* 细节

每次forword浏览器的地址栏不改变,永远停留在第一个Servlet处

URL是浏览器改变的,在同一个web服务器内不会改变,web故武器不会自己去改变URL

* 用户管理系统的继续开发

1. 增加到数据库去验证用户的功能

在sqlserver中建立一张表

create schema test

create table test.users(

id char(20) primary key,

username char(30) not null,

email char(20) not null,

grade smallint default 1,

--级别越高数字越大

password char(20) not null

)

select \* from test.users

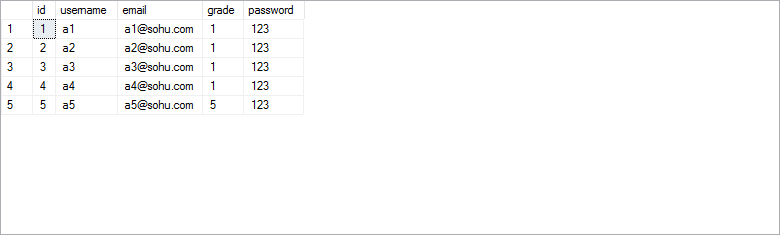
insert into test.users values('1','a1','a1@sohu.com','1','123')

insert into test.users values('2','a2','a2@sohu.com','1','123')

insert into test.users values('3','a3','a3@sohu.com','1','123')

insert into test.users values('4','a4','a4@sohu.com','1','123')

insert into test.users values('5','a5','a5@sohu.com','5','123')



2.在LoginclServlet中添加到数据库验证用户的功能

request.setCharacterEncoding("utf-8");

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

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

//到数据库中去验证

Connection cn=**null**;

ResultSet rs=**null**;

PreparedStatement st=**null**;

**try** {

//加载驱动

Class.*forName*("com.microsoft.sqlserver.jdbc.SQLServerDriver");

//得到连接

//第二/三个引号里直接写用户名和密码

cn=DriverManager.*getConnection*("jdbc:sqlserver://localhost:1433;database=Users","sa","mis\_123");

//创建PreparedStatement

System.*out*.println("数据库连接成功");

//查询语句中from数据库名.架构名.表名

String sql="select \* from Users.test.users where id=? and password=?";

//先创建PreparedStatement,再设定字符串内容

st=cn.prepareStatement(sql);

//给?赋值

st.setString(1, id);

st.setString(2, password);

//执行操作

rs=st.executeQuery();

//根据结果做处理

**if**(rs.next()){

//用户合法

request.getRequestDispatcher("/MainFrame").forward(request, response);

}**else**{

//用户不合法

request.getRequestDispatcher("/Login").forward(request, response);

}

} **catch** (Exception e) {

// **TODO**: handle exception

e.printStackTrace();

System.*out*.println("数据库连接成功");

}

**finally**{

//关闭资源

**if**(cn!=**null**){

**try** {

cn.close();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**if**(st!=**null**){

**try** {

st.close();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**if**(rs!=**null**){

**try** {

rs.close();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

1. 如果输入的用户ID/密码不正确,则给出提示