HttpServletResponse的应用

3.重定向(跳转)从一个资源跳转到另外一资源

示例: ForwardServlet跳转到succss.html

特点: 地址栏会发生变化

http://localhost:8080/day03/ForwardServlet

http://localhost:8080/day03/success.html

两种请求两次响应

重定向的另外一种用法:

response.sendRedirect("/day03/success.html");

4.解决response对象输出中文乱码问题的解决方案

(1)在使用response对象之前，设置response的编码

//设置response对应的编码

response.setCharacterEncoding("utf-8"); utf-8/GBK

如果浏览器采用不是utf-8格式的编码来预览内容，则会出现中文乱码,需要手动改变浏览器的预览编码，

改成来服务器相同的编码

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

第一层含义：服务器端（Servlet）采用GBK编码方案

第二层含义：通知浏览器以GBK编码来浏览内容

-------------------------------------------------------------------------------

HttpServletRequest 请求对象,获取客户端提交的数据

1.获取客户端表单提交过来的数据

键 值

请求 POST /day03/RegisterServlet HTTP/1.1

Accept text/html, application/xhtml+xml, \*/\*

Referer http://localhost:8080/day03/register.html

Accept-Language zh-CN

User-Agent Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko

Content-Type application/x-www-form-urlencoded

Accept-Encoding gzip, deflate

Host localhost:8080

Content-Length 167

DNT 1

Connection Keep-Alive

Cache-Control no-cache

实体数据:

username=zhangsan&password=&gender=%E7%94%B7&province=%E5%8C%97%E4%BA%AC&habby=%E4%B8%8A%E7%BD%91&habby=%E4%BF%9D%E5%81%A5&info=%09%09%09%09%09%09%0D%0A%09%09%09%09%09

request也有中文乱码问题，当浏览器提交过来的数据含有中文时，用request.getParamter()去获取时

就会有中文乱码，为什么?

因为tomcat接收数据时用的是request的getParameter方法，该方法返回一个字符串，getParameter返回的

字符串采用的是iso-8859-1。如果浏览端发送请求时采用是非iso-8859-1这种编码就是会有乱码问题。

如何request对象在接收中文时的乱码问题？

Post方式中文乱码解决方案:

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

在服务器端，将username按照ISO-8859-1还原成字节数组

然后我们利用utf-8编码，重新生成字符串

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

byte[] buffer=username.getBytes("ISO-8859-1");

username=new String(buffer,"utf-8");

(2)在使用request对象之前 request.setCharacterEncoding("utf-8");只解决post方式提交的数据乱码

Get请求方式:

是在地址栏中传递数据

http://localhost:8080/day03/RegisterServlet2?username=%E5%BC%A0%E4%B8%89&password=123456&gender=%E7%94%B7&province=%E9%87%8D%E5%BA%86&habby=%E4%B8%8A%E7%BD%91&habby=%E6%B3%A1%E5%A6%9E&habby=%E8%B4%AD%E7%89%A9&habby=%E4%BF%9D%E5%81%A5&info=%E4%B8%AD%E7%A7%8B%E6%94%BE%E5%81%87%09%09%09%09%09%09%0D%0A%09%09%09%09%09

对于不敏感数据可以get方式，对于敏感数据应用Post方式

对get方式传递的数据有大小限制，不同的浏览器规定的大小也尽相同。如例以IE为例get方式最大就是2KB,

多于2KB的剩于数据，默认浏览器会丢弃掉.而post方式是没有大小制度的。

GET方式中文乱解决方案:

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

在服务器端，将username按照ISO-8859-1还原成字节数组

然后我们利用utf-8编码，重新生成字符串

(2)修改tomcat默认解码集。

配置 tomcat/server.xml 在<Connector> 添加 URIEncoding="utf-8"

<Connector connectionTimeout="20000" port="8080" protocol="HTTP/1.1"

redirectPort="8443" URIEncoding="utf-8"/>

HttpServletRequest对象也能进行资源跳转(内部转发)

http://localhost:8080/day03/DemoForwardOne它执行了doGet方式，doGet又内部转发到了DemoForwardoTwo

执行DemoForwardoTwo中的doGet方法，所以在页面显示结果(我是DemoForwardOTwo)

内部转发的特点:

(1)地址栏不会发生变化

(2)可以利用request对象进行数据的传递，并且两个Servlet用的是同一套request和response对象

request对象可以调置键值对

setAttribute 向request域添加键值对

getAttribute 获取reqeust域获取指定key的值

removeAttribute 删除reqeust域中指定的key和值

内部转发和重定向的区别?

requestDispatcher.forward方法只能将请求转发给同一个WEB应用中的其它组件；

而HttpServletResponse.sendRedirect 方法还可以重定向到同一个站点上的其他应用程序中的资源，

甚至是使用绝对URL重定向到其他站点的资源。

如果传递给HttpServletResponse.sendRedirect 方法的相对URL以"/"开头，它是相对于当前WEB应用程序的根目录。

HttpServletResponse.sendRedirect方法对浏览器的请求直接作出响应，响应的结果就是告诉浏览器去重新发出

对另外一个URL的访问请求；

RequestDispatcher.forward方法在服务器端内部将请求转发给另外一个资源，

浏览器只知道发出了请求并得到了响应结果，并不知道在服务器程序内部发生了转发行为。

requestDispatcher.forward方法的调用者与被调用者之间共享相同的request对象和response对象，

它们属于同一个访问请求和响应过程；而HttpServletResponse.sendRedirect方法调用者与被调用者使用

各自的request对象和response对象，它们属于两个独立的访问请求和响应过程。

HttpServletRequest和HttpServletResponse对象是一对

HttpServletRequest和HttpServletRespons生命周期？

当在浏览器中输入http://localhost:8080/day03/DemoForwardOne回车，意味着向服务发了出了请求，

服务接收该请求之后，就立即创建与该请求有关的HttpServletRequest和HttpServletRespons对象

当命使用HttpServletRespons对象向浏览器端输出内容完毕后，HttpServletRequest和HttpServletRespons对象

就会被服务器销毁掉。

如果我再次访问http://localhost:8080/day03/DemoForwardOne回车,就意味着向服务器发出了新的请求

服务接收该请求之后，就立即创建与该请求有关的HttpServletRequest和HttpServletRespons对象

当命使用HttpServletRespons对象向浏览器端输出内容完毕后，HttpServletRequest和HttpServletRespons对象

就会被服务器销毁掉。