response对象：

response对象是哪一个接口的实例

response的主要作用

response：表示服务器端对客户端的回应，response本身是HttpServletResponse接口的对象，而HttpServletRespones本身是ServletResponse的子接口，和request对象时非常相识的。

response本身存在大量的方法，但是最主要的三种操作：设置头信息，跳转，Cookie。

response常用的方法：

|  |  |  |
| --- | --- | --- |
| **NO** | **方法** | **描述** |
| 1 | void setHeader(String name, String value) | 设置头信息的名字和内容 |
| 2 | void sendRedirect(String location) throws IOException | 跳转 |
| 3 | void addCookie(Cookie cookie) | 向客户端加入Cookie |
| 4 | void setContentType(String type) | 设置返回类型 |

头信息：

什么是头信息？

头信息就是随着内容一起提交到服务器，又从服务器随着内容一起返回发送到客户端

在jsp中可以直接使用request对象，取得头信息的全部内容，在HttpServletRequest中定义了一个方法，用于取得头信息：

·取得头信息的名称

|  |
| --- |
| Enumeration getHeaderNames() |

·取得每一个头信息的内容：

|  |
| --- |
| String getHeader(String name) |

现在我们通过代码，取得一个客户端发送给服务器的全部头信息

|  |
| --- |
| <%@page contentType=*"text/html; charset=GBK"*%>  <%@page import=*"java.util.\*"*%>  <%  //取得头信息名称  Enumeration em = request.getHeaderNames();    **while**(em.hasMoreElements()){  String name = (String)em.nextElement();  %>  <h3><%=name%>----<%=request.getHeader(name)%></h3>  <%  }  %> |

第一次运行：

|  |
| --- |
| **accept----image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, \*/\***  **referer----http://localhost/demoWeb/responseDemo/**  **accept-language----zh-cn**  **accept-encoding----gzip, deflate**  **user-agent----Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; InfoPath.2)**  **host----localhost**  **connection----Keep-Alive** |

第二次运行：

|  |
| --- |
| accept----\*/\*referer----http://localhost/demoWeb/responseDemo/accept-language----zh-cnaccept-encoding----gzip, deflateuser-agent----Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; InfoPath.2)host----localhostconnection----Keep-Alivecookie----JSESSIONID=05D11FDD2653B82386F9F3D58FF2A98B |

设置头信息：

以上取得的头信息都是Http自己提交过去的，用户也可以自己设置头信息，回应给客户端，但是头信息有这么多，头信息中最有用的就是定时刷新，如果想要定时刷新，就需要设置一个头信息的名称“refresh“

范例：

|  |
| --- |
| <%@page contentType=*"text/html; charset=GBK"*%>  <%@page import=*"java.util.\*"*%>  <%!  **int** temp = 0;  %>  <%    response.setHeader("refresh","1");  %>  <h1><%=temp++ %></h1> |

定时刷新，经过适当的设置，可以完成定时跳转的功能，可以让一个页面经过多少次之后跳转到另一个页面

|  |
| --- |
| <%@page contentType=*"text/html; charset=GBK"*%>  <%@page import=*"java.util.\*"*%>  <%  //设置5秒后定时跳转到其他页面  response.setHeader("refresh","5;URL=responseDemo01.jsp");  %>  <h2>本页面5秒后自动跳转到\*\*网站</h2>  <h2>如果没有跳转，请<a href=*"responseDemo01.jsp"*>点击这里</a></h2> |

定时跳转了之后，地址栏发生了改变，所以此跳转称为客户端跳转，当然在HTTP中也提供了这样跳转的功能，也是可以通过通信息的设置方式完成

|  |
| --- |
| <meta http-equiv=*"refresh"* content=*"5;URL=responseDemo01.jsp"*>  <h2>本页面5秒后自动跳转到\*\*网站</h2>  <h2>如果没有跳转，请<a href=*"responseDemo01.jsp"*>点击这里</a></h2> |

跳转：

在response中提供了专门的跳转执行，使用sendRedirect()完成。

范例：

|  |
| --- |
| <%@page contentType=*"text/html; charset=GBK"*%>  <%  response.sendRedirect("responseDemo01.jsp");  %> |

发现成功跳转后，地址栏已经改变，所以称这种跳转为客户端跳转。

这种跳转方式也可以通过URL重写方式向页面传递参数

·两种跳转的区别：

已经学过的两种跳转：

服务端跳转：<jsp:forward page=”页面”>

客户端跳转：response. sendRedirect(),setHeader()

通过以下代码观察两种跳转的区别：客户端

|  |
| --- |
| <%@page contentType=*"text/html; charset=GBK"*%>  <%  System.out.println("这是客户端跳转前");    %>  <%  response.sendRedirect("responseDemo09.jsp");  %>  <%  System.out.println("这是客户端跳转后");    %> |

服务端：

|  |
| --- |
| <%@page contentType=*"text/html; charset=GBK"*%>  <%  System.out.println("这是服务端跳转前");  %>  <jsp:forward page=*"responseDemo09.jsp"*></jsp:forward>  <%  System.out.println("这是服务端跳转后");  %> |

我们发现，客户端跳转是处理完页面后再进行跳转，而forward跳转属于无条件跳转，只要执行到这条语句，就立马进行跳转，但是使用forward可以接收request范围属性

Cookie:

Cookie的原本含义是“小甜饼干“，是服务端保存在客户端的一组数据。

但是由于Cookie保存在客户端，所以安全性较差。

jsp中，Cookie属于一个类，是在javax.servlet.http下面的一个类:

主要方法：

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **方法** | **类型** | **描述** |
| 1 | public Cookie(String name, String value) | 构造 | 实例对象 名称和值 |
| 2 | public String getName() | 普通 | 取得Cookie的名字 |
| 3 | public String getValue() | 普通 | 取得Cookie的值 |
| 4 | public void setValue(String newValue) | 普通 | 修改Cookie的值 |
| 5 | public void setMaxAge(int expiry) | 普通 | Cookie保存的最大时间 |

如果想要设置Cookie，则肯定要是用response对象

向客户端增加Cookie：

|  |
| --- |
| <%@page contentType=*"text/html; charset=GBK"*%>  <%  Cookie ck1 = **new** Cookie("username","JJM");  Cookie ck2 = **new** Cookie("password","jjm123");  response.addCookie(ck1);  response.addCookie(ck2);  %> |

如果想要取得Cookie，则肯定要使用request对象，

取得Cookie：

|  |
| --- |
| Cookie[] getCookies() |

范例：

|  |
| --- |
| <%@page contentType=*"text/html; charset=GBK"*%>  <%  Cookie cook[] = request.getCookies();  **for**(**int** i=0 ;i<cook.length;i++){  %>  <h3><%=cook[i].getName()%>-----<%=cook[i].getValue() %></h3>  <%  }  %> |

以上代码我们已经证明了可以设置Cookie，为什么我们设置了两个Cookie，而取得的确实三个，实际上JSESSIONID是服务端自动设置到客户端上面去的，表示的就是一个普通的用户，每一个JSESSONID都是不一样的，服务器就是依靠这个东西来区分不同的用户，如果想要深一步了解，就要等待下午的session对象的讲解。

但是我们发现，如果浏览器关闭，在开启的话，Cookie不能取得，而且报错，这是因为一个Cookie默认情况下只能保存在当前的页面，所以打开新的浏览器就不能再取的Cookie，为了解决这个问题，而实现Cookie真正的保存较长时间，必须使用设置Cookie 的最大保存时间。

|  |
| --- |
| <%@page contentType=*"text/html; charset=GBK"*%>  <%  Cookie ck1 = **new** Cookie("username","JJM");  Cookie ck2 = **new** Cookie("password","jjm123");  ck1.setMaxAge(60);  ck2.setMaxAge(30);  response.addCookie(ck1);  response.addCookie(ck2);    %> |

现在已经把Cookie保存了较长时间，Cookie实际保存的位置：C:\Documents and Settings\Administrator\Cookies

总结：设置Cookie和设置Cookie的保存时间

掌握两种跳转及区别

登陆：

|  |
| --- |
| <%@page contentType=*"text/html; charset=GBK"*%>  <%  Cookie cook[] = request.getCookies();  **if**(cook == **null**||cook.length==1){  %>  <center>  <h1>登陆程序</h1>  <form id=*"form1"* action=*"check.jsp"* method=*"post"*>  <table>  <tr>  <td>用户名：<input type=*"text"* name=*"username"*/></td>  </tr>  <tr>  <td>密&nbsp;&nbsp;码：<input type=*"text"* name=*"password"*/></td>  </tr>  <tr>  <td align=*"center"*>  <input type=*"submit"* value=*"登陆"*/>  <input type=*"reset"* value=*"重置"*/>  </td>  </tr>  </table>  </form>  </center>  <%    }**else**{  String username = cook[0].getValue();  String password = cook[1].getValue();  %>  <jsp:forward page=*"check.jsp"*>  <jsp:param value="<%=username%>" name=*"username"*/>  <jsp:param value="<%=password%>" name=*"password"*/>  </jsp:forward>  <%  }  %> |
| <%@page contentType=*"text/html; charset=GBK"*%>  <%  String username = request.getParameter("username");  String password = request.getParameter("password");  **if** (username.equals("jjm") && password.equals("jjm123")) {  Cookie cook[] = request.getCookies();  **if** (cook == **null** || cook.length == 1) {  Cookie ck1 = **new** Cookie("username", username);  Cookie ck2 = **new** Cookie("password", password);  ck1.setMaxAge(60);  ck2.setMaxAge(60);  response.addCookie(ck1);  response.addCookie(ck2);  }  %>  <h3>  登陆成功  </h3>  <%  } **else** {  %>  <h3>  登陆失败  </h3>  <%  }  %> |