# Servlet

1. Request。getParameter（s：请求头中的参数名称） 调用请求对象读取请求头中的参数，得到详细信息

# Html

1. HttpServletRequest request 请求协议包

Request 。 getParameterNames（）；读取当前请求包中所有的请求参数，并把所有参数封装到一个枚举中。

Enumeration parameterNames = request.getParameterNames();

2. request.getSession(); 拿到当前属性的柜子

HttpServletRequest于HttpServletResponse 详解

3 . session.getAttributeNames() 把session中所有共享数据名取出来，放到枚举中

4. response.sendRedirect(url（地址）);由响应对象将地址写入响应头中location属性中

5.1request.getRequestDispatcher(地址) 调用请求对象生成一个【资源申请调度对象】

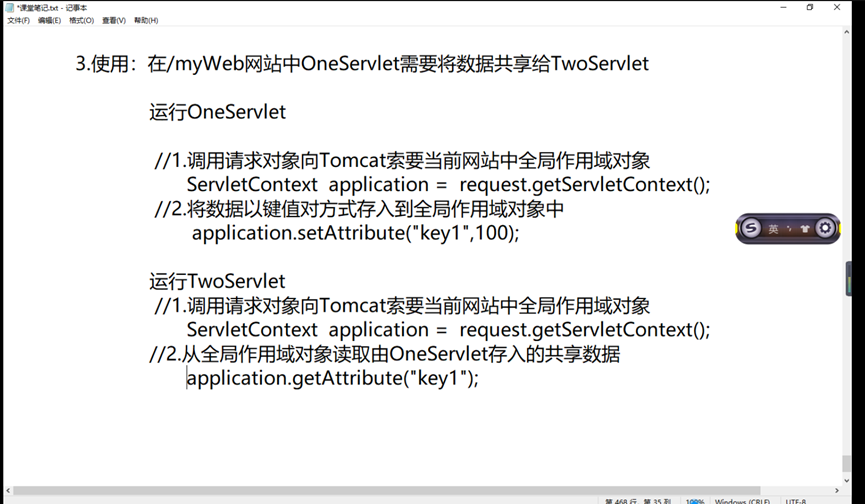
5.2report.forward(request请求，response响应);将【资源申请调度对象】发送给tomcat

6.1 request.getServletContext(); 调用请求对象向Tomcat索要当前网站中全局作用域对象

6.2application.setAttribute(“key1”,”value”)将数据以键值对方式存入到全局作用域对象中

6.3application.getAttribute(“key1”) 从全局作用域对象读取由Servlet存入的共享数据

7.innerHTML 属性用于获取或替换 HTML 元素的内容。



<http://c.biancheng.net/view/8040.html>

共享模式

1. ServletContext接口 （全局作用域对象） 实现类：ConcurrentHashMap 线程安全的

## HttpServletRequest 详解

请求

HttpServletRequest对象代表客户端的请求，当客户端通过HTTP协议访问服务器时，HTTP请求头中的所有信息都封装在这个对象中，通过这个对象提供的方法，可以获得客户端请求的所有信息。

获得客户机请求参数

getParameter(String name) 根据name获取请求参数(常用)

getParameterValues(String name) 根据name获取请求参数列表(常用)

getParameterMap() 返回的是一个Map类型的值，该返回值记录着前端（如jsp页面）所提交请求中的请求参数和请求参数值的映射关系。(编写框架时常用)

## HttpServletResponse 对象

响应

与 HttpServletRequest 相反，HttpServletResponse 对象是专门用来封装 HTTP 响应消息的。客户端发出了请求，得对人家有所回应呀，HttpServletResponse 就是做这项工作的。  
  
由于 HTTP 响应消息分为状态行、响应消息头、响应消息体三部分，因此，在 HttpServletResponse 接口中定义了向客户端发送响应状态码、响应消息头、响应消息体的方法。

#### 1) 发送状态码相关的方法

① void setStatus(int status)  
该方法用于设置 Http 响应消息的状态码，并生成响应状态行。  
  
由于响应状态行中的状态描述信息直接与状态码相关，而 HTTP 版本由服务器确定，因此，只要通过 setStatus() 方法设置了状态码，即可实现状态行的发送。

需要注意的是，在正常情况下，Web 服务器会默认产生一个状态码为 200 的状态行。

② void sendError(int sc)  
该方法用于发送表示错误信息的状态码。例如，404 状态码表示找不到客户端请求的资源。  
  
HttpServletResponse 对象提供了两个重载的 sendError(int sc) 方法，具体如下：

public void sendError(int code) throws java.io.IOException  
public void sendError(int code,String message)throws java.io.IOException

在上面重载的两个方法中，第一个方法只发送错误信息的状态码，而第二个方法除了发送状态码以外，还可以增加一条用于提示说明的文本信息，该文本信息将出现在发送给客户端的正文内容中。

#### 2) 发送响应消息头相关的方法

|  |  |
| --- | --- |
| **方法声明** | **功能描述** |
| void addHeader(String name,String value) | 这两个方法都是用来设置 HTTP 协议的响应头字段，其中：   * 参数 name 用于指定响应头字段的名称； * 参数 value 用于指定响应头字段的值。   不同的是，addHeader() 方法可以增加同名的响应头字段，而 setHeader() 方法则会覆盖同名的头字段。 |
| void setHeader(String name,String value) |
| void addIntHeader(String name,int value) | 这两个方法专门用于设置包含整数值的响应头，避免了「使用 addHeader() 与 setHeader() 方法时需要将 int 类型的设置值转换为 string 类型」的问题。 |
| void setIntHeader(String name,int value) |
| void setContentLength(int len) | 该方法用于设置响应消息的实体内容的大小，单位为字节。对于 HTTP 协议来说，这个方法就是设置 Content-Length 响应头字段的值。 |
| void setContentType(String type) | 该方法用于设置 Servlet 输出内容的 MIME 类型。对于 HTTP 协议来说，就是设置 Content-Type 响应头字段的值。举例说明：   * 如果发送到客户端的内容是 jpeg 格式的图像数据，就需要将响应头字段的类型设置为image/jpeg； * 如果响应的内容为文本，则该方法还可以设置字符编码，例如text/html;charset=UTF-8。 |
| void setLocale(Locale loc) | 该方法用于设置响应消息的本地化消息。对 HTTP 来说，就是设置 Content-Language 响头字段和 Content-Type 头字段中的字符集编码部分。  需要注意的是，如果 HTTP 消息没有设置 Content-Type 头字段，setLocale() 方法设置的字符编码则没有效果，如果调用 setCharacterEncoding() 或 setContentType() 方法指定了响应头内容的字符集编码，则 setLocale() 方法将不再具有指定字符集编码的功能。 |
| void setCharacterEncoding(String charset) | 该方法用于设置 out 输出流中所采用的编码。对 HTTP 协议来说，就是设置 Content-Type 头字段中的字符集编码部分。如果没有设置 Content-type 头字段，setCharacterEncoding() 方法设置的字符集编码则没有效果。  该方法的优先级比setContentType()和setLocale()方法要高，它的设置结果将覆盖setContentType()和setLocale()方法所设置的字符编码。 |

#### 3) 发送响应消息体相关的方法

① ServletOutputStream getOutputStream()方法  
该方法所获取的字节输出流对象为 ServletOutputStream 类型。  
  
由于 ServletOutputStream 是 OutputStream 的子类，它可以直接输出字节数组中的二进制数据，因此，要想输出二进制格式的响应正文，就需要使用 getOutputStream() 方法。  
  
② PrintWriter getWriter()方法  
它返回了一个可以向客户端发送文本的的 Java.io.PrintWriter 对象。  
  
由于 PrintWriter 类型的对象可以直接输出字符文本内容，因此，要想输出内容全为字符文本的网页文档，需要使用 getWriter() 方法。

默认情况下，PrintWriter 对象使用 ISO-8859-1 编码（该编码在输入中文时会发生乱码，乱码问题后面章节会讲到）。

注意：虽然HttpServletResponse 对象的 getOutSream() 和 getWriter() 方法都可以发送响应消息体，但是他们之间相互排斥，不可以同时使用，否则会发生异常。

# Cookie

1. 知识点1：setMaxAge(int seconds)

1、该方法用于设置cookie的生存时间，传入的参数表示生存时间，是int型的秒数值