

HttpServletResponse

教学导航

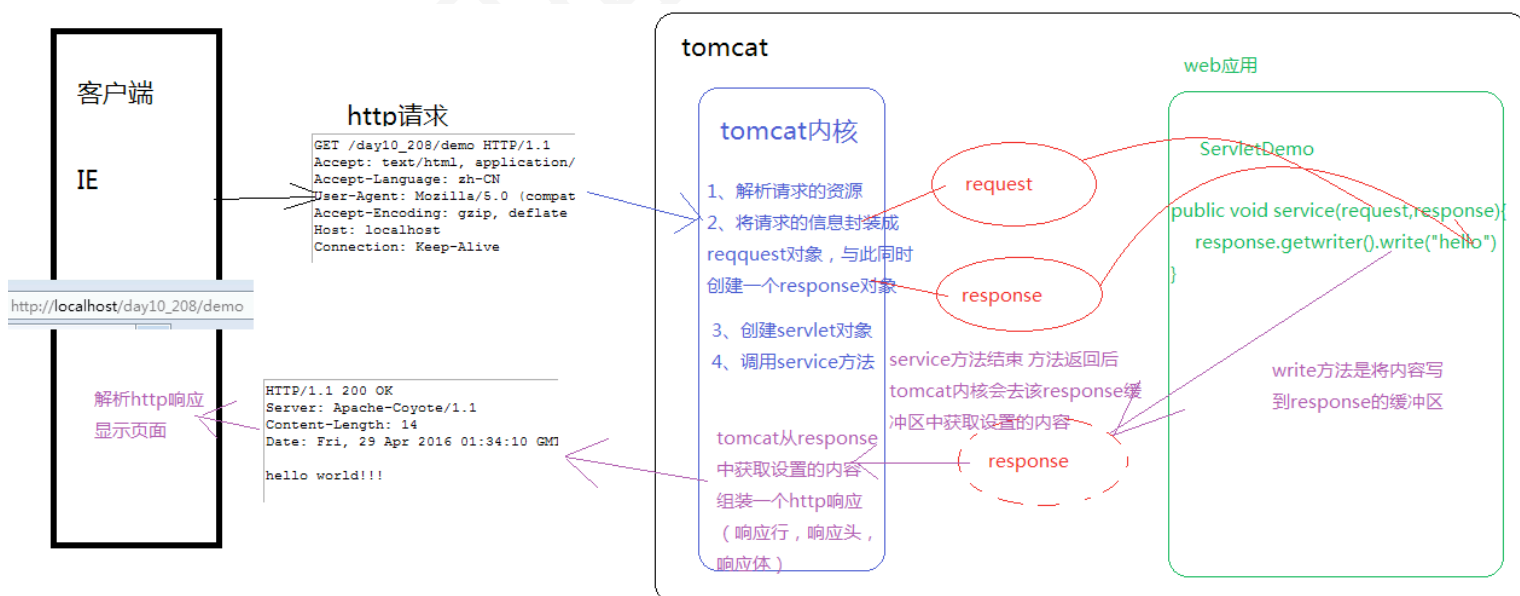
教学目标	案例一、完成文件下载 案例二、生成验证码（了解）
教学方法	

1. HttpServletResponse 概述

我们在创建 Servlet 时会覆盖 service()方法 ,或 doGet()/doPost(),这些方法都有两个参数，一个为代表请求的 request 和代表响应 response。

service 方法中的 response 的类型是 ServletResponse ,而 doGet/doPost 方法的 response 的类型是 HttpServletResponse ,HttpServletResponse 是 ServletResponse 的子接口，功能和方法更加强大，今天我们学习 HttpServletResponse。

2. response 的运行流程





3. 通过抓包工具抓取 Http 响应

Http响应

HTTP/1.1 200 OK

Http响应行

```
Server: Apache-Coyote/1.1
Accept-Ranges: bytes
ETag: W/"312-1467289802502"
Last-Modified: Thu, 30 Jun 2016 12:30:02 GMT
Content-Type: text/html
Content-Length: 312
Date: Thu, 30 Jun 2016 12:31:12 GMT
```

Http响应头

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
  <form action="#" method="post">
    <input type="text" name="username"><br>
    <input type="password" name="password"><br>
    <input type="submit" value="嫵嫵嫵http璇锋脛"><br>
  </form>
</body>
</html>
```

Http响应体

因为 response 代表响应，所以我们可以通过该对象分别设置 Http 响应的响应行，响应头和响应体

4. 通过 response 设置响应行

设置响应行的状态码

setStatus(int sc)

5. 通过 response 设置响应头

`addHeader(String name, String value)`

`addIntHeader(String name, int value)`

`addDateHeader(String name, long date)`

`setHeader(String name, String value)`

`setDateHeader(String name, long date)`

`setIntHeader(String name, int value)`

其中，add 表示添加，而 set 表示设置

6. 通过 response 设置响应体

(1) 响应体设置文本

`PrintWriter getWriter()`

获得字符流，通过字符流的 `write(String s)` 方法可以将字符串设置到 response 缓冲区中，随后 Tomcat 会将 response 缓冲区中的内容组装成 Http 响应返回给浏览器端。

关于设置中文的乱码问题

原因：response 缓冲区的默认编码是 iso8859-1，此码表中没有中文，可以通过 response 的 `setCharacterEncoding(String charset)` 设置 response 的编码

但我们发现客户端还是不能正常显示文字

原因：我们将 response 缓冲区的编码设置成 UTF-8，但浏览器的默认编码是本地系统的编码，因为我们都是中文系统，所以客户端浏览器的默认编码是 GBK，我们可以手动修改浏览器的编码是 UTF-8。

我们还可以在代码中指定浏览器解析页面的编码方式，

通过 response 的 `setContentType(String type)` 方法指定页面解析时的编码是 UTF-8
`response.setContentType("text/html;charset=UTF-8");`

上面的代码不仅可以指定浏览器解析页面时的编码，同时也内含

`setCharacterEncoding` 的功能，所以在实际开发中只要编写
`response.setContentType("text/html;charset=UTF-8");`

就可以解决页面输出中文乱码问题。

(2) 响应头设置字节

ServletOutputStream getOutputStream()

获得字节流，通过该字节流的 `write(byte[] bytes)` 可以向 `response` 缓冲区中写入字节，在由 Tomcat 服务器将字节内容组成 Http 响应返回给浏览器。

7. 案例-完成文件下载

文件下载的实质就是文件拷贝，将文件从服务器端拷贝到浏览器端。所以文件下载需要 IO 技术将服务器端的文件使用 `InputStream` 读取到，在使用 `ServletOutputStream` 写到 `response` 缓冲区中

代码如下：

```
//1、获得download文件夹下的a.mp3文件的绝对路径
String path = getServletContext().getRealPath("download/a.jpg");
//2、使用io读取到该文件
InputStream in = new FileInputStream(path);
//3、将该文件写到response的缓冲区中
OutputStream out = response.getOutputStream();
int len = 0;
byte[] buffer = new byte[1024];
while((len=in.read(buffer))>0){
    out.write(buffer, 0, len);
}
//4、关闭资源
in.close();
//out.close();//response获得流不用手动关闭 会自动关闭
```

上述代码可以将图片从服务器端传输到浏览器，但浏览器直接解析图片显示在页面上，而不是提供下载，我们需要设置两个响应头，告知浏览器文件的类型和文件的打开方式。

1) 告知浏览器文件的类型：response.setContentType(文件的 MIME 类型);

2) 告知浏览器文件的打开方式是下载：

response.setHeader("Content-Disposition","attachment;filename=文件名称");

代码如下：

//0、设置两个文件下载的头

```
response.setContentType(getServletContext().getMimeType("a.jpg"));
response.setHeader("Content-Disposition", "attachment;filename=a.jpg");
```

//1、获得download文件夹下的a.mp3文件的绝对路径

```
String path = getServletContext().getRealPath("download/a.jpg");
```

//2、使用io读取到该文件

```
InputStream in = new FileInputStream(path);
```

//3、将该文件写到response的缓冲区中

```
OutputStream out = response.getOutputStream();
```

```
int len = 0;
```

```
byte[] buffer = new byte[1024];
```

```
while((len=in.read(buffer))>0){
    out.write(buffer, 0, len);
```

```
}
```

//4、关闭资源

```
in.close();
```

```
//out.close();//response获得流不用手动关闭 会自动关闭
```

但是，如果下载中文文件，页面在下载时会出现中文乱码或不能显示文件名的情况，原因是不同的浏览器默认对下载文件的编码方式不同，ie 是 UTF-8 编码方式，而火狐浏览器是 Base64 编码方式。所以这里需要解决浏览器兼容性问题，解决浏览器兼容性问题的首要任务是要辨别访问者是 ie 还是火狐（其他），通过 Http 请求体中的一个属性可以辨别

```
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)
```

```
User-Agent: "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0"
```



解决乱码方法如下（不要记忆--了解）：

```
if (agent.contains("MSIE")) {  
    // IE 浏览器  
    filename = URLEncoder.encode(filename, "utf-8");  
    filename = filename.replace("+", " ");  
} else if (agent.contains("Firefox")) {  
    // 火狐浏览器  
    BASE64Encoder base64Encoder = new BASE64Encoder();  
    filename = "?utf-8?B?"  
        + base64Encoder.encode(filename.getBytes("utf-8")) + "?=";  
} else {  
    // 其它浏览器  
    filename = URLEncoder.encode(filename, "utf-8");  
}
```

其中 agent 就是请求头 User-Agent 的值

```
String filename = "美女.jpg";  
  
//获得浏览器的User-Agent  
String agent = request.getHeader("User-Agent");  
if (agent.contains("MSIE")) {  
    // IE浏览器  
    filename = URLEncoder.encode(filename, "utf-8");  
    filename = filename.replace("+", " ");  
} else if (agent.contains("Firefox")) {  
    // 火狐浏览器  
    BASE64Encoder base64Encoder = new BASE64Encoder();  
    filename = "?utf-8?B?"  
        + base64Encoder.encode(filename.getBytes("utf-8")) + "?=";  
} else {  
    // 其它浏览器  
    filename = URLEncoder.encode(filename, "utf-8");  
}  
  
//0、设置两个文件下载的头  
response.setContentType(getServletContext().getMimeType("美女.jpg"));  
response.setHeader("Content-Disposition", "attachment;filename="+filename);
```



黑马程序员
www.itheima.com

传智播客旗下
高端IT教育品牌

改变中国IT教育，我们正在行动

传智播客Java学院