

# Java EE 企业应用系统设计

## HTTP 响应处理编程

王晓东

[wangxiaodong@ouc.edu.cn](mailto:wangxiaodong@ouc.edu.cn)

中国海洋大学

November 30, 2017



# 参考书目

1. 吕海东，张坤编著，Java EE 企业级应用开发实例教程，清华大学出版社，2010 年 8 月



# 大纲

HTTP 响应的内容

HTTP 响应对象

响应对象功能和方法



# 接下来...

HTTP 响应的内容

HTTP 响应对象

响应对象功能和方法



# HTTP 响应的内容

在 Web 服务器接收请求处理后，向客户端发送 HTTP 响应 (Response)。响应的内容包括：

- ▶ 响应状态 (Status Code)
- ▶ 响应头 (Response Header)
- ▶ 响应体 (Response Body)

Java Web 中响应由响应对象 Response 完成。



# HTTP 响应状态行

表明响应的状态信息，如成功、失败、错误。

状态行组成：版本 / 状态代码 / 状态消息。

状态行例子：

HTTP/1.1 200 ok

1. 版本：使用的 HTTP 协议版本，如 HTTP/1.1；
2. 状态代码：3 位数字；
  - ▶ 1xx: 收到请求，没有处理完。
  - ▶ 2xx: 成功，响应完毕。
  - ▶ 3xx: 重定向，到另一个请求中去。
  - ▶ 4xx: 失败，没有请求的文档等。
  - ▶ 5xx: 内部错误，代码出现异常。
3. 状态描述。



# 响应头

Web 服务器在向客户端发送 HTTP 响应时也可以包含响应头，赖指示客户端如何处理响应体，主要用来告诉浏览器响应的类型、字符编码和字节大小等信息，以使得浏览器知道如何处理响应内容。

常见响应头内容：

1. 指示 HTTP 响应可以接收到的文档类型集：Accept
2. 告知客户可以接收的字符集：Accept-Charset
3. 所有响应的字符编码集：Accept-Encoding
4. 响应体的 MIME 类型：Content-Type
5. 响应体的语言类型：Context-Language
6. 响应体的长度和字节数：Content-Length
7. 通知客户端到期时间：Expires
8. 缓存情况：Cache-Control
9. 重定向到另一个 URL 地址：Redirect



## 响应体

响应体类型由响应头确定，可以使任何类型。浏览器在处理响应体之前，会收到响应头，根据响应头的信息，确定如何处理响应体，如响应头的 Content-Type 为 PDF，则浏览器会启动 PDF Reader 来处理此响应体以显示 PDF 文档。

常用类型包括：

1. 纯文本：text/plain
2. HTML：text/html
3. 图片：image/gif, image/jpeg
4. PDF：application/pdf

注意：

- ▶ 文本类型响应要求响应头中包含 MIME 类型和字符编码集，使用字符输出流向客户端发送响应体数据；
- ▶ 二进制数据类型响应需要在响应头中包含 MIME 类型，不要设置字符编码集，使用字节输出流向客户端发送响应体数据。





# 接下来...

HTTP 响应的内容

HTTP 响应对象

响应对象功能和方法



# 响应对象类型

## ❖ 响应对象类型

`javax.servlet.http.HttpServletResponse`

## ❖ 响应对象职责

- ▶ 设置状态行；
- ▶ 发送响应头；
- ▶ 向 Web 浏览器发送 HTTP 响应体；
- ▶ 控制页面的重定向，即将告知浏览器再发送一次请求。



# 响应对象生命周期

1. Web 容器自动为每次 Web 组件的请求生成一个响应对象。
2. Web 容器创建响应对象后，传入到 doGet 或 doPost 方法。
3. 通过响应对象向浏览器发响应。
4. 响应结束后，Web 容器销毁响应对象，释放所占用的内存。



# 接下来...

HTTP 响应的内容

HTTP 响应对象

响应对象功能和方法



## 设置响应状态码

一般情况下，Web 开发人员不需要通过编程来改变响应状态码，Web 服务器会根据请求处理的情况自动设置状态码，并发送到客户端浏览器。例如，当客户请求不存在的 URL 地址时，Web 服务器会自动设置状态码为 404，状态消息为 not found。

❖ `public void setStatus(int code)`

直接发送指定的响应状态码，没有设置状态消息，只有默认的状态消息，如果无对应状态消息则显示为空。

❖ `public void setStatus(int code, String message)`

设置指定的状态码，同时设定自定义的状态消息，可以修改默认的状态消息。该方法在 Servlet 2.5 后被舍弃，一般不要使用。



## 设置响应状态码

❖ `public void sendError(int sc) throws IOException`

向客户端发送指定的错误信息码，可以是任意定义的整数。

```
1 response.setCharacterEncoding("GBK");  
2 response.sendError(580);
```

❖ `public void sendError(int sc, String msg) throws IOException`

向客户端发送指定的错误信息码和自定义状态消息。

```
1 response.setCharacterEncoding("GBK");  
2 response.sendError(580, "自定义错误");
```



## 设置响应头

当客户端接收到响应状态为 200 时，浏览器会继续接收响应头信息，来确定响应体的类型和大小。

❖ `public void setHeader(String name, String value)`

将指定名称和值的响应头发送到客户端。

```
1 response.setHeader("Content-Type", "text/html");
```

❖ `public void setIntHeader(String name, int value)`

设置整数类型的响应头的名和值。

```
1 response.setHeader("Content-Length", 20);
```

实际项目中无需设定该响应头，Web 服务器会自动计算并发送给浏览器。

❖ `public void setDateHeader(String name, long date)`

设定日期类型的响应头，参数 `date` 为 GMT 格式的日期。



## 设置响应头的便捷方法

❖ `public void setContentType(String type)`

直接设置响应内容类型 MIME 响应头。

❖ `public void setContentLength(int len)`

设置响应体长度，以字节为单位。

❖ `void setCharacterEncoding(String charset)`

设置响应字符集，包括响应状态，响应头和响应体。





## 设置响应头的便捷方法

❖ `public void setBufferSize(int size)`

设定响应体的缓存字节数。

如设定响应体缓存为 4k:

```
1 response.setBufferSize(4096);
```

Servlet 在发送响应时，一般按照发送状态码、响应头和响应体的顺序进行，大的响应体缓存，可以允许 Servlet 有更多的时间发送状态码和响应头，这种情况发生在响应头和响应体同时写的情况。

提示：编程的时候最好先把响应头全部设定后，再发送响应体。



## 响应对象方法——向客户端传送 Cookie

❖ `public void addCookie(Cookie cookie);`

此方法功能将 Cookie 对象放置在响应头中，随响应内容到浏览器客户端，并保存到客户端的 PC 的本地目录中。

```
1 Cookie cookie01=new Cookie("userid", "9001");  
2 response.addCookie(cookie01);
```



## 响应对象方法——请求重定向

❖ `public void sendRedirect(String url)`

将对客户的响应重定向到新的 URL 上，让客户端浏览器对此 URL 进行请求。

重定向到登录页面，相当于在浏览器地址栏上再输入一次 URL 地址，进行一次 HTTP 请求：

```
1 String url="../admin/login.jsp";  
2 response.sendRedirect(url);
```



## 设置响应体发送功能

响应体即浏览器实际显示的具体内容，可以是 HTML 网页，也可以是其他文件格式，由响应头的 Content-Type 决定。

响应体的类型主要分为两大类，即文本类型和二进制类型。文本类型使用字符输出流 `PrintWriter` 的对象来实现；二进制类型由 `OutputStream` 的对象来实现。

❖ `public PrintWriter getWriter()`

取得字符输出流。

❖ `public ServletOutputStream getOutputStream()`

取得二进制输出流。



# 设置响应体——文本类型响应体发送编程

## 1. 设置响应类型 ContentType

```
1 response.setContentType("text/html"); //响应类型为 HTML 文档
```

## 2. 设置响应字符编码

```
1 response.setCharacterEncoding("GBK"); //字符编码使用 GBK
```

## 3. 取得字符输出流对象

```
1 PrintWriter out = response.getWriter();
```

## 4. 向流对象中发送文本数据

```
1 out.println("<html><body></body></html>"); //输出文本字符
```

## 5. 清空流中缓存的字符

```
1 out.flush();
```

## 6. 关闭流

```
1 out.close();
```



# 设置响应体——文本类型响应体发送编程

## CODE 示例

```
1 response.setContentType("text/html; charset=gb2312");
2 PrintWriter out = response.getWriter();

4 out.println("<html><head>");
5 out.println("</head><body>");
6 out.println("hello!");
7 out.println("</body></html>");
8 out.flush();
9 out.close();
```



# 设置响应体——二进制类型响应体发送编程

## 1. 设置响应类型 ContentType

```
1 response.setContentType("image/jpeg"); //响应类型为 JPEG 图片
```

## 2. 取得字节输出流对象

```
1 OutputStream out = response.getOutputStream(); //取得字节输出流
```

## 3. 向流对象中发送字节数据

```
1 out.println(200); //输出字节数据
```

## 4. 清空流中缓存的字节

```
1 out.flush();
```

## 5. 关闭流

```
1 out.close();
```

注意：二进制响应编程不需要设置字符编码。







# THE END

wangxiaodong@ouc.edu.cn

