

## 一，上午回顾

HTTPServletresponse

响应消息行

响应消息头

```
//不使用缓存
response.setHeader();
//刷新
response.setHeader("refresh","3;url=.....");
//告诉浏览器下载文件
String filename = path.substring(path.lastIndexOf("\\")+1);
//设置文件名的编码,将不安全的文件名改为utf-8格式
filename = URLEncoder.encode(filename,"utf-8");
response.setHeader("content-
disposition","attachment;filename="+filename);
```

响应消息正文

```
response.setHeader("content-type","text/html;charset=utf-8");告诉浏览器
使用什么编码
response.setCharacterEncoding("utf-8");//告诉服务器应该使用utf-8编码
response.setContentType("text/html;charset=utf-8");是对前俩句话的封装
getWriter().write("string")//字符输出流
getOutputStream().write("string".getBytes());//字节输出流
```

## 二，请求重定向

302告诉客户端要重定向新的资源

将第一个页面需要做的事情全部做完，再去执行重定向需要执行的代码

例如：从servlet4重定向到servlet5

servlet4中doget代码：

```
System.out.println("A:我要借钱!");
System.out.println("B:我没有，但是我可以告诉你谁有!");
//告诉客户端要重定向的资源
response.setStatus(302);
//告诉浏览器要去访问那个URL
response.setHeader("location","/newS/servlet/test5");
System.out.println("A:我去了!");
```

或者

```
response.sendRedirect(“请求重定向的资源路径”);  
    response.sendRedirect("/newS/servlet/test5");  
    System.out.println("A:我去了!");  
servlet5中doGet代码:  
    System.out.println("我有钱, 我借给你!");
```

## response细节

- 1, `getOutputStream`和`getWriter`方法分别用于输出二进制数据、输出文本数据的  
    `servletOutputStream`、`PrintWriter`对象
- 2, `getOutputStream`和`getWriter`这两个方法相互排斥, 调用了其中任何一个方法后, 就不能再调用另一个方法, 会抛异常
- 3, `servlet`程序向`servletOutputStream`或者`printWriter`对象中写入的数据将被  
    `servlet`引擎从  
        `response`里面获取, `servlet`引擎将这些数据当作响应消息的正文, 然后再与响应状态行  
        和各响应头组合后输出到客户端。
- 4, `servlet`的`service`方法结束后, `service`引擎将检查`getWriter`或  
    `getOutputStream`方法返回  
        的输出流对象是否已经调用过`close`方法, 如果没有, `service`引擎将调用  
    `close`方法关闭  
        该输出流对象