

# 通过 HttpServletResponse 在浏览器端下载文件

作者: [Wuhon](#)

原文链接: <https://ld246.com/article/1605076507618>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



## HttpServletResponse

web服务器接收到客户端的http请求，针对这个请求，分别创建一个代表请求的HttpServletRequest对象，代表响应的一个HttpServletResponse负责向浏览器发送数据的方法；

- 如果要获取客户端请求过来的参数：找HttpServletRequest
- 如果要给客户端响应一些信息：找HttpServletResponse-

---

### 向浏览器输出消息

```
response.getWriter().print();
```

### 下载文件

步骤:

1. 要获取下载文件的路径
2. 下载的文件名是啥?
3. 设置想办法让浏览器能够支持下载我们需要的东西
4. 获取下载文件的输入流
5. 创建缓冲区
6. 获取OutputStream对象
7. 将FileInputStream流写入到buffer缓冲区
8. 使用OutputStream将缓冲区中的数据输出到客户端!

```

protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    //获取文件存在的路径
    String realPath = "/Users/wuhongbin/IdeaProjects/JavaWeb_study/servlet_02/target/classes/1.png";
    System.out.println("文件下载路径: " + realPath);
    // 下载的文件名是什么
    String fillName = realPath.substring(realPath.lastIndexOf("/") + 1);
    // 设置想办法能让浏览器支持下载(Content-Disposition)我们需要的东西
    resp.setHeader("Content-Disposition", "attachment;filename="+URLEncoder.encode(fillName, "UTF-8"));
    // 获取下载文件的输入流
    FileInputStream in = new FileInputStream(realPath);
    // 创建缓冲区
    int len = 0;
    byte[] buffer = new byte[1024];
    // 获取OutputStream对象
    ServletOutputStream out = resp.getOutputStream();
    // 将fileInputStream流写入到buffer缓冲区
    while ((len = in.read(buffer)) > 0) {
        // 使用OutputStream流将数据输出到客户端
        out.write(buffer, 0, len);
    }
    // 关闭流
    in.close();
    out.close();
}

```

**这里我们需要注意的是设置浏览器支持的下载**

```

resp.setHeader("ContentDisposition", "attachment;filename="+URLEncoder.encode(fillName, "UTF-8"));

```

---