# HTTP协议

***第一章 HTTP协议的基础***

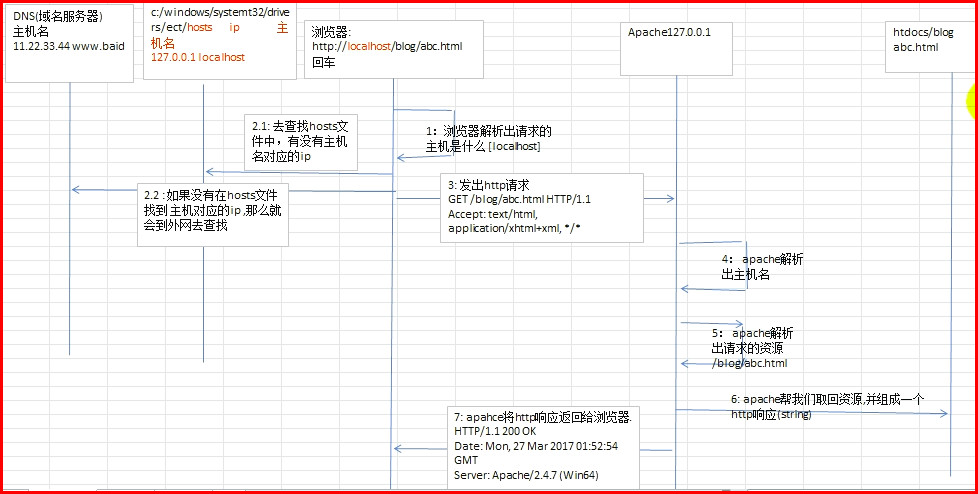
**1.1 问题引出**

问题:当我们在浏览器输入[http://localhost/FY/HTTP/myblog.html](http://localhost/FY/HTTP/test4.php)时,为什么可以看到myblog.html文件的内容



**1.2 浏览器如何显示hello,world!**

采用UML(用户建模语言,描叙一个功能是怎样按照时间的顺序完成的)时序图的方式来描叙一个



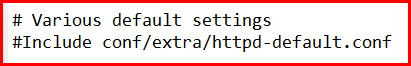
**1.3 通过一个实际的需求**

在我们实际访问网站的过程中,不可能使用<http://localhost:80>的方式去访问网站,实际上使用<http://www.baidu.com> 或<http://www.qq.com>的方式取访问网站,那么这是如何实现的呢?

**1.4 配置虚拟主机(网站)**

步骤:

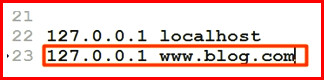
1.打开apache/conf/httpd.conf



2.找到hosts文件

C:/windows/system32/drivers/etc/hosts(windows)

etc/hosts(linux)



3.正式配置虚拟主机文件httpd-vhosts.conf

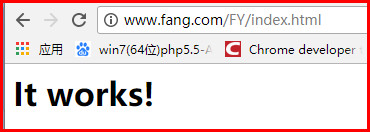
该文件在apache/conf/extra/ httpd-vhosts.conf

基本配置



4.重启apache服务器

5.我们就可以通过域名访问



**1.5 配置虚拟主机的细节说明**

1.如何查看错误日志

到apache目录下的/log/error.log 查看即可

2.如何配置默认首页面



3.如何配置访问某个网站的权限



***第二章 HTTP协议***

**2.1 HTTP协议的介绍**

**http(HyperText Transfer Protocol)协议**表示超文本传输协议,是互联网上最为广泛的一种网络协议.

**http协议是建立在tcp/ip协议的基础上的**,所有的WWW文件都必须遵循这个标准.

设计http最初的目的是为了提供一种发布和接手html页面的方法.

说明:

1.http**协议**表示超文本传输协议

2.http**协议**用于传输文本与图片(等文件)

3.http**协议**是建立在tcp/ip协议的基础上

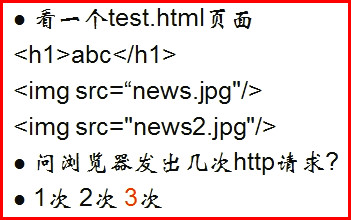
* 通过工具可以抓取http请求内容
* http1.0表示短连接,http1.1表示长连接
* http是tcp/ip协议的一个应用层协议,http也是我们web开发的基础

**2.1 HTTP协议内容的解析**

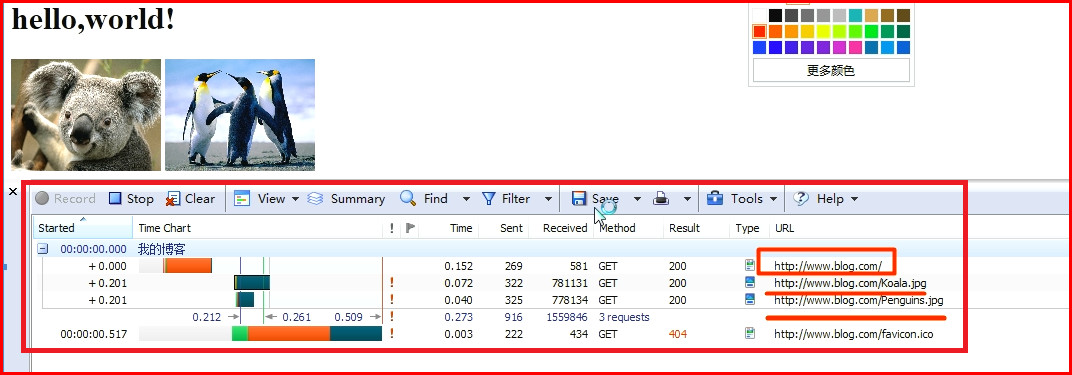
抓取http协议包(请求包,响应包)

谷歌浏览器-->F12开发者模式

* 问题导出:



一共发出3次请求

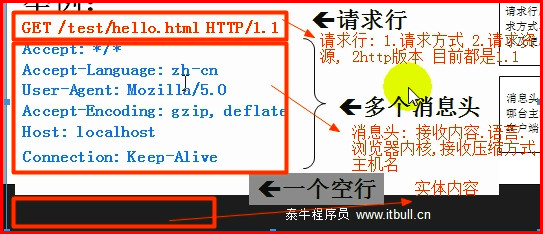


问题:如果我们的test.html还有js,css等等,还需要发请求吗?

回答: 有多少就继续发送多少请求

**2.2 HTTP协议细节-请求**

* 基本介绍
  + **浏览器连上服务器后,向服务器请求某个web资源,称之为浏览器向服务器发送一个http请求**,一个完整的http请求包括如下内容:(其中一些消息头与实体内容是可选的)消息头与实体内容之间用空行隔开:
    - **一个请求行**
    - **若干消息头**
    - **实体内容**

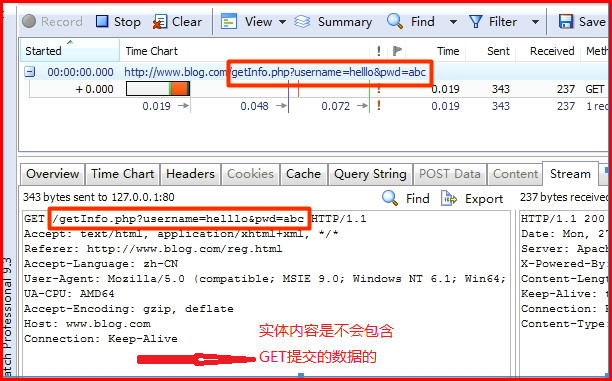


* **HTTP请求—请求行细节**
  + 请求方式说明:
    - POST,GET,HEAD,OPTIONS,DELETE,TRACE,PUT
* ***实际开发中使用最多的是post与get(区别)***

1.GET提交,请求的数据会跟在URL(同一资源定位)之后,而POST提交方式则不会,POST提交的数据会放在HTTP请求实体内容部分

举例说明:

GET提交方式



POST提交方式



小结:从上面的测试可以看出POST的安全性比GET安全性要高

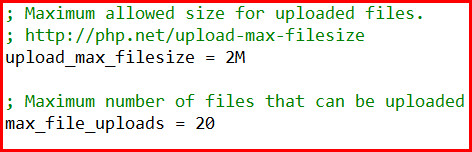
2.两种方式提交数据的长度大小

***HTTP协议本身没有对传输的数据大小进行限制,HTTP协议规范也没有对URL长度进行限制,***在***实际开发中存在的限制***主要有:

* GET:***特定浏览器和服务器对URL长度有限制***,比如IE对GET提交数据时的URL长度限制是2083字节(2k+35),对于其它浏览器比如火狐Firefox等,理论上没有限制,限制取决于操作系统支持
* POST:***POST提交数据不通过URL,理论上数据不受限***

说明:

1. GET的大小取决与浏览器和操作系统来限制,
2. POST的数据理论上没有限制
3. ***如果我们需要上传或下载文件,但是该文件一般不要大于2M,如果将来需要上传或下载很大的文件可以:1.开发插件 2.开发专门的客户端***



3.安全性:相对而言,POST提交安全高

4.GET提交更利于添加到收藏夹

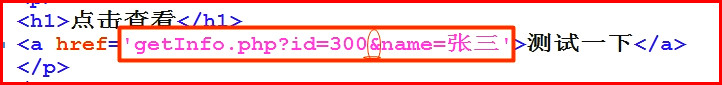
<http://www>.baidu.com/search.php?id=100

5.默认情况下,HTTP协议是GET请求

当一个form表单没有写method方式时,默认使用GET方式

6.如果时小数量数据,并不要求安全性,则选择GET,否则选择POST

7.如果是一个超链接带参数,则只能是GET方式提交

比如: 

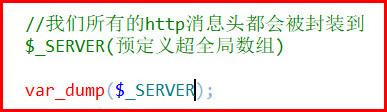
* **HTTP请求—消息头细节**

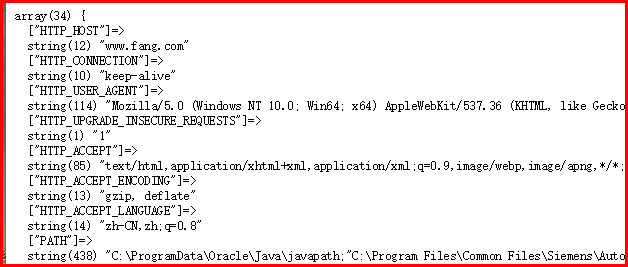
1.常用消息头



2.HTTP请求中的**常用头应用**案例

* 1.如何获取所有http请求的消息头(能拒绝某ip段)





$\_SERVER重要的数据

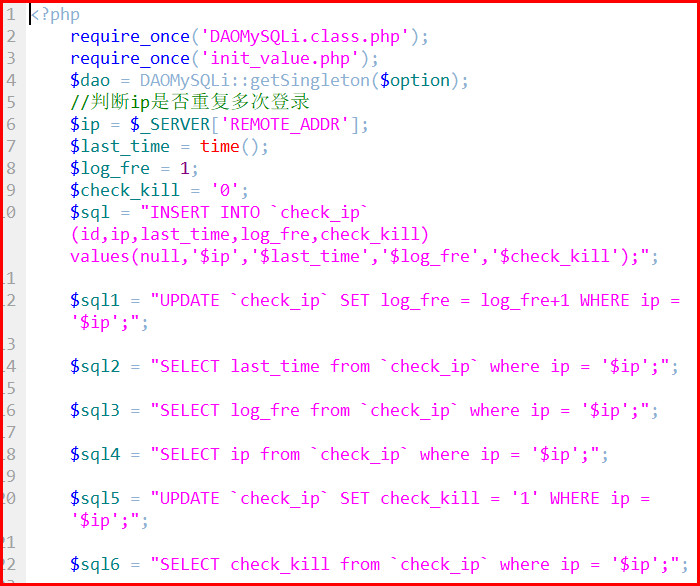
HTTP\_AGENT:浏览器内核,操作系统信息

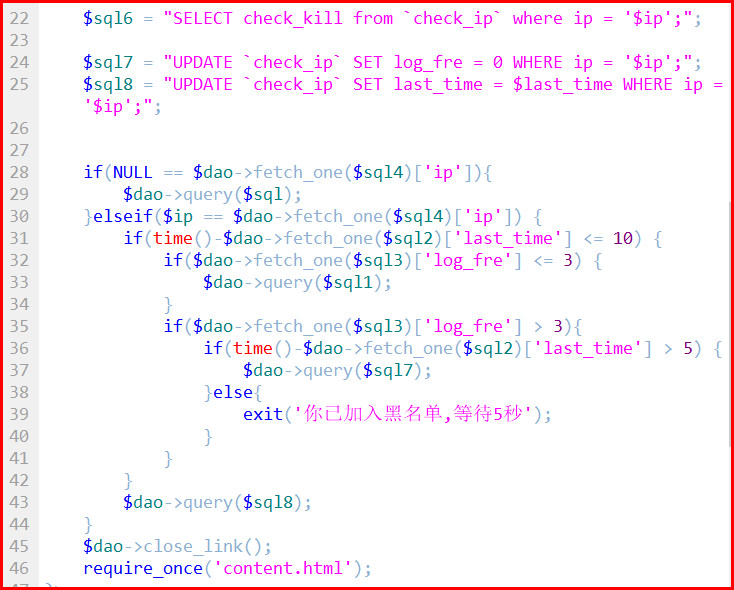
HTTP\_HOST:主机名称

REMOTE\_ADDR:访问该页面的ip

DOCUMENT\_ROOT:Apache的主目录

REQUEST\_URI:请求的资源名称

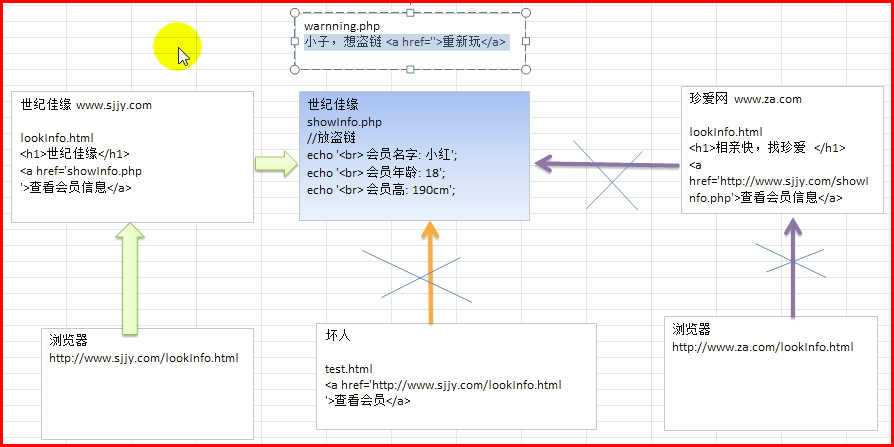




* 2.防止用户盗链某个页面

案例演示

图解说明防盗链(不包括图片)



(图片视频等文件防盗链,配置apache服务器,利用正则表达式)

代码截图如下



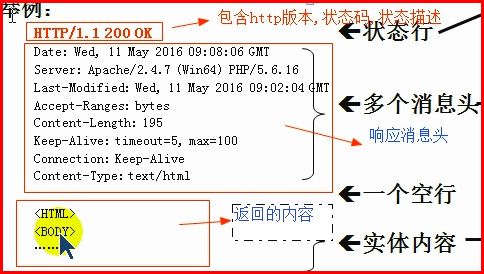
**2.3 HTTP协议细节-响应**

**1.基本介绍**

一个http响应代表服务器向客户端回送的数据,包括:

1.**一个状态行 2.若干消息头 3.以及实体内容**,

其中一些消息头和实体内容时可选的,消息头和实体内容之间要用空行隔开



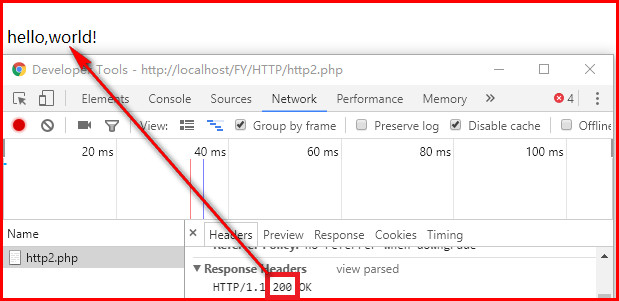
**2.http协议响应状态行详解**

* 基本结构
  + 格式: HTTP版本号 状态码 原因叙述<CRLF>
  + 举例: HTTP/1.1 200 OK
* 状态码用于表示服务器对请求的处理结果,他是一个三位的十进制数.响应状态码分为5类:

|  |  |
| --- | --- |
| 状态码 | 含义 |
| 100~199 | 表示成功接受请求,要求客户端继续提交洗一次请求才能完成整个处理 |
| 200~299 | 表示成功接受请求并已经完成整个处理过程,常用**200** |
| 300~399 | 为完成请求,客户需要进一步细化请求,比如:请求的资源已经移动到一个新地址,常用**302[重定向] 304[资源未修改]** |
| 400~499 | 客户端请求有错误,常用**404/403(禁止访问)** |
| 500~599 | 服务器端出现错误,常用**500** |

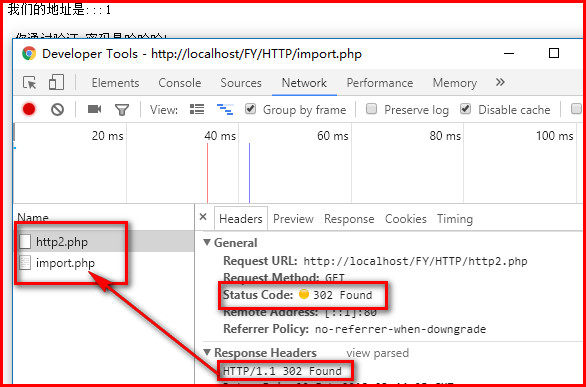
说明:**必须掌握状态码:**

* **1.200 成功**
* **2.302 重定向,告诉浏览器到另外一个URL请求资源**
* **3.304资源未修改 服务器说明浏览器本地的资源已经是最新的,不需要返回**
* **4.403 禁止访问 告诉浏览器你没有权限访问该资源**
* **5.404 没有该资源**
* **6.500 服务器端内部问题**
* 举例说明上述状态码:
  + 1.**200成功**

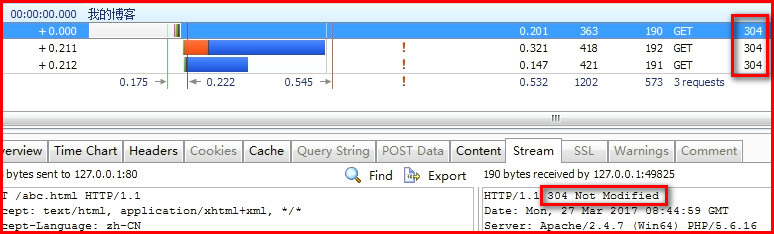


* + 2.**302重定向**

2018-02-02_115110.jpg

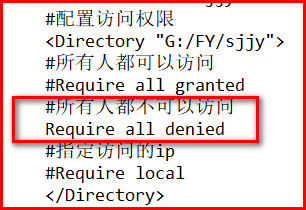


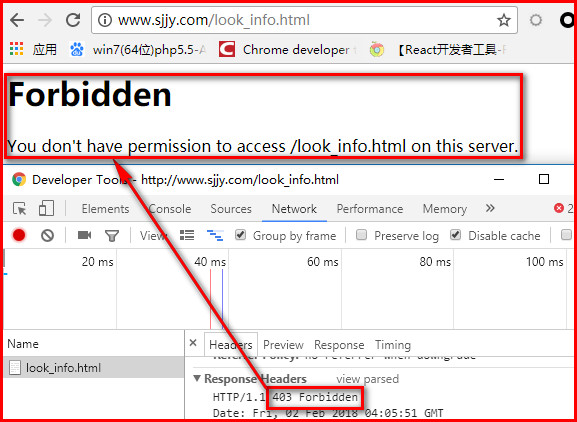
* + 3.304**资源未修改**



说明:当一个资源(html,css,js,jpg,png,gif),如果服务器没有修改过,则返回304码

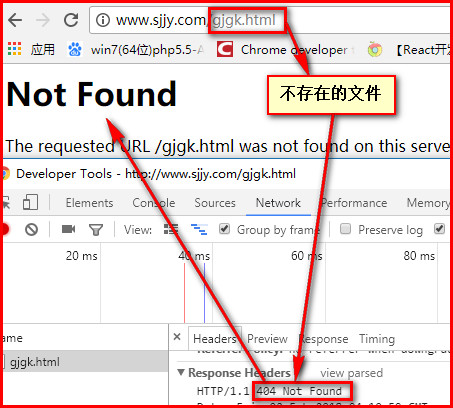
* + 4.**403禁止访问**





说明:这个表示禁止访问,与PHP代码是否错误无关

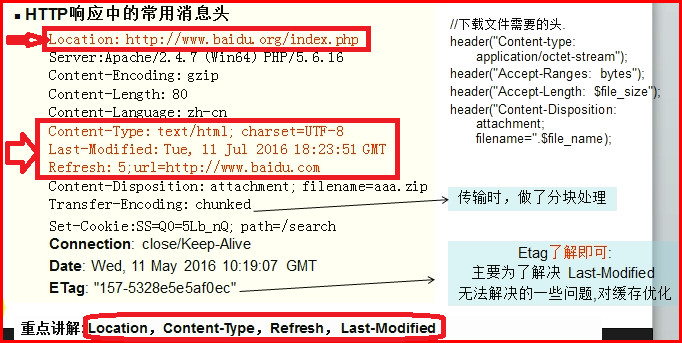
* + 5.**404没有该资源**



* + 6.**500服务器端内部问题**

***服务器内部错误***

**3.http协议响应消息头详解**



* **Expires**与**Cache-Control**与**Pragma 控制页面缓存**

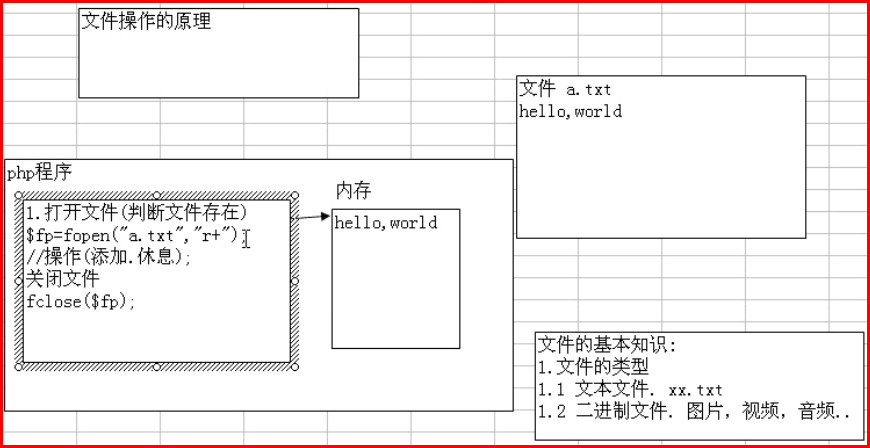
**默认情况浏览器会缓存页面**

**Expires:-1 Cache-Control:no-cache Pragma:no-cache**



* **HTTP关于文件下载**

**PHP操作文件的原理**

****

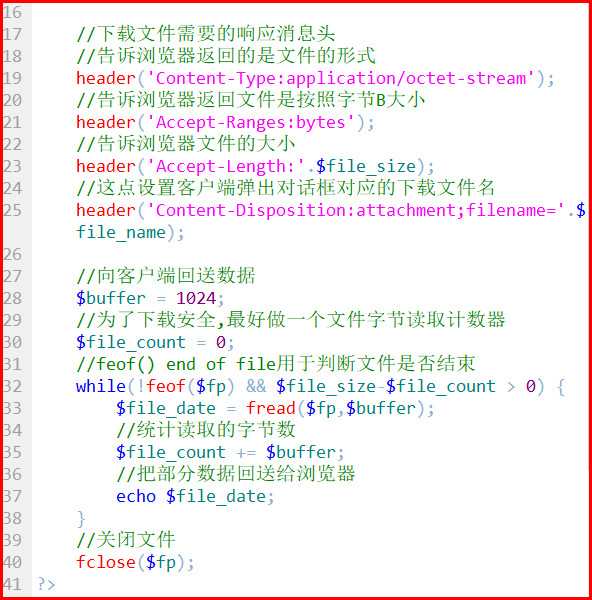
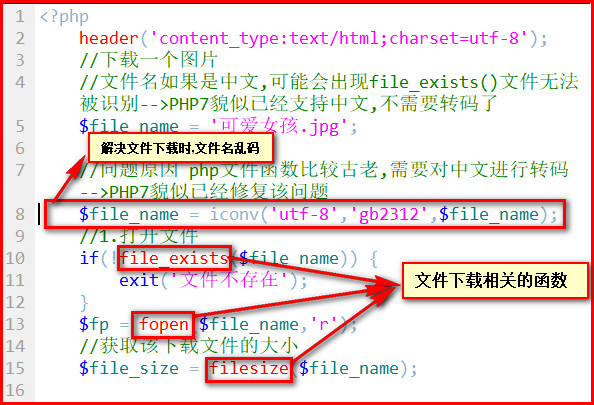
**下载文件需要的头**

**header(‘Content-Type:application/octet-stream’);**

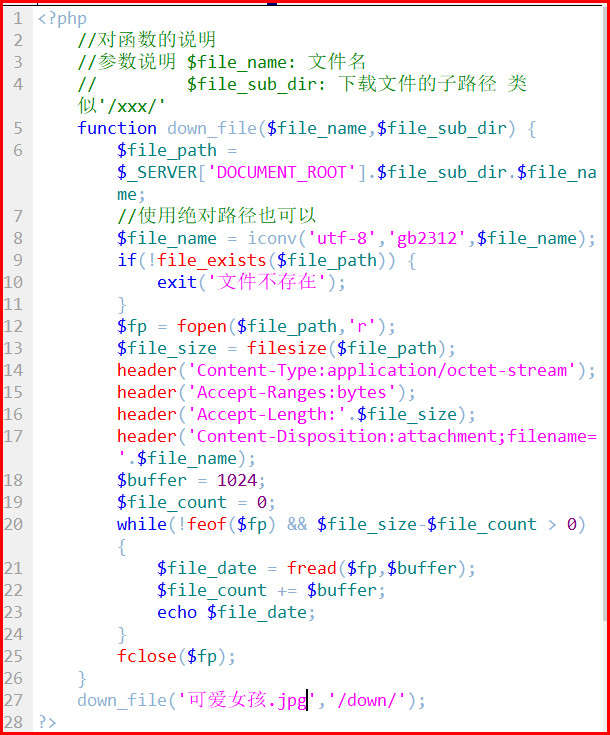
**header(‘Accept-Ranges:bytes’);**

**header(‘Accept-Length:$file\_size’);**

**header(“Content-Disposition:attachment;filename=$file\_name”);**

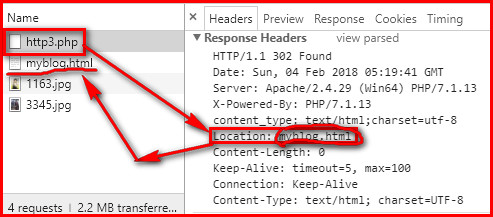
****

**封装到函数中,一定要fclose()!!!!!!!!!!**

****

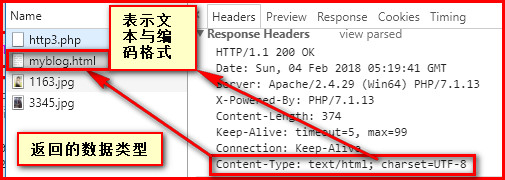
* **Location**消息头

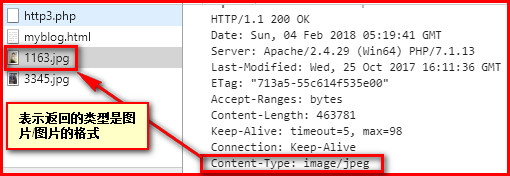
2018-02-04_132245.jpg



* **Content-Type**消息头

返回的数据类型是什么

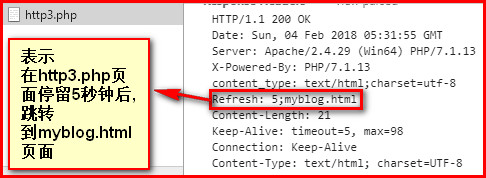




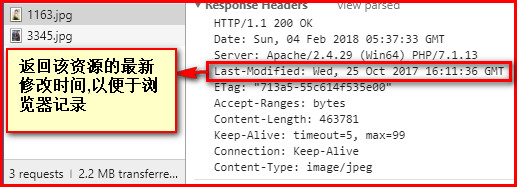
* **Refresh**消息头

Refresh:停留的时间(秒);url=需要跳转的URL





* **Last-Modified**消息头

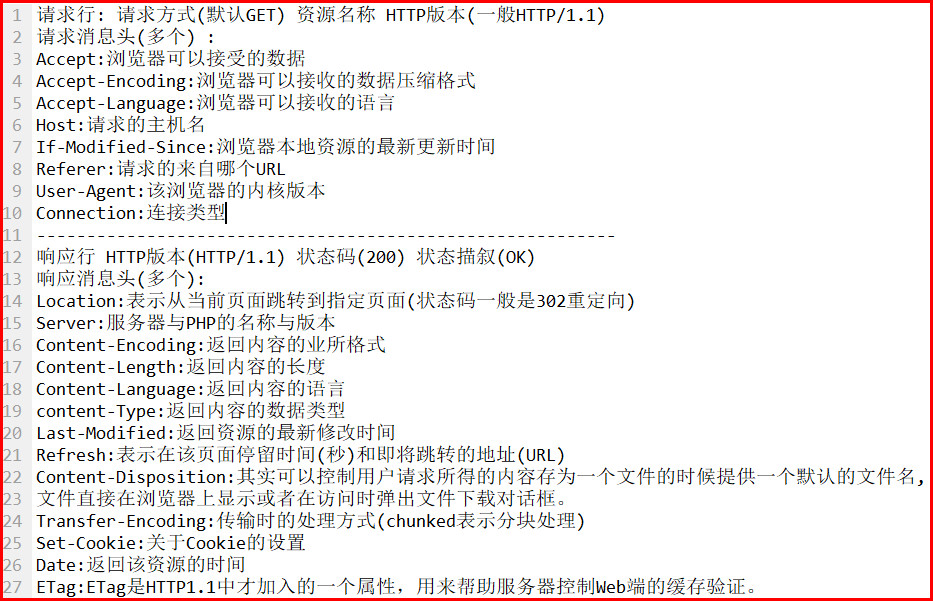


是304码是否出现有关

内容小结

**1.描述清楚HTTP请求头、响应头的格式**

**2.请求头和响应头中各个头字段的含义**

****

**Cathe\_control:是否缓存**

**3.请描述200、302、304、404和500等响应状态码所表示的意义。**

**200 成功 302 重定向 304 资源未修改 403 禁止访问(没有权限) 404 请求资源不存在 500 服务器内部错误**

**4.如果浏览器传递给WEB服务器的参数内容超过1K，应该使用那种方式发送请求消息？**

**最好使用POST,没有超过2k+32的情况下,使用IE勉强可用GET**

**5. 请详细说明一下 get 和 post 请求的区别**

**1.GET请求的携带的数据会被浏览器限制(HTTP协议不会限制)一般是2k+32,POST理论上携带数据是不限制的,但最好不超过2M**

**2.GET是通过在URL后面携带数据,POST是在实体内容处携带**

**3.POST的安全性高于GET**

**4.GET利于收藏页面**

**5.默认情况下都是GET请求,如果使用超链接携带数据只能是GET**

**6. Last-Modified: 响应消息头表示什么意思?**

**表示浏览器请求资源的最新修改时间,方便浏览器记录(不过HTTP/1.1引入了ETag来优化缓存)**

**7. Refresh: 响应消息头表示什么意思**

**表示浏览器在该页面停留一段时间后自动跳转到指定URL,一般格式是Refresh: 停留时间(5,单位秒);url=指定的跳转到的URL**

**8. 请画出 浏览器请求 php文件的详细 流程图**

