http协议详解

燕十八主讲

连接就是网络上的虚拟电路

Http只是一种协议，只要满足这种规范，用什么工具都可以发，所以除了浏览器，Telnet也可以发送Http请求

http执行流程：建立连接，发送请求给服务器，服务器响应，传回响应信息，结束连接

请求头信息

GET /http/hello.php HTTP/1.1

Host: localhost

Connection: keep-alive

Cache-Control: max-age=0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8

Upgrade-Insecure-Requests: 1

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.0.2661.102 Safari/537.36

Referer: http://localhost/http/

Accept-Encoding: gzip, deflate, sdch

Accept-Language: zh-CN,zh;q=0.8,en;q=0.6

Cookie: 9Uxj\_2132\_saltkey=YN66zIBp; 9Uxj\_2132\_lastvisit=1485426796; ECS[visit\_times]=7

响应头信息

HTTP/1.1 200 OK

Date: Tue, 14 Feb 2017 04:09:56 GMT

Server: Apache/2.4.10 (Win32) OpenSSL/0.9.8zb mod\_fcgid/2.3.9

X-Powered-By: PHP/5.3.29

Keep-Alive: timeout=5, max=100

Connection: Keep-Alive

Transfer-Encoding: chunked

Content-Type: text/html

implode(',',$array)将数组连接成字符串

$POST是一个超全局数组，他所存放的是用post方式发送过来的值

http协议方法与状态码

总结：

请求格式

POST /http/post.php HTTP/1.1 请求行（请求方法 路径 协议）/表示网站根目录

HOST:localhost 请求头信息（key：value）

Content-type:application/x-www-form-urlencoded

Content-Length:23（当请求主体信息达到指定长度后，自动发送请求）

空行

请求主体（也就是要发送的内容）可以没有

响应格式

http协议 状态码 状态文字

响应头信息

响应主体信息

Http请求方法：

OPTIONS / http/1.1

Host:localhost

OPTION方法可以获取服务器支持的所有请求方法

HEAD /http/hello.php HTTP/1.1

Host:localhost

HEAD与get相似，只是返回是否能请求成功的状态信息 并不返回网站内容，可用于调试确认一个页面

PUT 方法 在Apache中不支持 出现 method not allowed

也就是Http规定的方法，服务器未必支持

TRACE:用了代理上网，可以用trace查看代理有没有改变了你的请求信息

Apache默认支持以下请求方法

OPTIONS / http/1.1

Host:localhost

结果：

Allow: GET,HEAD,POST,OPTIONS,TRACE

content-type:application/x-www-form-urlencoded

Http响应中的状态码

2xx 系列 表示成功

3xx 系列 表示重定向

4xx 系列 表示客户端错误

5xx 系列 表示服务器错误

200 – 服务器成功返回网页

301/2- 永久/临时重定向

304 – 未修改 not modify

什么时候会出现304？

请求同个资源，如果缓存中已经存在，他会把这个资源的信息（比如 时间if-modified-since if-none-match）发送到服务器，与服务器的这个资源的信息（ETage last-modified）进行比对，如果相同则服务器不返回该资源，直接从缓存中取，否则返回该资源

所以并不是服务器知道客户端有这个资源，而是客户端把告诉服务器有这个资源，只是不确定这个资源有没有被修改过，所以要把客户端资源的一些信息与服务器端资源的信息进行比对，然后在确定是从服务器返回资源还是直接从缓存中取！！！

失败的状态码：

404 – 请求的网页不存在

503 – 服务器暂时不可用

500 – 服务器内部错误

GET /http/backgroup.gif HTTP/1.1

Host:localhost

307 重定向中保留原来的请求信息 可用于post请求，防止信息数据丢失

数据经post方式提交到02.php，然后转发到03.php

header('location:02.php',true,307);

exit;

由于在02.PHP中设置了307 所以在03.php中也能接收到post信息

socket编程发送GET请求

相当于文件操作，先要打开通道，也就是建立连接，然后通过这个通道往另一端写入内容

cookie原理：

当浏览器访问一个带有setCookie()的页面时，服务器会在响应头信息中返回Cookie：value

在这个会话中，当浏览器再次访问该服务器的其他页面时，在请求头，就把cookie信息传了过去，于是服务器就知道该用户 已经登录了

http防盗链技术

referer:xxx 这里保存的是网页的来源，也就是上一个页面的地址

如果直接从地址栏通过键入地址方式访问，请求头里没有referer项

可以通过Apache重写的方式，对图片进行防盗链，以防止别人盗用本网站图片

方法：打开Apache rewrite模块

建立一个.htaccess文件 内容为：

RewriteEngine on;

RewriteCond %{REQUEST\_FILENAME} .\*\.(jpeg|jpg|png|gif) [NC]

RewriteCond %{HTTP\_REFERER} !localhost [NC] //这是一个正则表达式，并不是严格的等于

RewriteRule .\* ./01.jpg

NC表示忽略大小写

服务器变量 ，引用方法是：

%{ NAME\_OF\_VARIABLE }

NAME\_OF\_VARIABLE可以是下表列出的字符串之一：

HTTP\_USER\_AGENT

HTTP\_REFERER

HTTP\_COOKIE

HTTP\_FORWARDED

HTTP\_HOST

HTTP\_PROXY\_CONNECTION

HTTP\_ACCEPT

REMOTE\_ADDR

REMOTE\_HOST

REMOTE\_PORT

REMOTE\_USER

REMOTE\_IDENT

REQUEST\_METHOD

SCRIPT\_FILENAME

PATH\_INFO

QUERY\_STRING

AUTH\_TYPE

DOCUMENT\_ROOT

SERVER\_ADMIN

SERVER\_NAME

SERVER\_ADDR

SERVER\_PORT

SERVER\_PROTOCOL

SERVER\_SOFTWARE

API\_VERSION

THE\_REQUEST

REQUEST\_URI

REQUEST\_FILENAME

IS\_SUBREQ

HTTPS

如何采集图片呢？

通过伪造referer的方法：也就是在请求头中加上 rererer:需要采集的网站

对于采集回来的图片，要想方法去掉http响应头信息和空行，把主体信息保存下来

需要用到的函数strstr('\r\n\r\n',$html);也就是两个回车之后的内容才是图片的全部信息

缓存详解：

Etag ,(签名可以理解图片的”指纹”).

如果该图片最新的Etag的值 和 If-None-Match的值不匹配,则重新请求.

HTTP协议与缓存控制

具体用法:

在主服务器, 打开apache的 expires扩展,利用该扩展来控制图片,css,html等文件

控制是否缓存,及缓存生命周期.

在.htaccess中,具体语法如下:

ExpiresByType type/encoding "<base> [plus] {<num> <type>}\*

可以设置缓存的类型，时间

首先打开Apache的expires模块

在htaccess中，写

ExpireActive on

ExpiresByType image/jpeg "now plus 30 days"

如果需要某一类文件不缓存，可以打开Apache的header模块

然后在.htaccess中

<FilesMatch "\.(gif)$">

header set Cache-Control:"no-store,must revalidate"

</FilesMatch>

http协议与内容压缩

服务器将内容压缩响应给浏览器，浏览器根据响应的压缩编码方式进行解压缩 目前浏览器普遍支持 gzip deflate compress 格式

如何在apache启用压缩功能?

1:开启deflate模块, 或gzip模块

2: 在conf文件中,写如下代码

<ifmodule mod\_deflate.c>

DeflateCompressionLevel 6 # 压缩级别为6, 可选1-9,推荐为6

AddOutputFilterByType DEFLATE text/plain # 压缩文本文件

AddOutputFilterByType DEFLATE text/html # 压缩html文件

AddOutputFilterByType DEFLATE text/xml # 压缩xml

</ifmodule>

小技巧: 当我们在采集时,可以不发送Accept-Encoding信息,这样采集直接是源码.

当然,也可以采集gzip(提高速度), 再用PHP解压gzip的内容.