# Http简单介绍

GET /test/hello.html HTTP/1.1

【表示发送的get请求, 请求资源是/test/hello.html 】

**Accept: \*/\* 【表示客户端可以接受任何数据】**

**Referer:** [**http://localhost:80/test/abc.html**](http://localhost:80/test/abc.html) **【1.表示我是从哪里来】**

**Accept-Language: zh-cn 【页面语言】**

**User-Agent: Mozilla/4.0 【告诉服务我的浏览器的内核，操作系统】**

**Accept-Encoding: gzip, deflate 【表示接受什么样的数据压缩格式.】**

**Host: localhost:80 【主机:80】**

**Connection: Keep-Alive** 【表示不要立即断掉我们的请求.】

现在我们想知道，客户端究竟给服务器发送的内容

如果我这个http1.php. 不希望 192.168.1.33 这个用户访问.

# 在服务器端，我们可以通过$\_SERVER来获取我们需要的信息

重要的有:

HTTP\_HOST

REMOTE\_ADDR 访问该页面的ip

DOCUMENT\_ROOT 可以获取 apche的主目录

REQUEST\_URI 可以获取 请求的资源名

http请求有两种主要的方式

get / post

get 和post的区别有哪些

1. 安全性 get 请求的数据会显示在 地址栏上, post请求的数据，放在http协议的消息体
2. 从可以提交数据的大小看.
   1. http协议本身并没有限制数据大小.
   2. 浏览器在对get 和post请求做显示, get 请求数据 2k+35 ,post没有显示.
3. get请求可以更好的添加到收藏夹.

现在我们实际使用一些http请求，完成一个防盗链的练习.

☞ http请求不是固定，是根据实际情况的，比如REFEER..

# Http响应（response）

简单定义: 一个Http响应代表服务器给浏览器回送的数据，同时告诉浏览器应当怎样处理数据.

1. 基本结构

**状态行**

**消息头信息**

**<--空行**

**实体信息**

**快速入门**

HTTP/1.1 200 OK 【200 ok 表示客户端请求成功】

Server: Microsoft-IIS/5.0 【表示告诉浏览器 服务器的情况 】

Date: Thu, 13 Jul 2000 05:46:53 GMT 【告诉浏览器 请求的页面 的时间 】

Content-Length: 2291 【表示回送的数据有 2291个字节】

Content-Type: text/html 【文档类型】

Cache-control: private 【缓存】

HTTP/1.1 304 Not Modified

Date: Tue, 13 Sep 2011 02:10:38 GMT

Server: Apache/2.2.17 (Win32) PHP/5.3.5

Connection: Keep-Alive

Keep-Alive: timeout=5, max=99

ETag: "12d00000000572c-6f69-3e1d849f35000"

下面是比较详细的一个Http响应

Location: http://www.baidu.org/index.php

Server:apache

Content-Encoding: gzip 【内容编码支持gzip压缩算法】

Content-Length: 80 【返回数据大小】

Content-Language: zh-cn

Content-Type: text/html; charset=GB2312

Last-Modified: Tue, 11 Jul 2000 18:23:51 GMT 【表示浏览器请求资源，最新时间】

Refresh: 1;url=http://www.baidu.com 【告诉浏览器，间隔1秒，重定向到 baidu】

Content-Disposition: attachment; filename=aaa.zip

Transfer-Encoding: chunked

Set-Cookie:SS=Q0=5Lb\_nQ; path=/search 【讲Cookie】

Expires: -1

**Cache-Control**: no-cache

**Pragma**: no-cache

**Connection**: close/Keep-Alive

**Date**: Tue, 11 Jul 2000 18:23:51 GMT

# 1.跳转界面

## 1. header("Location: 新的页面”);

// 跳转到另一个界面

header("Location: http://www.sohu.com");

**exit**();

# 2.停留几秒条状

//header(“Refresh: 3 ; url=http://www.sohu.com”);

//三秒后跳转

header("Refresh: 3 ; url=http://localhost/FirstPhP/three/three.php");

# 3.禁用缓存

//通过header来禁用缓存(ajax )

header("Expires: -1");

//为了保证兼容性，上面的那个是时间上的设置缓存，下面的这个就不是完全不缓存了

header("Cache-Control: no\_cache");

header("Pragma: no-cache");

**echo** "hello!cache";

# 4.下载

//对函数的说明

//参数说明 $file\_name 文件名

// $file\_sub\_dir: 下载文件的子路径 '"/xxx/xxx/"

function down\_file($file\_name,$file\_sub\_dir){

//死去活来，演示下载一个图片.

//如果文件是中文.

//原因 php文件函数，比较古老，需要对中文转码 gb2312 ，将中文转为 gb2312

$file\_name=iconv("utf-8","gb2312",$file\_name);

//相对路径

$path=".Http".$file\_name;

//绝对路径 server 里面有一个信息是表面这个的 DOCUMENT\_ROOT 是 www 的的地址 $file\_path=$\_SERVER['DOCUMENT\_ROOT'].$file\_sub\_dir.$file\_name;

// $file\_path=$\_SERVER['DOCUMENT\_ROOT']."/Http/Http/Sunset.jpg";

echo $\_SERVER;

echo $file\_path;

//如果你希望绝对路径

//1.打开文件

**if**(!file\_exists($file\_path)){

**echo** "文件不存在!";

**return** ;

}

//获得文件

$fp=fopen($file\_path,"r");

//获取下载文件的大小

$file\_size=filesize($file\_path);

**echo** $file\_size;

/\* if($file\_size>30){

echo "<script language='javascript'>window.alert('过大')</script>";

return ;

} \*/

//返回的文件

header("Content-type: application/octet-stream");

//按照字节大小返回

header("Accept-Ranges: bytes");

//返回文件大小

header("Accept-Length: $file\_size");

//这里客户端的弹出对话框，对应的文件名

header("Content-Disposition: attachment; filename=".$file\_name);

//向客户端回送数据

$buffer=1024;

//为了下载的安全，我们最好做一个文件字节读取计数器

$file\_count=0;

//这句话用于判断文件是否结束 feof($fp 最后一次读操作的时候的内容

**while**(!feof($fp) && ($file\_size-$file\_count>0) ){

$file\_data=fread($fp,$buffer);

//统计读了多少个字节

$file\_count+=$buffer;

//把部分数据回送给浏览器;

**echo** $file\_data;

}

//关闭文件

fclose($fp);

}

//测试函数是否可用

down\_file("Sunset.jpg","/Http/Http/");