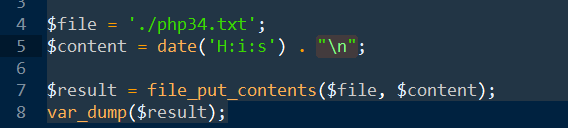
# 文件操作

## 写入字节数 = File\_put\_contents(文件地址, 内容)

写，文件不存在，自动创建。

默认为替换写：



支持使用追加写：

第三个参数使用FILE\_APPEND常量表示：



## File\_get\_contents(文件地址)

读

当操作的文件过大时？

不能一次性操作全部文件内容。上面的函数不适用！

使用下面的函数完成处理：

## Fopen(); 打开文件句柄（PHP与文件间数据流通道）

参数1：文件地址

参数2：打开方式（模式），打开文件后希望完成那种操作，可以在模式位置进行限制。

PHP提供了如下一些的打开模式：

R（read）读模式

W（write）替换写模式，将文件内容清零，自动创建不存在的文件。

A（append）追加写模式

X 替换写，不会自动创建不存在的文件。（只能打开已经存在的文件）

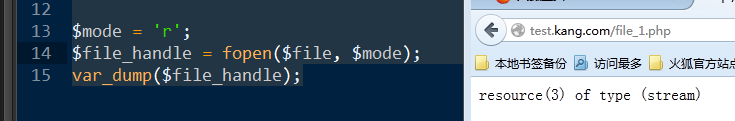
+扩展，扩展了操作。

R+ 读写模式，将文件指针放在文件开头。

W+ 读替换写模式，将文件内容清零，将文件指针放在文件开头，自动创建不存在的文件。

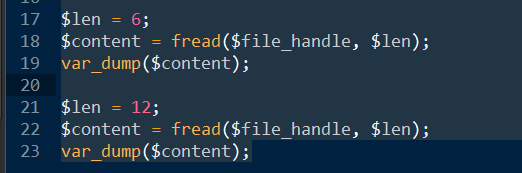
A+ 读追加写模式，写操作永远在文件末尾，读操作受限于文件指针。

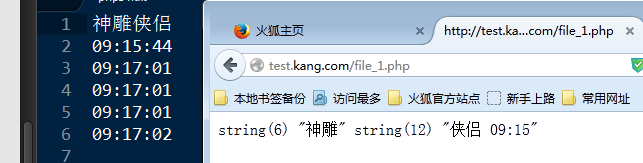
X+ 读替换写模式，将文件内容清零，将文件指针放在文件开头，不会自动创建不存在的文件。（只能打开已经存在的文件）



在都可以执行读写时，差异在于，打开文件时，初始化的工作不同，针对初始化的文件指针位置，文件内容。

## Fread(文件句柄, 长度); 读取，读取指定长度的字符串内容（单位字节）

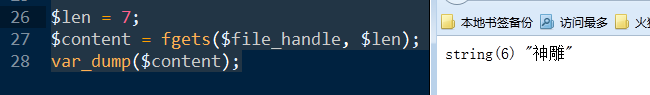




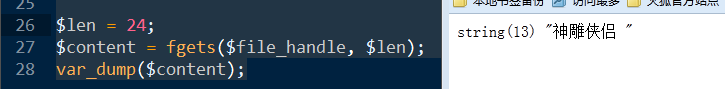
Tip：长度是有个最大值。为8192个字节。

## Fgets(文件句柄，长度); 读取，读取指定长度的字符串

长度，指的是会读取长度-1个字节。

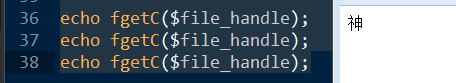


除了长度外，行末也是该函数的终止读操作条件。取决于先满足那个条件。



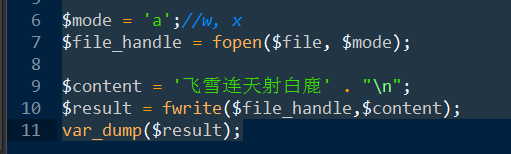
因此也称读行函数。最常用。fgetS($handle, 1024)

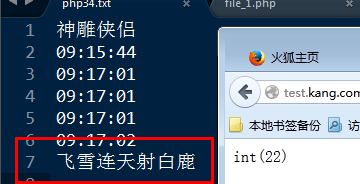
## fgetC(文件句柄); 读取,一次读取一个字节的数据



读取指针位置字符，操作时会移动指针。

## fwrite(文件句柄，写入内容); 写，在指定位置写入内容，通常由文件指针来指示，如果是a模式，不论指针在哪里，只能在末尾写。





如果w：

## fclose();

关闭句柄

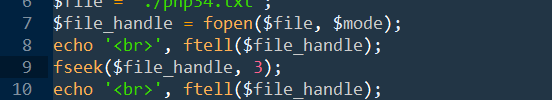


## Ftell()

获取指针位置

## Fseek()

设置指针位置



R+:

读写都受指针影响。写替换掉原有字节写。

W+:

清空内容

A+/a:

指针位置具有不确定因素。

A+,读受指针影响，写就是追加。

## Filemtime(文件地址)

文件的最后修改时间。

## Filesize(文件地址)



统计目录大小。

# HTTP协议

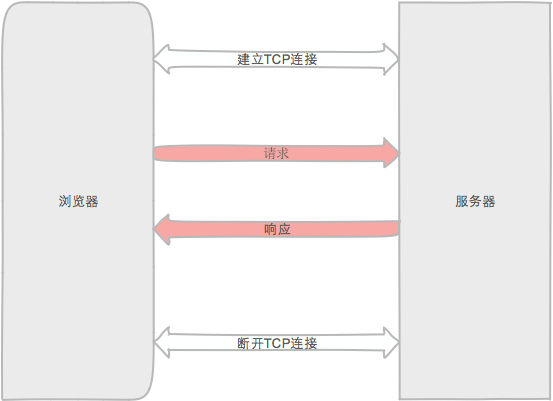
超文本传输协议（*HTTP*，HyperText Transfer Protocol)

最常见B/s架构中，使用，浏览器端与服务器端数据交互协议。

协议：约定好的一套语法规则。

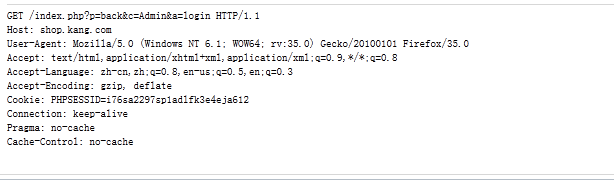
规定：请求数据格式，与响应数据格式。

是应用层协议



## 请求协议

规范，浏览器向服务器发送数据格式



以上内容，就是浏览器在向服务器的<http://shop.kang.com/index.php?p=back&c=Admin&a=login>请求时，所发送的数据。

请求数据分成三部分（列子中其实2个部分）：

### 请求行：request\_line

请求数据的第一行，用来说明当前请求的最基本信息。

由三块组成：

GET /index.php?p=back&c=Admin&a=login HTTP/1.1 换行

方式 资源地址（没有域名） 协议版本 \r\n(回车换行CRLF)

### 请求头：request\_header

请求头信息，请求标头

浏览器需要传输给服务器的属性数据。

# 主机，请求的主机。用于标识一台web服务器上的其中一个虚拟主机的。

Host: shop.kang.com

# 用户代理UA，什么发出的请求。

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:35.0) Gecko/20100101 Firefox/35.0

# 可以接受的内容类型

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

# 可以接受的语言

Accept-Language: zh-cn,zh;q=0.8,en-us;q=0.5,en;q=0.3

# 可以接受的数据编码类型（压缩编码）

Accept-Encoding: gzip, deflate

# 携带的cookie数据

Cookie: PHPSESSID=i76sa2297sp1adlfk3e4eja612

# tcp连接类型 keep-alive保持激活

Connection: keep-alive

# Keep-Alive: 30 保持连接的时间段秒

# 缓存控制选项

Pragma: no-cache

Cache-Control: no-cache

注意语法格式：

头，由标识与内容组成，使用冒号分隔（标识与冒号间不能存在空格）

每个头占用一行，行结尾，使用CRLF（回车换行\r\n）

可以存在多个头。

头信息结束使用空行标志

### 请求主体：request\_content

请求的主要用户数据。

就是POST数据。

如果方式为POST，则需要请求主体部分，（GET没有请求主体）

## 操作请求

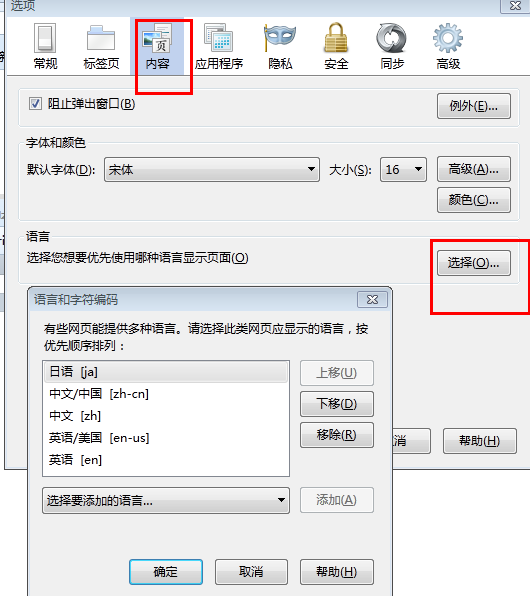
通过浏览器所携带属性，完成特定功能。

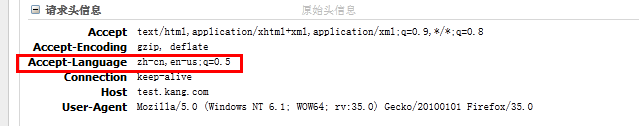
### 多语言程序（I18n，国际化程序）

项目：提供多套语言版本。

浏览器：请求时，携带属性，告知服务器，需要那种语言。

通过请求头：Accept-Language 告知服务器，希望使用的语言。

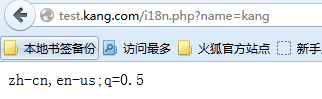




项目：可以获取浏览器需要的语言，载入相应的语言包，做展示。

获取：





找到最佳匹配：

项目存在默认的语言。（浏览器需要的，项目都没有提供）

确定浏览器可以接受的语言列表，找到服务器中，最匹配（权重依次匹配）

### PHP发出请求（模拟请求）

采集程序。

公共平台开发。

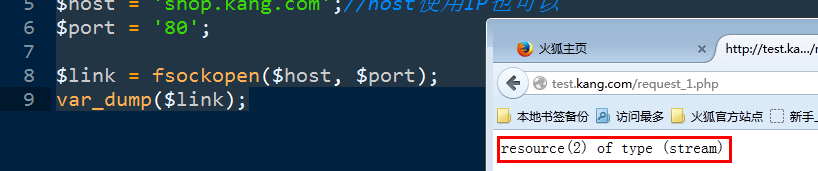
发出请求：

连接目标服务器，发送符合请求协议格式的数据。

服务器就会将其视为请求，发出响应！

连接：

Fsockopen();建立一个internet连接

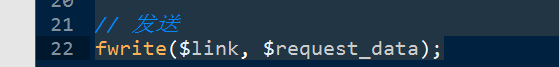


处理请求数据：



向服务器发送，发出请求。

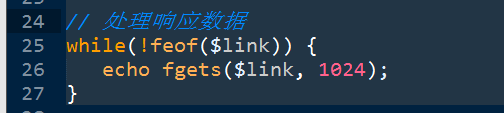
通过函数 fwrite即可完成。（fwrite可以向文件和网络资源位置发送数据）



获取服务器的响应数据

使用 fgetS()完成：

Feof，（end of file）用于检测是否到到数据流末尾。



断开连接



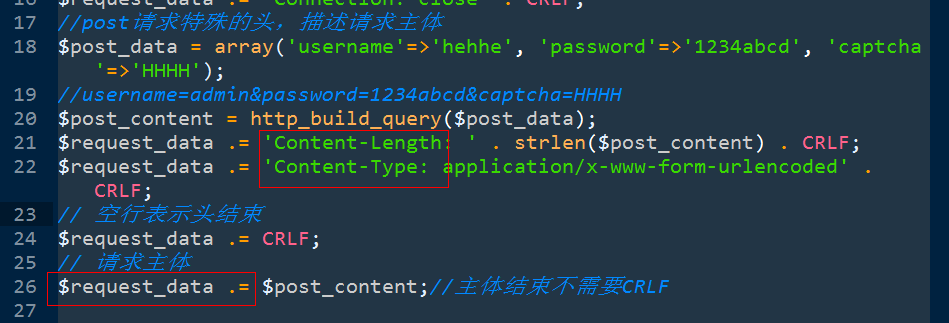
## POST请求

存在请求主体数据。

请求行：POST

请求头：存在对请求主体的描述

请求主体：post数据



## CURL-Client URL

PHP扩展。

可以用来模拟URL客户端（浏览器，请求代理）的工具扩展。

### 开启扩展



Apacherestart

Tip：

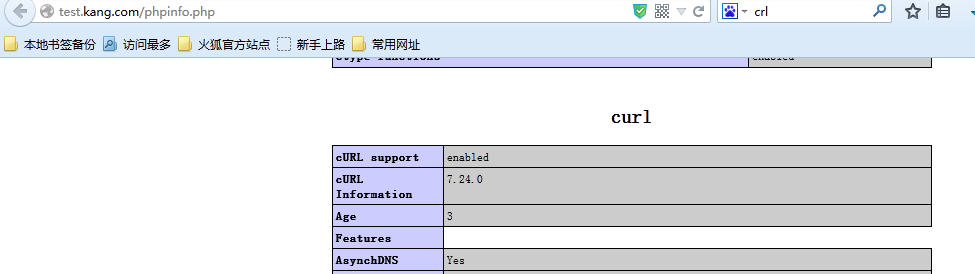
依赖类库：

Php目录：





将以上文件拷贝到系统目录（windows中），将php目录设置为path环境变量。



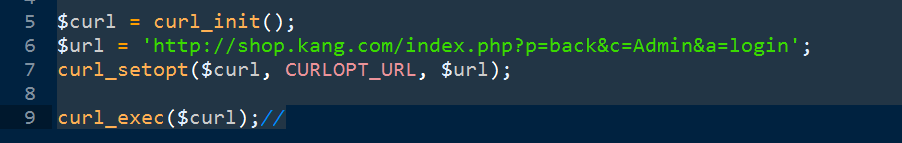
### 模拟GET

Curl资源 = curl\_init()初始化curl

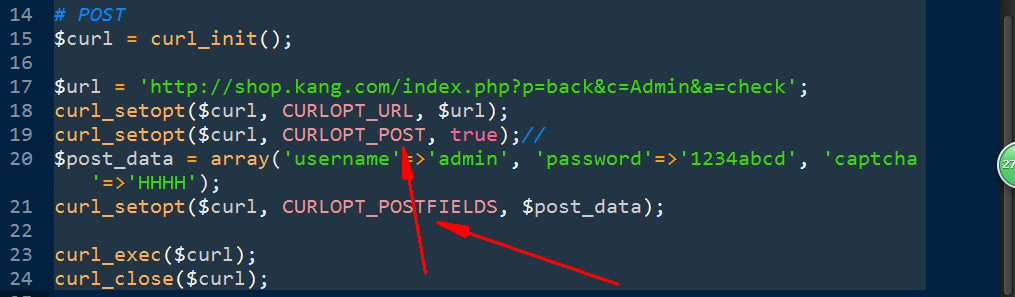
curl\_setopt(curl资源，选项标志，选项值)

curl\_exec(资源)发出请求

curl\_close()关闭资源



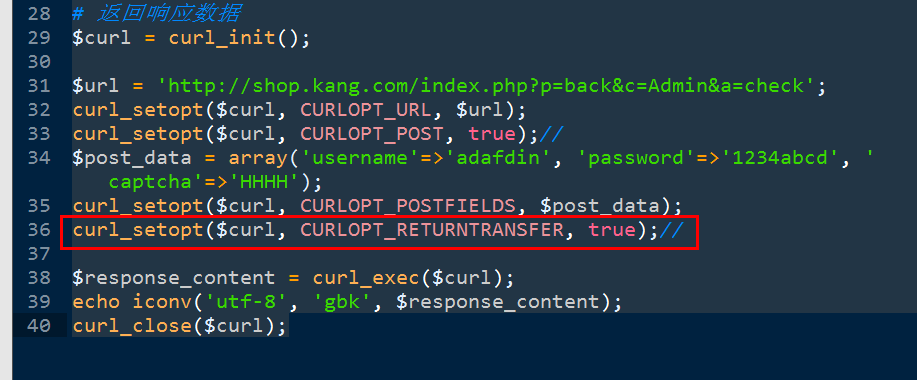
### POST请求



### 处理响应数据

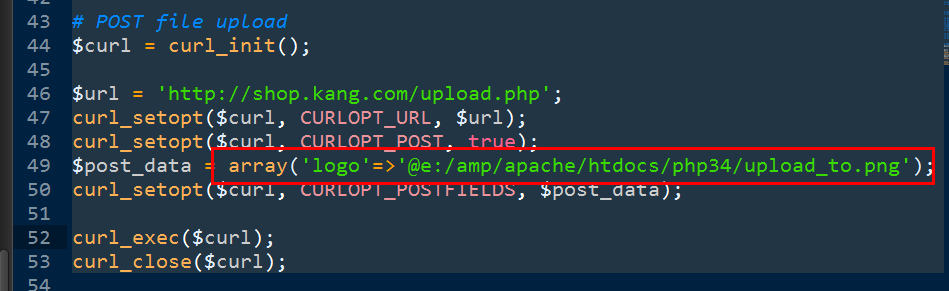
curl\_setopt($curl, CURLOPT\_RETURNTRANSFER, true);//

请求执行时，不是将响应直接输出，而是以返回值的形式处理。



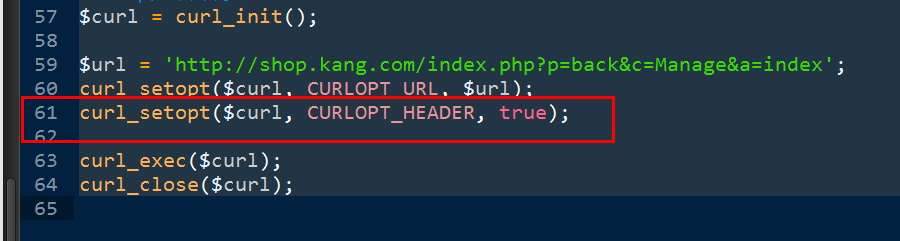
### POST文件上传

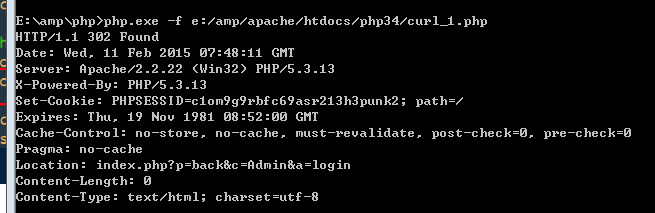
Post数据使用文件地址，前使用@标志为文件而不是字符串。



### 处理响应头

CURLOPT\_HEADER：是否获取响应头数据



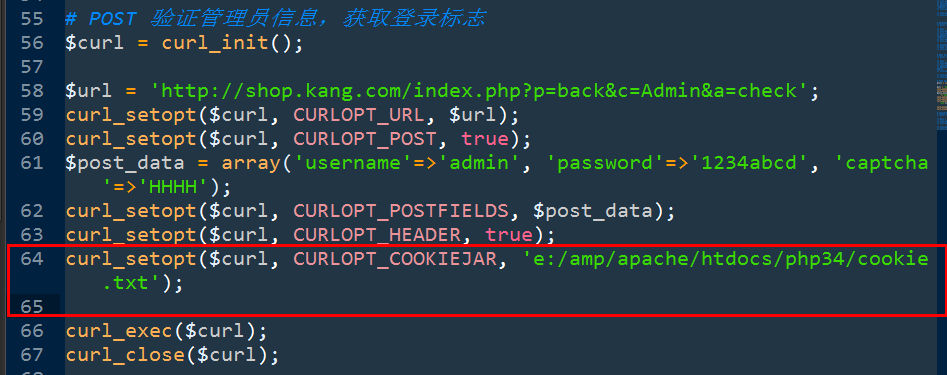


### 处理会话COOKIE

（模拟登录）

接收服务器设置的cookie，并存储于请求代理端：

CURLOPT\_COOKIEJAR 指定存储服务器所设置的cookie变量存储位置。



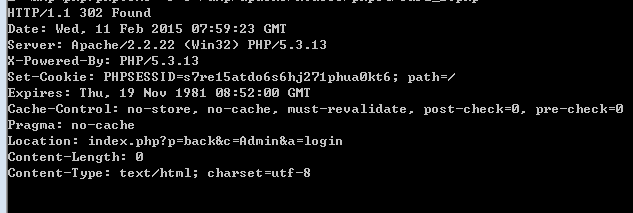
发出请求时携带拥有的cookie：

CURLOPT\_COOKIEFILE

请求时携带的cookie数据所在的位置。



## 响应协议



也是三部分：

1响应行2响应头3响应主体

### 响应行

响应数据的第一行，响应结果的概述

http/1.1 302 FoundCRLF

版本 响应状态码 响应消息

响应状态码，和响应消息，一一对应。

302 Found：重定向，配合响应头Location使用。

404 Not Found：请求资源不存在。

403 Forbidden：没有权限访问该地址

200 ok： 请求成功

500 Server Internal Error： 服务器内部错误

### 响应头

服务器告知浏览器属性信息。

### 响应主体

主体数据（通过浏览器的查看源代码所看到的内容）

## 操作响应

操作响应头，header()可以

操作响应主体，任何的输出，都是响应主体。（echo， print，var\_dump，PHP标签之外的所有HTML代码）

### 控制器浏览器缓存

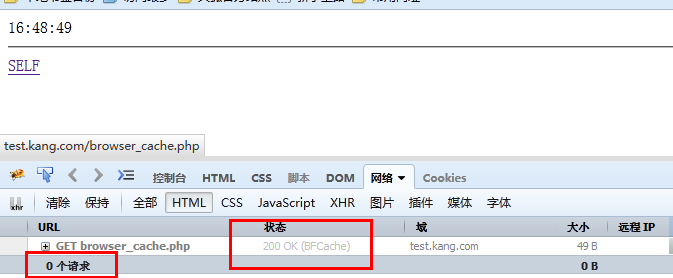


Expires：有效期，时间点，GMT时间，格林威治平时。

Gmdate() 将时间戳格式化为格林威治平时。

D，星期

M，月（三个英文月）



### HTTP下载

下载：文件从服务器端传输到浏览器端。发生于服务器响应时。

PHP仅仅需要将需要下载的内容作为响应主体输出即可。

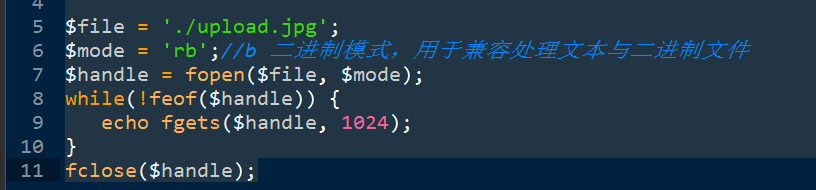
通过响应头：Content-Disposition：告知浏览器，接受到响应主体后的处理方式。

值：attachment 表示以附件的方式处理响应主体。

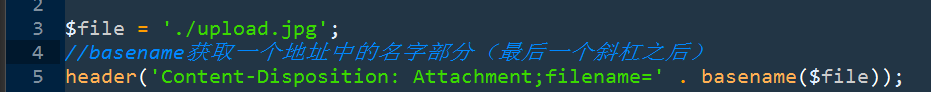


如果是文件的下载：

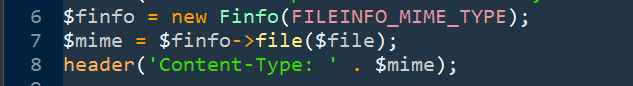
将文件内容作为响应主体输出即可！



控制下载的文件名（默认为请求的脚本名）



控制主体类型：



细致控制：

主体大小。

