**C U r l**

**1 什么是curl：**

Curl就是php中用来模仿浏览器去打开某个网站的功能类，curl是php内置的组件，都是通过方法的方式来操作的，比如初始化调用方法curl\_init()来实现。

**2.基本结构**

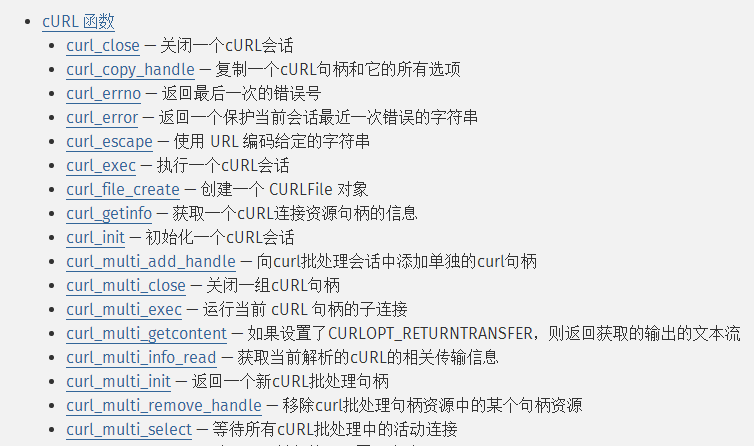
　　在学习更为复杂的功能之前，先来看一下在PHP中建立cURL请求的基本步骤：

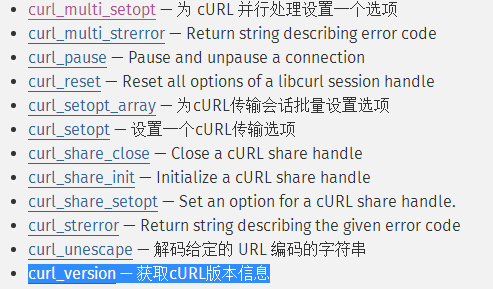
　　（1）初始化   
　　　　curl\_init()

　　（2）设置变量   
　　　　curl\_setopt() 。最为重要，一切玄妙均在此。有一长串cURL参数可供设置，它们能指定URL请求的各个细节。要一次性全部看完并理解可能比较困难，所以今天我们只试一下那些更常用也更有用的选项。

　　（3）执行并获取结果   
　　　　curl\_exec()

1. 释放cURL句柄  
   　　　　curl\_close()





**2.curl\_setopt()的值**

|  |  |  |
| --- | --- | --- |
| 选项 | 可选value值 | 备注 |
| **CURLOPT\_AUTOREFERER** | 当根据Location:重定向时，自动设置header中的Referer:信息。 |  |
| **CURLOPT\_BINARYTRANSFER** | 在启用**CURLOPT\_RETURNTRANSFER**的时候，返回原生的（Raw）输出。 |  |
| **CURLOPT\_COOKIESESSION** | 启用时curl会仅仅传递一个session cookie，忽略其他的cookie，默认状况下cURL会将所有的cookie返回给服务端。session cookie是指那些用来判断服务器端的session是否有效而存在的cookie。 |  |
| **CURLOPT\_CRLF** | 启用时将Unix的换行符转换成回车换行符。 |  |
| **CURLOPT\_DNS\_USE\_GLOBAL\_CACHE** | 启用时会启用一个全局的DNS缓存，此项为线程安全的，并且默认启用。 |  |
| **CURLOPT\_FAILONERROR** | 显示HTTP状态码，默认行为是忽略编号小于等于400的HTTP信息。 |  |
| **CURLOPT\_FILETIME** | 启用时会尝试修改远程文档中的信息。结果信息会通过curl\_getinfo()函数的CURLINFO\_FILETIME选项返回。 curl\_getinfo(). |  |
| **CURLOPT\_FOLLOWLOCATION** | 启用时会将服务器服务器返回的"Location: "放在header中递归的返回给服务器，使用**CURLOPT\_MAXREDIRS**可以限定递归返回的数量。 |  |
| **CURLOPT\_FORBID\_REUSE** | 在完成交互以后强迫断开连接，不能重用。 |  |
| **CURLOPT\_FRESH\_CONNECT** | 强制获取一个新的连接，替代缓存中的连接。 |  |
| **CURLOPT\_FTP\_USE\_EPRT** | 启用时当FTP下载时，使用EPRT (或 LPRT)命令。设置为**FALSE**时禁用EPRT和LPRT，使用PORT命令 only. |  |
| **CURLOPT\_FTP\_USE\_EPSV** | 启用时，在FTP传输过程中回复到PASV模式前首先尝试EPSV命令。设置为**FALSE**时禁用EPSV命令。 |  |
| **CURLOPT\_FTPAPPEND** | 启用时追加写入文件而不是覆盖它。 |  |
| **CURLOPT\_FTPASCII** | **CURLOPT\_TRANSFERTEXT**的别名。 |  |
| **CURLOPT\_FTPLISTONLY** | 启用时只列出FTP目录的名字。 |  |
| **CURLOPT\_HEADER** | 启用时会将头文件的信息作为数据流输出。 |  |
| **CURLINFO\_HEADER\_OUT** | 启用时追踪句柄的请求字符串。 | 从 PHP 5.1.3 开始可用。**CURLINFO\_**前缀是故意的(intentional)。 |
| **CURLOPT\_HTTPGET** | 启用时会设置HTTP的method为GET，因为GET是默认是，所以只在被修改的情况下使用。 |  |
| **CURLOPT\_HTTPPROXYTUNNEL** | 启用时会通过HTTP代理来传输。 |  |
| **CURLOPT\_MUTE** | 启用时将cURL函数中所有修改过的参数恢复默认值。 |  |
| **CURLOPT\_NETRC** | 在连接建立以后，访问~/.netrc文件获取用户名和密码信息连接远程站点。 |  |
| **CURLOPT\_NOBODY** | 启用时将不对HTML中的BODY部分进行输出。 |  |
| **CURLOPT\_NOPROGRESS** | 启用时关闭curl传输的进度条，此项的默认设置为启用。  **Note**:  PHP自动地设置这个选项为**TRUE**，这个选项仅仅应当在以调试为目的时被改变。 |  |
| **CURLOPT\_NOSIGNAL** | 启用时忽略所有的curl传递给php进行的信号。在SAPI多线程传输时此项被默认启用。 | cURL 7.10时被加入。 |
| **CURLOPT\_POST** | 启用时会发送一个常规的POST请求，类型为：application/x-www-form-urlencoded，就像表单提交的一样。 |  |
| **CURLOPT\_PUT** | 启用时允许HTTP发送文件，必须同时设置**CURLOPT\_INFILE**和**CURLOPT\_INFILESIZE**。 |  |
| **CURLOPT\_RETURNTRANSFER** | 将curl\_exec()获取的信息以文件流的形式返回，而不是直接输出。 |  |
| **CURLOPT\_SSL\_VERIFYPEER** | 禁用后cURL将终止从服务端进行验证。使用**CURLOPT\_CAINFO**选项设置证书使用**CURLOPT\_CAPATH**选项设置证书目录 如果**CURLOPT\_SSL\_VERIFYPEER**(默认值为2)被启用，**CURLOPT\_SSL\_VERIFYHOST**需要被设置成**TRUE**否则设置为**FALSE**。 | 自cURL 7.10开始默认为**TRUE**。从cURL 7.10开始默认绑定安装。 |
| **CURLOPT\_TRANSFERTEXT** | 启用后对FTP传输使用ASCII模式。对于LDAP，它检索纯文本信息而非HTML。在Windows系统上，系统不会把STDOUT设置成binary模式。 |  |
| **CURLOPT\_UNRESTRICTED\_AUTH** | 在使用**CURLOPT\_FOLLOWLOCATION**产生的header中的多个locations中持续追加用户名和密码信息，即使域名已发生改变。 |  |
| **CURLOPT\_UPLOAD** | 启用后允许文件上传。 |  |
| **CURLOPT\_VERBOSE** | 启用时会汇报所有的信息，存放在STDERR或指定的**CURLOPT\_STDERR**中。 |  |

**实例：**

Curl提交GET数据

Crul提交post数据

$ch = curl\_init(); //初始化一个CURL对象

curl\_setopt($ch, CURLOPT\_URL, "http://localhost:443/a.php?bb=fffff");

//设置你所需要抓取的URL

curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, 0);

//设置curl参数，要求结果是否输出到屏幕上，为true的时候是不返回到网页中

//假设上面的0换成1的话，那么接下来的$data就需要echo一下。

//post提交

curl\_setopt($ch, CURLOPT\_POSTFIELDS, $curlPost);

$data = curl\_exec($ch);

//运行curl,请求网页。

curl\_close($ch);

echo $data;

$ch = curl\_init(); //初始化一个CURL对象

$curlPost = "user=白痴&pass=傻瓜"; //这是post的数据

curl\_setopt($ch, CURLOPT\_URL, "http://localhost:443/a.php");

//设置你所需要抓取的URL

curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, 0);

//设置curl参数，要求结果是否输出到屏幕上，为true的时候是不返回到网页中

//假设上面的0换成1的话，那么接下来的$data就需要echo一下。

curl\_setopt($ch, CURLOPT\_POST, 1); //这句用来说明是用post方式提交数据

//post提交

curl\_setopt($ch, CURLOPT\_POSTFIELDS, $curlPost);

$data = curl\_exec($ch);

//运行curl,请求网页。

curl\_close($ch);

echo $data;