curl\_setopt

(PHP 4 >= 4.0.2, PHP 5, PHP 7)

curl\_setopt — 设置 cURL 传输选项

说明

curl\_setopt ( resource $ch , int $option , [mixed](https://www.php.net/manual/zh/language.pseudo-types.php#language.types.mixed) $value ) : bool

为 cURL 会话句柄设置选项。

参数

ch

由 [curl\_init()](https://www.php.net/manual/zh/function.curl-init.php) 返回的 cURL 句柄。

option

需要设置的*CURLOPT\_XXX*选项。

value

将设置在option选项上的值。

以下 option 参数的 value应该被设置成 [bool](https://www.php.net/manual/zh/language.types.boolean.php) 类型：

test.php

<?php  
echo "nihao";

test1.php

<?php  
$param=["nihao"=>"nihaovalue","status"=>1];  
$url="http://www.test.com:80/test.php";  
$param = json\_encode($param, *JSON\_UNESCAPED\_UNICODE*);  
print\_r(curl\_get($url));  
function curl\_get($url, $milliseconds = 30000)  
{  
 set\_time\_limit(0);  
 $start\_time = microtime(true);  
//句柄  
 $ch = curl\_init();  
 //这个参数很重要，设置这个才可以支持毫秒级的超时设置  
 curl\_setopt($ch, *CURLOPT\_NOSIGNAL*, 1);  
 curl\_setopt($ch, *CURLOPT\_HTTP\_VERSION*, *CURL\_HTTP\_VERSION\_1\_0*);  
 curl\_setopt($ch, *CURLOPT\_SSL\_VERIFYPEER*, FALSE);  
 curl\_setopt($ch, *CURLOPT\_RETURNTRANSFER*, TRUE);  
 curl\_setopt($ch, *CURLOPT\_FAILONERROR*, 0);  
 curl\_setopt($ch, *CURLOPT\_TIMEOUT\_MS*, $milliseconds);  
  
 curl\_setopt($ch, *CURLOPT\_SSL\_VERIFYHOST*, FALSE);  
 curl\_setopt($ch, *CURLOPT\_URL*, $url);  
 curl\_setopt($ch, *CURLOPT\_IPRESOLVE*, *CURL\_IPRESOLVE\_V4* );  
 $response = curl\_exec($ch);//执行  
 $http\_error\_code = curl\_errno($ch); // 检查是否有错误发生  
 curl\_close($ch); // 关闭句柄  
 if($http\_error\_code){//出错则显示错误信息  
 return $http\_error\_code;  
 }  
 return $response;  
}

# 本地服务CURL请求返回超时或卡死

#### 完美解决本地服务CURL请求返回超时或卡死

本地环境是windows+nginx。  
PHP+NGINX在windows下是不支持并发。  
fastcgi\_pass 都绑定了9000端口,  
所以两个服务就会有一个端口被占用，无法返回消息。

##### 解决办法

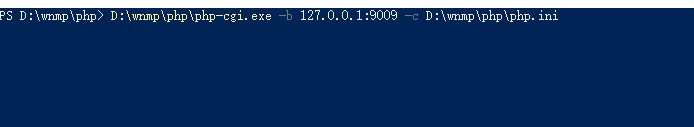
第一步 修改 fastcgi\_pass=127.0.0.1:9000 的配置：

这个配置表示 nginx 会通过 fastcgi\_pass 将请求的资源发给 127.0.0.1:9000 进行解析。

利用cmd命令启动两个服务，所以分别配置成不同的端口号如127.0.0.1:9009，然后保存重启 nginx。  
fastcgi\_pass = 127.0.0.1:9000  
fastcgi\_pass = 127.0.0.1:9009  
第二步 同时在php的 cmd 下开启多个 php-cgi 指向 php.ini 进行监听：  
进入php目录下, 运行命令:D:/wnmp/php/php-cgi.exe" -b 127.0.0.1:9009 -c "D:/wnmp/php/php.ini,

或者打开cmd

php-cgi.exe -b 127.0.0.1:9003 -c php.ini 运行这个命令 但是cmd 命令窗口不能关闭。

  
nginx域名配置如下:  
*server* {  
 *listen* 80;  
 *server\_name www*.*test*.*com* ;  
 *root* "D:\phpstudy\WWW/test";  
 *location* / {  
 *index index*.*html index*.*htm index*.*php*;  
 #autoindex on;  
 }  
 *location* ~ \.php(.\*)$ {  
 *fastcgi\_pass* 127.0.0.1:9000;  
 *fastcgi\_index index*.*php*;  
 *fastcgi\_split\_path\_info* ^((?*U*).+\.*php*)(/?.+)$;  
 *fastcgi\_param SCRIPT\_FILENAME* $document\_root$fastcgi\_script\_name;  
 *fastcgi\_param PATH\_INFO* $fastcgi\_path\_info;  
 *fastcgi\_param PATH\_TRANSLATED* $document\_root$fastcgi\_path\_info;  
 include *fastcgi\_params*;  
 }  
}  
*server* {  
 *listen* 80;  
 *server\_name www*.*test1*.*com* ;  
 *root* "D:\phpstudy\WWW/test1";  
 *location* / {  
 *index index*.*html index*.*htm index*.*php*;  
 #autoindex on;  
 }  
 *location* ~ \.php(.\*)$ {  
 *fastcgi\_pass* 127.0.0.1:9003;  
 *fastcgi\_index index*.*php*;  
 *fastcgi\_split\_path\_info* ^((?*U*).+\.*php*)(/?.+)$;  
 *fastcgi\_param SCRIPT\_FILENAME* $document\_root$fastcgi\_script\_name;  
 *fastcgi\_param PATH\_INFO* $fastcgi\_path\_info;  
 *fastcgi\_param PATH\_TRANSLATED* $document\_root$fastcgi\_path\_info;  
 include *fastcgi\_params*;  
 }  
}