

# phpcurl:PHP中的curl\_multi一族函数

疯狂代码 <http://CrazyCoder.cn/> <http://CrazyCoder.cn/Php/Article5509.html>

;;;相信许多人对php手册中语焉不详curl\_multi族头疼不已它们文档少给例子;更是简单让你无从借鉴我也曾经找了许多网页都没见个完整应用例子

```
curl_multi_add_handle ;
curl_multi_close ;
curl_multi_exec ;
curl_multi_getcontent ;
curl_multi_info_read ;
curl_multi_init ;
curl_multi_remove_handle ;
curl_multi_select ;
```

般来说想到要用这些时目显然应该是要同时请求多个url而不是个个依次请求否则不如自己循环去调curl\_exec好了

步骤整理总结如下:

第步:curl\_multi\_init

第 2步:循环curl\_multi\_add\_handle

这步需要注意的是curl\_multi\_add\_handle第 2个参数是由curl\_init而来子handle

第 3步:持续curl\_multi\_exec

第 4步:根据需要循环curl\_multi\_getcontent获取结果

第 5步:curl\_multi\_remove\_handle并为每个字handlecurl\_close

第 6步:curl\_multi\_close

这里有个网上找简单例子其作者称为dirty例子(稍后我会介绍说明为何dirty):

```
/*
Here' s ;a ;quick ;and ;dirty ;example ;for ;curl-multi ;from ;PHP, ;tested ;on ;PHP ;5.0.0RC1 ;CLI ;/
;FreeBSD ;5.2.1
*/
```

```
$connos ;= ;.gif' />(
"http://www.cnn.com/",
"http://www.canada.com/",
"http://www.yahoo.com/"
);
```

```
$mh ;= ;curl_multi_init;
```

```

foreach ;($connos ;as ;$i ;=> ;$url) ;{
    ;;;;;$conn[$i]=curl_init($url);
    ;;;;;curl_opt($conn[$i],CURLOPT_RETURNTRANSFER,1);
    ;;;;;curl_multi_add_handle ;($mh,$conn[$i]);
}

```

```

do ;{;$n=curl_multi_exec($mh,$active);;} ;while ;($active);

```

```

foreach ;($connos ;as ;$i ;=> ;$url) ;{
    ;;;;;$res[$i]=curl_multi_getcontent($conn[$i]);
    ;;;;;curl_close($conn[$i]);
}

```

```

pr_r($res);
;

```

整个使用过程差不多就是这样但是这个简单代码有个致命弱点就是在do循环那段在整个url请求期间是个死循环它会轻易导致CPU占用100%

现在我们来改进它这里要用到个几乎没有任何文档curl\_multi\_select了虽然Ccurl库对select有介绍说明但是php里接口和使用方法确和C中有区别

把上面do那段改成下面这样:

```

;;;;;;;;;;;;;do ;{
    ;;;;;;;;;;;;;;$mrc ;= ;curl_multi_exec($mh,$active);
    ;;;;;;;;;;;;;;} ;while ;($mrc ;;CURLM_CALL_MULTI_PERFORM);
    ;;;;;;;;;;;;;;} ;while ;($active ;and ;$mrc ;;CURLM_OK) ;{ [Page]
    ;;;;;;;;;;;;;;} ;(curl_multi_select($mh) ;!= ;-1) ;{
    ;;;;;;;;;;;;;;} ;do ;{
    ;;;;;;;;;;;;;;} ;$mrc ;= ;curl_multi_exec($mh, ;$active);
    ;;;;;;;;;;;;;;} ;while ;($mrc ;;CURLM_CALL_MULTI_PERFORM);
    ;;;;;;;;;;;;;;}
    ;;;;;;;;;;;;;;}

```

\$active要等全部url数据接受完毕才变成false所以这里用到了curl\_multi\_exec返回值判断是否还有数据当有数据时候就不停curl\_multi\_exec暂时没有数据就进入select阶段新数据来就可以被唤醒继续执行这里好处就是CPU无谓消耗没有了

另外:还有些细节地方可能有时候要遇到:

控制每个请求超时时间在curl\_multi\_add\_handle的前通过curl\_opt去做:

```
curl_opt($ch, ;CURLOPT_TIMEOUT, ;$timeout);
```

```
判断是否超时了或者其他在curl_multi_getcontent的前用:curl_error($conn[$i]);
```

```
疯狂代码 http://CrazyCoder.cn/
```

2009-2-12 3:56:42