

PHP cURL 函数

概述

PHP支持的由Daniel Stenberg创建的libcurl库允许你与各种的服务器使用各种类型的协议进行连接和通讯。

libcurl目前支持http、https、ftp、gopher、telnet、dict、file和ldap协议。libcurl同时也支持HTTPS认证、HTTP POST、HTTP PUT、FTP 上传(这个也能通过PHP的FTP扩展完成)、HTTP 基于表单的上传、代理、cookies和用户名+密码的认证。

PHP中使用cURL实现Get和Post请求的方法

这些函数在PHP 4.0.2中被引入。

需求

为了使用PHP的cURL函数，你需要安装 [» libcurl](#)包。

PHP需要使用libcurl 7.0.2-beta 或者更高版本。在PHP 4.2.3 里使用cURL，你需要安装7.9.0或更高版本的libcurl。从PHP 4.3.0 开始你需要安装7.9.0或更高版本的libcurl。从PHP 5.0.0开始你需要安装7.10.5或更高版本的libcurl。

安装

要使用PHP的cURL支持你必须在编译PHP时加上--with-curl=[DIR] 选项，DIR为包含lib和include的目录路径。在include目录中必须有一个名为curl，包含了easy.h和curl.h的文件夹。lib文件夹里应该有一个名为libcurl.a的文件。对于PHP 4.3.0你可以配置--with-curlwrappers 使cURL使用URL流。

注意: Win32用户注意 要在Windows环境下使用这个模块，libeay32.dll和ssleay32.dll必须放到PATH环境变量包含的目录下。不用cURL网站上的libcurl.dll。

资源类型

这个扩展定义了2中资源：cURL句柄和cURL批处理句柄。

PHP cURL 函数

以下包含了PHP cURL函数列表：

函数	描述
curl_close()	关闭一个cURL会话。
curl_copy_handle()	复制一个cURL句柄和它的所有选项。
curl_errno()	返回最后一次的错误号。
curl_error()	返回一个保护当前会话最近一次错误的字符串。

curl_escape()	返回转义字符串，对给定的字符串进行URL编码。
curl_exec()	执行一个cURL会话。
curl_file_create()	创建一个 CURLFile 对象。
curl_getinfo()	获取一个cURL连接资源句柄的信息。
curl_init()	初始化一个cURL会话。
curl_multi_add_handle()	向curl批处理会话中添加单独的curl句柄。
curl_multi_close()	关闭一组cURL句柄。
curl_multi_exec()	运行当前 cURL 句柄的子连接。
curl_multi_getcontent()	如果设置了CURLOPT_RETURNTRANSFER，则返回获取的输出的文本流。
curl_multi_info_read()	获取当前解析的cURL的相关传输信息。
curl_multi_init()	返回一个新cURL批处理句柄。
curl_multi_remove_handle()	移除curl批处理句柄资源中的某个句柄资源。
curl_multi_select()	等待所有cURL批处理中的活动连接。
curl_multi_setopt()	设置一个批处理cURL传输选项。
curl_multi_strerror()	返回描述错误码的字符串文本。
curl_pause()	暂停及恢复连接。
curl_reset()	重置libcurl的会话句柄的所有选项。
curl_setopt_array()	为cURL传输会话批量设置选项。
curl_setopt()	设置一个cURL传输选项。
curl_share_close()	关闭cURL共享句柄。
curl_share_init()	初始化cURL共享句柄。
curl_share_setopt()	设置一个共享句柄的cURL传输选项。
curl_strerror()	返回错误代码的字符串描述。
curl_unescape()	解码URL编码后的字符串。

[curl_version\(\)](#)

获取cURL版本信息。

[← PHP curl_init函数](#)[PHP curl_multi_add_handle函数 →](#)

1 篇笔记



写笔记



PHP 利用 curl 发送 post get del put patch 请求

因为需要在 php 开发中对接其它接口需要用 php curl 去对接其它接口 我把他们封装成函数 希望能对大家有所帮助。

这里面是封装好的会自动把 data 进行转成 json 格式，同时解码成 php 数组输出。

```
<?php
function geturl($url){
    $headerArray =array("Content-type:application/json;", "Accept:application/json");
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE);
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, FALSE);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($url,CURLOPT_HTTPHEADER,$headerArray);
    $output = curl_exec($ch);
    curl_close($ch);
    $output = json_decode($output,true);
    return $output;
}

function posturl($url,$data){
    $data = json_encode($data);
    $headerArray =array("Content-type:application/json;charset='utf-8'", "Accept:application/json");
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $url);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, FALSE);
    curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, FALSE);
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $data);
    curl_setopt($curl,CURLOPT_HTTPHEADER,$headerArray);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    $output = curl_exec($curl);
    curl_close($curl);
    return json_decode($output, true);
}
```

```
function puturl($url,$data){
    $data = json_encode($data);
    $ch = curl_init(); //初始化CURL句柄
    curl_setopt($ch, CURLOPT_URL, $url); //设置请求的URL
    curl_setopt ($ch, CURLOPT_HTTPHEADER, array('Content-type:application/json'));
    curl_setopt($ch, CURLOPT_RETURNTRANSFER,1); //设为TRUE把curl_exec()结果转化为字串，而不是直接输出

    curl_setopt($ch, CURLOPT_CUSTOMREQUEST,"PUT"); //设置请求方式
    curl_setopt($ch, CURLOPT_POSTFIELDS, $data); //设置提交的字符串
    $output = curl_exec($ch);
    curl_close($ch);
    return json_decode($output,true);
}

function delurl($url,$data){
    $data = json_encode($data);
    $ch = curl_init();
    curl_setopt ($ch,CURLOPT_URL,$url);
    curl_setopt ($ch, CURLOPT_HTTPHEADER, array('Content-type:application/json'));
    curl_setopt ($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt ($ch, CURLOPT_CUSTOMREQUEST, "DELETE");
    curl_setopt($ch, CURLOPT_POSTFIELDS,$data);
    $output = curl_exec($ch);
    curl_close($ch);
    $output = json_decode($output,true);
}

function patchurl($url,$data){
    $data = json_encode($data);
    $ch = curl_init();
    curl_setopt ($ch,CURLOPT_URL,$url);
    curl_setopt ($ch, CURLOPT_HTTPHEADER, array('Content-type:application/json'));
    curl_setopt ($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt ($ch, CURLOPT_CUSTOMREQUEST, "PATCH");
    curl_setopt($ch, CURLOPT_POSTFIELDS,$data); //20170611修改接口，用/id的方式传递，直接写在url中了
    $output = curl_exec($ch);
    curl_close($ch);
    $output = json_decode($output);
    return $output;
}
?>
```

llmrU 5个月前 (10-22)

