**MsgSrv V1 PHP Extension API DOC**

***V1.0.0*** ***Released***

msgsrv扩展是基于PHP的扩展标准使用Zend以及标准C API开发，用于为PHP提供快速方便连接msgsrv并使用msgsrv服务；该版本基于msgsrv v1版本协议实现，提供简单的API使用规则。总体操作分为两步：

1. 建立连接
2. 发送或请求操作
3. 关闭连接

具体操作API在后面的内容会讲解。

# 方法摘要

## 1 建立连接

msgsrv\_open — 创建一个与msgsrv的链接。

**说明：**

resource|bool **msgsrv\_open** ( string *$host*, int *$port*, string *$appname*, string *$username*, string *$passwrod* )

完成连接到msgsrv并自动登陆到msgsrv。

**参数：**

***host***

msgsrv所在的地址。

***port***

msgsrv所在计算机监听端口。

***appname***

当前连接在登陆msgsrv认证时提交。

***username***

当前连接在登陆msgsrv认证时提交使用的username。

***passwrod***

当前连接在登陆msgsrv认证时提交使用的password。

**返回值：**

如果成功则返回一个 msgsrv 连接标识，或者在失败时返回FALSE。

**范例：**

<?php

$link = msgsrv\_open('localhost', 6004, 'PHP', 'PHP', 'PHP');

if (!$link) {

die('Could not connect: ' . msgsrv\_last\_error());

}

echo 'Connected successfully!';

msgsrv\_close($link);

?>

## 发送消息

msgsrv\_send — 发送消息。

**说明：**

bool **msgsrv\_send** ( string *$target*, string *$cmd*, string *$content*,  resource *$link* )

通过msgsrv连接发送消息。

**参数：**

***target***

目标app名。

***cmd***

发送消息命令。

***content***

发送消息内容字段。

***link***

通过该msgsrv连接发送消息。

**返回值：**

发送成功则返回TRUE，失败返回 FALSE。

**范例：**

<?php

$link = msgsrv\_open('localhost', 6010, 'PHP', 'PHP', 'PHP');

if (!$link) {

die('Could not connect: ' . msgsrv\_last\_error());

}

echo 'Connected successfully!';

if (msgsrv\_send("root", "command", "content", $link) == FALSE) {

echo 'message send failed! ' . msgsrv\_last\_error();

exit();

}

echo 'message sent!';

msgsrv\_close($link);

?>

## 接收消息

msgsrv\_receive — 接收消息。

**说明：**

bool **msgsrv\_receive**( function *$callback* [, long *$timeout* [, resource *$link*] ])

通过msgsrv连接主动接收消息，该方法不能关联到消息所属请求。

**参数：**

***callback***

本次接收请求结果回调处理函数，可以使函数名，可以是函数定义。

***timeout***

等待请求结果返回超时时间，单位ms，默认值为php.ini中配置的msgsrv.request\_timeout参数。

***link***

制定尝试接收该连接的消息，不传递则尝试接收当前页面所有连接的消息。

**返回值：**

接收成功则返回TRUE，失败返回 FALSE。

**回调函数范例：**

function($status, $source, $command, $content, $link) {

if ($status != 1) {

echo 'Response:' . $source . $command . $content;

} else {

echo 'Request failed!' . $status;

}

}

备注说明：

$status：本次请求状态。（[参考](#_回调函数状态定义)）

$source：消息发送方

$command：消息命令字段

$content：消息内容字段

$link：请求对应的msgsrv连接

**范例：**

<?php

$link = msgsrv\_open('localhost', 6004, 'PHP', 'PHP', 'PHP');

if (!$link) {

die('Could not connect: ' . msgsrv\_last\_error());

}

echo 'Connected successfully!';

if (msgsrv\_send("root", "echo", "content", $link) == FALSE) {

echo 'message send failed! ' . msgsrv\_last\_error();

exit();

}

echo 'message sent!';

if (msgsrv\_receive(function() { }, 5000, $link) == FALSE) {

echo 'message send failed! ' . msgsrv\_last\_error();

}

msgsrv\_close($link);

?>

## 请求消息

msgsrv\_send — 发送消息。

**说明：**

array|bool **msgsrv\_request** ( string *$target*, string *$cmd*, string *$content*, resource *$link* [, long *$timeout* ] )

通过msgsrv连接发送消息。

**参数：**

***target***

目标app名。

***cmd***

发送消息命令。

***content***

发送消息内容字段。

***link***

通过该msgsrv连接发送消息。

***timeout***

等待请求结果返回超时时间，单位ms，默认值为php.ini中配置的msgsrv.request\_timeout参数。

**返回值：**

发送成功则返回消息对象数组，请求失败返回 FALSE。

**范例：**

<?php

$link = msgsrv\_open('localhost', 6010, 'PHP', 'PHP', 'PHP');

if (!$link) {

die('Could not connect: ' . msgsrv\_last\_error());

}

echo 'Connected successfully!';

if (($response = msgsrv\_request("root", "Echo", "content", $link)) == FALSE) {

echo 'request failed! ' . msgsrv\_last\_error();

}

var\_dump($response);

msgsrv\_close($link);

?>

**结果范例：**

array(3) {

["appname"]=>

string(15) "root.e.f6.M6010"

["command"]=>

string(4) "Echo"

["content"]=>

string(138) "content"

}

## 错误信息

msgsrv\_last\_error — 返回最新的异常的错误码。

**说明：**

int **msgsrv\_last\_error** ( void )

返回最近操作失败的错误码，详细定义参考附录。

**返回值：**

返回最近操作失败的错误码，详细定义参考附录。

## 关闭连接

msgsrv\_close — 关闭指定msgsrv连接。

**说明：**

void **msgsrv\_close** ( resource *$link* )

通过msgsrv连接发送消息。

**参数：**

***link***

需要关闭的msgsrv连接。

注：扩展在每个请求处理完后会自动关闭连接，但是还是建议在连接不再使用的时候及时调用该方法关闭连接。

# 安装配置

## 1 参数配置

***msgsrv.debug***

开启调试模式[1开启，0关闭（默认）]。开启调试模式后，扩展将通过syslog记录所有通信报文。

***msgsrv.read\_buffer\_size***

设置扩展读缓冲区大小，单位byte，默认1024。设置合理的socket读缓冲区大小有助于性能优化。

***msgsrv.request\_timeout***

配置扩展msgsrv请求时默认超时时间，单位ms，默认30000ms。请求超时也可以在使用msgsrv\_request函数时通过timeout指定，若未指定将会使用这里配置的时间。

***msgsrv.*** ***read\_timeout***

扩展每次尝试读取一条msgsrv协议时的超时时间，单位ms，默认30000ms。该参数会和msgsrv\_request中的timeout时间配合使用，。

***msgsrv.*** ***select\_timeout***

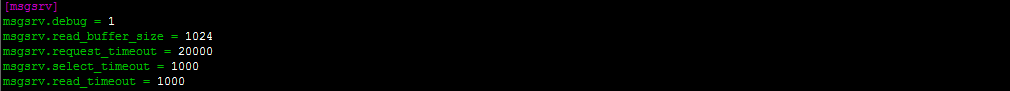
指定底层socket每次select超时时间，单位ms，默认1000ms。

## 2 安装并启用扩展

在php.ini中添加如下配置：



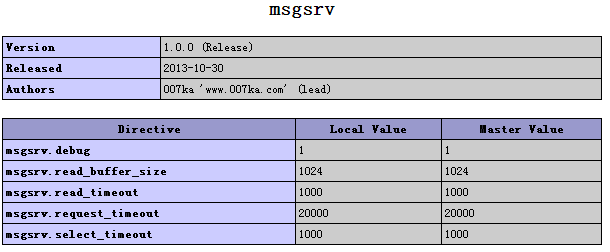
在php.ini中添加msgsrv参数如下：



所有都配置好后，保存配置然后重启apache即可。

## 3 验证安装成功

以上安装完成后，通过查看phpinfo找到msgsrv相关配置正常显示即安装成功，如下图：



# 附录

## 错误码定义

|  |  |  |
| --- | --- | --- |
| **类型** | **结果码** | **描述** |
| 通用 | 0 | 没有错误 |
| 3111 | 套接字读错误 |
| 3112 | 套接字读取超时 |
| 3113 | 套接字写错误 |
| 3501 | 扩展内部错误 |
| 登陆 | 3002 | 收到无效数据包 |
| 3003 | MsgSrv登陆认证失败 |
| 3004 | MsgSrv返回错误 |
| 3101 | 套接字创建失败 |
| 3102 | 套接字连接失败 |
|  | 3103 | 套接字出现未知错误 |
| 发送/请求 | 3404 | 目标App未连接 |

## 回调函数状态定义

|  |  |  |
| --- | --- | --- |
| **类型** | **结果码** | **描述** |
| 通用 | 1 | 接收成功 |
| 失败 | 0 | 没有连接可用（在未指定msgsrv连接时返回） |
| -1 | 连接已关闭 |
| -2 | 读取结果返回超时 |
| -3 | 套接字读取错误 |
| -4 | 扩展内部错误 |
| -5 | 目前App未找到 |
| -6 | MsgSrv返回错误 |