MsgSrv PHP Extension

***v1.2.0*** ***Release***

msgsrv扩展是基于PHP的扩展标准使用Zend API开发，为PHP提供快速方便的msgsrv服务调用方案，并支持msgsrv长连接。

1. 建立连接
2. 发送或请求操作
3. 关闭连接

# 方法摘要

## 1 建立连接

resource msgsrv\_connect(string $host, int $port, string $app, string $user, string $pass);

**说明：**

连接到msgsrv并返回可用连接，连接失败返回**FALSE**。

注：为兼容v1.0保留方法msgsrv\_open,和该方法参数一致。

**参数：**

***host***

msgsrv服务器地址。

***port***

msgsrv监听端口。

***app***

登录msgsrv陆认证appname。

***user***

登录msgsrv陆认证username。

***pass***

登录msgsrv陆认证password。

**返回值：**

如果成功则返回一个 msgsrv 连接，或者在失败时返回**FALSE**。

**范例：**

<?php

$link = msgsrv\_connect('10.2.10.202', 6008, 'PHP', 'PHP', 'PHP');

if (!$link) {

die('Could not connect msgsrv…');

}

echo 'Connected successfully!';

msgsrv\_close($link);

?>

## 建立连接(兼容)

resource msgsrv\_open(string $host, int $port, string $app, string $user, string $pass[, int persistent]);

**说明：**

连接到msgsrv并返回可用连接，连接失败返回**FALSE**。

注：为兼容v1.0保留方法，persistent默认值0。

**参数：**

***host***

msgsrv服务器地址。

***port***

msgsrv监听端口。

***app***

登录msgsrv陆认证appname。

***user***

登录msgsrv陆认证username。

***pass***

登录msgsrv陆认证password。

***persistent[可选]***

长连接标识，1长连接，0段连接。

**返回值：**

如果成功则返回一个 msgsrv 连接，或者在失败时返回**FALSE**。

**范例：**

<?php

$link = msgsrv\_open('10.2.10.202', 6008, 'PHP', 'PHP', 'PHP');

if (!$link) {

die('Could not connect msgsrv…');

}

echo 'Connected successfully!';

msgsrv\_close($link);

?>

## 建立连接(持久化)

resource msgsrv\_pconnect(string $host, int $port, string $app, string $user, string $pass);

**说明：**

连接到msgsrv并返回可用连接，连接失败返回**FALSE**。如果当前PHP环境不支持**ZTS**将自动转为普通连接。每次连接创建随机使用使用连接**1~ msgsrv.max\_pool\_size**的连接，

1. 若连接不存在，则建立连接，并返回长连接；
2. 若连接存在，且不在使用中，则测试连接可用，可用则返回该连接作为长连接；测试不可用时则释放连接并重新尝试建立长连接。
3. 若连接存在，且使用中，则直接转使用普通连接。

**参数：**

***host***

msgsrv服务器地址。

***port***

msgsrv监听端口。

***app***

登录msgsrv陆认证appname。

***user***

登录msgsrv陆认证username。

***pass***

登录msgsrv陆认证password。

**返回值：**

如果成功则返回一个 msgsrv 连接，或者在失败时返回FALSE。

**范例：**

<?php

$plink = msgsrv\_pconnect('10.2.10.202', 6008, 'PHP', 'PHP', 'PHP');

if (!$plink) {

die('Could not connect msgsrv…');

}

echo 'Connected successfully!';

msgsrv\_close($plink);

?>

## 消息请求

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

**说明：**

通过连接发送指定消息，并在timeout时间内阻塞等待结果返回，若成功返回返回结果数组，失败返回**FALSE**。

返回Array原型：

array(3) {

["from"]=> string(15) "root.e.f6.M6010"

["cmd"]=> string(4) "Echo"

["body"]=> string(138) "body" /\* 如果返回数据没有body字段，则数组也无该成员！ \*/

}

**参数：**

***target***

目标app名。

***cmd***

发送消息命令。

***body***

发送消息内容字段。

***link***

通过该msgsrv连接发送消息。

***timeout[可选]***

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

**返回值：**

如果成功则返回应答数组，或者在失败时返回FALSE。

**范例：**

<?php

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

if (!$link) {

die('Could not connect…');

}

echo 'Connected successfully!';

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

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

}

msgsrv\_close($link);

?>

## 发送消息

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

**说明：**

通过连接发送指定消息，成功发送后返回，发送成功返回**TRUE**，失败返回**FALSE**。

**参数：**

***target***

目标app名。

***cmd***

发送消息命令。

***body***

发送消息内容字段。

***link***

通过该msgsrv连接发送消息。

**范例：**

<?php

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

if (!$link) {

die('Could not connect…');

}

echo 'Connected successfully!';

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

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

exit();

}

echo 'message sent!';

msgsrv\_close($link);

?>

## 接收消息

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

**说明：**

主动接收消息，接收所有成功返回**TRUE**，出现异常返回**FALSE**。

返回callback原型：

function ($status, $from, $cmd, $body, $link) {

if ($status == 1) {

echo 'Response:' . $from . ' ' . $cmd . ' ' . $body;

} else {

echo 'Request failed!' . $status;

}

}

注：若发生找不到目标APP，也会触发callback回调。

**参数：**

***callback***

处理结果回调函数。

***timeout[可选]***

等待结果超时时间，单位s，默认值为php.ini中配置的msgsrv.request\_timeout参数。

***limit[可选]***

接收消息条数限制，默认1条。

***link[可选]***

指定只接收该连接的消息。默认接收所有连接的消息。

## 连接地址

string **msgsrv\_full\_app**(resource $link)

说明：

返回获取指定连接的full\_app。

## 错误信息

int **msgsrv\_last\_error**(resource $link)

说明：

返回指定连接最近发生错误的错误码。

## 关闭连接

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

说明：

关闭指定msgsrv连接，

*注：对于长连接不会断开，仅释放连接资源，并返回连接到连接池。*

# 安装配置

## 1 参数配置

***msgsrv.trance\_mode***

开启跟踪模式[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.allow\_persistent***

允许长连接开关，[1允许，0关闭(默认)]。开启后，允许msgsrv连接公用。关闭后，所有msgsrv\_pconnect打开的连接都默认使用短连接。

***msgsrv.max\_persistent***

扩展允许最大长连接数量，包括长连接与短连接至和。当超过后连接将不允许建立失败。当为-1时不限制；默认-1。

***msgsrv.max\_links***

扩展允许最大连接数量[>0, -1(默认)]，包括长连接与短连接至和。当超过后连接将不允许建立失败。当为-1时不限制；默认-1。

***msgsrv.max\_pool\_size***

配置每个app的长连接池数量大小，当超过连接数量大小后，将转使用普通短连接。

***msgsrv.connect\_timeout***

连接超时时间，在扩展连接到msgsrv的超时时间；单位s，默认10s。

***msgsrv.request\_timeout***

请求时默认超时时间，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；单位s，默认10s。

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

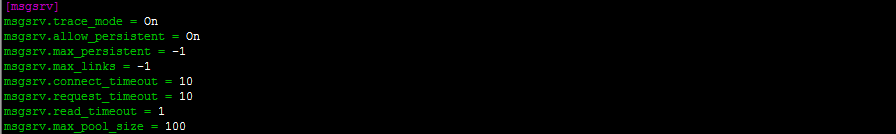
类似select函数中的每次轮询超时时间；单位s，默认2s。

## 2 安装并启用扩展

在php.ini中添加如下配置：



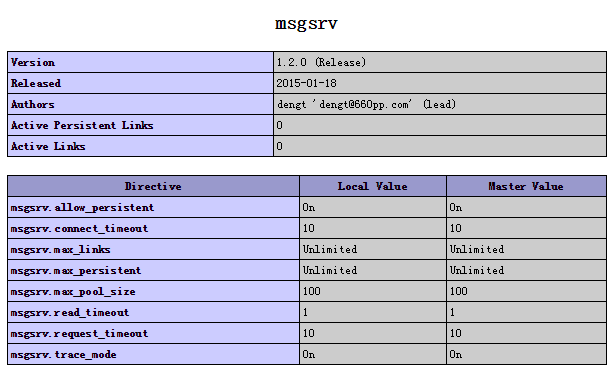
在php.ini中添加msgsrv参数如下：



所有都配置好后，保存配置然后重启apache/php-fpm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** |
| 连接 | 0 | 没有错误 |
| 3112 | 数据读取超时 |
| 3111 | 数据读取错误 |
| 3113 | 数据写入失败 |
| 操作 | 3102 | 连接已断开 |
| 3501 | 其他未知错误，可能是bug。 |
| 3002 | 底层协议错误，可能是bug。 |
| 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 | 消息接受成功 |
| 失败 | 0 | 读取消息超时 |
| -1 | 连接已关闭 |
| -2 | 数据读取出现异常 |
| -3 | 底层协议错误 |
| -4 | 目标APP未连接 |
| -9 | 其他未知错误，可能是bug。 |