

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);
?>
```

2 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);
?>
```

3 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
$link = msgsrv_pconnect('10.2.10.202', 6008, 'PHP', 'PHP', 'PHP');
if (!$link) {
    die('Could not connect msgsrv...');
}
echo 'Connected successfully!';
msgsrv_close($link);
?>
```

4 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);
?>
```

5 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);
?>
```

6 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。

7 连接地址

string **msgsrv_full_app**(resource \$link)

说明：

返回获取指定连接的 full_app。

8 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。

9 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.trace_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:

```
configuration for php msgsrv module
extension=msgsrv.so
```

在 php.ini 中添加 msgsrv 参数如下:

```
[msgsrv]
msgsrv.trace_mode = On
msgsrv.allow_persistent = On
msgsrv.max_persistent = -1
msgsrv.max_links = -1
msgsrv.connect_timeout = 10
msgsrv.request_timeout = 10
msgsrv.read_timeout = 1
msgsrv.max_pool_size = 100
```

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：

msgsrv

Version	1.2.0 (Release)
Released	2015-01-18
Authors	dengt 'dengt@660pp.com' (lead)
Active Persistent Links	0
Active Links	0

Directive	Local Value	Master Value
msgsrv.allow_persistent	On	On
msgsrv.connect_timeout	10	10
msgsrv.max_links	Unlimited	Unlimited
msgsrv.max_persistent	Unlimited	Unlimited
msgsrv.max_pool_size	100	100
msgsrv.read_timeout	1	1
msgsrv.request_timeout	10	10
msgsrv.trace_mode	On	On

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。