

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 *\$password*)

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。

password

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);
?>
```

2 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);
?>
```

3 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)

\$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);
?>
```

4 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"
}
```

5 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。

6 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：

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

在 php.ini 中添加 msgsrv 参数如下：

```
[msgsrv]
msgsrv.debug = 1
msgsrv.read_buffer_size = 1024
msgsrv.request_timeout = 20000
msgsrv.select_timeout = 1000
msgsrv.read_timeout = 1000
```

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：

msgsrv

Version	1.0.0 (Release)	
Released	2013-10-30	
Authors	007ka 'www.007ka.com' (Lead)	

Directive	Local Value	Master Value
msgsrv.debug	1	1
msgsrv.read_buffer_size	1024	1024
msgsrv.read_timeout	1000	1000
msgsrv.request_timeout	20000	20000
msgsrv.select_timeout	1000	1000

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