



链滴

# PHP Socket 的使用

作者: [JinFengYi](#)

原文链接: <https://ld246.com/article/1545384633887>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# PHP Socket的使用

最近在做的项目有一项需要耗时任务在后台运行的功能，虽然PHP并不是非常适合做常驻后台的守护程，但是由于项目主要代码都是基于PHP实现，如果运行在后台的守护进程改换别的语言会非常不方便。所以不可避免会涉及到Web端和Daemon部分的通信，Socket是一个不错的方式。

## Socket是什么

socket的英文原义是“孔”或“插座”。作为BSD UNIX的进程通信机制，取后一种意思。通常也称“套接字”，用于描述IP地址和端口，是一个通信链的句柄。在Internet上的主机一般运行了多个服务软件，同时提供几种服务。每种服务都打开一个Socket，并绑定到一个端口上，不同的端口对应于不同的服务。

以上内容来自[百度百科](#)

简单说来，socket可以帮助不同的服务在不同的端口进行通信。

## PHP中的实现

### 服务端

```
<?php
set_time_limit(0);
// 设置主机和端口
$host = "127.0.0.1";
$port = 12387;
// 创建一个tcp流
$socket = socket_create(AF_INET, SOCK_STREAM, SOL_TCP)
    or die("socket_create() failed:" . socket_strerror(socket_last_error()));

// 设置阻塞模式
socket_set_block($socket)
    or die("socket_set_block() failed:" . socket_strerror(socket_last_error()));

// 绑定到端口
socket_bind($socket, $host, $port)
    or die("socket_bind() failed:" . socket_strerror(socket_last_error()));

// 开始监听
socket_listen($socket, 4)
    or die("socket_listen() failed:" . socket_strerror(socket_last_error()));

echo "Binding the socket on $host:$port ... \n";

while (true) {

    // 接收连接请求并调用一个子连接Socket来处理客户端和服务端间的信息
    if (($msgsock = socket_accept($socket)) < 0) {
        echo "socket_accept() failed:" . socket_strerror(socket_last_error());
    }else{
        // 读数据
```

```

$out = "";
while($buf = socket_read($msgsock,8192)){
    $out .= $buf;
}

// 写数据
$in = "数据是 $out";
socket_write($msgsock, $in, strlen($in));
}
// 结束通信
socket_close($msgsock);
}
socket_close($socket);
?>

```

服务端主要进行了以下步骤：

- 创建Socket的监听，等待连接
- 当链接到来时，开启一个子连接处理IO
- 接收来自客户端的传输数据
- 将结果写回给客户端

## 客户端

```

<?php
set_time_limit(0);
$host = "127.0.0.1";
$port = 12387;

// 创建一个tcp流
$socket = socket_create(AF_INET, SOCK_STREAM, SOL_TCP)
    or die("socket_create() failed:" . socket_strerror(socket_last_error()));

echo "try to connect to $host:$port...\n";
$result = socket_connect($socket, $host, $port)
    or die("socket_connect() failed:" . socket_strerror(socket_last_error()));

$in = "hello \n";
if(!socket_write($socket, $in, strlen($in))) {
    echo "socket_write() failed:" . socket_strerror($socket);
}else {
    echo "发送成功! \n";
}

$out = "";
while($buf = socket_read($socket, 8192)) {
    $out .= $buf;
}
echo "接受内容为: $out \n";
socket_close($socket);
?>

```

客户端主要有以下步骤：

- 连接到服务端Socket
- 向服务端写数据
- 接收来自服务端的数据