参考：

<http://doc2.workerman.net/send-to-uid.html>

github资料

<https://github.com/walkor/GatewayWorker>

## GatewayWorker

### 安装

composer require workerman/gateway-worker

GatewayWorker基于Workerman开发的一个项目框架，用于快速开发TCP长连接应用，例如app推送服务端、即时IM服务端、游戏服务端、物联网、智能家居等等

## linux安装

### 环境要求

Linux用户只能使用Linux版本的Workerman。

1、安装PHP>=5.3.3，并安装了pcntl、posix扩展

2、建议安装event或者libevent扩展，但不是必须的（注意event扩展需要PHP>=5.4）

### Posix缺失

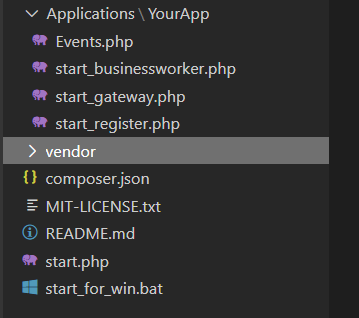
yum install -y php-process

默认安装的php 5.6的扩展

本人服务器php为7.2

所以使用yum install php72w-process 安装

## gateway框架目录



### gateway

文件位置：GatewayWorker/Gateway.php

Gateway类用于初始化Gateway进程。Gateway进程是暴露给客户端的让其连接的进程。所有客户端的请求都是由Gateway接收然后分发给BusinessWorker处理，同样BusinessWorker也会将要发给客户端的响应通过Gateway转发出去。

### register

Register类其实也是基于基础的Worker开发的。Gateway进程和BusinessWorker进程启动后分别向Register进程注册自己的通讯地址，Gateway进程和BusinessWorker通过Register进程得到通讯地址后，就可以建立起连接并通讯了。

### BusinessWorker

BusinessWorker类其实也是基于基础的Worker开发的。BusinessWorker是运行业务逻辑的进程，BusinessWorker收到Gateway转发来的事件及请求时会默认调用Events.php中的onConnect onMessage onClose方法处理事件及数据，开发者正是通过实现这些回调控制业务及流程。

## 启动流程

运行php start.php start

## 工作原理

1、Register、Gateway、BusinessWorker进程启动

2、Gateway、BusinessWorker进程启动后向Register服务进程发起长连接注册自己

3、Register服务收到Gateway的注册后，把所有Gateway的通讯地址保存在内存中

4、Register服务收到BusinessWorker的注册后，把内存中所有的Gateway的通讯地址发给BusinessWorker

5、BusinessWorker进程得到所有的Gateway内部通讯地址后尝试连接Gateway

6、如果运行过程中有新的Gateway服务注册到Register（一般是分布式部署加机器），则将新的Gateway内部通讯地址列表将广播给所有BusinessWorker，BusinessWorker收到后建立连接

7、如果有Gateway下线，则Register服务会收到通知，会将对应的内部通讯地址删除，然后广播新的内部通讯地址列表给所有BusinessWorker，BusinessWorker不再连接下线的Gateway

8、至此Gateway与BusinessWorker通过Register已经建立起长连接

9、客户端的事件及数据全部由Gateway转发给BusinessWorker处理，BusinessWorker默认调用Events.php中的onConnect onMessage onClose处理业务逻辑。

10、BusinessWorker的业务逻辑入口全部在Events.php中，包括onWorkerStart进程启动事件(进程事件)、onConnect连接事件(客户端事件)、onMessage消息事件（客户端事件）、onClose连接关闭事件（客户端事件）、onWorkerStop进程退出事件（进程事件）

Tp5使用 安装gatewayclient

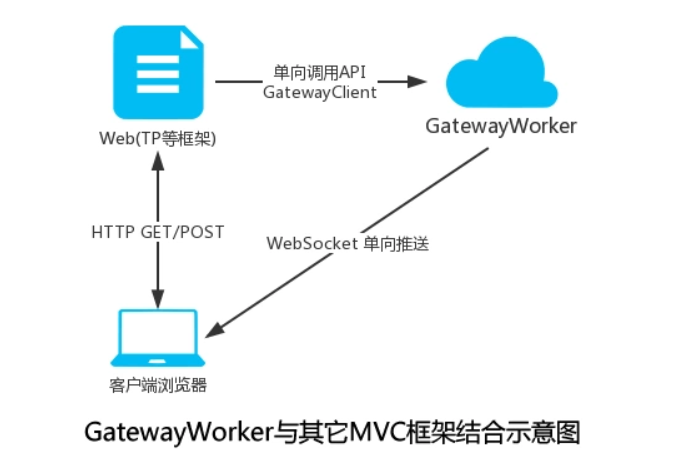
## Gatewayclient

composer require workerman/gatewayclient

参考：<https://github.com/walkor/GatewayClient>

## getwayWorker连接mysql

参考：<http://doc.workerman.net/components/workerman-mysql.html>



**前端建立websocket连接，sendmessage触发gatewaywork onmessage方法此时绑定$clientid和userid，tp5 sendtoUId 所有的客户端都会收到消息 ，即前端**

### onConnect

void Events::onConnect(string $client\_id);

当建立连接的时候系统会默认给进程分配一个独立的、唯一的$client\_id

### sendToClient

void Gateway::sendToClient(string $client\_id, string $send\_data);

向客户端$client\_id发送数据，如果对应的client\_id对应的客户端不存在或不在线则自动丢弃发送数据。

### SendtoUid

#### 说明

向uid绑定的所有**在线**client\_id发送数据。

注意：默认uid与client\_id是一对多的关系，如果当前uid下绑定了多个client\_id，则多个client\_id对应的客户端都会收到消息，这类似于PC QQ和手机QQ同时在线接收消息。

#### 参数

$uid

uid可以是字符串、数字、或者包含uid的数组。如果为数组，则是给数组内所有uid发送数据

$message

要发送的数据（字符串类型），此数据会被Gateway所使用协议的encode方法打包后再发送给客户端

#### 返回值

因为数据发送是异步进行的，所以没有返回值。一般来说只要uid在线就可以发送成功。

发送前先可以用Gateway::getClientIdByUid判断下uid是否有在线的client\_id

## gateWorker实时聊天demo

## 分布式部署