


```

const CHANNEL = '__keyevent@0__:expired';
const AUTH    = "yuechain!@#\$";
}

class TimeWheel
{
    protected $_redisObj;
    protected $_channel;

    /**
     * @desc
     * @access
     * @param string $ip      redis的ip地址
     *          number $port   redis的端口号
     *          string $channel redis消息通道
     * @return
     * @example
     * @date 2016年11月17日
     * @author benjamin
     */
    function __construct($ip=TimeWheelConfig::IP,$port=TimeWheelConfig::PORT,$channel=TimeWheelConfig::CHANNEL,$auth=TimeWheelConfig::AUTH)
    {
        ini_set('default_socket_timeout', -1);
        $this->_redisObj = new Redis();
        $this->_redisObj->connect($ip, $port);
        $this->_redisObj->auth($auth);
        $this->_channel = $channel;
    }

    /**
     * @desc 业务逻辑回调
     * @access
     * @param string $callback 匿名回调函数
     * @return
     * @example
     * @date 2016年11月17日
     * @author benjamin
     * @memo
     * 注意点一: callback回调函数有固定格式, 格式如下所示
     * function($instance, $channelName, $message){

                // $instance 是redis实例, 系统自动传递该参数, 无需手动设置
                // $channelName 是消息通道名字
                // $message 是redis的key
                // 所有业务逻辑围绕message展开

            }

    * 注意点二: message设计参考如下
    * 数据库名: 表名: unique字段名: unique字段值: 其他需要处理的业务逻辑
    *
    */
    public function subscribe($callback)
    {
        $this->_redisObj->subscribe(array($this->_channel), $callback);
    }

    /**
     * @desc
     * @param string $key key
     * @param string $value value
     * @param string $expire 超时时间, 单位秒

```

```
* @return
* @example
* @date 2017年3月28日
* @author benjamin
*/
public function setRedis($key,$value,$expire)
{
    $this->_redisObj->set("timewheel:".$key,$value,$expire);
}
}
```

php客户端

```
import("Vendor.Timewheel.Timewheel");
$timewheelObj = new \Timewheel();
$timewheelObj->subscribe(function($instance, $channelName, $message){
//    这里写业务逻辑
    ob_flush();//如果有输出的话，这里一定要写
});
```

shell脚本

```
#文件名base-daemon-timewheel.sh
#!/bin/sh
while true;
do
    count=$(ps -ef | grep -c Wheel) #查找当前的进程中，计算server程序的数量
    if [ $count -lt 4 ]; then        #判断服务器进程的数量是否小于3（根据实际填上你的服务器进程数量）
        cd /www/shanpiao.tao3w.com
        php index.php Api Shanpiao timewheel &          #这里填入需要重启的服务器进程
    fi
    sleep 2          #睡眠2s，周期性地检测服务器程序是不是崩溃了
done
```

运行shell脚本

```
[root@iz23yrxtsktZ ~]# nohup bash ./base-daemon-timewheel.sh &
```

参考

- <http://www.mooc.com/article/10431>
- <https://my.oschina.net/itblog/blog/601284>