

# 欢迎使用



## 微信开发

# 目 录

WeChatDeveloper	
目录结构及文件描述	
环境准备及安装使用	
接口实例及配置参数	
推送事件及消息回复	
微信客服消息管理	
微信模板消息管理	
微信卡券支持管理	
微信自定义菜单管理	
微信网页授权管理	
微信网页JS SDK开发	
媒体素材图文管理	
微信粉丝标签管理	
微信蓝牙摇一摇周边	
生成带参数的二维码	
微信扫一扫管理	
微信服务号和小程序支付	
统一下单	
客户端发起支付（新）	
查询订单	
关闭订单	
申请退款	
查询退款	
支付通知	
微信小程序服务接口支持（新）	
encryptedData解码	
小程序模板消息	
微信小程序二维码	
小程序数据统计	
小程序地址位置	
小程序插件管理	

# WeChatDeveloper

unstable dev-master · stable v1.2.9 · downloads 4.99 k · license MIT

## WeChatDeveloper

- WeChatDeveloper 是基于 [wechat-php-sdk](#) 重构，优化并完善；
- 微信 SDK 运行最底要求 PHP 版本 5.4，建议在 PHP7 上运行以获取最佳性能；
- WeChatDeveloper 针对 access\_token 失效增加了自动刷新机制；
- 微信的部分接口需要缓存数据在本地，因此对目录需要有写权限；
- 我们鼓励大家使用 composer 来管理您的第三方库，方便后期更新操作；
- WeChatDeveloper 已历经数个线上项目考验，欢迎 fork 或 star 此项目。

## 功能描述

- 微信小程序，服务端接口支持
- 微信认证服务号，服务端接口支持
- 微信支付（账单、卡券、红包、退款、转账、App支付、JSAPI支付、Web支付、扫码支付等）
- 支付宝支付（账单、转账、App支付、刷卡支付、扫码支付、Web支付、Wap支付等）

## 技术帮助

PHP开发技术交流（QQ群 513350915）



WeChatDeveloper 是基于官方接口封装，在做微信开发前，必需先阅读微信官方文档。

微信官方文档：<http://mp.weixin.qq.com/wiki>

商户支付文档：<https://pay.weixin.qq.com/wiki/doc/api/index.html>

针对 WeChatDeveloper 也有一准备了帮助资料可供参考。

看云文档地址：<http://www.kancloud.cn/zoujingli/wechat-developer>

Think.Admin：<https://github.com/zoujingli/Think.Admin>

## 代码仓库

WeChatDeveloper 为开源项目，允许把它用于任何地方，不受任何约束，欢迎 fork 项目。

GitHub 托管地址：<https://github.com/zoujingli/WeChatDeveloper>

OSChina 托管地址：<http://git.oschina.net/zoujingli/WeChatDeveloper>

## 文件说明

文件名	类名	描述	类型	加载 ①
App.php	AliPay\App	支付宝App支付	支付宝	\We::AliPayApp()
Bill.php	AliPay\Bill	支付宝账单下载	支付宝	\We::AliPayBill()
Pos.php	AliPay\Pos	支付宝刷卡支付	支付宝	\We::AliPayPos()
Scan.php	AliPay\Scan	支付宝扫码支付	支付宝	\We::AliPayScan()
Transfer.php	AliPay\Transfer	支付宝转账	支付宝	\We::AliPayTransfer()
Wap.php	AliPay\Wap	支付宝Wap支付	支付宝	\We::AliPayWap()
Web.php	AliPay\Web	支付宝Web支付	支付宝	\We::AliPayWeb()
Card.php	WeChat\Card	微信卡券接口支持	认证服务号	\We::WeChatCard()
Custom.php	WeChat\Custom	微信客服消息接口支持	认证服务号	\We::WeChatCustom()
Media.php	WeChat\Media	微信媒体素材接口支持	认证服务号	\We::WeChatMedia()
Oauth.php	WeChat\Oauth	微信网页授权消息类接口	认证服务号	\We::WeChatOauth()
Pay.php	WeChat\Pay	微信支付类接口	认证服务号	\We::WeChatPay()
Product.php	WeChat\Product	微信商店类接口	认证服务号	\We::WeChatProduct()
Qrcode.php	WeChat\Qrcode	微信二维码接口支持	认证服务号	\We::WeChatQrcode()
Receive.php	WeChat\Receive	微信推送事件消息处理支持	认证服务号	\We::WeChatReceive()
Scan.php	WeChat\Scan	微信扫一扫接口支持	认证服务号	\We::WeChatScan()
Script.php	WeChat\Script	微信前端JSSDK支持	认证服务号	\We::WeChatScript()
Shake.php	WeChat\Shake	微信蓝牙设备	认证服务号	\We::WeChatShake()

Shake.php	ke	摇一摇接口	认证服务号	tShake()
Tags.php	WeChat\Tags	微信粉丝标签接口支持	认证服务号	\We::WeChatTags()
Template.php	WeChat\Template	微信模板消息接口支持	认证服务号	\We::WeChatTemplate()
User.php	WeChat\User	微信粉丝管理接口支持	认证服务号	\We::WeChatCard()
Wifi.php	WeChat\Wifi	微信门店WIFI管理支持	认证服务号	\We::WeChatWifi()
Bill.php	WePay\Bill	微信商户账单及评论	微信支付	\We::WePayBill()
Coupon.php	WePay\Coupon	微信商户代金券	微信支付	\We::WePayCoupon()
Order.php	WePay\Order	微信商户订单	微信支付	\We::WePayOrder()
Redpack.php	WePay\Redpack	微信红包支持	微信支付	\We::WePayRedpack()
Refund.php	WePay\Refund	微信商户退款	微信支付	\We::WePayRefund()
Transfers.php	WePay\Transfers	微信商户打款到零钱	微信支付	\We::WePayTransfers()
TransfersBank.php	WePay\TransfersBank	微信商户打款到银行卡	微信支付	\We::WePayTransfersBank()
Crypt.php	WeMini\Crypt	微信小程序数据加密处理	微信小程序	\We::WeMiniCrypt()
Plugs.php	WeMini\Plugs	微信小程序插件管理	微信小程序	\We::WeMiniPlugs()
Poi.php	WeMini\Poi	微信小程序地址管理	微信小程序	\We::WeMiniPoi()
Qrcode.php	WeMini\Qrcode	微信小程序二维码管理	微信小程序	\We::WeMiniCrypt()
Template.php	WeMini\Template	微信小程序模板消息支持	微信小程序	\We::WeMiniTemplate()
Total.php	WeMini\Total	微信小程序数据接口	微信小程序	\We::WeMiniTotal()

## 开源协议

- WeChatDeveloper 基于 MIT 协议发布，任何人可以用在任何地方，不受约束
- WeChatDeveloper 部分代码来自互联网，若有异议，可以联系作者进行删除

## 赞助支持

---

微信扫一扫 支付



向Anyon (\*\*立) 转账

支付宝扫一扫 支付



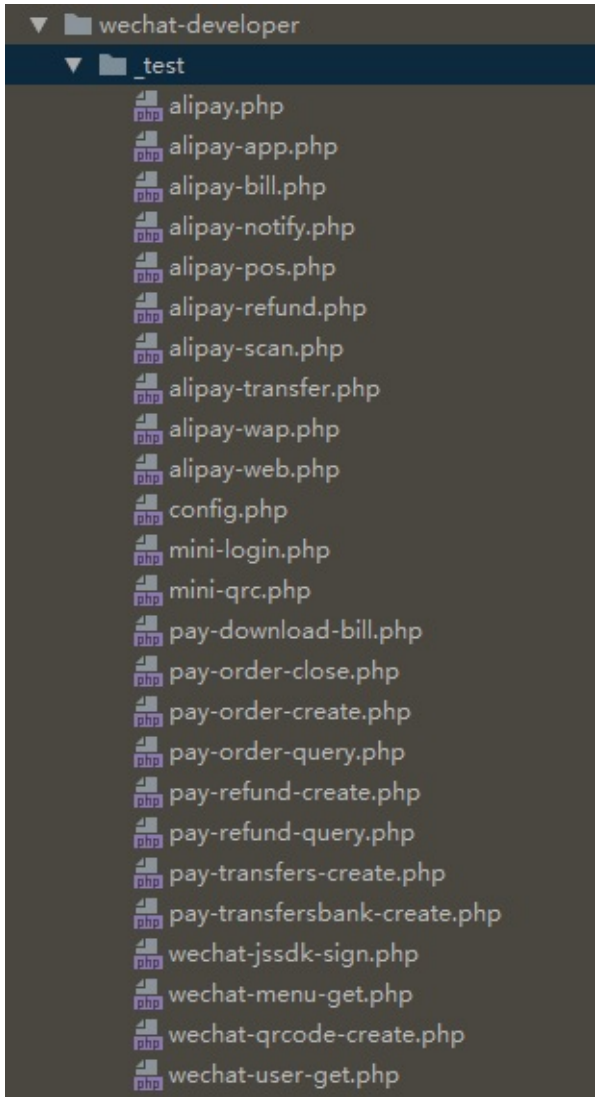
向Anyon (\*\*立) 转账

# 目录结构及文件描述

## 一、目录结构及文件描述

目录	描述
_test	SDK功能测试DEMO代码
Cache	默认缓存目录，需要拥有读写权限
AliPay	支付宝支付支持
WeChat	认证服务号支持
WeMini	微信小程序支持
WePay	微信支付支持
WeChat/Contracts	WeChat 基础支持类，通用外界不需要使用
WeChat/Exceptions	WeChat 自定义异常类，在调用接口时可以使用 try 来处理异常

## 二、类文件及功能描述（部分功能可以查看源码中的\_test测试文件）



文件名	类名	描述	类型	加载 ①
App.php	AliPay\App	支付宝App支付	支付宝	\We::AliPayApp()
Bill.php	AliPay\Bill	支付宝账单下载	支付宝	\We::AliPayBill()
Pos.php	AliPay\Pos	支付宝刷卡支付	支付宝	\We::AliPayPos()
Scan.php	AliPay\Scan	支付宝扫码支付	支付宝	\We::AliPayScan()
Transfer.php	AliPay\Transfer	支付宝转账	支付宝	\We::AliPayTransfer()
Wap.php	AliPay\Wap	支付宝Wap支付	支付宝	\We::AliPayWap()
Web.php	AliPay\Web	支付宝Web支付	支付宝	\We::AliPayWeb()
Card.php	WeChat\Card	微信卡券接口支持	认证服务号	\We::WeChatCard()



Custom.php	WeChat\Custom	微信客服消息接口支持	认证服务号	\We::WeChatCustom()
Media.php	WeChat\Media	微信媒体素材接口支持	认证服务号	\We::WeChatMedia()
Oauth.php	WeChat\Oauth	微信网页授权消息类接口	认证服务号	\We::WeChatOauth()
Pay.php	WeChat\Pay	微信支付类接口	认证服务号	\We::WeChatPay()
Product.php	WeChat\Product	微信商店类接口	认证服务号	\We::WeChatProduct()
Qrcode.php	WeChat\Qrcode	微信二维码接口支持	认证服务号	\We::WeChatQrcode()
Receive.php	WeChat\Receive	微信推送事件消息处理支持	认证服务号	\We::WeChatReceive()
Scan.php	WeChat\Scan	微信扫一扫接口支持	认证服务号	\We::WeChatScan()
Script.php	WeChat\Script	微信前端JSSDK支持	认证服务号	\We::WeChatScript()
Shake.php	WeChat\Shake	微信蓝牙设备摇一摇接口	认证服务号	\We::WeChatShake()
Tags.php	WeChat\Tags	微信粉丝标签接口支持	认证服务号	\We::WeChatTags()
Template.php	WeChat\Template	微信模板消息接口支持	认证服务号	\We::WeChatTemplate()
User.php	WeChat\User	微信粉丝管理接口支持	认证服务号	\We::WeChatCard()
Wifi.php	WeChat\Wifi	微信门店WIFI管理支持	认证服务号	\We::WeChatWifi()
Bill.php	WePay\Bill	微信商户账单及评论	微信支付	\We::WePayBill()
Coupon.php	WePay\Coupon	微信商户代金券	微信支付	\We::WePayCoupon()
Order.php	WePay\Order	微信商户订单	微信支付	\We::WePayOrder()
Redpack.php	WePay\Redpack	微信红包支持	微信支付	\We::WePayRedpack()
Refund.php	WePay\Refund	微信商户退款	微信支付	\We::WePayRefund()

Transfers.php	WePay\Transfers	微信商户打款到零钱	微信支付	\We::WePayTransfers()
TransfersBank.php	WePay\TransfersBank	微信商户打款到银行卡	微信支付	\We::WePayTransfersBank()
Crypt.php	WeMini\Crypt	微信小程序数据加密处理	微信小程序	\We::WeMiniCrypt()
Plugs.php	WeMini\Plugs	微信小程序插件管理	微信小程序	\We::WeMiniPlugs()
Poi.php	WeMini\Poi	微信小程序地址管理	微信小程序	\We::WeMiniPoi()
Qrcode.php	WeMini\Qrcode	微信小程序二维码管理	微信小程序	\We::WeMiniCrypt()
Template.php	WeMini\Template	微信小程序模板消息支持	微信小程序	\We::WeMiniTemplate()
Total.php	WeMini\Total	微信小程序数据接口	微信小程序	\We::WeMiniTotal()

建议有兴趣的朋友阅读下源码 ^\_^ 以方便更好的整合开发。

# 环境准备及安装使用

## 环境准备及加载安装

WeChatDeveloper 遵循 PSR-2 命名规范和 PSR-4 自动加载规范，我们强烈推荐用户使用 composer 来管理项目扩展库。

## Composer 准备工作

如果还没有安装 Composer，你可以按 [这里](#)提到方法安装 在 Linux 和 Mac OS X 中可以运行如下命令：

```
curl -sS https://getcomposer.org/installer | php
mv composer.phar /usr/local/bin/composer
```

在 Windows 中，你需要下载并运行 [Composer-Setup.exe](#)；

如果遇到任何问题或者想更深入地学习 Composer，请参考 [Composer 文档 \(英文\)](#)，[Composer 中文](#)。

由于众所周知的原因，国外的网站连接速度很慢。因此安装的时间可能会比较长，我们建议通过下面的方式使用国内镜像。

打开命令行窗口（windows用户）或控制台（Linux、Mac 用户）并执行如下命令：

```
composer config -g repo.packagist composer https://packagist.phpcomposer.com
```

## 通过 Composer 安装 WeChatDeveloper

进入项目根目录下，通常会安装在 `vender/zoujingli/wechat-developer` 目录

```
# 首次安装 线上版本（稳定）
composer require zoujingli/wechat-developer

# 首次安装 开发版本
composer require zoujingli/wechat-developer dev-master

# 更新 WeChatDeveloper
composer update zoujingli/wechat-developer
```

## 不使用 Composer 直接加载 WeChatDeveloper

如果实在不想使用 Composer 来管理 WeChatDeveloper，也可以直接 include 进入

1. 下载 WeChatDeveloper ，并解压到项目中
2. 在项目中引入初始化文件（通常在入口的位置）

```
include "您的目录/WeChatDeveloper/include.php";
```

具体使用请参与相应文档章节。

# 接口实例及配置参数

## 接口实例及配置参数

WeChatDeveloper 所需的配置参数，需要提前准备好，方便在实例接口对接时使用。

```
$config = [
    'token'           => 'test',
    'appid'           => 'wx60a43dd8161666d4',
    'appsecret'       => '71308e96a204296c57d7cd4b21b883e8',
    'encodingaeskey' => 'BJIUzE0gqlWy0GxfPp4J1oPTBm0rNDIGPNav1YFH5Z5',
    // 配置商户支付参数（可选，在使用支付功能时需要）
    'mch_id'          => "1235704602",
    'mch_key'          => 'IKI4kpHjU94ji3oqre5zYaQMwLHuZPmj',
    // 配置商户支付双向证书目录（可选，在使用退款|打款|红包时需要）
    'ssl_key'          => '',
    'ssl_cer'          => '',
    // 缓存目录配置（可选，需拥有读写权限）
    'cache_path'       => '',
];
```

- WeChatDeveloper 最少需要配置 `appid`，`appsecret` 两个参数；在配置微信开发接口时，还需要配置 `token` 才能通过配置；如果接口配置开启了安全加密，则必需配置 `encodingaeskey`；
- 如果需要用到支付（Pay），刚还需要配置 `mch_id`，`mch_key` 两个商户参数（注意需要与 `appid` 对应），使用退款功能、企业打款到零钱、企业打款到银行卡等商户到用户的交易功能时，必需配置双向证书（这里是配置key及cert两个文件的位置目录）。

## ① 实例接口对象举例，如获取已关注用户列表

```
try {

    // 实例对应的接口对象
    $user = new \WeChat\User($config);

    // 调用接口对象方法
    $list = $user->getUserList();

    // 处理返回的结果
    echo '<pre>';
    var_export($list);

} catch (Exception $e) {

    // 出错啦，处理下吧
```

```
    echo $e->getMessage() . PHP_EOL;

}
```

## ② 实例接口对象举例，如获取已关注用户列表

```
try {

    // 实例对应的接口对象
    $user = \We::WeChatUser($config);

    // 调用接口对象方法
    $list = $user->getUserList();

    // 处理返回的结果
    echo '<pre>';
    var_export($list);

} catch (Exception $e) {

    // 出错啦，处理下吧
    echo $e->getMessage() . PHP_EOL;

}
```

# 推送事件及消息回复

## 推送事件处理及回复

- 服务端处理代码，就算没有消息回复，也要输出 success 字符串内容；
- 如果不想实时回复内容，也可以选择发送客服消息给微信端，发送多条消息或业务处理比较慢时可以选择这样做（异步发送）；
- 具体的数据格式可以参与微信官方的文件，有些接口需要传入数组，具体请查看接口参数类型。

```
try {  
  
    // 实例接口，同时实现接口配置验证与解密处理  
    $api = new \WeChat\Receive($config);  
  
    // 第二方法实例接口  
    // $api = \We::WeChatReceive($config);  
  
    // 获取当前推送接口类型 ( text,image,loction,event... )  
    $msgType = $api->getMsgType();  
  
    // 获取当前推送来源用户的openid  
    $openid = $api->getOpenid();  
  
    // 获取当前推送的所有数据  
    $data = $api->getReceive();  
  
    // 回复文本消息  
    $api->text($content)->reply();  
  
    // 回复图文消息（高级图文或普通图文，数组）  
    $api->news($news)->reply();  
  
    // 回复图片消息（需先上传到微信服务器生成 media_id）  
    $api->image($media_id)->reply();  
  
    // 回复语音消息（需先上传到微信服务器生成 media_id）  
    $api->voice($media_id)->reply();  
  
    // 回复视频消息（需先上传到微信服务器生成 media_id）  
    $api->video($media_id,$title,$desc)->reply();  
  
    // 回复音乐消息  
    $api->music($title,$desc,$musicUrl,$hgMusicUrl,$thumbe)->reply();  
  
    // 将消息转发给多客服务  
    $api->transferCustomerService($account)->reply();  
}
```

```
} catch (\Exception $e) {  
    // 处理异常  
    echo $e->getMessage();  
}
```

更多功能需要根据消息类型及配置进行处理。



# 微信客服消息管理

## 微信客服消息管理

- 微信客服消息管理
  - 创建客服账号
  - 更新客服账号
  - 删除客服账号
  - 获取所有客服账号
  - 发送客服消息
  - 客服输入状态
  - 根据标签进行群发【订阅号与服务号认证后均可用】
  - 根据OpenID列表群发【订阅号不可用，服务号认证后可用】
  - 删除群发【订阅号与服务号认证后均可用】
  - 预览接口【订阅号与服务号认证后均可用】
  - 查询群发消息发送状态【订阅号与服务号认证后均可用】
  - 获取群发速度
  - 设置群发速度

### 创建客服账号

- @param string \$kf\_account 客服账号
- @param string \$nickname 客服昵称

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Custom($config);  
  
    // 执行创建用户  
    $wechat->addAccount($account,$nickname);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

### 更新客服账号

- @param string \$kf\_account 客服账号
- @param string \$nickname 客服昵称

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Custom($config);  
  
    // 执行更新用户  
    $wechat->updateAccount($account,$nickname);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 删除客服账号

- @param string \$kf\_account 客服账号

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Custom($config);  
  
    // 执行删除用户  
    $wechat->deleteAccount($account);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 获取所有客服账号

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Custom($config);  
  
    // 执行获取客服账号  
    $list = $wechat->getAccountList();  
  
}
```

```
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 发送客服消息

---

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Custom($config);  
  
    // 执行发送客服消息  
    $wechat->send($msg);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 客服输入状态

---

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Custom($config);  
  
    // 执行客服输入状态  
    $wechat->typing($openid,$command);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 根据标签进行群发【订阅号与服务号认证后均可用】

---

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Custom($config);  
  
    // 执行
```

```
$wechat->massSendAll($data);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 根据OpenID列表群发【订阅号不可用，服务号认证后可用】

```
try {

    // 实例接口
    $wechat = new \WeChat\Custom($config);

    // 执行
    $wechat->massSend($data);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 删除群发【订阅号与服务号认证后均可用】

```
try {

    // 实例接口
    $wechat = new \WeChat\Custom($config);

    // 执行
    $wechat->massDelete($msg_id, $article_idx);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 预览接口【订阅号与服务号认证后均可用】

```
try {

    // 实例接口
    $wechat = new \WeChat\Custom($config);
```

```
// 执行
$wechat->massPreview($data);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 查询群发消息发送状态【订阅号与服务号认证后均可用】

---

```
try {

    // 实例接口
    $wechat = new \WeChat\Custom($config);

    // 执行
    $wechat->massGet($msg_id);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 获取群发速度

---

```
try {

    // 实例接口
    $wechat = new \WeChat\Custom($config);

    // 执行
    $wechat->massGetSeed();

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 设置群发速度

---

```
try {
```

```
// 实例接口
$wechat = new \WeChat\Custom($config);

// 执行
$wechat->massSetSeed($speed);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

# 微信模板消息管理

## 微信模板消息管理

- 微信模板消息管理
  - 设置所属行业
  - 获取设置的行业信息
  - 获得模板ID
  - 获取模板列表
  - 删除模板ID
  - 发送模板消息

### 设置所属行业

- @param string \$industry\_id1 公众号模板消息所属行业编号
- @param string \$industry\_id2 公众号模板消息所属行业编号

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Template($config);  
  
    // 执行操作  
    $result = $wechat->setIndustry($industry_id1, $industry_id2);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

### 获取设置的行业信息

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Template($config);  
  
    // 执行操作  
    $result = $wechat->getIndustry();  
  
}
```

```

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}

```

## 获得模板ID

- @param string \$tpl\_id 板库中模板的编号，有“TM\*\*”和“OPENTMTM\*\*”等形式

```

try {

    // 实例接口
    $wechat = new \WeChat\Template($config);

    // 执行操作
    $result = $wechat->addTemplate($tpl_id);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}

```

## 获取模板列表

```

try {

    // 实例接口
    $wechat = new \WeChat\Template($config);

    // 执行操作
    $result = $wechat->getAllPrivateTemplate();

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}

```

## 删除模板ID

- @param string \$tpl\_id 公众帐号下模板消息ID

```

try {

```



```
// 实例接口
$wechat = new \WeChat\Template($config);

// 执行操作
$result = $wechat->delPrivateTemplate($tpl_id);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 发送模板消息

---

```
try {

    // 实例接口
    $wechat = new \WeChat\Template($config);

    // 执行操作
    $result = $wechat->send($data);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

# 微信卡券支持管理

## 微信卡券支持管理

- 微信卡券支持管理
  - 创建卡券
  - 设置买单接口
  - 设置自助核销接口
  - 创建二维码接口
  - 创建货架接口
  - 导入自定义code
  - 查询导入code数目
  - 核查code接口
  - 图文消息群发卡券
  - 设置测试白名单
  - 线下核销查询Code
  - 线下核销核销Code
  - Code解码接口
  - 获取用户已领取卡券接口
  - 查看卡券详情
  - 批量查询卡券列表
  - 更改卡券信息接口
  - 修改库存接口
  - 更改Code接口
  - 删除卡券接口
  - 设置卡券失效接口
  - 拉取卡券概况数据接口
  - 获取免费券数据接口
  - 【会员卡】激活会员卡
  - 【会员卡】用户激活时需要填写的选项
  - 【会员卡】根据activate\_ticket获取到用户填写的信息
  - 【会员卡】拉取会员卡概况数据接口
  - 【会员卡】更新会员信息
  - 【会员卡】拉取会员信息（积分查询）接口

- [设置支付后投放卡券接口](#)
- [删除支付后投放卡券规则](#)
- [查询支付后投放卡券规则详情](#)
- [批量查询支付后投放卡券规则](#)
- [创建支付后领取立减金活动](#)
- [开通券点账户接口](#)
- [对优惠券批价](#)
- [查询券点余额接口](#)
- [确认兑换库存接口](#)
- [充值券点接口](#)
- [查询订单详情接口](#)
- [查询券点流水详情接口](#)

## 创建卡券

---

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Card($config);  
  
    // 执行操作  
    $result = $wechat->create($array);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 设置买单接口

---

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Card($config);  
  
    // 执行操作  
    $result = $wechat->setPaycell($card_id, $is_open);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 设置自助核销接口

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Card($config);  
  
    // 执行操作  
    $result = $wechat->setConsumeCell($card_id, $is_open);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 创建二维码接口

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Card($config);  
  
    // 执行操作  
    $result = $wechat->createQrc($data);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 创建货架接口

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Card($config);  
  
    // 执行操作  
    $result = $wechat->createLandingPage($data);  
  
} catch (Exception $e){  
    // 异常处理
```

```
    echo $e->getMessage();  
}
```

## 导入自定义code

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Card($config);  
  
    // 执行操作  
    $result = $wechat->deposit($card_id, $code);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 查询导入code数目

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Card($config);  
  
    // 执行操作  
    $result = $wechat->getDepositCount($card_id);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 核查code接口

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Card($config);  
  
    // 执行操作  
    $result = $wechat->checkCode($card_id, $code);  
}
```

```
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 图文消息群发卡券

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Card($config);  
  
    // 执行操作  
    $result = $wechat->getNewsHtml($card_id);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 设置测试白名单

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Card($config);  
  
    // 执行操作  
    $result = $wechat->setTestWhiteList($openids, $usernames);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 线下核销查询Code

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Card($config);  
  
    // 执行操作
```

```
$result = $wechat->getCode($code, $card_id , $check_consume);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 线下核销核销Code

```
try {

    // 实例接口
    $wechat = new \WeChat\Card($config);

    // 执行操作
    $result = $wechat->consume($code, $card_id);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## Code解码接口

```
try {

    // 实例接口
    $wechat = new \WeChat\Card($config);

    // 执行操作
    $result = $wechat->decrypt($encrypt_code);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 获取用户已领取卡券接口

```
try {

    // 实例接口
    $wechat = new \WeChat\Card($config);
```

```
// 执行操作
$result = $wechat->getCardList($openid, $card_id);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 查看卡券详情

---

```
try {

    // 实例接口
    $wechat = new \WeChat\Card($config);

    // 执行操作
    $result = $wechat->getCard($card_id);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 批量查询卡券列表

---

```
try {

    // 实例接口
    $wechat = new \WeChat\Card($config);

    // 执行操作
    $result = $wechat->batchGet($offset, $count, array $status_list);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 更改卡券信息接口

---

```
try {
```



```
// 实例接口
$wechat = new \WeChat\Card($config);

// 执行操作
$result = $wechat->updateCard($card_id, $member_card);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 修改库存接口

```
try {

    // 实例接口
    $wechat = new \WeChat\Card($config);

    // 执行操作
    $result = $wechat->modifyStock($card_id, $increase_stock_value, $reduce_stock_value);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 更改Code接口

```
try {

    // 实例接口
    $wechat = new \WeChat\Card($config);

    // 执行操作
    $result = $wechat->updateCode($code, $new_code, $card_id);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 删除卡券接口

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Card($config);  
  
    // 执行操作  
    $result = $wechat->deleteCard($card_id);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 设置卡券失效接口

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Card($config);  
  
    // 执行操作  
    $result = $wechat->unAvailable($code, $card_id, $reason);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 拉取卡券概况数据接口

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Card($config);  
  
    // 执行操作  
    $result = $wechat->getCardBizuininfo($begin_date, $end_date, $cond_source);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 获取免费券数据接口

```
try {

    // 实例接口
    $wechat = new \WeChat\Card($config);

    // 执行操作
    $result = $wechat->getCardCardinfo($begin_date, $end_date, $cond_source, $card_id);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 【会员卡】激活会员卡

```
try {

    // 实例接口
    $wechat = new \WeChat\Card($config);

    // 执行操作
    $result = $wechat->activateMemberCard(array $data);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 【会员卡】用户激活时需要填写的选项

```
try {

    // 实例接口
    $wechat = new \WeChat\Card($config);

    // 执行操作
    $result = $wechat->setActivateMemberCardUser(array $data);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

```
}
```

## 【会员卡】根据activate\_ticket获取到用户填写的信息

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Card($config);  
  
    // 执行操作  
    $result = $wechat->setActivateMemberCardUser(array $data);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 【会员卡】拉取会员卡概况数据接口

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Card($config);  
  
    // 执行操作  
    $result = $wechat->getActivateMemberCardTempinfo($activate_ticket);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 【会员卡】更新会员信息

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Card($config);  
  
    // 执行操作  
    $result = $wechat->updateMemberCardUser(array $data);  
  
} catch (Exception $e){  
    // 异常处理
```

```
    echo $e->getMessage();  
}
```

## 【会员卡】拉取会员信息（积分查询）接口

---

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Card($config);  
  
    // 执行操作  
    $result = $wechat->getCardMemberCard($card_id, $code);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 设置支付后投放卡券接口

---

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Card($config);  
  
    // 执行操作  
    $result = $wechat->payGiftCard($data);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 删除支付后投放卡券规则

---

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Card($config);  
  
    // 执行操作  
    $result = $wechat->delPayGiftCard($rule_id);  
}
```

```
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 查询支付后投放卡券规则详情

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Card($config);  
  
    // 执行操作  
    $result = $wechat->getPayGiftCard($rule_id);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 批量查询支付后投放卡券规则

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Card($config);  
  
    // 执行操作  
    $result = $wechat->batchGetPayGiftCard($offset, $count, $effective);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 创建支付后领取立减金活动

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Card($config);  
  
    // 执行操作
```

```
$result = $wechat->addActivity($data);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 开通券点账户接口

```
try {

    // 实例接口
    $wechat = new \WeChat\Card($config);

    // 执行操作
    $result = $wechat->payActivate();

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 对优惠券批价

```
try {

    // 实例接口
    $wechat = new \WeChat\Card($config);

    // 执行操作
    $result = $wechat->getPayprice($card_id, $quantity);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 查询券点余额接口

```
try {

    // 实例接口
    $wechat = new \WeChat\Card($config);
```

```
// 执行操作
$result = $wechat->getCoinsInfo();

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 确认兑换库存接口

---

```
try {

    // 实例接口
    $wechat = new \WeChat\Card($config);

    // 执行操作
    $result = $wechat->payConfirm($card_id, $quantity, $order_id);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 充值券点接口

---

```
try {

    // 实例接口
    $wechat = new \WeChat\Card($config);

    // 执行操作
    $result = $wechat->payRecharge($coin_count);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 查询订单详情接口

---

```
try {
```



```
// 实例接口
$wechat = new \WeChat\Card($config);

// 执行操作
$result = $wechat->payGetOrder($order_id);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 查询券点流水详情接口

---

```
try {

    // 实例接口
    $wechat = new \WeChat\Card($config);

    // 执行操作
    $result = $wechat->payGetList($data);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

# 微信自定义菜单管理

## 微信自定义菜单管理

- 微信自定义菜单管理
  - [创建微信自定义菜单](#)
  - [获取微信自定义菜单](#)
  - [删除微信自定义菜单](#)
- 微信个性化菜单管理
  - [创建个性化菜单管理](#)
  - [获取微信个性化菜单](#)
  - [删除所有微信自定义菜单](#)
  - [删除指定微信个性化菜单](#)
  - [测试微信个性化菜单](#)

### 创建微信自定义菜单

微信菜单数据 `$data` 可以参考官方文档生成，这里就不详细举例了。

```
try {  
  
    // 实例接口  
    $menu = new \WeChat\Menu($config);  
  
    // 执行创建菜单  
    $menu->create($data);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

### 获取微信自定义菜单

```
try {  
  
    // 实例接口  
    $menu = new \WeChat\Menu($config);  
  
    // 执行获取菜单
```

```
$data = $menu->get();

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 删除微信自定义菜单

```
try {

    // 实例接口
    $menu = new \WeChat\Menu($config);

    // 执行删除菜单
    $data = $menu->delete();

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

# 微信个性化菜单管理

## 创建个性化菜单管理

微信菜单数据 \$data 可以参考官方文档生成，这里就不详细举例了。

```
try {

    // 实例接口
    $menu = new \WeChat\Menu($config);

    // 执行创建菜单
    $menu->addConditional($data);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 获取微信个性化菜单

```
try {

    // 实例接口
    $menu = new \WeChat\Menu($config);

    // 执行获取菜单
    $data = $menu->get();

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 删除所有微信自定义菜单

---

```
try {

    // 实例接口
    $menu = new \WeChat\Menu($config);

    // 执行删除菜单
    $data = $menu->delete();

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 删除指定微信个性化菜单

---

```
try {

    // 实例接口
    $menu = new \WeChat\Menu($config);

    // 执行删除菜单
    $data = $menu->delConditional($menuid);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 测试微信个性化菜单

---

```
try {  
  
    // 实例接口  
    $menu = new \WeChat\Menu($config);  
  
    // 执行删除菜单  
    $data = $menu->tryConditional($openid);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

# 微信网页授权管理

## 微信网页授权管理

- 微信网页授权管理
  - [OAuth授权跳转接口](#)
  - [通过code获取AccessToken和openid](#)
  - [刷新AccessToken并续期](#)
  - [检验授权凭证\(access\\_token\)是否有效](#)
  - [拉取用户信息\(需scope为snsapi\\_userinfo\)](#)

### OAuth授权跳转接口

- @param string \$redirect\_url 授权回跳地址
- @param string \$state 为重定向后会带上state参数 ( 填写a-zA-Z0-9的参数值, 最多128字节 )
- @param string \$scope 授权类类型(可选值snsapi\_base|snsapi\_userinfo)

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\OAuth($config);  
  
    // 执行操作  
    $result = $wechat->getOAuthRedirect($redirect_url, $state, $scope);  
  
} catch (Exception $e){  
  
    // 异常处理  
    echo $e->getMessage();  
  
}
```

### 通过code获取AccessToken和openid

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\OAuth($config);  
  
    // 执行操作  
    $result = $wechat->getOAuthAccessToken();  
  
}
```

```

} catch (Exception $e){

    // 异常处理
    echo $e->getMessage();

}

```

## 刷新AccessToken并续期

```

try {

    // 实例接口
    $wechat = new \WeChat\Oauth($config);

    // 执行操作
    $result = $wechat->getOauthRefreshToken($refresh_token);

} catch (Exception $e){

    // 异常处理
    echo $e->getMessage();

}

```

## 检验授权凭证(access\_token)是否有效

- @param string \$access\_token 网页授权接口调用凭证
- @param string \$openid 用户的唯一标识

```

try {

    // 实例接口
    $wechat = new \WeChat\Oauth($config);

    // 执行操作
    $result = $wechat->checkOauthAccessToken($access_token, $openid);

} catch (Exception $e){

    // 异常处理
    echo $e->getMessage();

}

```

## 拉取用户信息(需scope为snsapi\_userinfo)

- @param string \$access\_token 网页授权接口调用凭证
- @param string \$openid 用户的唯一标识
- @param string \$lang 返回国家地区语言版本，zh\_CN 简体，zh\_TW 繁体，en 英语

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\OAuth($config);  
  
    // 执行操作  
    $result = $wechat->getUserInfo($access_token,$openid, $lang);  
  
} catch (Exception $e){  
  
    // 异常处理  
    echo $e->getMessage();  
  
}
```



# 微信网页JSSDK开发

## 微信网页JSSDK开发

- 微信网页JSSDK开发
  - 获取JSAPI\_TICKET接口
  - 删除JSAPI授权TICKET
  - 获取JsApi使用签名

### 获取JSAPI\_TICKET接口

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Script($config);  
  
    // 执行操作  
    $result = $wechat->getTicket($type, $appid);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

### 删除JSAPI授权TICKET

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Script($config);  
  
    // 执行操作  
    $result = $wechat->delTicket($type, $appid);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

### 获取JsApi使用签名

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Script($config);  
  
    // 执行操作  
    $result = $wechat->getJsSign($url, $appid , $ticket);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

# 媒体素材图文管理

## 媒体素材图文管理

- 媒体素材图文管理
  - 新增临时素材
  - 获取临时素材
  - 新增图文素材
  - 更新图文素材
  - 上传图文消息内的图片获取URL
  - 新增其他类型永久素材
  - 获取永久素材
  - 删除永久素材
  - 获取素材总数
  - 获取素材列表

### 新增临时素材

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Media($config);  
  
    // 执行操作  
    $result = $wechat->add($filename, $type);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

### 获取临时素材

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Media($config);  
  
    // 执行操作
```

```
$result = $wechat->get($media_id);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 新增图文素材

```
try {

    // 实例接口
    $wechat = new \WeChat\Media($config);

    // 执行操作
    $result = $wechat->addNews($data);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 更新图文素材

```
try {

    // 实例接口
    $wechat = new \WeChat\Media($config);

    // 执行操作
    $result = $wechat->updateNews($media_id, $index, $news);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 上传图文消息内的图片获取URL

```
try {

    // 实例接口
    $wechat = new \WeChat\Media($config);
```

```
// 执行操作
$result = $wechat->uploadImg($filename);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 新增其他类型永久素材

---

```
try {

    // 实例接口
    $wechat = new \WeChat\Media($config);

    // 执行操作
    $result = $wechat->addMaterial($filename, $type, $description);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 获取永久素材

---

```
try {

    // 实例接口
    $wechat = new \WeChat\Media($config);

    // 执行操作
    $result = $wechat->getMaterial($media_id);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 删除永久素材

---

```
try {
```

```
// 实例接口
$wechat = new \WeChat\Media($config);

// 执行操作
$result = $wechat->delMaterial($media_id);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 获取素材总数

---

```
try {

    // 实例接口
    $wechat = new \WeChat\Media($config);

    // 执行操作
    $result = $wechat->getMaterialCount();

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 获取素材列表

---

```
try {

    // 实例接口
    $wechat = new \WeChat\Media($config);

    // 执行操作
    $result = $wechat->batchGetMaterial($type, $offset, $count);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

# 微信粉丝标签管理

## 微信粉丝标签管理

- 微信粉丝标签管理
- 用户标签管理
  - 获取粉丝标签列表
  - 创建粉丝标签
  - 删除粉丝标签
  - 批量为用户打标签
  - 批量为用户取消标签
  - 获取用户身上的标签列表
- 用户粉丝管理
  - 设置用户备注名
  - 获取用户基本信息（包括UnionID机制）
  - 批量获取用户基本信息
  - 获取用户列表
  - 获取公众号的黑名单列表
  - 批量拉黑用户
  - 批量取消拉黑用户

## 用户标签管理

### 获取粉丝标签列表

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Tags($config);  
  
    // 执行操作  
    $result = $wechat->getTags();  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 创建粉丝标签

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Tags($config);  
  
    // 执行操作  
    $result = $wechat->createTags($name);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 删除粉丝标签

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Tags($config);  
  
    // 执行操作  
    $result = $wechat->deleteTags($tagId);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 批量为用户打标签

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Tags($config);  
  
    // 执行操作  
    $result = $wechat->batchTagging($openids, $tagId);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```



## 批量为用户取消标签

```
try {

    // 实例接口
    $wechat = new \WeChat\Tags($config);

    // 执行操作
    $result = $wechat->batchUntagging($openids, $tagId);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 获取用户身上的标签列表

```
try {

    // 实例接口
    $wechat = new \WeChat\Tags($config);

    // 执行操作
    $result = $wechat->getUserTagId($openid);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

# 用户粉丝管理

## 设置用户备注名

string \$openid 需要操作的粉丝用户 openid  
string \$remark 设置指定用户备注的内容

```
try {

    // 实例接口
```

```

$wechat = new \WeChat\User($config);

// 执行操作
$result = $wechat->updateMark($openid, $remark);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}

```

## 获取用户基本信息（包括UnionID机制）

string \$openid 需要操作的粉丝用户 openid

string \$lang 指定获取资料的语言类型（可选，默认 zh-CN）

```

try {

    // 实例接口
    $wechat = new \WeChat\User($config);

    // 执行操作
    $result = $wechat->getUserInfo($openid, $lang);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}

```

## 批量获取用户基本信息

string \$openid 需要操作的粉丝用户 openid

string \$lang 指定获取资料的语言类型（可选，默认 zh-CN）

```

try {

    // 实例接口
    $wechat = new \WeChat\User($config);

    // 执行操作
    $result = $wechat->getBatchUserInfo($openids, $lang);

} catch (Exception $e){
    // 异常处理

```

```
    echo $e->getMessage();  
}
```

## 获取用户列表

string \$next\_openid 获取列表下个起始位置（可选，默认为 ""）

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\User($config);  
  
    // 执行操作  
    $result = $wechat->getUserList($next_openid);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 获取公众号的黑名单列表

string \$begin\_openid 获取列表下个起始位置（可选，默认为 ""）

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\User($config);  
  
    // 执行操作  
    $result = $wechat->getBlackList($begin_openid);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 批量拉黑用户

array \$openids 需要批量拉黑的粉丝OPENID

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\User($config);  
  
    // 执行操作  
    $result = $wechat->batchBlackList($openids);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 批量取消拉黑用户

array \$openids 需要批量取消拉黑的粉丝OPENID

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\User($config);  
  
    // 执行操作  
    $result = $wechat->batchUnblackList($openids);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

# 微信蓝牙摇一摇周边

# 微信蓝牙摇一摇周边

- 微信蓝牙摇一摇周边
  - [创建客服账号](#)
  - [查询审核状态](#)
  - [申请设备ID](#)
  - [申请设备ID](#)
  - [查询设备ID申请审核状态](#)
  - [编辑设备信息](#)
  - [配置设备与门店的关联关系](#)
  - [查询设备列表](#)
  - [页面管理](#)
  - [编辑页面信息](#)
  - [查询页面列表](#)
  - [删除页面](#)
  - [上传图片素材](#)
  - [配置设备与页面的关联关系](#)
  - [查询设备与页面的关联关系](#)
  - [以设备为维度的数据统计接口](#)
  - [批量查询设备统计数据接口](#)
  - [以页面为维度的数据统计接口](#)
  - [编辑分组信息](#)
  - [删除分组](#)
  - [查询分组列表](#)
  - [查询分组详情](#)
  - [添加设备到分组](#)
  - [从分组中移除设备](#)

## 创建客服账号

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Shake($config);
```

```
// 执行操作
$wechat->register($data);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 查询审核状态

---

```
try {

    // 实例接口
    $wechat = new \WeChat\Shake($config);

    // 执行操作
    $wechat->auditStatus();

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 申请设备ID

---

```
try {

    // 实例接口
    $wechat = new \WeChat\Shake($config);

    // 执行操作
    $wechat->createApply($quantity, $apply_reason, $comment, $poi_id);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 申请设备ID

---

```
try {
```

```
// 实例接口
$wechat = new \WeChat\Shake($config);

// 执行操作
$wechat->createApply($quantity, $apply_reason, $comment, $poi_id);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 查询设备ID申请审核状态

---

```
try {

    // 实例接口
    $wechat = new \WeChat\Shake($config);

    // 执行操作
    $wechat->getApplyStatus($applyId);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 编辑设备信息

---

```
try {

    // 实例接口
    $wechat = new \WeChat\Shake($config);

    // 执行操作
    $wechat->updateApply($data);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 配置设备与门店的关联关系

---

```
try {

    // 实例接口
    $wechat = new \WeChat\Shake($config);

    // 执行操作
    $wechat->bindLocation($data);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 查询设备列表

---

```
try {

    // 实例接口
    $wechat = new \WeChat\Shake($config);

    // 执行操作
    $wechat->search($data);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 页面管理

---

```
try {

    // 实例接口
    $wechat = new \WeChat\Shake($config);

    // 执行操作
    $wechat->createPage($data);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```



## 编辑页面信息

---

```
try {

    // 实例接口
    $wechat = new \WeChat\Shake($config);

    // 执行操作
    $wechat->updatePage($data);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 查询页面列表

---

```
try {

    // 实例接口
    $wechat = new \WeChat\Shake($config);

    // 执行操作
    $wechat->searchPage($data);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 删除页面

---

```
try {

    // 实例接口
    $wechat = new \WeChat\Shake($config);

    // 执行操作
    $wechat->deletePage($data);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 上传图片素材

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Shake($config);  
  
    // 执行操作  
    $wechat->upload($filename, $type);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 配置设备与页面的关联关系

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Shake($config);  
  
    // 执行操作  
    $wechat->bindPage($data);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 查询设备与页面的关联关系

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Shake($config);  
  
    // 执行操作  
    $wechat->queryPage($data);  
  
} catch (Exception $e){  
    // 异常处理
```

```
    echo $e->getMessage();  
}
```

## 以设备为维度的数据统计接口

---

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Shake($config);  
  
    // 执行操作  
    $wechat->totalDevice($data);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 批量查询设备统计数据接口

---

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Shake($config);  
  
    // 执行操作  
    $wechat->totalDeviceList($date, $page_index);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 以页面为维度的数据统计接口

---

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Shake($config);  
  
    // 执行操作  
    $wechat->totalDeviceList($page_id, $begin_date, $end_date);  
}
```

```
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 编辑分组信息

---

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Shake($config);  
  
    // 执行操作  
    $wechat->updateGroup($group_id, $group_name);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 删除分组

---

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Shake($config);  
  
    // 执行操作  
    $wechat->deleteGroup($group_id);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 查询分组列表

---

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Shake($config);  
  
    // 执行操作
```

```
$wechat->getGroupList($begin, $count);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 查询分组详情

---

```
try {

    // 实例接口
    $wechat = new \WeChat\Shake($config);

    // 执行操作
    $wechat->getGroupDetail($group_id, $begin, $count );

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 添加设备到分组

---

```
try {

    // 实例接口
    $wechat = new \WeChat\Shake($config);

    // 执行操作
    $wechat->addDeviceGroup($data);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 从分组中移除设备

---

```
try {

    // 实例接口
    $wechat = new \WeChat\Shake($config);
```

```
// 执行操作
$wechat->deleteDeviceGroup($data);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

# 生成带参数的二维码

## 生成带参数的二维码

- [生成带参数的二维码](#)
  - [创建二维码ticket](#)
  - [通过ticket换取二维码](#)
  - [长链接转短链接接口](#)

### 创建二维码ticket

- @param string|integer \$scene 场景值（数字类型与字符串类型自动转换）
- @param int \$expire\_seconds 有效时间（可选，有值时为临时二维码）

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Qrcode($config);  
  
    // 执行操作  
    $result = $wechat->create($scene, $expire_seconds);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

### 通过ticket换取二维码

- @param string \$ticket 获取的二维码ticket

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Qrcode($config);  
  
    // 执行操作  
    $result = $wechat->url($ticket);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

```
}
```

## 长链接转短链接接口

---

- @param string \$longUrl 需要转换的长链接

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Qrcode($config);  
  
    // 执行操作  
    $result = $wechat->shortUrl($longUrl);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```



# 微信扫一扫管理

## 微信扫一扫管理

- 微信扫一扫管理
  - 获取商户信息
  - 创建商品
  - 商品发布
  - 设置测试人员白名单
  - 获取商品二维码
  - 查询商品信息
  - 批量查询商品信息
  - 更新商品信息
  - 清除商品信息
  - 检查wxticket参数
  - 清除扫码记录

### 获取商户信息

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Scan($config);  
  
    // 执行操作  
    $wechat->getMerchantInfo();  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

### 创建商品

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Scan($config);  
  
    // 执行操作
```

```
$wechat->addProduct($data);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 商品发布

```
try {

    // 实例接口
    $wechat = new \WeChat\Scan($config);

    // 执行操作
    $wechat->modProduct($keystandard, $keystr, $status);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 设置测试人员白名单

```
try {

    // 实例接口
    $wechat = new \WeChat\Scan($config);

    // 执行操作
    $wechat->setTestWhiteList($openids, $usernames);

} catch (Exception $e){
    // 异常处理
    echo $e->getMessage();
}
```

## 获取商品二维码

```
try {

    // 实例接口
    $wechat = new \WeChat\Scan($config);

    // 执行操作
    $wechat->getQrc($keystandard, $keystr, $extinfo , $qrcode_size);

}
```

```
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 查询商品信息

---

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Scan($config);  
  
    // 执行操作  
    $wechat->getProductInfo($keystandard, $keystr);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 批量查询商品信息

---

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Scan($config);  
  
    // 执行操作  
    $wechat->getProductList($offset , $limit , $status , $keystr);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 更新商品信息

---

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Scan($config);  
  
    // 执行操作  
    $wechat->updateProduct($data);  
}
```

```
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 清除商品信息

---

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Scan($config);  
  
    // 执行操作  
    $wechat->clearProduct($keystandard, $keyst);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 检查wxticket参数

---

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Scan($config);  
  
    // 执行操作  
    $wechat->checkTicket($ticket);  
  
} catch (Exception $e){  
    // 异常处理  
    echo $e->getMessage();  
}
```

## 清除扫码记录

---

```
try {  
  
    // 实例接口  
    $wechat = new \WeChat\Scan($config);  
  
    // 执行操作  
    $wechat->clearScanTicket($keystandard, $keyst, $extinfo);  
  
} catch (Exception $e){
```

```
// 异常处理
echo $e->getMessage();
}
```

# 微信服务号和小程序支付

---

## 公众号支付

---

- [统一下单](#)
- [查询订单](#)
- [关闭订单](#)
- [申请退款](#)
- [查询退款](#)
- [支付通知](#)

# 统一下单

## 统一下单

发起统一下单申请，获取支付订单凭证，  
统一使用签名方式为 MD5，发起支付时注意对应好。

更多参数请参阅：[https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=9\\_1](https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=9_1)

```
try {  
  
    // 创建接口实例  
    $wechat = new \WeChat\Pay($config);  
  
    // 组装参数，可以参考官方商户文档  
    $options = [  
        'body' => '测试商品',  
        'out_trade_no' => time(),  
        'total_fee' => '1',  
        'openid' => 'o38gpszoJoC9oJYz3UHHf6bEp0Lo',  
        'trade_type' => 'JSAPI',  
        'notify_url' => 'http://a.com/text.html',  
        'spbill_create_ip' => '127.0.0.1',  
    ];  
  
    // 尝试创建订单  
    $result = $wechat->createOrder($options);  
  
    // 订单数据处理  
    var_export($result);  
  
} catch (Exception $e) {  
  
    // 出错啦，处理下吧  
    echo $e->getMessage() . PHP_EOL;  
  
}
```

### 返回结果（已转为数组）

```
array (  
    'return_code' => 'SUCCESS',  
    'return_msg' => 'OK',  
)
```

```
'appid' => 'wx60a43dd8161666d4',
'mch_id' => '1332187001',
'nonce_str' => 'pYpUew3a5UsPoo0m',
'sign' => '920B9F6A3DBEBBF4BE20FAC6E9AFC47E8396042E0A10F8CAAEBAE150F6F37EA2',
'result_code' => 'SUCCESS',
'prepay_id' => 'wx20180202200449ea27133ab70182105407',
'trade_type' => 'JSAPI',
);
```

错误码

名称	描述	原因	解决方案
NOAUTH	商户无此接口权限	商户未开通此接口权限	请商户前往申请此接口权限
NOTENOUGH	余额不足	用户帐号余额不足	用户帐号余额不足，请用户充值或更换支付卡后再支付
ORDERPAID	商户订单已支付	商户订单已支付，无需重复操作	商户订单已支付，无需更多操作
ORDERCLOSED	订单已关闭	当前订单已关闭，无法支付	当前订单已关闭，请重新下单
SYSTEMERROR	系统错误	系统超时	系统异常，请用相同参数重新调用
APPID_NOT_EXIST	APPID不存在	参数中缺少APPID	请检查APPID是否正确
MCHID_NOT_EXIST	MCHID不存在	参数中缺少MCHID	请检查MCHID是否正确
APPID_MCHID_NOT_MATCH	appid和mch_id不匹配	appid和mch_id不匹配	请确认appid和mch_id是否匹配
LACK_PARAMS	缺少参数	缺少必要的请求参数	请检查参数是否齐全
OUT_TRADE_NO_USED	商户订单号重复	同一笔交易不能多次提交	请核实商户订单号是否重复提交
SIGNERROR	签名错误	参数签名结果不正确	请检查签名参数和方法是否都符合签名算法要求
XML_FORMAT_ERROR	XML格式错误	XML格式错误	请检查XML参数格式是否正确
REQUIRE_POST_	请使用post方法	未使用post传递	请检查请求参数是否通过post方法



METHOD	请使用post方法	参数	提交
POST_DATA_EMPTY	post数据为空	post数据不能为空	请检查post数据是否为空
NOT_UTF8	编码格式错误	未使用指定编码格式	请使用UTF-8编码格式

# 客户端发起支付（新）

## 客户端发起支付（新版本 v1.1.2 开始支持）

- prepay\_id 由统一下单获取到的预支付码

```
// 创建接口实例
$wechat = new \WeChat\Pay($config);

// 创建微信端发起支付参数及签名
$options = $wechat->createParamsForJsApi($prepay_id);

// 微信端发起支付参数及签名JSON化
$params = json_encode($options);
```

- 基于微信 JSAPI 的 js 代码

```
wx.ready(function () {

    // 这里获取到PHP生成签名参数包，注意是JSON格式
    var options = <{$params}>;

    // 支付成功后的操作
    options.success = function () {
        window.location.href = "支付成功后的跳转地址，通常是订单列表";
    };

    // 取消支付的操作
    options.cancel = function () {

    };

    // 支付失败的处理
    options.fail = function () {

    };

    // 传入参数，发起JSAPI支付
    wx.chooseWXPay(options);
});
```

# 查询订单

## 查询订单

查询订单 transaction\_id 与 out\_trade\_no 参数二选一

更多参数请参阅：[https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=9\\_2](https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=9_2)

```
try {  
  
    // 创建接口实例  
    $wechat = new \WeChat\Pay($config);  
  
    // 组装参数，可以参考官方商户文档  
    $options = [  
        'transaction_id' => '1008450740201411110005820873',  
        // 'out_trade_no'   => '商户订单号',  
    ];  
  
    // 尝试创建订单  
    $result = $wechat->queryOrder($options);  
  
    // 订单数据处理  
    var_export($result);  
  
} catch(Exception $e) {  
  
    // 出错啦，处理下吧  
    echo $e->getMessage() . PHP_EOL;  
  
}
```

### 返回结果（已转为数组）

```
<xml>  
  <return_code><![CDATA[SUCCESS]]></return_code>  
  <return_msg><![CDATA[OK]]></return_msg>  
  <appid><![CDATA[wX2421b1c4370ec43b]]></appid>  
  <mch_id><![CDATA[10000100]]></mch_id>  
  <device_info><![CDATA[1000]]></device_info>  
  <nonce_str><![CDATA[TN55w09Pba5yEN18]]></nonce_str>  
  <sign><![CDATA[BDF0099C15FF7BC6B1585FBB110AB635]]></sign>  
  <result_code><![CDATA[SUCCESS]]></result_code>  
  <openid><![CDATA[oUpF8uN95-Ptaags6E_roPHg7AG0]]></openid>
```

```
<is_subscribe><![CDATA[Y]]></is_subscribe>
<trade_type><![CDATA[MICROPAY]]></trade_type>
<bank_type><![CDATA[CCB_DEBIT]]></bank_type>
<total_fee>1</total_fee>
<fee_type><![CDATA[CNY]]></fee_type>
<transaction_id><![CDATA[1008450740201411110005820873]]></transaction_id>
<out_trade_no><![CDATA[1415757673]]></out_trade_no>
<attach><![CDATA[订单额外描述]]></attach>
<time_end><![CDATA[20141111170043]]></time_end>
<trade_state><![CDATA[SUCCESS]]></trade_state>
</xml>
```

错误码

名称	描述	原因	解决方案
ORDERNOTEXIST	此交易订单号不存在	查询系统中不存在此交易订单号	该API只能查提交支付交易返回成功的订单，请商户检查需要查询的订单号是否正确
SYSTEMERROR	系统错误	后台系统返回错误	系统异常，请再调用发起查询

# 关闭订单

## 关闭订单

更多参数请参阅：[https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=9\\_3](https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=9_3)

```
try {

    // 创建接口实例
    $wechat = new \WeChat\Pay($config);

    // 组装参数，可以参考官方商户文档
    $options => '1217752501201407033233368018';

    // 尝试创建订单
    $result = $wechat->closeOrder($options);

    // 订单数据处理
    var_export($result);

} catch(Exception $e) {

    // 出错啦，处理下吧
    echo $e->getMessage() . PHP_EOL;

}
```

### 返回结果（已转为数组）

```
<xml>
  <return_code><![CDATA[SUCCESS]]></return_code>
  <return_msg><![CDATA[OK]]></return_msg>
  <appid><![CDATA[wx2421b1c4370ec43b]]></appid>
  <mch_id><![CDATA[10000100]]></mch_id>
  <nonce_str><![CDATA[BFK89FC6rxKC0jLX]]></nonce_str>
  <sign><![CDATA[72B321D92A7BFA0B2509F3D13C7B1631]]></sign>
  <result_code><![CDATA[SUCCESS]]></result_code>
  <result_msg><![CDATA[OK]]></result_msg>
</xml>
```

### 错误码

名称	描述	原因	解决方案
----	----	----	------

ORDERPAID	订单已支付	订单已支付，不能发起关单	订单已支付，不能发起关单，请当作已支付的正常交易
SYSTEMERROR	系统错误	系统错误	系统异常，请重新调用该API
ORDERCLOSED	订单已关闭	订单已关闭，无法重复关闭	订单已关闭，无需继续调用
SIGNERROR	签名错误	参数签名结果不正确	请检查签名参数和方法是否都符合签名算法要求
REQUIRE_POST_METHOD	请使用post方法	未使用post传递参数	请检查请求参数是否通过post方法提交
XML_FORMAT_ERROR	XML格式错误	XML格式错误	请检查XML参数格式是否正确

# 申请退款

## 申请退款（需要双向证书）

更多参数请参阅：[https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=9\\_4](https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=9_4)

```
try {  
  
    // 创建接口实例  
    $wechat = new \WeChat\Pay($config);  
  
    // 组装参数，可以参考官方商户文档  
    $options = [  
        'transaction_id' => '1008450740201411110005820873',  
        'out_refund_no'   => '商户退款单号',  
        'total_fee'       => '1',  
        'refund_fee'      => '1',  
    ];  
  
    // 尝试创建订单  
    $result = $wechat->createRefund($options);  
  
    // 订单数据处理  
    var_export($result);  
  
} catch(Exception $e) {  
  
    // 出错啦，处理下吧  
    echo $e->getMessage() . PHP_EOL;  
  
}
```

## 返回结果（已转为数组）

```
<xml>  
  <appid>wx2421b1c4370ec43b</appid>  
  <mch_id>10000100</mch_id>  
  <nonce_str>6cefdb308e1e2e8aabd48cf79e546a02</nonce_str>  
  <out_refund_no>1415701182</out_refund_no>  
  <out_trade_no>1415757673</out_trade_no>  
  <refund_fee>1</refund_fee>  
  <total_fee>1</total_fee>  
  <transaction_id></transaction_id>  
  <sign>FE56DD4AA85C0EECA82C35595A69E153</sign>
```

```
</xml>
```

错误码

名称	描述	原因	解决方案
SYSTEMERROR	接口返回错误	系统超时等	请不要更换商户退款单号，请使用相同参数再次调用API。
BIZERR_NEED_RETRY	退款业务流程错误，需要商户触发重试来解决	并发情况下，业务被拒绝，商户重试即可解决	请不要更
TRADE_OVERDUE	订单已经超过退款期限	订单已经超过可退款的最大期限(支付后一年内可退款)	请选择其他方式自行退款
ERROR	业务错误	申请退款业务发生错误	该错误都会返回具体的错误原因，请根据实际返回做相应处理。
USER_ACCOUNT_ABNORMAL	退款请求失败	用户帐号注销	此状态代表退款申请失败，商户可自行处理退款。
INVALID_REQ_TOO_MUCH	无效请求过多	连续错误请求数过多被系统短暂屏蔽	请检查业务是否正常，确认业务正常后请在1分钟后再来重试
NOTENOUGH	余额不足	商户可用退款余额不足	此状态代表退款申请失败，商户可根据具体的错误提示做相应的处理。
INVALID_TRANSACTIONID	无效transaction_id	请求参数未按指引进行填写	请求参数错误，检查原交易号是否存在或发起支付交易接口返回失败
PARAM_ERROR	参数错误	请求参数未按指引进行填写	请求参数错误，请重新检查再调用退款申请
APPID_NOT_EXIST	APPID不存在	参数中缺少APPID	请检查APPID是否正确



MCHID_NOT_EXIST	MCHID不存在	参数中缺少MCHID	请检查MCHID是否正确
REQUIRE_POST_METHOD	请使用post方法	未使用post传递参数	请检查请求参数是否通过post方法提交
SIGNERROR	签名错误	参数签名结果不正确	请检查签名参数和方法是否都符合签名算法要求
XML_FORMAT_ERROR	XML格式错误	XML格式错误	请检查XML参数格式是否正确
FREQUENCY_LIMITED	频率限制	2个月之前的订单申请退款有频率限制	该笔退款未受理，请降低频率后重试

# 查询退款

## 查询退款

更多参数请参阅：[https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=9\\_5](https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=9_5)

```
try {  
  
    // 创建接口实例  
    $wechat = new \WeChat\Pay($config);  
  
    // 组装参数，可以参考官方商户文档  
    $options = [  
        'transaction_id' => '1008450740201411110005820873',  
        // 'out_trade_no'   => '商户订单号',  
        // 'out_refund_no'  => '商户退款单号'  
        // 'refund_id'     => '微信退款单号',  
    ];  
  
    // 尝试创建订单  
    $result = $wechat->queryRefund($options);  
  
    // 订单数据处理  
    var_export($result);  
  
} catch(Exception $e) {  
  
    // 出错啦，处理下吧  
    echo $e->getMessage() . PHP_EOL;  
  
}
```

### 返回结果（已转为数组）

```
<xml>  
  <return_code><![CDATA[SUCCESS]]></return_code>  
  <return_msg><![CDATA[OK]]></return_msg>  
  <appid><![CDATA[wx2421b1c4370ec43b]]></appid>  
  <mch_id><![CDATA[10000100]]></mch_id>  
  <nonce_str><![CDATA[NfsMFbUFpdbEhPXP]]></nonce_str>  
  <sign><![CDATA[B7274EB9F8925EB93100DD2085FA56C0]]></sign>  
  <result_code><![CDATA[SUCCESS]]></result_code>  
  <transaction_id><![CDATA[1008450740201411110005820873]]></transaction_id>  
  <out_trade_no><![CDATA[1415757673]]></out_trade_no>  
  <out_refund_no><![CDATA[1415701182]]></out_refund_no>
```

```
<refund_id><![CDATA[2008450740201411110000174436]]></refund_id>
<refund_channel><![CDATA[]]></refund_channel>
<refund_fee>1</refund_fee>
</xml>
```

错误码

名称	描述	原因	解决方案
SYSTEMERROR	接口返回错误	系统超时	请尝试再次掉调用API。
REFUNDNOTEXIST	退款订单查询失败	订单号错误或订单状态不正确	请检查订单号是否有误以及订单状态是否正确，如：未支付、已支付未退款
INVALID_TRANSACTIONID	无效transaction_id	请求参数未按指引进行填写	请求参数错误，检查原交易号是否存在或发起支付交易接口返回失败
PARAM_ERROR	参数错误	请求参数未按指引进行填写	请求参数错误，请检查参数再调用退款申请
APPID_NOT_EXIST	APPID不存在	参数中缺少APPID	请检查APPID是否正确
MCHID_NOT_EXIST	MCHID不存在	参数中缺少MCHID	请检查MCHID是否正确
REQUIRE_POST_METHOD	请使用post方法	未使用post传递参数	请检查请求参数是否通过post方法提交
SIGNERROR	签名错误	参数签名结果不正确	请检查签名参数和方法是否都符合签名算法要求
XML_FORMAT_ERROR	XML格式错误	XML格式错误	请检查XML参数格式是否正确

# 支付通知

## 支付结果通知

更多参数请参阅：[https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=9\\_7](https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=9_7)

```
try {  
  
    // 创建接口实例  
    $wechat = new \WeChat\Pay($config);  
  
    // 尝试创建订单  
    $result = $wechat->getNotify();  
  
    // 订单数据处理  
    var_export($result);  
  
} catch(Exception $e) {  
  
    // 出错啦，处理下吧  
    echo $e->getMessage() . PHP_EOL;  
  
}
```

### 返回结果（已转为数组）

```
<xml>  
  <appid><![CDATA[wx2421b1c4370ec43b]]></appid>  
  <attach><![CDATA[支付测试]]></attach>  
  <bank_type><![CDATA[CFT]]></bank_type>  
  <fee_type><![CDATA[CNY]]></fee_type>  
  <is_subscribe><![CDATA[Y]]></is_subscribe>  
  <mch_id><![CDATA[10000100]]></mch_id>  
  <nonce_str><![CDATA[5d2b6c2a8db53831f7eda20af46e531c]]></nonce_str>  
  <openid><![CDATA[oUpF8uMEb4qRXf22hE3X68TekukE]]></openid>  
  <out_trade_no><![CDATA[1409811653]]></out_trade_no>  
  <result_code><![CDATA[SUCCESS]]></result_code>  
  <return_code><![CDATA[SUCCESS]]></return_code>  
  <sign><![CDATA[B552ED6B279343CB493C5DD0D78AB241]]></sign>  
  <sub_mch_id><![CDATA[10000100]]></sub_mch_id>  
  <time_end><![CDATA[20140903131540]]></time_end>  
  <total_fee>1</total_fee>  
  <coupon_fee><![CDATA[10]]></coupon_fee>  
  <coupon_count><![CDATA[1]]></coupon_count>  
  <coupon_type><![CDATA[CASH]]></coupon_type>
```

```
<coupon_id><![CDATA[10000]]></coupon_id>  
<coupon_fee><![CDATA[100]]></coupon_fee>  
<trade_type><![CDATA[JSAPI]]></trade_type>  
<transaction_id><![CDATA[1004400740201409030005092168]]></transaction_id>  
</xml>
```

# 微信小程序服务接口支持（新）

## 微信小程序

- [encryptedData解码](#)
- [小程序模板消息](#)
- [微信小程序二维码](#)
- [小程序数据统计](#)
- [小程序地址位置](#)

小程序支付与服务号支付相近，服务端统一下单，返回参数给小程序就可以了.

如果还有遗落的接口，可以加入QQ群或在git上提交issue

# encryptedData解码

## encryptedData解码

直接解码数据，获取用户资料

- 通过小程序客户端API获取需要的数据

```
wx.login({
  success: res => {
    if (res.code) {
      wx.getUserInfo({
        lang: 'zh_CN',
        withCredentials: true,
        success: ret => {
          ret.code = res.code;
          console.log(ret);
        }
      })
    }
  }
})
```

从上面可以得到以下数据

```
{
  "errMsg": "getUserInfo:ok",
  "rawData": "{\\"nickName\\":\\"Anyon\\",\\"gender\\":1,\\"language\\":\\"zh_CN\\",\\"city\\":\\"广州\\",\\"province\\":\\"广东\\",\\"country\\":\\"中国\\",\\"avatarUrl\\":\\"https://wx.qlogo.cn/mmopen/vi_32/JSmwB5ERvyXDUGib2yIcPwFuZicQWDpB3cnRayRKew66zZTGStbyhjoj2DM0nuxJAsY7BMz5nXjRne1MImHnrU2A/0\\"}",
  "userInfo": {
    "nickName": "Anyon",
    "gender": 1,
    "language": "zh_CN",
    "city": "广州",
    "province": "广东",
    "country": "中国",
    "avatarUrl": "https://wx.qlogo.cn/mmopen/vi_32/JSmwB5ERvyXDUGib2yIcPwFuZicQWDpB3cnRayRKew66zZTGStbyhjoj2DM0nuxJAsY7BMz5nXjRne1MImHnrU2A/0"
  },
  "signature": "da1ec007c47956d266348fc8cef0ed068115d50d",
  "encryptedData": "wNni4RgfFYXHHwe2oTqkGQ/hQHbeqK0cq65o0db9fWqmdARqphmXUcZZG94XiEuVWNLl0vxJlJK+hoWwewmzcpA0PH3YD1Sag1qtIbUfst27anT5nDdn0S0ZBEEI3SafvZ07WkEPaV3/3JNVmp2lTNYZlAFCXl+s7hbCse3NagXGhs57SCfb+CTg3yNFzN5NgV+30RY0woHP4M4A55nooZ0WCeQuGX4LEhNcQ+wL9Kg7TISPD1mxdCsAzzFTxZZLSSyfv/qNB+QYlD1ApFunZ7RBkR01uJQnYwSA9SH"
```

```
YIy6yreGioXbZBSN+lHshje1JYpG/Q3zdtf6SU0dh7mbCjCsguEzEy1lIKxsRkLQh9n/RFws7KckAmS
v04Ai7/8nTUtSMgjJmzquaAllT+aU1Fn3TSMEV+Fy7iGnfWcqLOS14Gk0DKXqTa/iyKxntwJyR9Jjk
itwjy8q4WvfIeCotA==",
    "iv": "R5vsQykg1idxADhU3+XmWw==",
    "code": "003YiWlt0YIgfB18ubMt04G5Mt0YiWLe"
}
```

把其中的 `code` , `iv` , `encryptedData` 传递到SDK进行解码

```
// 引入自动加载
include '../include.php';

// 小程序配置
$config = [
    'appid'      => 'wx6bb7b70258da09c6',
    'appsecret' => '78b7b8d65bd67b078babf951d4342b42',
];

// 实例SDK
$mini = new WeMini\Crypt($config);

// 数据解码
try {
    $data = $mini->decode($iv, $sessionKey, $encryptedData);
    print_r($data);
} catch (Exception $e) {
    print_r($e);
}
```

## 二、当然你也可以按步来操作

- 1. 通过 wx.login 获取来 code , 在服务器换取 session\_key

```
// 引入自动加载
include '../include.php';

// 小程序配置
$config = [
    'appid'      => 'wx6bb7b70258da09c6',
    'appsecret' => '78b7b8d65bd67b078babf951d4342b42',
];

// 实例SDK
$mini = new WeMini\Crypt($config);
try {
    $data = $mini->session($iv, $sessionKey, $encryptedData);
}
```



```
    print_r($data);  
} catch (Exception $e) {  
  
}
```

- 2. 通过 wx.getUserInfo 获取 encryptedData 及 iv , 在服务器进行解码

```
// 实例SDK  
$mini = new WeMini\Crypt($config);  
try {  
    $data = $mini->decode($iv, $sessionKey, $encryptedData);  
    print_r($data);  
} catch (Exception $e) {  
  
}
```

# 小程序模板消息

## 小程序模板消息

- [小程序模板消息](#)
- [获取小程序模板库标题列表](#)
- [获取模板库某个模板标题下关键词库](#)
- [组合模板并添加至帐号下的个人模板库](#)
- [获取帐号下已存在的模板列表](#)
- [删除模板消息](#)
- [发送模板消息](#)

## 获取小程序模板库标题列表

```
// 引入自动加载
include '../include.php';

// 小程序配置
$config = [
    'appid'      => 'wx6bb7b70258da09c6',
    'appsecret' => '78b7b8d65bd67b078babf951d4342b42',
];

// 实例SDK
$mini = new WeMini\Template($config);

try {
    $data = $mini->getTemplateLibraryList();
    print_r($data);
} catch (Exception $e) {
    print_r($e);
}
```

## 获取模板库某个模板标题下关键词库

- \$template\_id 模板标题id，可通过接口获取，也可登录小程序后台查看获取

```
// 引入自动加载
include '../include.php';

// 小程序配置
$config = [
```

```

        'appid'      => 'wx6bb7b70258da09c6',
        'appsecret' => '78b7b8d65bd67b078babf951d4342b42',
    ];

    // 实例SDK
    $mini = new WeMini\Template($config);

    try {
        $data = $mini->getTemplateLibrary($template_id);
        print_r($data);
    } catch (Exception $e) {
        print_r($e);
    }

```

## 组合模板并添加至帐号下的个人模板库

- \$template\_id 模板标题id，可通过接口获取，也可登录小程序后台查看获取
- \$keyword\_id\_list 开发者自行组合好的模板关键词列表，关键词顺序可以自由搭配（例如[3,5,4]或[4,5,3]），最多支持10个关键词组合

```

// 引入自动加载
include '../include.php';

// 小程序配置
$config = [
    'appid'      => 'wx6bb7b70258da09c6',
    'appsecret' => '78b7b8d65bd67b078babf951d4342b42',
];

// 实例SDK
$mini = new WeMini\Template($config);

try {
    $data = $mini->addTemplate($template_id,$keyword_id_list);
    print_r($data);
} catch (Exception $e) {
    print_r($e);
}

```

## 获取帐号下已存在的模板列表

```

// 引入自动加载
include '../include.php';

```

```
// 小程序配置
$config = [
    'appid'      => 'wx6bb7b70258da09c6',
    'appsecret' => '78b7b8d65bd67b078babf951d4342b42',
];

// 实例SDK
$mini = new WeMini\Template($config);

try {
    $data = $mini->getTemplateList();
    print_r($data);
} catch (Exception $e) {
    print_r($e);
}
```

## 删除模板消息

- \$template\_id 要删除的模板id

```
// 引入自动加载
include '../include.php';

// 小程序配置
$config = [
    'appid'      => 'wx6bb7b70258da09c6',
    'appsecret' => '78b7b8d65bd67b078babf951d4342b42',
];

// 实例SDK
$mini = new WeMini\Template($config);

try {
    $data = $mini->delTemplate($template_id);
    print_r($data);
} catch (Exception $e) {
    print_r($e);
}
```

## 发送模板消息

- \$data 参与官方文档，需要传数组

```
// 引入自动加载
include '../include.php';
```

```
// 小程序配置
$config = [
    'appid'      => 'wx6bb7b70258da09c6',
    'appsecret' => '78b7b8d65bd67b078babf951d4342b42',
];

// 实例SDK
$mini = new WeMini\Template($config);

try {
    $data = $mini->send($data);
    print_r($data);
} catch (Exception $e) {
    print_r($e);
}
```

# 微信小程序二维码

## 微信小程序二维码

- [微信小程序二维码](#)
- [获取小程序码（永久有效）](#)
- [获取小程序码（永久有效）](#)
- [获取小程序码（永久有效）](#)

## 获取小程序码（永久有效）

- 接口A: 适用于需要的码数量较少的业务场景
- \$path 不能为空，最大长度 128 字节
- \$width 二维码的宽度
- \$auto\_color 自动配置线条颜色，如果颜色依然是黑色，则说明不建议配置主色调
- \$line\_color 为 false 时生效，默认：["r" => "0", "g" => "0", "b" => "0"]

```
// 引入自动加载
include '../include.php';

// 小程序配置
$config = [
    'appid'      => 'wx6bb7b70258da09c6',
    'appsecret' => '78b7b8d65bd67b078babf951d4342b42',
];

// 实例SDK
$mini = new WeMini\Qrcode($config);

try {
    $data = $mini->createMiniPath($path, $width, $auto_color, $line_color);
    print_r($data);
} catch (Exception $e) {
    print_r($e);
}
```

## 获取小程序码（永久有效）

- 接口B：适用于需要的码数量极多的业务场景
- \$scene 最大32个可见字符，只支持数字
- \$path 不能为空，最大长度 128 字节

- \$width 二维码的宽度
- \$auto\_color 自动配置线条颜色，如果颜色依然是黑色，则说明不建议配置主色调
- \$line\_color 为 false 时生效，默认：["r" => "0", "g" => "0", "b" => "0"]

```
// 引入自动加载
include '../include.php';

// 小程序配置
$config = [
    'appid'      => 'wx6bb7b70258da09c6',
    'appsecret' => '78b7b8d65bd67b078babf951d4342b42',
];

// 实例SDK
$mini = new WeMini\Qrcode($config);

try {
    $data = $mini->createMiniScene($scene, $path, $width, $auto_color, $line_color);
    print_r($data);
} catch (Exception $e) {
    print_r($e);
}
```

## 获取小程序码（永久有效）

- 接口C：适用于需要的码数量较少的业务场景
- \$path 不能为空，最大长度 128 字节
- \$width 二维码的宽度

```
// 引入自动加载
include '../include.php';

// 小程序配置
$config = [
    'appid'      => 'wx6bb7b70258da09c6',
    'appsecret' => '78b7b8d65bd67b078babf951d4342b42',
];

// 实例SDK
$mini = new WeMini\Qrcode($config);

try {
    $data = $mini->createDefault($path, $width);
    print_r($data);
}
```

```
} catch (Exception $e) {  
    print_r($e);  
}
```



# 小程序数据统计

---

看源码吧，不常用。

源码文件：`WeMini/Total.php`

如果还有遗落的接口，可以加入QQ群或在git上提交issue

# 小程序地址位置

---

看源码吧，不常用。

源码文件：`WeMini/Poi.php`

如果还有遗落的接口，可以加入QQ群或在git上提交issue

# 小程序插件管理

---

看源码吧，不常用。

源码文件：`WeMini/Plugs.php`

如果还有遗落的接口，可以加入QQ群或在git上提交issue