



微信 APP 支付接口文档 V1.6

APP 支付接口文档 V1.6



版本说明

版本	更改说明	更新时间
V1.6	更新实例代码	2014-05-03



1.微信支付简介	
1.1 功能简介	4
1.2 平台帐号注册	5
1.2.1 申请流程指引图	5
1.2.2 申请接入步骤详细说明	5
1.3 支付账户	6
2.微信 APP 支付接入	8
2.1 系统架构	8
2.2 基本交互	8
2.3 支付功能开发介绍	9
3.接口介绍	10
3.1 获取 access_token	10
3.2 生成预支付订单	11
3.3 订单详情 (package) 扩展字符串定义	12
3.4 支付签名 (app_signature) 生成方法	14
3.5 添加 prepayid 再次签名	15
3.6APP 端开发说明	16
3.6.1 初始化设置	16
3.6.2 注册您的 APPID	16
3.6.3 用服务器返回的参数调起支付	17
4.通知接口说明	18
4.1 通知接口简介	18

4.2 补单机制.....	18
4.3 通知接口参数.....	18
4.4 后台通知结果返回.....	21
4.5 后台通知签名方式.....	21
5.API 接口说明.....	23
5.1API 接口简介.....	23
5.2API 列表.....	23
5.2.1 发货通知 delivernotify.....	23
5.2.2 订单查询 orderquery.....	25
6. 常见问题和注意事项.....	27
6.1 帮助 SDK.....	27
6.2 常见基本概念疑惑.....	28
6.3 常见错误现象及解决方法.....	28
6.4 常见注意事项.....	29
6.5 最新接口文档下载.....	29
6.6 联系我们.....	30

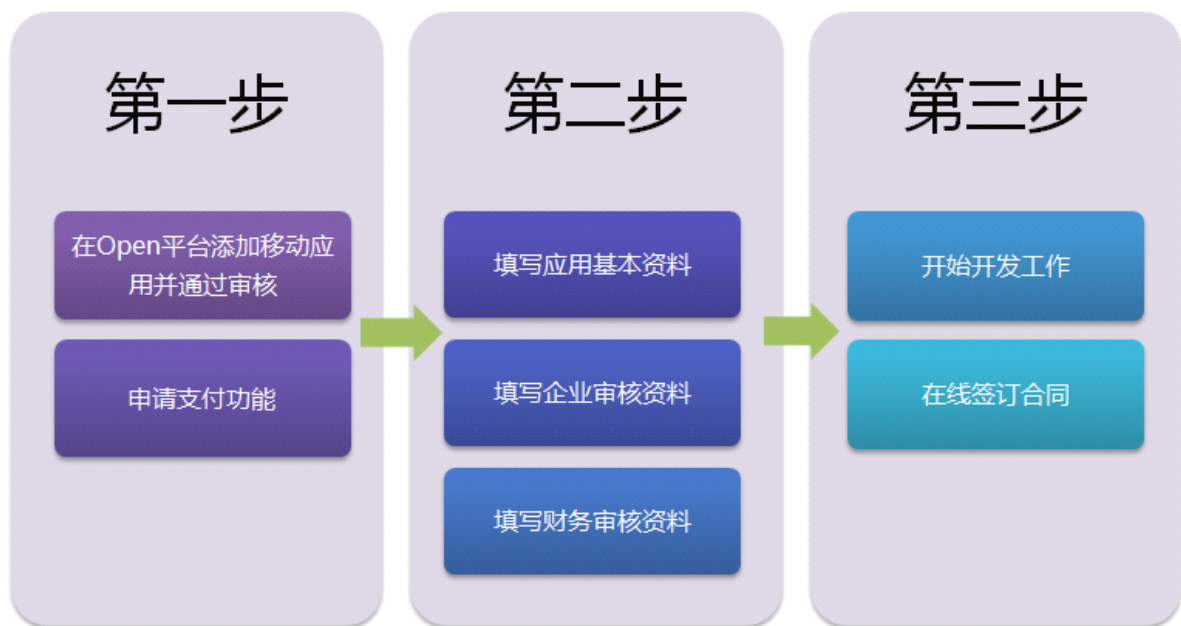
1.微信支付简介

1.1 功能简介

微信 APP 支付，是基于微信客户端提供的支付服务功能。在第三方 APP 集成微信支付模块后即可完成商户购买支付流程。本文将全面介绍微信 APP 支付的技术方案。

1.2 平台帐号注册

1.2.1 申请流程指引图



1.2.2 申请接入步骤详细说明

a. 目前仅对通过微信开放平台移动应用审核的 APP 开放微信支付申请入口，请登录微信开放平台，添加移动应用，等待审核。

b. 在微信开放平台-管理中心，为通过审核的移动应用申请支付功能，提交相关资料；



c.提交相关的资料后，等待审核，微信团队会在 7 个工作日给出审核结果，结果会发送到开放平台通知中心和邮箱中；邮箱是在商户在填写“企业审核资料”时设置的。

微信 · 开放平台

首页 资源中心 管理中心

1 资料准备

2 资料填写

3 预览并提交

商户负责人信息

负责人姓名
如遇突发情况或需要联系商家，微信希望第一时间联系到负责人

联系电话
为及时取得联系，请最好填写手机号码

重要邮箱
该邮箱将接收与微信支付相关的全部重要信息，强烈建议使用企业邮箱

商户基本信息

申请主体类别

通过审核后，商户将收到邮件通知，邮件中包含重要开发参数，请牢记申请时填写的邮箱地址（如上图示）。

1.3 支付账户

商户注册微信开放平台，创建应用后可以获取到 APPID 和 APPSecret，申请开通支付功能，提交企业以及银行账户资料，商户功能审核通过后，可以获得以下账号资料（包含财付通的商户账户）。

帐号	作用
appId	在开放平台查看，标识申请的应用
paySignKey	支付请求中用于加密的密钥 Key，可验证商户唯一身份，PaySignKey 对应

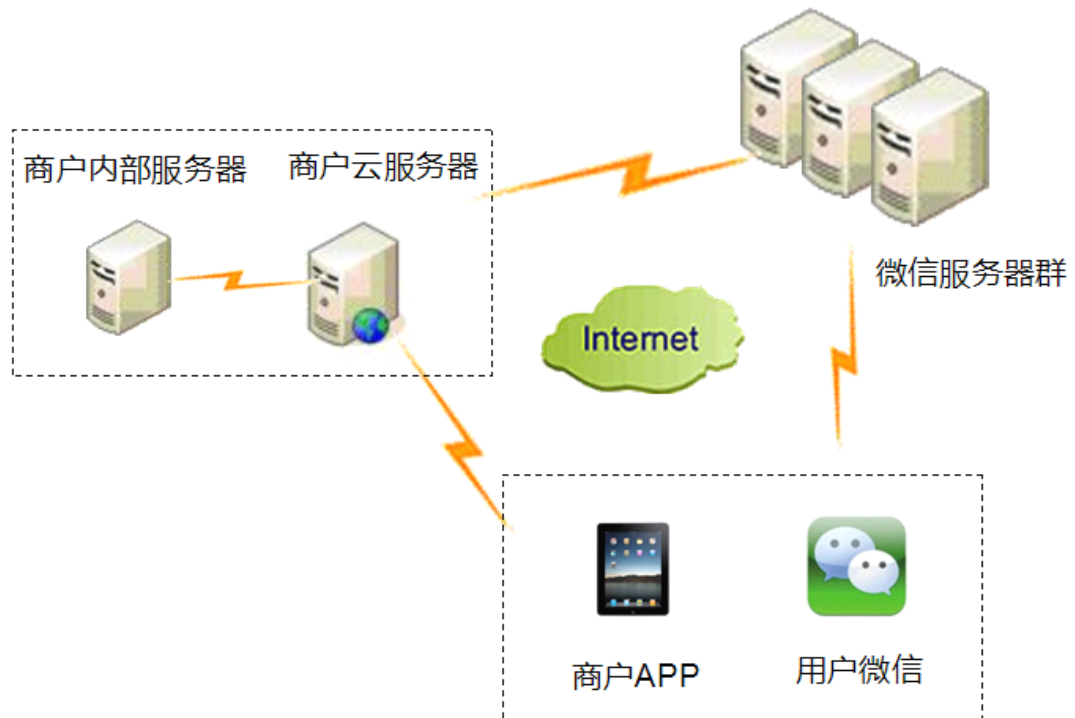
	于支付场景中的 appKey 值。 审核通过后，在微信发送的邮件中查看。
appSecret	除了支付请求需要用到 paySignKey，API 的权限获取所需密钥 Key，在使用所有公众平台 API 时，都需要先用它去换取 access_token，然后再进行调用（详情参考文档 API 接口部分）。在开放平台中查看。
partnerId	财付通商户身份的标识。前三项审核通过后，在财付通发送的邮件中查看。
partnerKey	财付通商户权限密钥 Key。前三项审核通过后，在财付通发送的邮件中查看。

注意：appSecret、paySignKey、partnerKey 是验证商户唯一性的安全标识，请妥善保管。

对于 appSecret 和 paySignKey 的区别，可以这样认为：appSecret 是 API 使用时的登录密码，会在网络中传播的；而 paySignKey 是在所有支付相关数据传输时用于加密并进行身份校验的密钥，仅保留在第三方后台和微信后台，不会在网络中传播，而且 paySignKey 仅用于支付请求。

2. 微信 APP 支付接入

2.1 系统架构



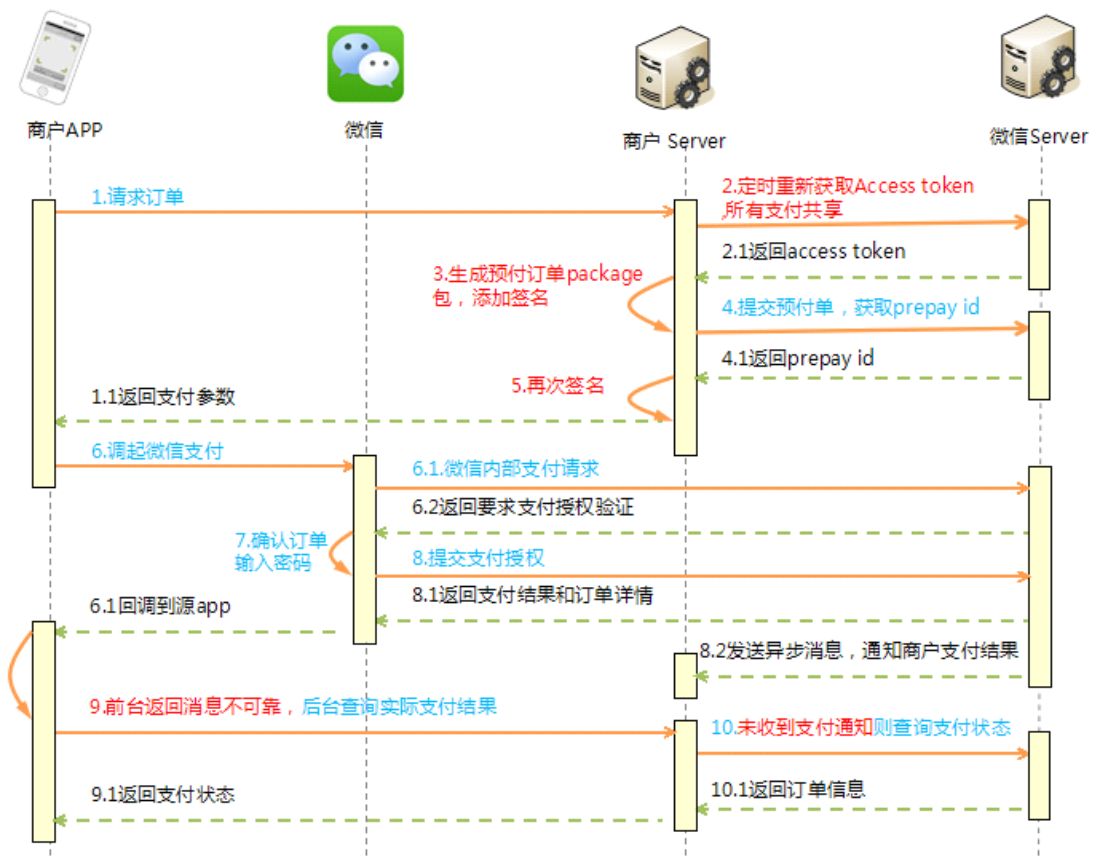
商户接入 APP 支付，系统架构如上图。商户 APP 直接与商户后台服务器通信，涉及支付接口调用都是在商户云服务器与微信后台服务器之间通信。商户服务器生成支付订单，签名后返回给 APP，再通过微信 SDK 提供的 API 调起微信支付。

2.2 基本交互

APP 支付的用户交互如下图：



参看以下的交互时序图，更直观、详细讲述 APP 支付在各端的交互流程。



2.3 支付功能开发介绍

开发工作分为两步，服务器端和客户端开发。商户针对不同的手机系统，开发不同的版本的 APP(比如 IOS、Android 和 WPF)，在服务器端的处理，则是一致的。下面我们依次介绍服务器开发和不同移动系统终端的微信开放平台 API 的使用。

3.接口介绍

3.1 获取 access_token

access_token 对应于公众号是全局唯一的票据，每天有获取次数限制，所有 API 公用，正常情况下 access_token 有效期为 7200 秒，重复获取将导致上次获取的 access_token 失效。它的获取方式为：

http 请求方式: GET

地址：

https://api.weixin.qq.com/cgi-bin/token?grant_type=client_credential&appid=APPID&secret=APPSECRET

请求参数含义：

grant_type：获取 access_token，此处填写 client_credential

appid：第三方用户唯一凭证

secret：第三方用户唯一凭证密钥，既 appsecret

返回结果说明：

正确的 Json 返回示例:

```
{"access_token":"ACCESS_TOKEN","expires_in":7200}
```

返回参数说明：

access_token：获取到的凭证（最大长度为 512 字节）

expires_in：凭证有效时间，单位：秒。正常情况下 access_token 有效期为 7200 秒，重复获取将导致上次获取的 access_token 失效。

错误的 Json 返回示例:

```
{"errcode":40013,"errmsg":"invalid appid"}
```



3.2 生成预支付订单

获取到 access_token 后，通过微信开放平台接口生成预支付订单。

http 请求方式：POST

地址：https://api.weixin.qq.com/pay/genprepay?access_token=ACCESS_TOKEN

Url 中的参数只包含目前微信公众平台凭证 access_token，详细的订单数据放在 PostData

中，格式为 json，示例如下：

```
{
  "appid": "wxd930ea5d5a258f4f",
  "traceid": "crestxu_1398746574",
  "noncestr": "139042a4157a773f209847829d80894d",
  "timestamp": 1398746574,
  "package": "bank_type=WX&body=%E5%AE%89%E5%8D%93%E7%AD%BE%E5%90%8D%E6%B5%8B%E8%AF%95&fee_type=1&input_charset=UTF-8&notify_url=http%3A%2F%2Fweixin.qq.com&out_trade_no=1cdf14d1e3699d61d237cf76ce1c2dca&partner=1900000109&spbill_create_ip=196.168.1.1&total_fee=1&sign=1F4D39942AEE58A0A5B47421ED4A3767",
  "sign_method": "sha1",
  "app_signature": "426f41a7594badcca405a398020c3c7c8f279404"
}
```

其中，各字段含义如下：

appid：开放平台账户的唯一标识

traceid：由开发者自定义，可用于订单的查询与跟踪，建议根据支付用户信息生成此 id

noncestr：32 位内的随机串，防重发

package：订单详情（具体生成方法见后文 3.3 节）

timestamp：时间戳，为自 1970 年 1 月 1 日 00:00(时区：东八区)至当前时间的秒数

app_signature：签名（具体生成方法见后文 3.4 节）

sign_method：密方式，默认为 sha1

返回结果说明：

正确的 Json 返回示例：

```
{"prepayid":"1101000000140429eb40476f8896f4c9","errcode":0,"errmsg":"Success"}
```

错误的 Json 返回示例：

```
{"errcode":48001,"errmsg":"api unauthorized"}
```

3.3 订单详情 (package) 扩展字符串定义

下面将定义 package 的所需字段列表以及签名方法。package 所需字段列表

参数	名称	必填	格式	说明
bank_type	银行通道类型	是	字符串类型，固定为 "WX"，注意大写	固定为 "WX"；
body	商品描述	是	字符串类型，128 字节以下	商品描述；
attach	附加数据	否	字符串类型，128 字节以下	附加数据，原样返回；
partner	商户号	是	字符串类型	注册时分配的财付通商户号 partnerId；
out_trade_no	商户订单号	是	字符串类型，32 字节以下	商户系统内部的订单号，32 个字符内、可包含字母， 确保在商户系统唯一 ；
total_fee	订单总金额	是	字符串类型	订单总金额， 单位为分 ；
fee_type	支付币种	是	字符串类型，默认值是 "1"	取值：1（人民币），暂只支持 1；
notify_url	通知 URL	是	字符串类型，255 字节以下	在支付完成后，接收微信通知支付结果的 URL，需给绝对路径，255 字符内，格式如：http://weixin.qq.com/notiry；不能带查询字符串
spbill_create_ip	订单生成的机器 IP	是	字符串类型，15 字节以下	指用户浏览器端 IP，不是商户服务器 IP，格式为 IPV4；
time_start	交易起始时间	否	字符串类型，14 字节以下	订单生成时间，格式为 yyyyMMddHHmmss，如 2009 年 12 月 25 日 9 点 10 分 10 秒表示为 20091225091010，时区为 GMT+8 beijing；该时间取自商户服务器；
time_expire	交易结束时间	否	字符串类型，14 字节以下	订单失效时间，格式为 yyyyMMddHHmmss，如 2009 年 12 月 27 日 9 点 10 分 10 秒表示为 20091227091010，时区为 GMT+8 beijing；该时间取自商户服务器；
transport_fee	物流费用	否	字符串类型	物流费用，单位为分。如果有值，必须保证 transport_fee + product_fee = total_fee；



product_fee	商品费用	否	字符串类型	商品费用，单位为分。如果有值，必须保证 $\text{transport_fee} + \text{product_fee} = \text{total_fee}$ ；
goods_tag	商品标记	否	字符串类型	商品标记，优惠券时可能用到；
input_charset	传入参数字符编码	是	字符串类型	取值范围："GBK"、"UTF-8"，默认："GBK"

package 生成方法

由于 package 中携带了生成订单的详细信息，因此在微信将对 package 里面的内容进行鉴权，确定 package 携带的信息是真实、有效、合理的。因此，这里将定义生成 package 字符串的方法。

a. 对所有传入参数按照字段名的 ASCII 码**从小到大排序（字典序）**后，使用 URL 键值对的格式（即 $\text{key1}=\text{value1}\&\text{key2}=\text{value2}\dots$ ）拼接成字符串 string1；

b. 在 string1 **最后拼接上 key=paternerKey** 得到 stringSignTemp 字符串，并对 stringSignTemp **进行 md5 运算**，再将得到的**字符串所有字符转换为大写**，得到 sign 值 signValue。

c. 对传入参数所有键值对的 value 进行 **urlencode 转码（注意！进行 urlencode 时要将空格转化为 %20 而不是+）** 后重新拼接成字符串 string2。

d. 将 **sign=signValue 拼接**到 **string2 后面**得到最终的 package 字符串。

下面定义了一段生成 package 字符串的示范过程：

假设以下为 package 传入参数：

```
bank_type=WX
body=安卓签名测试
fee_type=1
input_charset=UTF-8
notify_url=http://weixin.qq.com
out_trade_no=1cdf14d1e3699d61d237cf76ce1c2dca
partner=1900000109
spbill_create_ip=196.168.1.1
total_fee=1
```

i : 经过 a 过程 URL 键值对字典序排序后的字符串 string1 为 :

```
bank_type=WX&body=      安      卓      签      名      测      试
&fee_type=1&input_charset=UTF-8&notify_url=http://weixin.qq.com&out_trade_no=1cdf14d1e
3699d61d237cf76ce1c2dca&partner=1900000109&spbill_create_ip=196.168.1.1&total_fee=1
```

ii : 经过 b 过程后得到 sign 为 :

```
sign
=md5(string1&key=8934e7d15453e97507ef794cf7b0519d).toUpperCase
=md5(bank_type=WX&body=      安      卓      签      名      测      试
&fee_type=1&input_charset=UTF-8&notify_url=http://weixin.qq.com&out_trade_no=1cdf14d1e
3699d61d237cf76ce1c2dca&partner=1900000109&spbill_create_ip=196.168.1.1&total_fee=1&k
ey=8934e7d15453e97507ef794cf7b0519d).toUpperCase()
="1f4d39942aee58a0a5b47421ed4a3767".toUpperCase()
="1F4D39942AEE58A0A5B47421ED4A3767"
```

iii : 再对传入参数中的每一个键值对中的 value 进行 urlencode 编码后得到 :

```
bank_type=WX&body=%E5%AE%89%E5%8D%93%E7%AD%BE%E5%90%8D%E6%B
5%8B%E8%AF%95&fee_type=1&input_charset=UTF-8&notify_url=http%3A%2F%2Fweixin.q
q.com&out_trade_no=1cdf14d1e3699d61d237cf76ce1c2dca&partner=1900000109&spbill_create
_ip=196.168.1.1&total_fee=1
```

iv : 拼接上 sign 后得到最终 package 结果 :

```
bank_type=WX&body=%E5%AE%89%E5%8D%93%E7%AD%BE%E5%90%8D%E6%B
5%8B%E8%AF%95&fee_type=1&input_charset=UTF-8&notify_url=http%3A%2F%2Fweixin.q
q.com&out_trade_no=1cdf14d1e3699d61d237cf76ce1c2dca&partner=1900000109&spbill_create
_ip=196.168.1.1&total_fee=1&sign=1F4D39942AEE58A0A5B47421ED4A3767
```

3.4 支付签名 (app_signature) 生成方法

app_signature 字段是对下订单的行为进行鉴权 , 只有通过了 app_signature 鉴权 , 才能继
续对 package 鉴权并生成预支付单。这里将定义 app_signature 的生成规则。

参与 app_signature 签名的字段包括 : appid、appkey(即 paySignkey)、noncestr、package、
timestamp 以及 traceid。

a. 对所有待签名参数按照字段名的 **ASCII 码从小到大排序 (字典序)** 后 , 使用 URL 键
值对的格式 (即 key1=value1&key2=value2...) 拼接成字符串 string1。



b.对 string1 作签名算法。具体签名算法为 $\text{app_signature} = \text{SHA1}(\text{string1})$ 。

这里给出生成 app_signature 的具体示例如下：

假设参数如下：

```
appid=wx930ea5d5a258f4f
appkey=L8LrMqqeGRxST5reouB0K66CaYAWpqhAVsq7ggKkxHCOastWksvuX1uvmvQcl
xaHoYd3EINBrNO2DHnnzgfVG9Qs473M3DToZug5er46FhuGofumV8H2FVR9qkjSIC5K
noncestr=139042a4157a773f209847829d80894d
package=bank_type=WX&body=%E5%AE%89%E5%8D%93%E7%AD%BE%E5%90%8D
%E6%B5%8B%E8%AF%95&fee_type=1&input_charset=UTF-8&notify_url=http%3A%2F%2F
weixin.qq.com&out_trade_no=1cdf14d1e3699d61d237cf76ce1c2dca&partner=1900000109&spbi
ll_create_ip=196.168.1.1&total_fee=1&sign=1F4D39942AEE58A0A5B47421ED4A3767
timestamp=1398746574
traceid=crestxu_1398746574
```

i：经过 a 过程键值对排序后得到 string1 为：

```
appid=wx930ea5d5a258f4f&appkey=L8LrMqqeGRxST5reouB0K66CaYAWpqhAVsq7gg
KkxHCOastWksvuX1uvmvQclxaHoYd3EINBrNO2DHnnzgfVG9Qs473M3DToZug5er46FhuGo
fumV8H2FVR9qkjSIC5K&noncestr=139042a4157a773f209847829d80894d&package=bank_typ
e=WX&body=%E5%AE%89%E5%8D%93%E7%AD%BE%E5%90%8D%E6%B5%8B%E8%A
F%95&fee_type=1&input_charset=UTF-8&notify_url=http%3A%2F%2Fweixin.qq.com&out_tra
de_no=1cdf14d1e3699d61d237cf76ce1c2dca&partner=1900000109&spbill_create_ip=196.168.1.
1&total_fee=1&sign=1F4D39942AEE58A0A5B47421ED4A3767&timestamp=1398746574&trac
eid=crestxu_1398746574
```

ii：经过 b 过程签名后可得到：

```
paySign=SHA1(appid=wx930ea5d5a258f4f&appkey=L8LrMqqeGRxST5reouB0K66CaYA
WpqhAVsq7ggKkxHCOastWksvuX1uvmvQclxaHoYd3EINBrNO2DHnnzgfVG9Qs473M3DTo
Zug5er46FhuGofumV8H2FVR9qkjSIC5K&noncestr=139042a4157a773f209847829d80894d&pa
ckage=bank_type=WX&body=%E5%AE%89%E5%8D%93%E7%AD%BE%E5%90%8D%E6%
B5%8B%E8%AF%95&fee_type=1&input_charset=UTF-8&notify_url=http%3A%2F%2Fweixin.
qq.com&out_trade_no=1cdf14d1e3699d61d237cf76ce1c2dca&partner=1900000109&spbill_creat
e_ip=196.168.1.1&total_fee=1&sign=1F4D39942AEE58A0A5B47421ED4A3767&timestamp=1
398746574&traceid=crestxu_1398746574)
=426f41a7594badcca405a398020c3c7c8f279404
```

注意：以上操作可通过沙箱测试验证签名的有效性，沙箱地址：

<http://mp.weixin.qq.com/debug/cgi-bin/readtmpl?t=pay/index>

3.5 添加 prepayid 再次签名

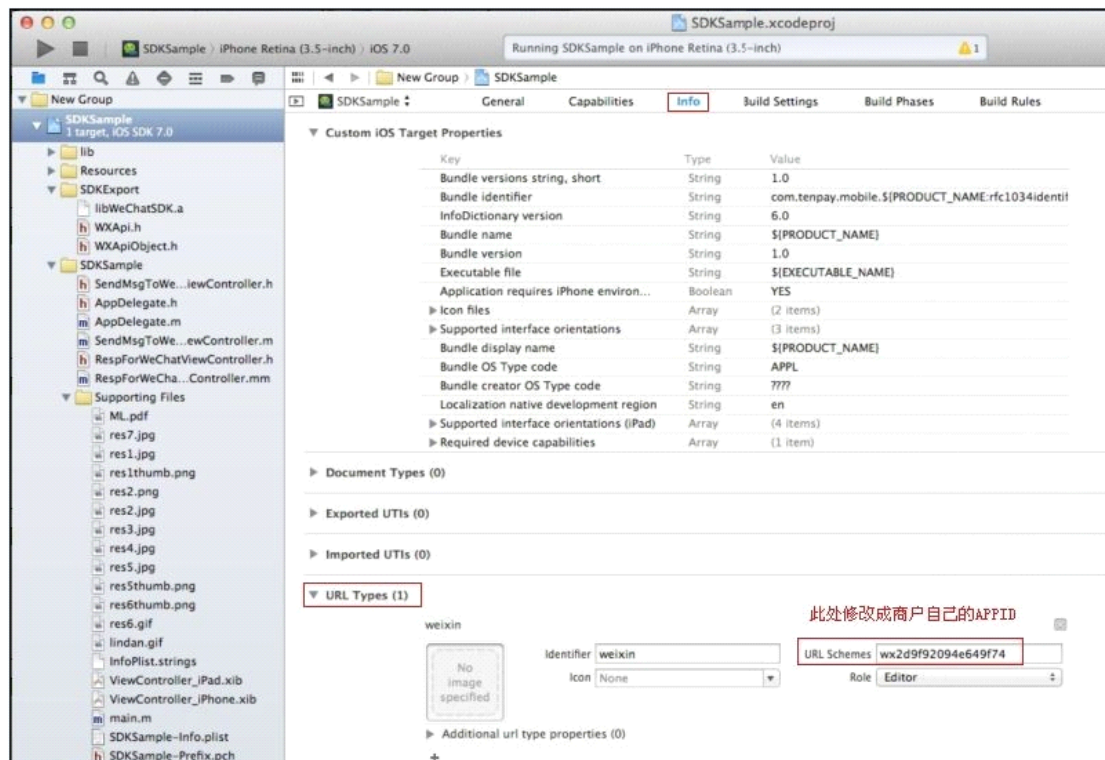
获取到 prepayid 后，将参数 appid、appkey、noncestr、package（注意：此处应置为 Sign=WXPAY）、partnerid、prepayid、timestamp 签名后返回给 APP，签名方法跟 [3.4 节](#) app_signature 说明一致。得到参数列表如下，通过这些参数即可在客户端调起支付。

```
{
  "appid": "wxd930ea5d5a258f4f",
  "noncestr": "139042a4157a773f209847829d80894d",
  "package": "Sign=WXPAY";
  "partnerid": "1900000109"
  "prepayid": "1101000000140429eb40476f8896f4c9",
  "sign": "7ffecb600d7157c5aa49810d2d8f28bc2811827b",
  "timestamp": "1398746574"
}
```

3.6 APP 端开发说明

3.6.1 初始化设置

在 Xcode 中打开项目，引入微信 lib 库和头文件后，设置项目属性中的 URL Schemes 为您的 APPID，如下图所示。我们提供开发实例，请参见附件中的 DEMO。API 详细说明请参考：https://open.weixin.qq.com/zh_CN/htmledition/res/dev/document/sdk/ios/index.html



3.6.2 注册您的 APPID

API 调用前，需要先向微信注册您的 APP，代码如下：

```
[WXApi registerApp:@"wxd930ea5d5a258f4f" withDescription:@"demo 2.0"];
```

3.6.3 用服务器返回的参数调起支付

获取到服务器的订单参数后，以下主要代码即可调起微信支付。

```
PayReq *request = [[[PayReq alloc] init] autorelease];
request.partnerId = @"1900000109";
request.prepayId = @"1101000000140415649af9fc314aa427";
request.package = @"Sign=WXpay";
request.nonceStr = @"a462b76e7436e98e0ed6e13c64b4fd1c";
request.timeStamp = @"1397527777";
request.sign = @"582282d72dd2b03ad892830965f428cb16e7a256";
[WXApi safeSendReq:request];
```

3.6.4 接收支付返回结果

参照微信 SDK Sample，在 WXPAYEntryActivity 类中实现 onResp 函数，处理支付结果

的通知和下一步界面操作。注意由客户端返回的支付结果不能作为最终支付的可信结果，应

以服务器端的支付结果通知为准。代码示例如下：

```
- (void)onResp:(BaseResp *)resp {
    if ([resp isKindOfClass:[PayResp class]]) {
        PayResp *response = (PayResp *)resp;
        switch (response.errCode) {
            case WXSuccess:
                if (_delegate && [_delegate respondsToSelector:@selector(PaySuccess)]) {
                    {
                        [_delegate PaySuccess];
                    }
                    break;
                }
            default:
                if (_delegate && [_delegate respondsToSelector:@selector(PayFail:)]) {
                    {
                        [_delegate PayFail:response.errCode];
                    }
                    break;
                }
        }
    }
}
```

4.通知接口说明

4.1 通知接口简介

用户在成功完成支付后，微信后台通知（POST）商户服务器（notify_url）支付结果。

商户 APP 可轮询后台的支付通知结果后展示。

4.2 补单机制

对后台通知交互时，如果微信收到商户的应答不是 success 或超时，微信认为通知失败，微信会通过一定的策略（如 30 分钟共 8 次）定期重新发起通知，尽可能提高通知的成功率，但微信不保证通知最终能成功。



由于存在重新发送后台通知的情况，因此同样的通知可能会多次发送给商户系统。**商户**

系统必须能够正确处理重复的通知。

微信推荐的做法是，当收到通知进行处理时，首先检查对应业务数据的状态，判断该通知是否已经处理过，如果没有处理过再进行处理，如果处理过直接返回 success。在对业务数据进行状态检查和处理之前，要采用数据锁进行并发控制，以避免函数重入造成的数据混乱。

4.3 通知接口参数

后台通知通过请求中的 notify_url 进行，采用 POST 机制。返回通知中的参数一致，URL

包含如下内容：

字段名	变量名	必填	类型	说明
协议参数				
签名方式	sign_type	否	String(8)	签名类型，取值：MD5、RSA，默认：MD5
字符集	input_charset	否	String(8)	字符编码，取值：GBK、UTF-8，默认：GBK。
签名	sign	是	String(32)	签名
业务参数				
交易模式	trade_mode	是	Int	1-即时到账 其他保留
交易状态	trade_state	是	Int	支付结果： 0—成功 其他保留
商户号	partner	是	String(10)	商户号，也即之前步骤的 partnerid，由微信统一分配的 10 位正整数 (120XXXXXXX)号
付款银行	bank_type	是	String(16)	银行类型，在微信中使用 WX
银行订单号	bank_billno	否	String(32)	银行订单号
总金额	total_fee	是	Int	支付金额，单位为分，如果 discount 有值，通知的 total_fee + discount = 请求的 total_fee
币种	fee_type	是	Int	现金支付币种，目前只支持人民币，默认值是 1-人民币
通知 ID	notify_id	是	String(128)	支付结果通知 id，对于某些特定商



				户，只返回通知 id，要求商户据此查询交易结果
订单号	transaction_id	是	String(28)	交易号，28 位长的数值，其中前 10 位为商户号，之后 8 位为订单产生的日期，如 20090415，最后 10 位是流水号。
商户订单号	out_trade_no	是	String(32)	商户系统的订单号，与请求一致。
商户数据包	attach	否	String(127)	商户数据包，原样返回，空参数不传递
支付完成时间	time_end	是	String(14)	支付完成时间，格式为 yyyyMMddhhmmss，如 2009 年 12 月 27 日 9 点 10 分 10 秒表示为 20091227091010。时区为 GMT+8 beijing。
物流费用	transport_fee	否	Int	物流费用，单位分，默认 0。如果有值，必须保证 $\text{transport_fee} + \text{product_fee} = \text{total_fee}$
物品费用	product_fee	否	Int	物品费用，单位分。如果有值，必须保证 $\text{transport_fee} + \text{product_fee} = \text{total_fee}$
折扣价格	discount	否	Int	折扣价格，单位分，如果有值，通知的 $\text{total_fee} + \text{discount} =$ 请求的 total_fee

上文中提交过来的参数，需要校验签名，sign 的签名方式与 [3.3 节](#) 介绍一致，将生成的签名与提交过来的签名 sign 对比，相同则验证为微信发的消息，否则扔弃不处理。同时，在 postData 中还将包含 xml 数据。各字段定义如下：

参数	必填	说明
AppId	是	字段名称：公众号 id；字段来源：商户注册具有支付权限的公众号成功后即可获得；传入方式：由商户直接传入。
TimeStamp	是	字段名称：时间戳；字段来源：商户生成从 1970 年 1 月 1 日 00:00:00 至今的秒数，即当前的时间；由商户生成后传入。取值范围：32 字符以下
NonceStr	是	字段名称：随机字符串；字段来源：商户生成的随机字符串；取值范围：长度为 32 个字符以下。由商户生成后传入。取值范围：32 字符以下
OpenId	是	支付该笔订单的用户 ID，商户可通过公众号其他接口为付款用户服务。

AppSignature	是	字段名称：签名；字段来源：对前面的其他字段与 appKey 按照字典序排序后，使用 SHA1 算法得到的结果。由商户生成后传入。
IsSubscribe	是	用户是否关注了公众号。1 为关注，0 为未关注。

AppSignature 依然是根据 [3.4 节](#) 支付签名 (paySign) 签名方式生成，参与签名的字段为：

appid、appkey、timestamp、noncestr、openid、issubscribe。

从以上信息可以看出，URL 参数中携带订单相关信息，**postData** 中携带该次支付的用户相关信息，这将便于商户拿到 openid，以便后续提供更好的售后服务。

回调实例：

http://URL/pub/notify/notify.php?bank_type=3122&discount=0&fee_type=1&input_charset=GBK¬ify_id=vdjzT_7d3i22q4oERvwGNXAKl4iGJ9Y66lvswAb7_DvD7_amrD5Q-vYIKxliGw79dNfoeyOOn8PIYirkd98a_LR1wdQYJckt&out_trade_no=2914136776&partner=1217752501&product_fee=1&sign=0830039FE9E82E6E98F3E7746301FEBC&sign_type=MD5&time_end=20140430132820&total_fee=1&trade_mode=1&trade_state=0&transaction_id=1217752501201404303163570075&transport_fee=0

POST 内容：

```
<xml>
<OpenId><![CDATA[oUpF8uN95-Ptaags6E_roPHg7AG0]]></OpenId>
<AppId><![CDATA[wx2421b1c4370ec43b]]></AppId>
<IsSubscribe>1</IsSubscribe>
<TimeStamp>1398835924</TimeStamp>
<NonceStr><![CDATA[1n0TsquoVMnCSkjs]]></NonceStr>
<AppSignature><![CDATA[3b00e45a02018dd25bc1ef7a2b4101cba435cbab]]></AppSignature>
<SignMethod><![CDATA[sha1]]></SignMethod>
</xml>
```

4.4 后台通知结果返回

微信后台通过 notify_url 通知商户，商户做业务处理后，需要**以字符串的形式反馈处理**

结果，内容如下：



返回结果	结果说明
success	处理成功，微信系统收到此结果后不再进行后续通知
fail 或其它字符	处理不成功，微信收到此结果或者没有收到任何结果，系统通过补单机制再次通知

5.API 接口说明

5.1 API 接口简介

为了更好地接入支付的整个流程，包括购买、通知、发货等，微信提供了一系列的支付相关 API，以供第三方调用。

5.2 API 列表

5.2.1 发货通知 delivernotify

为了更好地跟踪订单的情况，需要第三方在收到最终支付通知之后，调用发货通知 API 告知微信后台该订单的发货状态。

发货时间限制：虚拟、服务类 24 小时内，实物类 72 小时内。

请在收到支付通知后，按时发货，并使用发货通知接口将相关信息同步到微信后台。若平台在规定时间内没有收到，将视作发货超时处理。

API 的 URL 为：https://api.weixin.qq.com/pay/delivernotify?access_token=xxxxxx

URL 中的参数只包含目前微信公众平台凭证 access_token，而发货通知的真正的数据是放在 postData 中的，格式为 json，如下：

```
{
  "appid": "wwwb4f85f3a797777",
  "openid": "oX99MDgNcgwnz3zFN3DNmo8uwa-w",
  "transid": "1111122223333",
  "out_trade_no": "555666uuu",
```

```
{
  "deliver_timestamp": "1369745073",
  "deliver_status": "1",
  "deliver_msg": "ok",
  "app_signature": "53cca9d47b883bd4a5c85a9300df3da0cb48565c",
  "sign_method": "sha1"
}
```

其中，

appid 是公众平台账户的 AppId；

openid 是购买用户的 OpenId，这个已经放在最终支付结果通知的 PostData 里了；

transid 是交易单号；

out_trade_no 是第三方订单号；

deliver_timestamp 是发货时间戳，这里指的是 Linux 时间戳；

deliver_status 是发货状态，1 表明成功，0 表明失败，失败时需要在 deliver_msg 填上失败原因；

deliver_msg 是发货状态信息，失败时可以填上 UTF8 编码的错误提示信息，比如“该商品已退款”；

app_signature 依然是根据支付签名（paySign）生成方法中所讲的签名方式生成的，参加签名字段为：appid、appkey、openid、transid、out_trade_no、deliver_timestamp、deliver_status、deliver_msg；

sign_method 是签名方法（不计入签名生成）；

微信公众平台在校验 ok 之后，会返回数据表明是否通知成功，例如：

```
{"errcode":0,"errmsg":"ok"}
```

如果有异常，会在 errcode 和 errmsg 描述出来，如果成功 errcode 就为 0。

5.2.2 订单查询 orderquery

因为某一方技术的原因，可能导致商户在预期时间内都收不到最终支付通知，此时商户



可以通过该 API 来查询订单的详细支付状态。

API 的 URL 为：https://api.weixin.qq.com/pay/orderquery?access_token=xxxxxx

URL 中的参数只包含目前微信公众平台凭证 access_token，而订单查询的真正数据是放在 PostData 中的，格式为 json，如下：

```
{
  "appid": "wwwb4f85f3a797777",
  "package": "out_trade_no=11122&partner=1900090055&sign=4e8d0df3da0c3d0df38f",
  "timestamp": "1369745073",
  "app_signature": "53cca9d47b883bd4a5c85a9300df3da0cb48565c",
  "sign_method": "sha1"
}
```

其中，

appid 是公众平台账户的 AppId；

package 是查询订单的关键信息数据，包含第三方唯一订单号 out_trade_no、财付通商户身份标识 partner（即前文所述的 partnerid）、签名 sign，其中 sign 是对参数字典序排序并使用&联合起来，最后加上&key=partnerkey（唯一分配），进行 md5 运算，再转成全大写，最终得到 sign，对于示例，就是：

sign=md5（out_trade_no=11122&partner=1900090055&key=xxxxxx）.toupper；

timestamp 是 linux 时间戳；

app_signature 依然是根据支付签名（paySign）生成方法中所讲的签名方式生成的，参加签名字段为：appid、appkey、package、timestamp；

sign_method 是签名方法（不计入签名生成）；

微信公众平台在校验 ok 之后，会返回数据表明是否通知成功，例如：

```
{"errcode":0,"errmsg":"ok", ..... }
```

如果有异常，会在 errcode 和 errmsg 描述出来，如果成功 errcode 就为 0。

如果查询成功，会返回详细的订单数据，如下：


```
{
  "errcode":0,
  "errmsg":"ok",
  "order_info":
  {
    "ret_code":0,
    "ret_msg": "",
    "input_charset":"GBK",
    "trade_state":"0",
    "trade_mode":"1",
    "partner":"1900000109",
    "bank_type":"CMB_FP",
    "bank_billno":"207029722724",
    "total_fee":"1",
    "fee_type":"1",
    "transaction_id":"1900000109201307020305773741",
    "out_trade_no":"2986872580246457300",
    "is_split":"false",
    "is_refund":"false",
    "attach": "",
    "time_end":"20130702175943",
    "transport_fee":"0",
    "product_fee":"1",
    "discount":"0",
    "rmb_total_fee":""
  }
}
```

对于详细的订单信息，放在 order_info 中的 json 数据中，各个字段的含义如下：

ret_code 是查询结果状态码，0 表明成功，其他表明错误；

ret_msg 是查询结果出错信息；

input_charset 是返回信息中的编码方式；

trade_state 是订单状态，0 为成功，其他为失败；

trade_mode 是交易模式，1 为即时到帐，其他保留；

partner 是财付通商户号，即前文的 partnerid；

bank_type 是银行类型；

bank_billno 是银行订单号；



total_fee 是总金额，单位为分；

fee_type 是币种，1 为人民币；

transaction_id 是财付通订单号；

out_trade_no 是第三方订单号；

is_split 表明是否分账，false 为无分账，true 为有分账；

is_refund 表明是否退款，false 为无退款，ture 为退款；

attach 是商户数据包，即生成订单 package 时商户填入的 attach；

time_end 是支付完成时间；

transport_fee 是物流费用，单位为分；

product_fee 是物品费用，单位为分；

discount 是折扣价格，单位为分；

rmb_total_fee 是换算成人民币之后的总金额，单位为分，一般看 total_fee 即可。

6. 常见问题和注意事项

6.1 帮助 SDK

为了帮助商户更好地完成支付功能的开发，我们提供了生成支付请求的辅助 SDK，里

面封装了相关的参数签名算法，可供商户参考。下载地址为：

http://mp.weixin.qq.com/cgi-bin/readtemplate?t=business/faq_tmpl&lang=zh_CN

6.2 常见基本概念疑惑

(1) 还没拿到正式号，如何调试测试？

答：只有在“商户功能”审核通过以后，收到了微信和财付通的相关邮件，才可以进行

开发。

6.3 常见错误现象及解决方法

(1) 支付提示“众账号支付使用了无效的商户号，无法发起该笔交易”

答：请检查是否使用了正确的商户号，确认开放平台前三项审核结果均为“审核通过”。

(2) 点击支付按钮，提示“支付请求参数错误，请核实再试” or “交易出错，请稍后再试”

答：请检查 package 参数是否为空，并注意格式，检查签名是否有误。可参考6.1中提到的帮助 SDK。

(3) 点击支付按钮，提示“该支付签名无效，无法发起该笔交易”

答：调起支付的 SHA1签名错误，请检查相关签名。可参考6.1中提到的帮助 SDK。

(4) 获取 token 过程中，提示“out of freq”

答：获取 token 频率超限。Token 有效期为7200秒，不需要、也不要每次支付都重新获取一次，建议进行保存，即将过期时或已经过期时，再重新获取。

(5) Notify url 无法接收通知

答：不支持非 80 端口，同时注意不要被防火墙拦截。

(6) Notify url 通知收到的参数缺少 PostData 部分

答：可尝试使用此方法接收：`file_get_contents()`

6.4 常见注意事项

(1) 参数大小写问题



请留意文档中要求的字符大小写问题，如“md5 运算后，字符串的字符要转换为大写”，

“bank_type 填写时，WX 大写”，“参与签名的字符串要小写”等。

(2) 参数格式问题

所有传入参数，均为字符串类型，请注意文档中各处的具体要求。

(3) 签名问题

建议使用“帮助 SDK”生成签名；时间戳 timeStamp 和随即字符串 nonceStr，在调用 JS API 和参与 paySign 签名时要保持一致。请不要使用文档中的参数值进行签名调试（数值均已失效）。

(4) 时间戳问题

请使用 Linux 时间戳，注意为字符串格式。精确到秒，不需要到毫秒，即 10 位数字。

(5) 同一商户订单号支付问题

强烈建议商户的 out trade no 每一单均为唯一数值，调试过程中，请注意两次尝试使用的订单号是否不同。

6.5 联系我们

如果您在开发中遇到问题，可以进入到我们的论坛（网址：<http://mp.weixin.qq.com/qa/index.php?qa=questions>），上面有用户成功解决的方案，更高效解决您的问题。如果未找到解决方案，可以在上面提问，我们会第一时间回复。

获取最新文档和支付体验，请关注公众号：WXPayService “微信支付商户助手”。