公众号支付接口文档 V2.5

1.微信支付简介

1.1 功能简介

微信支付，是基于微信客户端提供的支付服务功能。同时向商户提供销售经营分析、账户和资金管理的功能支持。用户通过扫描二维码、微信内打开商品页面购买等多种方式调起微信支付模块完成支付。

微信支持公众号内支付，即基于公众号向用户收款，公众号相当于收款的商户。其中支付方式，可以分为 JS API（网页内）支付、Native（原生）支付。商户可以结合业务场景，自主选择支付方式。

本文将全面介绍公众号支付的技术方案。

1.2 支付账户

商户向微信公众平台提交企业以及银行账户资料，商户功能审核通过后，可以获得以下帐户（包含财付通的商户账户），用于公众号支付。

帐号	**作用**
appId	公众号身份的唯一标识。前三项审核通过后，在微信发送的邮件中查看。
paySignKey	公众号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**。前三项审核通过后，在微信发送的邮件中查看。
appSecret	除了支付请求需要用到 paySignKey，公众平台接口 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）。前三项审核通过后，在微

	信发送的邮件中查看。
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。

1.3 支付方式

公众号支付有 2 种方式：

JS API（网页内）支付：是指用户打开图文消息或者扫描二维码，在微信内置浏览器打开网页进行的支付。商户网页前端通过使用微信提供的 JS API，调用微信支付模块。这种方式，适合需要在商户网页进行选购下单的购买流程。

Native（原生）支付：是指商户组成符合 Native（原生）支付规则的 URL 链接，用户可通过在会话中点击链接或者扫描对应的二维码直接进入微信支付模块（客户端界面），即可进行支付。这种方式，适合无需选购直接支付的购买流程。

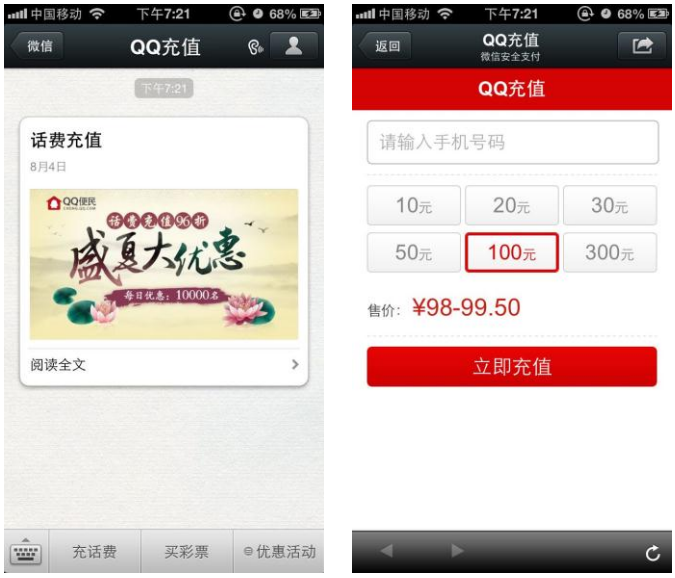
以上两种支付方式，**最大的差别在于是否需要经过网页调起支付。**以下是两种支付方式的基本交互。

1.3.1 网页内支付场景---JS API（网页内）支付接口

商户已有 H5 商城网站 ,在微信内打开网页时 ,可以调用微信支付完成下单购买的流程。

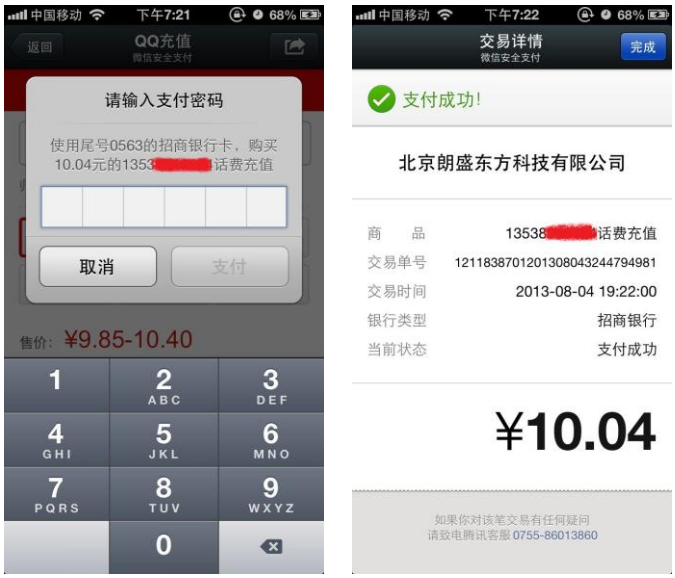
步骤（1）：左图，商户下发图文消息或者通过自定义菜单吸引用户点击进入商户网页。

步骤（2）：右图，进入家网页，用户选择购买，完成选购流程。



步骤（3）：左图，调起微信支付控件，用户开始输入支付密码。

步骤（4）：右图，密码验证通过，支付成功。商户后台得到支付成功的通知。



步骤（5）：左图，返回商户页面，显示购买成功。该页面由商户自定义。

步骤（6）：右图，公众号下发消息，提示发货成功。该步骤可选。

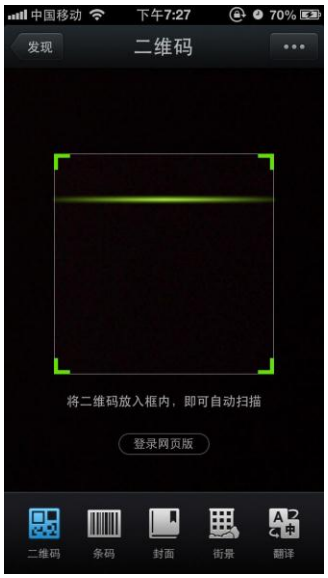
注意：商户也可以把商品网页的链接生成**二维码**，用户扫一扫打开后即可完成购买支付。

1.3.2 线下扫码购买场景---Native（原生）支付接口

与网页内支付场景不同，部分商户不需要经过网页选购，可以直接下单购买。

步骤（1）：左图，商户根据微信支付的规则，为不同商品生成不同的二维码，张贴在各种场景，便于用户扫描购买。

步骤（2）：右图，用户使用微信扫描二维码后，获取商品信息，同时到商户后台下单。



步骤（3）：左图，用户开始支付，输入支付密码。

步骤（4）：右图，支付成功，商户后台得到通知，进行发货处理。

2.JS API（网页内）支付接口

2.1 支付场景

以下是支付场景的交互细节，请认真阅读，并设计商户页面的逻辑：

（1）用户打开商户网页选购商品，发起支付，在网页通过 JavaScript 调用 getBrandWCPayRequest 接口，发起微信支付请求，用户进入支付流程。

（2）用户成功支付点击完成按钮后，商户的前端会收到 JavaScript 的返回值。商户可直接跳转到支付成功的静态页面进行展示。

（3）商户后台收到来自微信开放平台的支付成功回调通知，标志该笔订单支付成功。

注：

（2）和（3）的触发不保证遵循严格的时序。JS API 返回值作为触发商户网页跳转的标志，但商户后台应该**只在收到微信后台的支付成功回调通知后，才做真正的支付成功的处理**。

JS API 返回值目前只在支付成功时返回，后续版本将扩展返回值，以便商户做更多个

性化的展示。

2.2 功能交互

2.3 获取当前微信版本号

由于微信 5.0 版本后才加入微信支付模块，**低版本用户调用微信支付功能将无效。**因此，建议商户通过 user agent 来确定用户当前的版本号后再调用支付接口。以 iPhone 版本为例，可以通过 user agent 可获取如下微信版本示例信息：

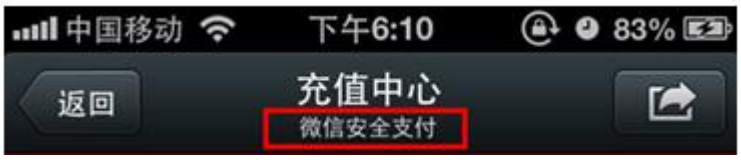
"Mozilla/5.0(iphone;CPU iphone OS 5_1_1 like Mac OS X) AppleWebKit/534.46(KHTML,like Geocko) Mobile/9B206 MicroMessenger/5.0"

其中 5.0 为用户安装的微信版本号，商户可以判定版本号是否高于或者等于 5.0。

2.4 显示微信安全支付标题

对于商户具有支付权限且需要调用微信支付的页面，为了让用户增加购买信心，确认交

易环境安全，**微信强烈建议商户使用“微信安全支付”标题。**安全支付标题的如下图。



显示支付安全标题，需将原始链接添加上"showwxpaytitle=1"的尾串。通过这种方式，商户的页面将出现微信安全支付的标识。例如，原始 URL 为：http://weixin.qq.com,显示安全支付标题的 URL 为：http://weixin.qq.com?showwxpaytitle=1。

当用户在微信里打开 http://weixin.qq.com 不会直接出现微信安全支付的标题，而打开 http://weixin.qq.com?showwxpaytitle=1 后将出现微信安全支付标题。

2.5JS API 支付接口（getBrandWCPayRequest）定义

微信 JS API 只能在微信内置浏览器中使用，其他浏览器调用无效。微信提供 getBrandWCPayRequest 接口供商户前端网页调用，调用之前微信会鉴定商户支付权限，若商户具有调起支付的权限，则将开始支付流程。这里主要介绍支付前的接口调用规则，支付状态消息通知机制请参加下文。

接口需要注意：**所有传入参数都是字符串类型！**

getBrandWCPayRequest 参数以及返回值定义

参数列表

参数	名称	必填	格式	说明
appId	公众号 id	是	字符串类型	商户注册具有支付权限的公众号成功后即可获得；
timeStamp	时间戳	是	字符串类型，32 个字节以下	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，且最终需要转换为字符串形式；
nonceStr	随机字符串	是	字符串类型，32 个字节以下	商户生成的随机字符串；
package	订单详情扩展字符串	是	字符串类型，4096 个字节以下	商户将订单信息组成该字符串，具体组成方案参见接口使用说明中 package 组包帮助；由商

				户按照规范拼接后传入；
signType	签名方式	是	字符串类型，参数取值"SHA1"	按照文档中所示填入，目前仅支持 SHA1；
paySign	签名	是	字符串类型	商户将接口列表中的参数按照指定方式进行签名，签名方式使用 signType 中标示的签名方式，具体签名方案参见接口使用说明中签名帮助；由商户按照规范签名后传入；

返回结果

返回值	说明
err_msg	get_brand_wcpay_request:ok 支付成功
	get_brand_wcpay_request:cancel 支付过程中用户取消
	get_brand_wcpay_request:fail 支付失败

注：JS API 的返回结果 get_brand_wcpay_request:ok 仅在用户成功完成支付时返回。由于前端交互复杂，get_brand_wcpay_request:cancel 或者 get_brand_wcpay_request:fail 可以统一处理为用户遇到错误或者主动放弃，不必细化区分。

2.6 订单详情（package）扩展字符串定义

在商户调起 JS API 时，商户需要此时确定该笔订单详情，并将该订单详情通过一定的方式进行组合放入 package。JS API 调用后，微信将通过 package 的内容生成预支付单。下面将定义 package 的所需字段列表以及签名方法。

接口需要注意：**所有传入参数都是字符串类型！**

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://wap.tenpay.com/tenpay.asp；
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 transport_fee + product_fee=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 key1=value1&key2=value2...）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 string1 中的所有键值对中的 value 进行 **urlencode 转码**，按照 a 步骤重新拼接成字符串，得到 string2。对于 JS 前端程序，一定要使用函数 encodeURIComponent 进行 urlencode 编码（**注意！进行 urlencode 时要将空格转化为%20 而不是+**）。

d.将 **sign=signValue 拼接到 string1 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,
fee_type=1,
body=XXX,
input_charset=GBK,
partner=1900000109,
total_fee=1,
spbill_create_ip=127.0.0.1,
out_trade_no=16642817866003386000,
notify_url=http://www.qq.com

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=XXX&fee_type=1&input_charset=GBK&notify_url=http://www.qq.com&out_trade_no=16642817866003386000&partner=1900000109&spbill_create_ip=127.0.0.1&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=XXX&fee_type=1&input_charset=GBK&notify_url=http://www.qq.com&out_trade_no=16642817866003386000&partner=1900000109&spbill_create_ip=127.0.0.1&total_fee=1&key=8934e7d15453e97507ef794cf7b0519d).toUpperCase()
="beef37ad19575d92e191c1e4b1474ca9".toUpperCase()
="BEEF37AD19575D92E191C1E4B1474CA9"

iii：再对 string1 中的每一个键值对中的 value 进行 urlencode 编码后得到：

bank_type=WX&body=XXX&fee_type=1&input_charset=GBK&notify_url=http%3a%2f%2fwww.qq.com&out_trade_no=16642817866003386000&partner=1900000109&spbill_create_ip=127.0.0.1&total_fee=1

iv：拼接上 sign 后得到最终 package 结果：

bank_type=WX&body=XXX&fee_type=1&input_charset=GBK&notify_url=http%3a%2f%2fwww.qq.com&out_trade_no=16642817866003386000&partner=1900000109&spbill_create_ip=127.0.0.1&total_fee=1&sign=BEEF37AD19575D92E191C1E4B1474CA9

2.7 支付签名（paySign）生成方法

paySign 字段是对本次发起 JS API 的行为进行鉴权，只有通过了 paySign 鉴权，才能继续对 package 鉴权并生成预支付单。这里将定义 paySign 的生成规则。

参与 paySign 签名的字段包括：appid、timestamp、noncestr、package 以及 appkey（即 paySignkey）。这里 signType 并不参与签名。

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。这里需要注意的是**所有参数名均为小写字符**，例如 appId 在排序后字符串则为 appid；

对 string1 作签名算法，字段名和字段值都采用原始值（**此时 package 的 value 就对应了使用 2.6 中描述的方式生成的 package**），**不进行 URL 转义**。具体签名算法为 paySign = SHA1(string)。

这里给出生成 paySign 的具体示例如下：

假设参数如下：

"appId":"wxf8b4f85f3a794e77",
"timeStamp":"189026618",
"nonceStr":"adssdasssd13d",
"package":"bank_type=WX&body=XXX&fee_type=1&input_charset=GBK&notify_url=http%3a%2f%2fwww.qq.com&out_trade_no=16642817866003386000&partner=1900000109&spbill_create_ip=127.0.0.1&total_fee=1&sign=BEEF37AD19575D92E191C1E4B1474CA9"
"appKey":"2Wozy2aksie1puXUBpWD8oZxiD1DfQuEaiC7KcRATv1Ino3mdopKaPGQQ7TtkNySuAmCaDCrw4xhPY5qKTBl7Fzm0RgR3c0WaVYIXZARsxzHV2x7iwPPzOz94dnwPWSn"

i：经过 a 过程键值对排序后得到 string1 为：

appid=wxf8b4f85f3a794e77&appkey=2Wozy2aksie1puXUBpWD8oZxiD1DfQuEaiC7KcRATv1Ino3mdopKaPGQQ7TtkNySuAmCaDCrw4xhPY5qKTBl7Fzm0RgR3c0WaVYIXZARsxzHV2x7iwPPzOz94dnwPWSn&noncestr=adssdasssd13d&package=bank_type=WX&body=XXX&f

ee_type=1&input_charset=GBK&notify_url=http%3a%2f%2fwww.qq.com&out_trade_no=16642817866003386000&partner=1900000109&spbill_create_ip=127.0.0.1&total_fee=1&sign=BEEF37AD19575D92E191C1E4B1474CA9&timestamp=189026618

ii：经过 b 过程签名后可得到：

paySign=SHA1(appid=wxf8b4f85f3a794e77&appkey=2Wozy2aksie1puXUBpWD8oZxiD1DfQuEaiC7KcRATv1Ino3mdopKaPGQQ7TtkNySuAmCaDCrw4xhPY5qKTBl7Fzm0RgR3c0WaVYIXZARsxzHV2x7iwPPzOz94dnwPWSn&noncestr=adssdasssd13d&package=bank_type=WX&body=XXX&fee_type=1&input_charset=GBK&notify_url=http%3a%2f%2fwww.qq.com&out_trade_no=16642817866003386000&partner=1900000109&spbill_create_ip=127.0.0.1&total_fee=1&sign=BEEF37AD19575D92E191C1E4B1474CA9&timestamp=189026618)
=7717231c335a05165b1874658306fa431fe9a0de

2.8 接口使用示例

接口需要注意：**所有传入参数都是字符串类型！**使用 JavaScript、PHP 等弱类型语言需要关注一下。

示例代码如下：

```
//传入公众号名称，时间戳，随机串，Package 扩展字段，签名方式和 PaySign 签名
WeixinJSBridge.invoke('getBrandWCPayRequest',{
    "appId" : "wxf8b4f85f3a794e77",
    "timeStamp" : "189026618",
    "nonceStr" : "adssdasssd13d",
    "package" :
    "bank_type=WX&body=XXX&fee_type=1&input_charset=GBK&notify_url=http%3a%2f%2fwww.qq.com&out_trade_no=16642817866003386000&partner=1900000109&spbill_create_ip=127.0.0.1&total_fee=1&sign=BEEF37AD19575D92E191C1E4B1474CA9",
    "signType" : "SHA1",
    "paySign" : "7717231c335a05165b1874658306fa431fe9a0de"
        },function(res){
    // 返回 res.err_msg,取值
    // get_brand_wcpay_request:cancel 用户取消
    // get_brand_wcpay_request:fail 发送失败
    // get_brand_wcpay_request:ok 发送成功
    WeixinJSBridge.log(res.err_msg);
    alert(res.err_code+res.err_desc);
```

```
});
```

3.Native（原生）支付接口

3.1 支付场景

（1）商户想使用 Native（原生）支付方式，除了填写申请支付的资料，还需要**提供一个获取订单 package 的回调 URL**（该 URL 在公众平台后台登记），以便微信后台通过 POST 的方式去获取该笔支付的订单信息。

（2）商户根据微信指定的规则生成原生支付的 URL 字符串，当用户点击该字符串，或者通过扫描该字符串对应的二维码时，微信后台开始进入预支付流程。

（3）微信后台鉴定权限后，**通过 POST 方式向商户后台获取生成订单的必要信息**，再次鉴定获取的信息成功后，客户端将进入支付流程。

（4）支付成功后，微信后台将通过 POST 机制通知商户后台该笔交易已成功。

3.2 基本交互

3.3Native（原生）支付接口描述

Native（原生）支付过程中，首先需要商户定义符合 Native（原生）支付规范的 URL，也就是 Native(原生)支付 URL ,同时在微信后台 POST 商户后台时需要提供 package 内容。

因此这里将重点介绍支付前的接口调用，支付通知等信息请查看下文。

3.4Native（原生）支付 URL 定义

Native（原生）支付 URL 是一系列具有 weixin://wxpay/bizpayurl?前缀的 URL，同时后面紧跟着一系列辨别商户的键值对。Native（原生）支付 URL 的规则如下：

weixin://wxpay/bizpayurl?sign=XXXXX&appid=XXXXXX&productid=XXXXXX&timestamp=XXXXXX&noncestr=XXXXXX

其中 XXXXX 为商户需要填写的内容，具体参数定义如下：

参数	名称	必填	格式	说明
appid	公众号 id；	是	字符串类型	商户注册具有支付权限的公众号成功后即可获得；
timestamp	时间戳	是	字符串类型，32 字符以下	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，且最终需要转换为字符串形式：
noncestr	随机字符串	是	字符串类型，32 字符以下	商户生成的随机字符串；
productid	商品唯一 id	是	字符串类型，32 字符以下	商户需要定义并维护自己的商品 id ,这个 id 与一张订单等价，微信后台凭借该 id 通过 POST 商户后台获取交易必须信息；
sign	签名	是	字符串类型	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；

3.5Native（原生）支付 URL 签名方式

参与 sign 签名的字段包括：appid、timestamp、noncestr、productid 以及 appkey。

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。这里需要注意的是**所有参数名均为小写字符**，即 appId 排序后的字符串则为 appid；

b.对 string1 作签名算法，字段名和字段值都采用原始值，**不进行 URL 转义。**具体签名算法为 sign = SHA1(string)。

这里给出生成 sign 的具体示例如下：

假设参数如下：

"appid":"wxf8b4f85f3a794e77",
"timestamp":"189026618",
"noncestr":"adssdasssd13d",
"productid":"123456",
"appkey":"2Wozy2aksie1puXUBpWD8oZxiD1DfQuEaiC7KcRATv1Ino3mdopKaPGQQ7Tt

kNySuAmCaDCrw4xhPY5qKTBl7Fzm0RgR3c0WaVYIXZARsxzHV2x7iwPPzOz94dnwPWSn"

则经过 a 过程后得到：

string1=
"appid=wxf8b4f85f3a794e77&appkey=2Wozy2aksie1puXUBpWD8oZxiD1DfQuEaiC7KcRATv1Ino3mdopKaPGQQ7TtkNySuAmCaDCrw4xhPY5qKTBl7Fzm0RgR3c0WaVYIXZARsxzHV2x7iwPPzOz94dnwPWSn&noncestr=adssdasssd13d&productid=123456&timestamp=189026618"

则经过 SHA1 处理后得到：

sign= 18c6122878f0e946ae294e016eddda9468de80df

3.6Native（原生）支付回调商户后台获取 package

在公众平台接到用户点击上述特殊 Native（原生）支付的 URL 之后，会调用注册时填写的商户获取订单 Package 的回调 URL。

假设回调 URL 为 https://www.outdomain.com/cgi-bin/bizpaygetpackage

微信公众平台调用时会使用 POST 方式，推送 xml 格式的 PostData，形如：

```
<xml>
    <OpenId><![CDATA[111222]]></OpenId>
    <AppId><![CDATA[wwwwb4f85f3a797777]]></AppId>
    <IsSubscribe>1</IsSubscribe>
    <ProductId><![CDATA[777111666]]></ProductId>
    <TimeStamp> 1369743908</TimeStamp>
    <NonceStr><![CDATA[YvMZOX28YQkoU1i4NdOnlXB1]]></NonceStr>
    <AppSignature><![CDATA[a9274e4032a0fec8285f147730d88400392acb9e]]></AppSignature>
    <SignMethod><![CDATA[sha1]]></ SignMethod >
</xml>
```

参数说明：

（1）OpenId，点击链接准备购买商品的用户 OpenId

（2）AppId，公众帐号的 appid

（3）IsSubscribe，标记用户是否订阅该公众帐号，1 为关注，0 为未关注

（4）ProductId，第三方的商品 ID 号

（5）TimeStamp，时间戳

（6）NonceStr，随机串

（7）AppSignature，参数的加密签名，是根据 2.7 支付签名（paySign）生成方法中所讲的签名方式生成的签名

（8）SignMethod，签名方式，目前只支持“SHA1”，该字段不参与签名

第三方为了确保是来自于微信公众平台的合法请求，需要使用同样的方式生成签名，并与 AppSignature 的值进行对比。

参与签名的字段为：appid、appkey、productid、timestamp、noncestr、openid、issubscribe。

为了返回 Package 数据，回调 URL 必须返回一个 xml 格式的返回数据，形如：

```
<xml>
    <AppId><![CDATA[wwwwb4f85f3a797777]]></AppId>
    <Package><![CDATA[a=1&url=http%3A%2F%2Fwww.qq.com]]></Package>
    <TimeStamp> 1369745073</TimeStamp>
    <NonceStr><![CDATA[iuytxA0cH6PyTAVISB28]]></NonceStr>
    <RetCode>0</RetCode>
    <RetErrMsg><![CDATA[ok]]></ RetErrMsg>
    <AppSignature><![CDATA[53cca9d47b883bd4a5c85a9300df3da0cb48565c]]>
    </AppSignature>
    <SignMethod><![CDATA[sha1]]></ SignMethod >
</xml>
```

其中，AppSignature 依然是根据前文 paySign 所讲的签名方式生成的签名，参与签名的字段为：appid、appkey、package、timestamp、noncestr、retcode、reterrmsg。

package 的生成规则请参考 JS API 所定义的 package 生成规则。这里就不再赘述了。

其中，对于一些第三方觉得商品已经过期或者其他错误的情况，可以在 RetCode 和 RetErrMsg 中体现出来，RetCode 为 0 表明正确，可以定义其他错误；当定义其他错误时，可以在 RetErrMsg 中填上 UTF8 编码的错误提示信息，比如“该商品已经下架”，客户端会直接提示出来。

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。商户可以使用 notify_url 的通知结果进行个性化页面的展示。

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。

目前补单机制的间隔时间为：8s、10s、10s、30s、30s、60s、120s、360s、1000s。

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
接口版本	service_version	否	String(8)	版本号，默认为 1.0
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
密钥序号	sign_key_index	否	Int	多密钥支持的密钥序号，默认 1
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
支付结果信息	pay_info	否	String(64)	支付结果信息，支付成功时为空
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
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 transport_fee + product_fee = 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 transport_fee +

				product_fee=total_fee
折扣价格	discount	否	Int	折扣价格，单位分，如果有值，通知的 total_fee + discount = 请求的 total_fee
买家别名	buyer_alias	否	String(64)	对应买家账号的一个加密串

同时，在 postData 中还将包含 xml 数据。数据如下：

```
<xml>
    <OpenId><![CDATA[111222]]></OpenId>
    <AppId><![CDATA[wwwwb4f85f3a797777]]></AppId>
    <IsSubscribe>1</IsSubscribe>
    <TimeStamp> 1369743511</TimeStamp>
    <NonceStr><![CDATA[jALldRTHAFd5Tgs5]]></NonceStr>
    <AppSignature><![CDATA[bafe07f060f22dcda0bfdb4b5ff756f973aecffa]]>
    </AppSignature>
    <SignMethod><![CDATA[sha1]]></ SignMethod >
</xml>
```

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 2.7 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。**

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

4.5 后台通知签名方式

对于 URL 中签名原始串按以下方式组装成字符串：

a.除 sign 字段外，所有参数按照字段名的 ascii 码从小到大排序后使用 QueryString 的格式（即 key1=value1&key2=value2...）拼接而成字符串 string1，空值不传递，不参与签名组串。

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 string1 进行 urlencode 转码，得到 string2。

d.将 sign=signValue 拼接到 string1 后面得到最终的 notifyargs 字符串。

e.所有参数是指通信过程中实际出现的所有非空参数，即使是接口中无描述的字段，也需要参与签名组串 。

f.**签名原始串中，字段名和字段值都采用原始值，不进行 URL Encode。**

g.**微信通知消息可能会由于升级增加参数，请验证应答签名时注意允许这种情况。**

下面定义了一段生成 notifyargs 字符串的示范过程：

假设以下为 notifyargs 传入参数：

bank_billno=206064184488,
bank_type=0,
discount=0,
fee_type=1,
input_charset=GBK,
notify_id=WE37gwCoFBcAKdkH34Y1nW94r_vao2ljmwE3oAHEeAP690xSVhRleOMfhsgjwVGDpluT-vdS79kbDbkDnjYg4qsmTdSjuJxl,
out_trade_no=843254536943809900,
partner=1900000109,
product_fee=1,
sign_type=MD5,
time_end=20130606015331,
total_fee=1,
trade_mode=1,
trade_state=0,
transaction_id=190000010920130606028255539 7,
transport_fee=0

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

bank_billno=206064184488&bank_type=0&discount=0&fee_type=1&input_charset=GBK&notify_id=WE37gwCoFBcAKdkH34Y1nW94r_vao2ljmwE3oAHEeAP690xSVhRleOMfhsgjwVGDpluT-vdS79kbDbkDnjYg4qsmTdSjuJxl&out_trade_no=843254536943809900&partner=1900000109&product_fee=1&sign_type=MD5&time_end=20130606015331&total_fee=1&trade_mode=1&trade_state=0&transaction_id=190000010920130606028255539 7&transport_fee=0

ii：经过 b 过程后得到 sign 为：

sign=
md5(string1&key=8934e7d15453e97507ef794cf7b0519d).toUpperCase
=md5(bank_billno=206064184488&bank_type=0&discount=0&fee_type=1&input_charset=GBK&notify_id=WE37gwCoFBcAKdkH34Y1nW94r_vao2ljmwE3oAHEeAP690xSVhRleOMfhsgjwVGDpluT-vdS79kbDbkDnjYg4qsmTdSjuJxl&out_trade_no=843254536943809900&partner=1900000109&product_fee=1&sign_type=MD5&time_end=20130606015331&total_fee=1&trade_mode=1&trade_state=0&transaction_id=190000010920130606028255539 7&transport_fee=0&key=8934e7d15453e97507ef794cf7b0519d).toUpperCase()
="8ef1f69d5d9d4ec39d3787526f27924e".toUpperCase()
="8EF1F69D5D9D4EC39D3787526F27924E"

iii：再对 string1 经过 c 过程 urlencode 编码后得到：

bank_billno=206064184488&bank_type=0&discount=0&fee_type=1&input_charset=GBK&notify_id=WE37gwCoFBcAKdkH34Y1nW94r_vao2ljmwE3oAHEeAP690xSVhRleOMfhsgjwVGDpluT-vdS79kbDbkDnjYg4qsmTdSjuJxl&out_trade_no=84325453694380990 0&partner=1900000109&product_fee=1&sign_type=MD5&time_end=20130606015331&total_fee=1&trade_mode=1&trade_state=0&transaction_id=1900000109201306060282555397&transport_fee=0

iv：拼接上 sign 后得到最终 package 结果：

bank_billno=206064184488&bank_type=0&discount=0&fee_type=1&input_charset=GBK&notify_id=WE37gwCoFBcAKdkH34Y1nW94r_vao2ljmwE3oAHEeAP690xSVhRleOMfhsgjwVGDpluT-vdS79kbDbkDnjYg4qsmTdSjuJxl&out_trade_no=84325453694380990 0&partner=1900000109&product_fee=1&sign_type=MD5&time_end=20130606015331&total_fee=1&trade_mode=1&trade_state=0&transaction_id=1900000109201306060282555397&transport_fee=0&sign=8EF1F69D5D9D4EC39D3787526F27924E

5.API 接口说明

5.1API 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。

5.2API 使用方式

在使用 API 之前，需要拥有 API 使用过程中用到的具有时效性的凭证 access_token。

这个获取方式就是使用前文提到的 appid 和 appsecret 调用 token 这个 api 获取。更详细的文档请参考：

http://mp.weixin.qq.com/wiki/index.php?title=%E8%8E%B7%E5%8F%96access_token 说明文档中的通用接口。

只有拥有了有效 access_token（即相当于具有了一定时间内的 API 登录态），后续的所有 API 调用才会成功。

5.3API 列表

5.3.1 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" : "wwwwb4f85f3a797777",
    "openid" : "oX99MDgNcgwnz3zFN3DNmo8uwa-w",
    "transid" : "111112222233333",
    "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。

5.3.2 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" : "wwwwb4f85f3a797777",
    "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":"190000010920130702030577374l",
        "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。

5.3.3 告警通知

为了及时通知商户异常，提高商户在微信平台的服务质量。微信后台会向商户推送告警通知，包括发货延迟、调用失败、通知失败等情况，通知的地址是商户在申请支付时填写的告警通知 URL，在“公众平台-服务-服务中心-商户功能-商户基本资料-告警通知 URL”可以查看。**商户接收到告警通知后请尽快修复其中提到的问题，以免影响线上经营。**（发货时间要求请参考 5.3.1）

告警通知 URL 接收的 postData 中还将含 xml 数据，格式如下：

```
<xml>
    <AppId><![CDATA[wxf8b4f85f3a794e77]]></AppId>
    <ErrorType>1001</ErrorType>
    <Description><![CDATA[错误描述]]></Description>
    <AlarmContent><![CDATA[错误详情]]></AlarmContent>
    <TimeStamp>1393860740</TimeStamp>
    <AppSignature><![CDATA[f8164781a303f4d5a944a2dfc68411a8c7e4fbea]]></AppSignature>
    <SignMethod><![CDATA[sha1]]></SignMethod>
</xml>
```

错误描述

ErrorType	Description	AlarmContent
1001	发货超时	transaction_id=XXXXX

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：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。

（2） 支付授权目录如何使用？

答：支付授权目录是支付功能正式上线后，商户后台发起支付请求的页面所在的目录。这个目录在注册填写时，需要精确到最细一级的目录，且在使用时，目录名称后直接加文件名，不可再增加 or 删减目录。

举例：发起请求的页面 url 为 http://pay.weixin.com/weixin/pay/payment.php?XXXXX，则填写的目录应该为 http://pay.weixin.com/weixin/pay/；反之，若填写的目录为 http://pay.weixin.com/weixin/pay/，则发起请求的页面 url 只能在目录后增加文件名，不可增减、修改任何目录。

（3） 支付测试目录和支付授权目录？

答：支付授权目录将会在产品上线审核时，以及上线后长期使用的正式目录。支付测试目录是提供给开发者，在开发测试期间使用的临时目录。这两个目录都是发起支付请求

的页面文件所在的位置。

6.3 常见错误现象及解决方法

（1） 点击支付按钮，调用 JS API 没反应？

答：尝试发起支付的页面 url，不在支付授权目录下，请检查 url 与支付授权目录是否对应。

（2） 点击支付按钮，提示“access_not_allow”

答：参与测试人员的微信号没有在白名单中，将测试用户加入白名单。操作在“mp.weixin.qq.com——功能——商户功能——支付测试”。

（3） 点击支付按钮，提示“access_denied”

答：尝试发起支付的页面 url，不在支付授权目录下，请检查 url 与支付授权目录是否对应。

（4） 点击支付页面链接后，没有反应？

答：在开发调试阶段，测试链接需要在公众号内点击打开。操作方法可以是：白名单用户在公众号内向公众号发一条消息，消息内容即为测试链接，然后点击打开。

（5） 点击支付按钮，提示“当前公众号没有权限支付本次交易”

答：请确认使用的 APPID 是否正确，确认在 MP 平台前三项审核结果均为“审核通过”。

（6） 点击支付按钮，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 MP 平台前三项审核结果均为“审核通过”。

（7） 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。

（8） 点击支付按钮，提示“该公众号支付签名无效，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。

（9） 点击支付按钮，大多数情况正常，只是很偶尔情况提示签名错误

答：有可能某些运营商会直接修改请求中的 IP，最保险的办法是“商户服务器向 H5页面传 package 时，将 ip 写成整数或其他非 ip 形式，H5接到后，再换成正规的 ip 地址”**（注意，只有在传输的时候需要改，其余时候均使用标准正规 ip 格式）**

（10） 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。

（11） 用户成功支付，点击“完成”，又再次跳转至输入密码页面，仍可支付并二次扣费

答：用户在商户的 H5 页面点击了两次“微信支付”，生成了两笔订单，需要在 H5 页面微信支付按钮上增加防二次点击的机制。

（12） 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。

（13） 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 最新接口文档下载

https://mp.weixin.qq.com/cgi-bin/readtemplate?t=business/course2_tmpl#1

6.6 联系我们

微信支付开发者 QQ 群：271381735