

微信 SDK 统一支付接口文档

文档版本：2.1.2

接口版本：2.0

目录

1	引言.....	3
1.1	文档概述.....	3
1.2	阅读对象.....	3
1.3	业务术语.....	3
2	方案概述.....	4
2.1	行业背景.....	4
2.2	业务实现流程.....	4
3	数据格式.....	6
3.1	提交数据.....	6
3.2	XML 数据格式.....	6
4	数字签名.....	7
4.1	签名原始串.....	7
4.2	签名算法.....	8
5	补单机制.....	9
6	统一预下单支付接口.....	10
6.1	预下单接口（非原态）.....	10
6.1.1	业务功能.....	10
6.1.2	交互模式.....	10
6.1.3	请求参数列表.....	10
6.1.4	返回结果.....	11
6.2	预下单接口（原生态）.....	12
6.2.1	业务功能.....	12
6.2.2	交换模式.....	12
6.2.3	请求参数列表.....	12
6.2.4	返回结果.....	14
6.3	支付通知接口.....	15
6.3.1	通知结果参数列表.....	15
6.3.2	后台通知结果反馈.....	16
7	查询退款接口.....	17
7.1	订单查询接口.....	17
7.1.1	业务功能.....	17
7.1.2	交互模式.....	17
7.1.3	请求参数列表.....	17
7.1.4	返回结果.....	18
7.2	退款接口.....	20
7.2.1	业务功能.....	20
7.2.2	交互模式.....	20
7.2.3	请求参数列表.....	20
7.2.4	返回结果.....	21
7.3	退款查询接口.....	22
7.3.1	请求参数列表.....	22
7.3.2	返回结果.....	23

8 关闭订单接口..... 24

8.1 关闭订单接口..... 24

8.1.1 业务功能..... 24

8.1.2 交互模式..... 24

8.1.3 请求参数列表..... 24

8.1.4 返回结果..... 25

9 注意事项..... 26

1 引言

1.1 文档概述

本文件针对移动支付行业提供接口解决方案。供内部相关开发和技术支持人员与商户平台服务方技术人员使用。文档分别从交互模式、签名、接口、注意事项等方面详细介绍了接口的开发流程，可以帮助开发人员快速入门并掌握开发技能，同时也可以作为日后接口参数以及参数类型的速查手册。

1.2 阅读对象

供商户平台服务方技术或业务人员参考和查询

1.3 业务术语

术语	值	说明
测试商户号	755437000006	商户调用接口时可用此测试商户号对应参数 mch_id
测试密钥	7daa4babae15ae17eee90c9e	为保证通讯不被篡改,平台与商户之间约定的 32 位或 24 位字符串，算签名 sign 时使用
商户 appid	wx2a5538052969956e	
金额	1000	金额，默认为 RMB，以分为单位。1000 表示 RMB10.00。测试商户号最大限额 100 分/笔（即 1 元/笔）

2 方案概述

2.1 行业背景

用于商户在移动端 APP 中集成微信支付功能。

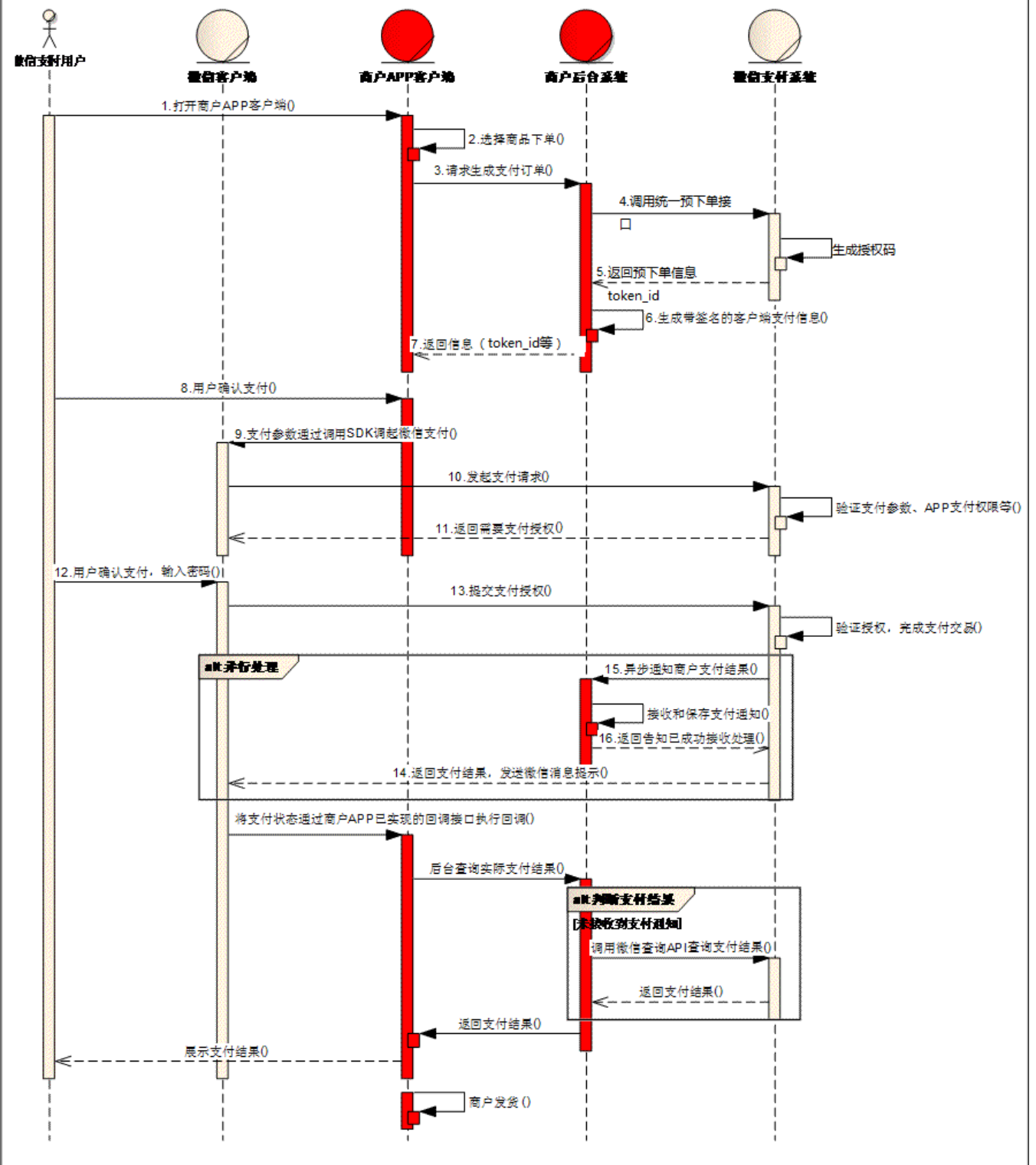
商户 APP 调用 SDK 微信支付模块，商户 APP 会跳转到微信中完成支付，支付完后跳回到商户 APP 内，最后展示支付结果。

2.2 业务实现流程

接口调用时序：

统一预下单 API、支付结果通知 API 和查询订单 API 等都涉及签名过程，调用都必须在商户服务器端完成。
时序图如下：

sd APP支付模式



3 数据格式

3.1 提交数据

采用 HTTPS 标准的 POST 协议，为了保证接收方接收数据正确，传输数据必须签名。

```
<xml>
<service>pay.weixin.native </service>
<attach><![CDATA[att]]></attach>
<body><![CDATA[支付测试]]></body>
<device_info>1000</device_info>
<mch_id>10000100</mch_id>
<nonce_str>adf880d5c8986bd0deb6423c92c9d948</nonce_str>
<out_trade_no>1406046836</out_trade_no>
<spbill_create_ip>127.0.0.1</spbill_create_ip>
<total_fee>1</total_fee>
<sign><![CDATA[F53145E553092CE52E4CAA4D2B49A91C]]></sign>
</xml>
```

3.2 XML 数据格式

采用标准 XML 协议，所有参数只存在一级节点中，不采用多级节点嵌套。

协议级错误返回：

```
<xml>
<status>500</status>
<message><![CDATA[SYSERR]]></message>
</xml>
```

正确返回数据：

```
<xml>
<status>0</status>
<message><![CDATA[OK]]></message>
<appid><![CDATA[wX2421b1c4370ec43b]]></appid>
<mch_id><![CDATA[10000100]]></mch_id>
<device_info><![CDATA[1000]]></device_info>
<nonce_str><![CDATA[FvYsnPuFFPkAr77M]]></nonce_str>
<sign><![CDATA[63238039D6E43634297CF2A6EB5F3B72]]></sign>
<result_code>0</result_code>
<openid><![CDATA[oUpF8uN95-Ptaags6E_roPHg7AG0]]></openid>
<is_subscribe><![CDATA[Y]]></is_subscribe>
```

```
<trade_type><![CDATA[MICROPAY]]></trade_type>
<bank_type><![CDATA[CCB_CREDIT]]></bank_type>
<total_fee>1</total_fee>
<coupon_fee>0</coupon_fee>
<fee_type><![CDATA[CNY]]></fee_type>
<transaction_id><![CDATA[1008450740201407220000058756]]></transaction_id>
<out_trade_no><![CDATA[1406033828]]></out_trade_no>
<attach><![CDATA[att]]></attach>
<time_end><![CDATA[20140722160655]]></time_end>
</xml>
```

业务级错误返回：

```
<xml>
<status>0</status>
<message><![CDATA[OK]]></message>
<appid><![CDATA[wx2421b1c4370ec43b]]></appid>
<mch_id><![CDATA[10000100]]></mch_id>
<device_info><![CDATA[1000]]></device_info>
<nonce_str><![CDATA[sthBJ9QyUG6vkrjJ]]></nonce_str>
<sign><![CDATA[6277A96D7875D4FF23AA7B6A4C3046AB]]></sign>
<result_code>1</result_code>
<err_code><![CDATA[AUTHCODE_EXPIRE]]></err_code>
<err_code_des><![CDATA[二维码已过期，请刷新再试]]></err_code_des>
</xml>
```

一般有返回有 status 参数，0 表示调用成功；非 0 表示调用失败。

4 数字签名

为了保证数据传输过程中的数据真实性和完整性，我们需要对数据进行数字签名，在接收签名数据之后进行签名校验。

数字签名有两个步骤，先按一定规则拼接要签名的原始串，再选择具体的算法和密钥计算出签名结果。

一般失败的结果不签名。

4.1 签名原始串

无论是请求还是应答，签名原始串按以下方式组装成字符串：

- 1、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，空值不传递，不参与签名组串。
- 2、签名原始串中，字段名和字段值都采用原始值，不进行 URL Encode。

3、平台返回的应答或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。

举例：

调用某个接口，接口有如下字段：

```
<xml>
<body><![CDATA[测试支付]]></body>
<mch_create_ip><![CDATA[127.0.0.1]]></mch_create_ip>
<mch_id><![CDATA[755437000006]]></mch_id>
<nonce_str><![CDATA[1409196838]]></nonce_str>
<notify_url><![CDATA[http://227.0.0.1:9001/javak/]]></notify_url>
<out_trade_no><![CDATA[141903606228]]></out_trade_no>
<service><![CDATA[unified.trade.pay]]></service>
<sign><![CDATA[6DD83E271779D6D885748A2C2A4D9CFD]]></sign>
<total_fee><![CDATA[1]]></total_fee>
</xml>
```

正确的签名字段排序为：

```
body=测试支付
&mch_create_ip=127.0.0.1&mch_id=755437000006&nonce_str=1409196838&notify_url=CD
ATA[http://227.0.0.1:9001/javak/&out_trade_no=141903606228&service=unified.trad
e.pay&total_fee=1
```

4.2 签名算法

目前暂只支持 MD5 签名

MD5 签名

MD5 是一种摘要生成算法，通过在签名原始串后加上商户通信密钥的内容，进行 MD5 运算，形成的摘要字符串即为签名结果。为了方便比较，签名结果统一转换为大写字符。

注意：签名时将字符串转化成字节流时指定的编码字符集应与参数 **charset** 一致。

MD5 签名计算公式：

$$\text{sign} = \text{Md5}(\text{原字符串} \& \text{key} = \text{商户密钥}). \text{toUpperCase}$$

如：

假设以下为 XML 传入参数：

```
<xml>
<body><![CDATA[测试支付]]></body>
<mch_create_ip><![CDATA[127.0.0.1]]></mch_create_ip>
<mch_id><![CDATA[755437000006]]></mch_id>
<nonce_str><![CDATA[1409196838]]></nonce_str>
<notify_url><![CDATA[http://227.0.0.1:9001/javak/]]></notify_url>
<out_trade_no><![CDATA[141903606228]]></out_trade_no>
<service><![CDATA[unified.trade.pay]]></service>
```



```
<sign><![CDATA[6DD83E271779D6D885748A2C2A4D9CFD]]></sign>
<total_fee><![CDATA[1]]></total_fee>
</xml>
```

假设商户密钥为：7daa4babae15ae17eee90c9e

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

body=测试支付

&mch_create_ip=127.0.0.1&mch_id=755437000006&nonce_str=1409196838¬ify_url=http://227.0.0.1:9001/javak/&out_trade_no=141903606228&service=unified.trade.pay&total_fee=1

ii: 经过 b 过程后得到 sign 为:

sign

=md5(string1&key=7daa4babae15ae17eee90c9e).toUpperCase

=md5(body=测试支付

&mch_create_ip=127.0.0.1&mch_id=755437000006&nonce_str=1409196838¬ify_url=http://227.0.0.1:9001/javak/&out_trade_no=141903606228&service=unified.trade.pay&total_fee=1&key=

7daa4babae15ae17eee90c9e).toUpperCase()

="6DD83E271779D6D885748A2C2A4D9CFD"

5 补单机制

注意：对后台通知交互模式，如果平台收到商户的应答不是**纯字符串 success**或**超过 5 秒**后返回时，平台认为通知失败。平台会通过一定的策略（通知频率为 0/15/15/30/180/1800/1800/1800/1800/3600，单位：秒）**间接性**重新发起通知，尽可能提高通知的成功率，但平台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。平台推荐的做法是，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**纯字符串 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**重复插入数据**造成的数据混乱。

6 统一预下单支付接口

6.1 预下单接口（非原态）

6.1.1 业务功能

初始化支付请求，通过该请求得到支持的支付类型和授权码。

6.1.2 交互模式

请求：后台请求交互模式

返回结果+通知：后台请求交互模式+后台通知交互模式

6.1.3 请求参数列表

请求 url: <https://pay.swiftpass.cn/pay/gateway>

POST XML 内容体进行请求

字段名	变量名	必填	类型	说明
业务参数				
接口类型	service	是	String(32)	接口类型：unified.trade.pay
版本号	version	否	String(8)	版本号，version 默认值是 2.0
字符集	charset	否	String(8)	可选值 UTF-8 ，默认为 UTF-8
签名方式	sign_type	否	String(8)	签名类型，取值：MD5 默认：MD5
商户号	mch_id	是	String(32)	商户号，由平台分配
商户订单号	out_trade_no	是	String(32)	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
设备号	device_info	否	String(32)	终端设备号
商品描述	body	是	String(127)	商品描述
附加信息	attach	否	String(127)	附加数据，在查询接口和支付通知中原样返回，该字段主要用于商户携带订单的自定义

				数据
总金额	total_fee	是	Int	总金额，以分为单位，不允许包含任何字、符号
终端 IP	mch_create_ip	是	String(16)	订单生成的机器 IP
通知地址	notify_url	是	String(255)	接收平台通知的 URL，需给绝对路径，255 字符 内 格 式 如:http://wap.tenpay.com/tenpay.asp
限制信用卡	limit_credit_pay	否	String(1)	限定用户使用微信支付时能否使用信用卡，值为 1，禁用信用卡，值为 0 或者不传此参数则不禁用
订单生成时间	time_start	否	String(14)	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。注：订单生成时间与超时时间需要同时传入才会生效。
订单超时时间	time_expire	否	String(14)	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。注：订单生成时间与超时时间需要同时传入才会生效。
随机字符串	nonce_str	是	String(32)	随机字符串，不长于 32 位
签名	sign	是	String(32)	MD5 签名结果，详见“第 4 章 数字签名”
商户 APP 应用 ID	sub_appid	是	String(32)	微信开放平台审核通过的移动应用 AppID
商品标识	goods_tag	否	String(32)	商品标识，微信平台配置的商品标识，用于优惠券或者满减使用

6.1.4 返回结果

数据按 XML 的格式实时返回

字段名	变量名	必填	类型	说明
版本号	version	是	String(8)	版本号，version 默认值是 2.0。
字符集	charset	是	String(8)	可选值 UTF-8，默认为 UTF-8。

签名方式	sign_type	是	String(8)	签名类型，取值：MD5 默认：MD5
返回状态码	status	是	String(16)	0 表示成功，非 0 表示失败
返回信息	message	否	String(128)	返回信息，如非空，为错误原因签名失败参数格式校验错误
以下字段在 status 为 0 的时候有返回				
支持的支付类型	services	是	String	支持的支付类型，多个以“ ”连接
授权码	token_id	是	String (64)	支付授权码
签名	sign	是	String(32)	MD5 签名结果，详见“第 4 章 MD5 签名规则”

6.2 预下单接口（原生态）

6.2.1 业务功能

用户在商家 app 内选择微信支付下单成功后，商家通过接口获取到微信支付的参数信息，然后在商家 app 内唤起微信支付，用户在微信内确认支付后会自动返回到用户 app 内。

6.2.2 交换模式

请求：后台请求交互模式
返回结果：后台通知交互模式

6.2.3 请求参数列表

请求 url：https://pay.swiftpass.cn/pay/gateway
POST XML 内容体进行请求

字段名	变量名	必填	类型	说明
接口类型	service	是	String(32)	接口类型：pay.weixin.raw.app
版本号	version	否	String(8)	版本号，version 默认值是 2.0。
字符集	charset	否	String(8)	可选值 UTF-8 ，默认为 UTF-8。

签名方式	sign_type	否	String(8)	签名类型，取值：MD5 默认：MD5
商户号	mch_id	是	String(32)	商户号，由平台分配
appid	appid	是	String(32)	商户 app 对应的微信开放平台移动应用 APPID
商户订单号	out_trade_no	是	String(32)	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
设备号	device_info	否	String(32)	终端设备号
商品描述	body	是	String(127)	商品描述
附加信息	attach	否	String(127)	商品附加信息，可做扩展参数
总金额	total_fee	是	Int	总金额，以分为单位，不允许包含任何字、符号
终端 I P	mch_create_ip	是	String(16)	订单生成的机器 IP
通知地址	notify_url	是	String(255)	接收平台通知的 URL，需给绝对路径，255 字符内格式，确保平台能通过互联网访问该地址
订单生成时间	time_start	否	String(14)	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+8Beijing。该时间取自商户服务器。增加 注：订单生成时间与超时时间需要同时传入才会生效。
订单超时时间	time_expire	否	String(14)	订单超时时间，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+8Beijing。该时间取自商户服务器。增加 注：订单生成时间与超时时间需要同时传入才会生效。
操作员	op_user_id	否	String(32)	操作员帐号, 默认为商户号
门店编号	op_shop_id	否	String(32)	
限制信用卡	limit_credit_pay	否	String(1)	限定用户使用微信支付时能否使用信用卡，值为 1，禁用信用卡，值为 0 或者不传此参数则不禁用
商品标记	goods_tag	否	String(32)	商品标记
随机字符串	nonce_str	是	String(32)	随机字符串，不长于 32 位
商户 APP 应用 ID	sub_appid	是	String(32)	微信开放平台审核通过的移动应用 AppID
签名	sign	是	String(32)	MD5 签名结果，详见“安全规范”

6.2.4 返回结果

数据按 XML 格式实时返回

字段名	变量名	必填	类型	说明
版本号	version	是	String(32)	版本号，version 默认值是 2.0。
字符集	charset	是	String(8)	可选值 UTF-8 ，默认为 UTF-8。
签名方式	sign_type	否	String(8)	签名类型，取值：MD5 默认：MD5
返回状态码	status	是	String(8)	0 表示交易成功，非 0 表示失败，此字段是通信标识，非交易标识，交易是否成功需要看你 result_code 来判断
返回信息	message	否	String(32)	返回信息，如非空，为错误原因签名失败参数格式校验错误
以下字段在 status 为 0 的时候有返回				
业务结果	result_code	是	String(16)	0 表示成功，非 0 表示失败
商户号	mch_id	是	String(32)	商户号，由平台分配
设备号	device_info	否	String(32)	平台支付分配的终端设备号
随机字符串	nonce_str	是	String(32)	随机字符串，不长于 32 位
错误代码	err_code	否	String(32)	具体错误码请看错误码列表
错误代码描述	err_msg	否	String(128)	结果信息描述
签名	sign	是	String(32)	MD5 签名结果，详见“安全规范”
以下字段在 status 和 result_code 都为 0 的时候有返回				
支付信息	pay_info	是	String	Json 格式的字符串，微信官方 SDK 中需要的参数信息
商户订单号	out_trade_no	是	String	商户订单号
平台订单号	transaction_id	是	String	平台订单号

6.3 支付通知接口

6.3.1 通知结果参数列表

通知 URL 是 6.1 节中提交的参数 `notify_url`，支付完成后，平台会把相关支付和用户信息发送到该 URL，商户需要接收处理信息。对后台通知交互时，如果平台收到商户的应答不是**纯字符串 success** 或**超过 5 秒**后返回时，平台认为通知失败。平台会通过一定的策略（**通知频率为 0/15/15/30/180/1800/1800/1800/1800/3600，单位：秒**）**间接性**重新发起通知，尽可能提高通知的成功率，但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。**

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结果成功。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。

特别注意：商户需对接收到的通知参数中订单号和金额与自身业务系统中的单号和金额对比校验，校验一致后方可更新自身系统订单状态

后台通知通过请求中的 `notify_url` 进行，post 给商户系统（通知参数内容为 xml 的字符串）

字段名	变量名	必填	类型	说明
版本号	version	是	String(8)	版本号，version 默认值是 2.0。
字符集	charset	是	String(8)	可选值 UTF-8，默认为 UTF-8。
签名方式	sign_type	否	String(8)	签名类型，取值：MD5 默认：MD5
返回状态码	status	是	String(16)	0 表示成功，非 0 表示失败 此字段是通信标识，非交易标识，交易是否成功需要查看 result_code 来判断
返回信息	message	否	String(128)	返回信息，如非空，为错误原因签名失败参数格式校验错误
以下字段在 status 为 0 的时候有返回				
业务结果	result_code	是	String(16)	0 表示成功，非 0 表示失败
商户号	mch_id	是	String(32)	商户号，由平台分配

设备号	device_info	否	String(32)	终端设备号
随机字符串	nonce_str	是	String(32)	随机字符串，不长于 32 位
错误代码	err_code	否	String(32)	参考错误码
错误代码描述	err_msg	否	String (128)	结果信息描述
签名	sign	是	String(32)	MD5 签名结果，详见“第 4 章 MD5 签名规则”
以下字段在 status 和 result_code 都为 0 的时候有返回				
用户标识	openid	是	String(128)	用户在商户 appid 下的唯一标识
交易类型	trade_type	是	String(32)	pay.weixin.app
支付结果	pay_result	是	Int	支付结果：0—成功；其它—失败
支付结果信息	pay_info	否	String(64)	支付结果信息，支付成功时空
平台订单号	transaction_id	是	String(32)	平台交易号
第三方支付单	out_transaction_id	否	String(32)	如：微信支付单号，支付宝支付单号
商户订单号	out_trade_no	是	String(32)	商户系统内部的定单号，32 个字符内、可包含字母
总金额	total_fee	是	Int	总金额，以分为单位，不允许包含任何字、符号
货币种类	fee_type	否	String(8)	货币类型，符合 ISO 4217 标准的三位字母代码，默认人民币：CNY
附加信息	attach	否	String(127)	商家数据包，原样返回预下单时自定义数据
付款银行	bank_type	是	String(16)	银行类型
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。该时间取自平台服务器

6.3.2 后台通知结果反馈

平台服务器发送通知，post 发送 XML 数据流，商户 notify_url 地址接收通知结果，接收方法 demo 有写（如 php 中的 callback 方法，c#中的 notify.aspx 文件，java 中的 TestPayResultServlet 方法），商户做业务处理后，需要以纯字符串的形式反馈处理结果，内容如下：

	结果说明
--	------

返回结果	
success	处理成功，平台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，平台收到此结果或者没有收到任何结果，系统通过补单机制（详见第 5 节）再次通知

7 查询退款接口

7.1 订单查询接口

7.1.1 业务功能

根据商户订单号或者平台订单号查询具体订单信息。

7.1.2 交互模式

后台系统调用交互模式

7.1.3 请求参数列表

请求 url: <https://pay.swiftpass.cn/pay/gateway>

通过 POST XML 内容体进行请求

字段名	变量名	必填	类型	说明
接口类型	service	是	String(32)	接口类型: unified.trade.query
版本号	version	否	String(8)	版本号, version 默认值是 2.0。
字符集	charset	否	String(8)	可选值 UTF-8 , 默认为 UTF-8。
签名方式	sign_type	否	String(8)	签名类型, 取值: MD5 默认: MD5
商户号	mch_id	是	String(32)	商户号, 由平台分配
商户订单号	out_trade_no	否	String(32)	商户系统内部的订单号, out_trade_no 和 transaction_id 至少一个必填, 同时存在时 transaction_id 优先

平台订单号	transaction_id	否	String(32)	平台交易号，out_trade_no 和 transaction_id 至少一个必填，同时存在时 transaction_id 优先。
随机字符串	nonce_str	是	String(32)	随机字符串，不长于 32 位
签名	sign	是	String(32)	MD5 签名结果，详见“第 4 章 MD5 签名规则”

7.1.4 返回结果

数据按 XML 的格式实时返回

字段名	变量名	必填	类型	说明
版本号	version	是	String(8)	版本号，version 默认值是 2.0。
字符集	charset	是	String(8)	可选值 UTF-8，默认为 UTF-8。
签名方式	sign_type	是	String(8)	签名类型，取值：MD5 默认：MD5
返回状态码	status	是	String(16)	0 表示成功，非 0 表示失败 此字段是通信标识，非交易标识，交易是否成功需要查看 trade_state 来判断
返回信息	message	否	String(128)	返回信息，如非空，为错误原因签名失败参数格式校验错误
以下字段在 status 为 0 的时候有返回				
业务结果	result_code	是	String(16)	0 表示成功，非 0 表示失败
商户号	mch_id	是	String(32)	商户号，由平台分配
设备号	device_info	否	String(32)	终端设备号
随机字符串	nonce_str	是	String(32)	随机字符串，不长于 32 位
错误代码	err_code	否	String(32)	参考错误码
错误代码描述	err_msg	否	String (128)	结果信息描述
签名	sign	是	String(32)	MD5 签名结果，详见“第 4 章 MD5 签名规则”
以下字段在 status 和 result_code 都为 0 的时候有返回				
交易状态	trade_state	是	String(32)	SUCCESS—支付成功 REFUND—转入退款

				NOTPAY—未支付 CLOSED—已关闭 PAYERROR—支付失败(其他原因, 如银行返回失败)
以下字段在 trade_state 为 SUCCESS 的时候有返回				
交易类型	trade_type	是	String(32)	pay.weixin.native
公众号 appid	appid	是	String(32)	服务商公众号 appid
用户标识	openid	是	String(128)	用户在服务商 appid 下的唯一标识
是否关注公众账号	is_subscribe	是	String(1)	用户是否关注服务商公众账号, Y-关注, N-未关注
平台订单号	transaction_id	是	String(32)	平台订单号
第三方支付单	out_transaction_id	否	String(32)	如: 微信支付单号, 支付宝支付单号
商户订单号	out_trade_no	是	String(32)	商户系统内部的定单号, 32 个字符内、可包含字母
总金额	total_fee	是	Int	总金额, 以分为单位, 不允许包含任何字、符号
现金券金额	coupon_fee	否	Int	现金券支付金额<=订单总金额, 订单总金额-现金券金额为现金支付金额
货币种类	fee_type	是	String(8)	货币类型, 符合 ISO 4217 标准的三位字母代码, 默认人民币: CNY
附加信息	attach	否	String(127)	商家数据包, 原样返回预下单时自定义数据
付款银行	bank_type	是	String(16)	银行类型
银行订单号	bank_billno	否	String(32)	银行订单号, 若为微信支付则为空
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。该时间取自平台服务器

7.2 退款接口

7.2.1 业务功能

商户针对某一个已经成功支付的订单发起退款，操作结果在同一会话中同步返回。

一、退款方式

目前只支持原路返回退款与退款至微信余额账户两种。

说明：退到银行卡则是非实时的，每个银行的处理速度不同，一般发起退款后 1-3 个工作日内到账。

同一笔单的部分退款需要设置相同的订单号和不同的 out_refund_no。一笔退款失败后重新提交，要采用原来的 out_refund_no。总退款金额不能超过用户实际支付金额(现金券金额不能退款)

二、退款限制

商户在退款操作时应该注意退款限制，避免发起不会成功的退款请求，下面是主要的退款限制：

- 1、 在平台中，只要退款累计金额不超过交易单支付总额，一笔交易单可以多次退款，退款申请单号（退款接口中有此参数）唯一确定一次退款，而不是交易单号确定一次退款。退款申请单号由商户生成，所以商户一定要保证退款申请单的唯一性。商家在退款过程中要特别注意，只有在能确定退款失败的情况下，才能重新发起另一笔退款。
- 2、 请求频率限制：150qps，即每秒钟正常的申请退款请求次数不超过 150 次。
- 3、 错误或无效请求频率限制：6qps，即每秒钟异常或错误的退款申请请求不超过 6 次。
- 4、 每个支付订单的部分退款次数不能超过 50 次。

7.2.2 交互模式

后台系统调用交互模式

7.2.3 请求参数列表

请求url: <https://pay.swiftpass.cn/pay/gateway>

POST XML 内容体进行请求

字段名	变量名	必填	类型	说明
接口类型	service	是	String(32)	接口类型：unified.trade.refund
版本号	version	否	String(8)	版本号，version 默认值是 2.0。
字符集	charset	否	String(8)	可选值 UTF-8，默认为 UTF-8。
签名方式	sign_type	否	String(8)	签名类型，取值：MD5 默认：MD5
商户号	mch_id	是	String(32)	商户号，由平台分配
商户订单号	out_trade_no	否	String(32)	商户系统内部的订单号，out_trade_no 和 transaction_id 至少一个必填，同时存在时 transaction_id 优先

平台订单号	transaction_id	否	String(32)	平台订单号, out_trade_no 和 transaction_id 至少一个必填, 同时存在时 transaction_id 优先
商户退款单号	out_refund_no	是	String(32)	商户退款单号, 32 个字符内、可包含字母, 确保在商户系统唯一。同个退款单号多次请求, 平台当一个单处理, 只会退一次款。如果出现退款不成功, 请采用原退款单号重新发起, 避免出现重复退款。
总金额	total_fee	是	Int	订单总金额, 单位为分
退款金额	refund_fee	是	Int	退款总金额, 单位为分, 可以做部分退款
操作员	op_user_id	是	String(32)	操作员帐号, 默认为商户号
退款渠道	refund_channel	否	String(16)	ORIGINAL-原路退款, 默认
随机字符串	nonce_str	是	String(32)	随机字符串, 不长于 32 位
签名	sign	是	String(32)	MD5 签名结果, 详见“第 4 章 MD5 签名规则”

7.2.4 返回结果

数据按 XML 的格式实时返回

字段名	变量名	必填	类型	说明
版本号	version	是	String(8)	版本号, version 默认值是 2.0。
字符集	charset	是	String(8)	可选值 UTF-8, 默认为 UTF-8。
签名方式	sign_type	是	String(8)	签名类型, 取值: MD5 默认: MD5
返回状态码	status	是	String(16)	0 表示成功, 非 0 表示失败 此字段是通信标识, 非交易标识, 交易是否成功需要查看 result_code 来判断
返回信息	message	否	String(128)	返回信息, 如非空, 为错误原因签名失败参数格式校验错误
以下字段在 status 为 0 的时候有返回				
业务结果	result_code	是	String(16)	0 表示成功, 非 0 表示失败 注: 此处返回 0 表示退款申请接收成功, 实际的退款结果根据退款查询接口查询
商户号	mch_id	是	String(32)	商户号, 由平台分配
设备号	device_info	否	String(32)	终端设备号
随机字符串	nonce_str	是	String(32)	随机字符串, 不长于 32 位
错误代码	err_code	否	String(32)	参考错误码
签名	sign	是	String(32)	MD5 签名结果, 详见“第 4 章 MD5 签名规则”
以下字段在 status 和 result_code 都为 0 的时候有返回				

平台订单号	transaction_id	是	String(32)	平台交易号
商户订单号	out_trade_no	是	String(32)	商户系统内部的订单号
商户退款单号	out_refund_no	是	String(32)	商户退款单号
平台退款单号	refund_id	是	String(32)	平台退款单号
退款渠道	refund_channel	是	String(16)	ORIGINAL—原路退款，默认
退款金额	refund_fee	是	Int	退款总金额,单位为分,可以做部分退款
现金券退款金额	coupon_refund_fee	否	Int	现金券退款金额 <= 退款金额， 退款金额-现金券退款金额为现金

7.3 退款查询接口

提交退款申请后， 通过调用该接口查询退款状态。 退款有一定延时， 请在 3 个工作日后重新查询退款状态。

7.3.1 请求参数列表

请求url: <https://pay.swiftpass.cn/pay/gateway>

POST XML 内容体进行请求

字段名	变量名	必填	类型	说明
接口类型	service	是	String(32)	接口类型: unified.trade.refundquery
版本号	version	否	String(8)	版本号, version 默认值是 2.0。
字符集	charset	否	String(8)	可选值 UTF-8 , 默认为 UTF-8。
签名方式	sign_type	否	String(8)	签名类型, 取值: MD5 默认: MD5
商户号	mch_id	是	String(32)	商户号, 由平台分配
商户订单号	out_trade_no	否	String(32)	商户系统内部的订单号, out_trade_no 和 transaction_id 至少一个必填, 同时存在时 transaction_id 优先
平台订单号	transaction_id	否	String(32)	平台订单号, out_trade_no 和 transaction_id 至少一个必填, 同时存在时 transaction_id 优先
商户退款单号	out_refund_no	否	String(32)	商户退款单号, 32 个字符内、可包含字母, 确保在商户系统唯一。
平台退款单号	refund_id	否	String(32)	平台退款单号 refund_id、out_refund_no、out_trade_no 、transaction_id 四个参数必填一个, 如果同时存在优先级为:

				refund_id>out_refund_no>transaction_id>out_trade_no
随机字符串	nonce_str	是	String(32)	随机字符串，不长于 32 位
签名	sign	是	String(32)	MD5 签名结果，详见“第 4 章 MD5 签名规则”

7.3.2 返回结果

数据按 XML 的格式实时返回

字段名	变量名	必填	类型	说明
版本号	version	是	String(8)	版本号，version 默认值是 2.0。
字符集	charset	是	String(8)	可选值 UTF-8，默认为 UTF-8。
签名方式	sign_type	是	String(8)	签名类型，取值：MD5 默认：MD5
返回状态码	status	是	String(16)	0 表示成功，非 0 表示失败 此字段是通信标识，非交易标识，交易是否成功需要查看 result_code 来判断
返回信息	message	否	String(128)	返回信息，如非空，为错误原因签名失败参数格式校验错误
以下字段在 status 为 0 的时候有返回				
业务结果	result_code	是	String(16)	0 表示成功，非 0 表示失败
商户号	mch_id	是	String(32)	商户号，由平台分配
设备号	device_info	否	String(32)	终端设备号
随机字符串	nonce_str	是	String(32)	随机字符串，不长于 32 位
错误代码	err_code	否	String(32)	参考错误码
签名	sign	是	String(32)	MD5 签名结果，详见“第 4 章 MD5 签名规则”
以下字段在 status 和 result_code 都为 0 的时候有返回				
平台订单号	transaction_id	是	String(32)	平台交易号
商户订单号	out_trade_no	是	String(32)	商户系统内部的订单号
退款笔数	refund_count	是	Int	退款记录数
商户退款单号	out_refund_no_\$n	是	String(32)	商户退款单号
平台退款单号	refund_id_\$n	是	String(32)	平台退款单号
退款渠道	refund_channel_\$n	是	String(16)	ORIGINAL—原路退款，默认
退款金额	refund_fee_\$n	是	Int	退款总金额,单位为分,可以做部分退款
现金券退款金额	coupon_refund_fee_\$n	否	Int	现金券退款金额 <= 退款金额，退款金额-现金券退款金额为现金
退款时间	refund_time_\$n	否	String(14)	yyyyMMddHHmmss

退款状态	refund_status_\$n	是	String(16)	退款状态： SUCCESS—退款成功 FAIL—退款失败 PROCESSING—退款处理中 NOTSURE—未确定， 需要商户原退款单号重新发起 CHANGE—转入代发，退款到银行发现用户的卡作废或者冻结了，导致原路退款银行卡失败，资金回流到商户的现金帐号，需要商户人工干预，通过线下或者平台转账的方式进行退款。
\$n 表示记录的序号，取值为 0~(\$ refund_count -1)，例如 refund_count 指示返回的退款记录有 2 条。第一条序号为“0”，第二条序号为“1”。				

8 关闭订单接口

8.1 关闭订单接口

8.1.1 业务功能

商户订单支付失败需要生成新单号重新发起支付，要对原订单号调用关单，避免重复支付；系统下单后，用户支付超时，系统退出不再受理，避免用户继续，请调用关单接口。

8.1.2 交互模式

请求：后台请求交互模式

返回结果：后台请求交互模式

8.1.3 请求参数列表

请求 url: <https://pay.swiftpass.cn/pay/gateway>

POST XML 内容体进行请求

字段名	变量名	必填	类型	说明
业务参数				
接口类型	service	是	String(32)	接口类型：unified.trade.close
版本号	version	否	String(8)	版本号，version 默认值是 2.0
字符集	charset	否	String(8)	可选值 UTF-8 ，默认为 UTF-8
签名方式	sign_type	否	String(8)	签名类型，取值：MD5 默认：MD5
商户号	mch_id	是	String(32)	商户号，由平台分配
商户订单号	out_trade_no	是	String(32)	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
随机字符串	nonce_str	是	String(32)	随机字符串，不长于 32 位
签名	sign	是	String(32)	MD5 签名结果，详见“第 4 章 MD5 签名规则”

8.1.4 返回结果

数据按 XML 的格式实时返回

字段名	变量名	必填	类型	说明
版本号	version	是	String(8)	版本号，version 默认值是 2.0。
字符集	charset	是	String(8)	可选值 UTF-8 ，默认为 UTF-8。
签名方式	sign_type	是	String(8)	签名类型，取值：MD5 默认：MD5
返回状态码	status	是	String(16)	0 表示成功，非 0 表示失败 此字段是通信标识，非交易标识，交易是否成功需要查看 result_code 来判断
返回信息	message	否	String(128)	返回信息，如非空，为错误原因签名失败参数格式校验错误
以下字段在 status 为 0 的时候有返回				
业务结果	result_code	是	String(16)	0 表示成功，非 0 表示失败；0 表示关单成功，此笔订单不能再发起支付；非 0 或其它表示关单接口异常，需再次发起关单操作；
商户号	mch_id	是	String(32)	商户号，由平台分配
随机字符串	nonce_str	是	String(32)	随机字符串，不长于 32 位
错误代码	err_code	否	String(32)	具体错误码请看文档最后错误码列表
错误代码描述	err_msg	否	String (128)	结果信息描述

签名	sign	是	String(32)	MD5 签名结果，详见“第 4 章 MD5 签名规则”
----	------	---	------------	-----------------------------

9 注意事项

- 1、所有涉及到金额的单位都是分，最小的单位是 1 分，不能有小数出现
- 2、notify_url 是平台服务器从后台直接发起请求到商户服务器，商户处理时不能检查用户的 cookie 或 session；商户更新 DB 等发货流程需要在 notify_url 完成后，以确保掉单时，平台补单能成功补上
- 3、notify_url 有可能重复通知，商户需要做去重处理，避免多次发货
- 4、notify_url 收到的通知，商户处理成功或检查订单已经处理，需要返回处理成功的标志纯字符串 success，字符串 success 不区分大小写；如果我们没有收到返回的 success，我方服务器继续向你们发送通知，在三小时后将不再通知；假设所有订单都没有返回 success，将增加我们服务器的通知负载，最坏的情况，可能就是导致正常通知商户的通知发生延迟；另外我们会催促你们完善，如果长期不改善，研发或运维技术将对贵公司开通的支付接口采取控制措施。
- 5、文档中请求接口时传的参数，必填为是的参数是必须要传的（如有缺少会报错），必填为否的参数可以传也可以不传
- 6、返回参数中必填为是的参数是一定会返回的，必填为否的参数则不一定返回，因升级或配置等情况实际返回参数可能不会跟文档完全一致，必须以实际接收到的参数为准
- 7、其它注意事项

（1）参数大小写问题

请注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”。

（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）时间戳问题

请使用 Linux 时间戳，注意为字符串格式。

（4）同一商户订单号支付问题

商户订单支付失败需要生成新单号重新发起支付，要对原订单号调用关单，避免重复支付；系统下单后，用户支付超时，系统退出不再受理，避免用户继续，请调用关单接口。

注意：订单生成后不能马上调用关单接口，最短调用时间间隔为 5 分钟。