



Merchant API Documentation

WeChat APP Payment API Specification

Document version: 1.5.3

API version: 2.0

TABLE OF CONTENTS

1	Introduction.....	4
1.1	Abstract.....	4
1.2	Audience.....	4
2	Program Overview.....	5
2.1	Industry Background.....	5
2.2	Business realization process.....	6
3	Data Format.....	6
3.1	Request data.....	6
3.2	XML data format.....	7
4	Digital Signature.....	9
4.2	Original string of signature.....	9
4.2	Method of signature.....	10
4.2.1	MD5 signature.....	10
4.2.2	SHA256 signature.....	11
4.2.3	RSA signature.....	12
5	Mechanism to Supplement Order.....	15
6	APP Interface.....	16
6.1	Pre-Order Interface.....	16
6.1.1	Business function.....	16
6.1.2	Interactive mode.....	16
6.1.3	Request Parameters.....	16
6.1.4	Response result.....	19
6.2	Notification of App pay.....	21
6.2.1	Notification request parameters.....	21
6.2.2	Response of notification.....	24
6.3	Retrieve transaction result interface.....	25
6.3.1	Business function.....	25
6.3.2	Interactive mode.....	25
6.3.3	Request Parameters.....	25
6.3.4	Response result.....	26
6.4	Refund Interface.....	29
6.4.1	Business function.....	29
6.4.2	Interactive mode.....	30
6.4.3	Request Parameters.....	30
6.4.4	Response result.....	31
6.5	Close Order Interface.....	33
6.5.1	Business function.....	33
6.5.2	Interactive mode.....	33
6.5.3	Request Parameters.....	34

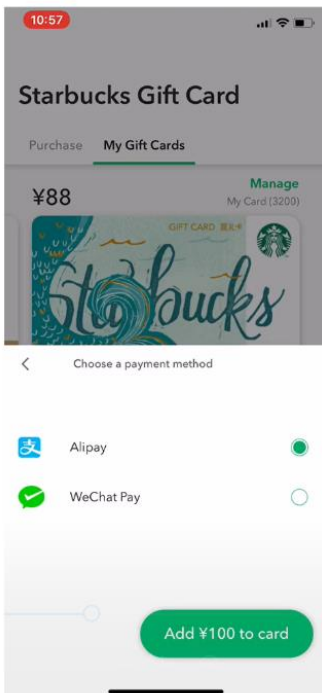
	6.5.4 Response result.....	35
7	Notes	36
8	Error Code.....	38
	8.1 Swiftpass error code.....	38
	8.2 WeChat error code.....	41

Document Changes		
Date	Version	Description
2018-05-09	1.0	First draft
2018-06-01	1.1	Request URL is changed to https://gateway.wepayez.com/pay/gateway
2019-06-12	1.2	Update SHA256 & RSA signature algorithm.
2022-07-06	1.5	Update description of fields mch_id, device_info, time_start, time_expire, appid, pay_info, trade_state, etc.
2022-07-06	1.5	Add SDK download address
2022-07-06	1.5	Add the precautions for using the query interface
2022-07-06	1.5	Add gateway error code
2022-08-16	1.5.1	Added swiftpass gateway request timeout description
2022-08-16	1.5.1	Add parameter "sign_agentno"
2022-08-16	1.5.1	Modify the description of the parameter "Body"
2022-09-26	1.5.2	Added Demo of Aggregator Mode and deleted "channel id"
2022-12-30	1.5.3	Update fields trade_type, local_fee_type, local_total_fee, order_fee, etc
2023-01-03	1.5.3	Update the field fee_type, delete the duplicated field order_fee
2023-01-06	1.5.3	Update the field cash_fee, cash_fee_type, rate
2023-02-24	1.5.3	Updated the diagram, signature sample message of SHA256. Added MD5 signature method; Updated the following fields: fields charset, sign_type, out_trade_no, device_info, attach, notify_url, err_code, err_msg, out_trade_no, transaction_id, refund_id, out_trade_refund;

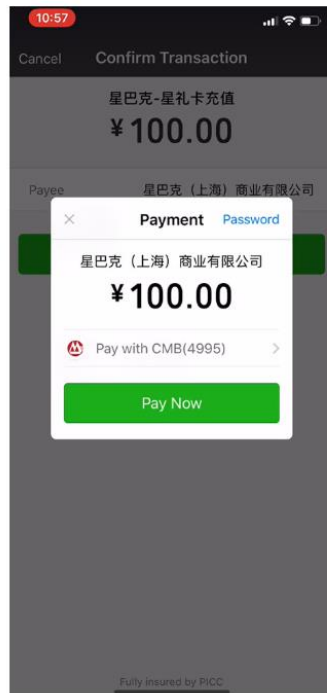
1 Introduction

1.1 Abstract

WeChat SDK Pay (App Pay) is a payment method that customers initiate transaction in merchant's App and then jump to WeChatApp to complete the payment.



Starbucks



WeChat



WeChat



Starbucks

1.2 Audience

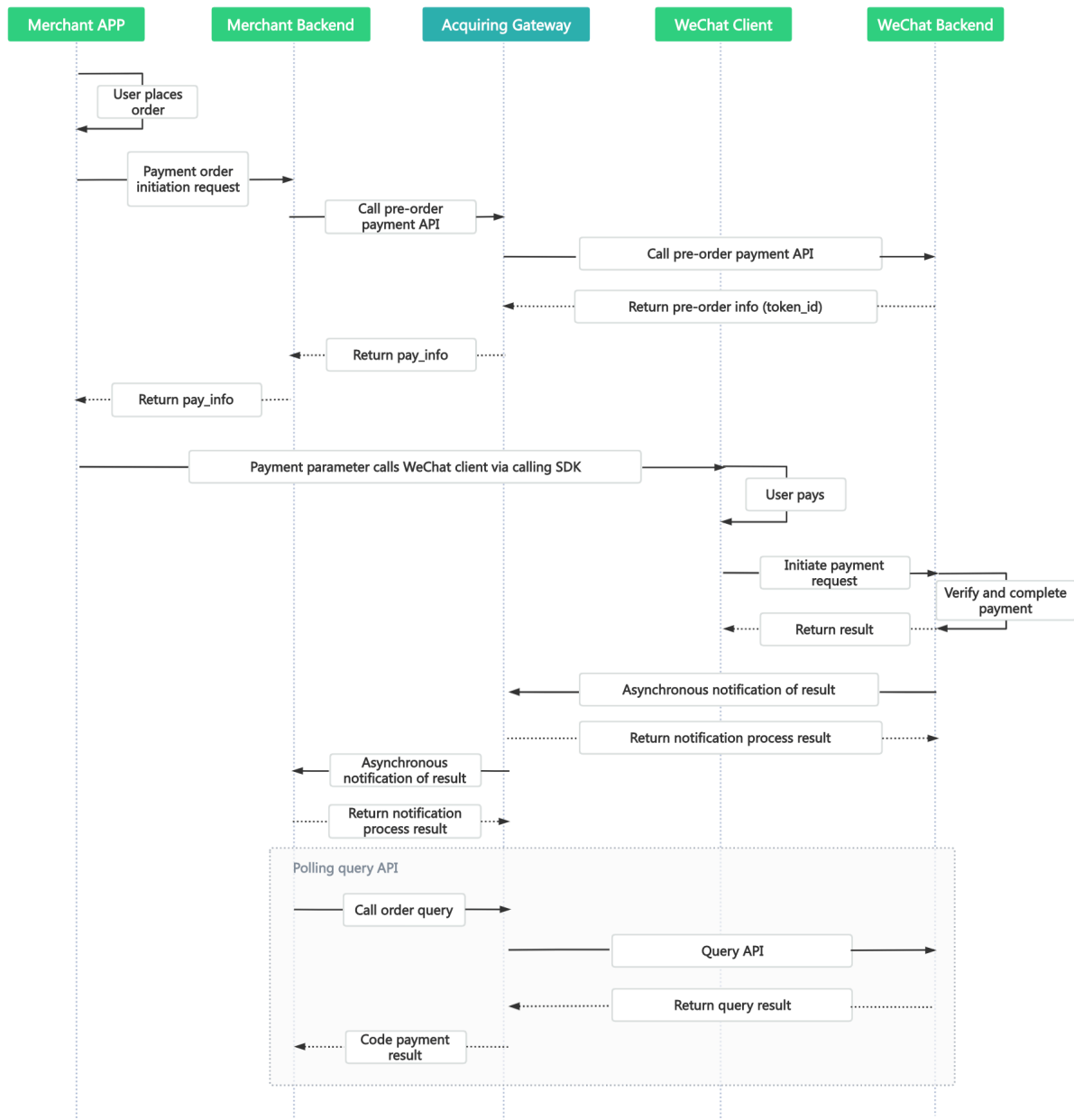
This document is provided to technical and business staff of merchants for reference.

2 Program Overview

2.1 Industry Background

WeChat payment is a payment service function which is provided based on WeChat application, and meanwhile it provides the commercial tenants with such support functions as sales, marketing analysis and management of account.

2.2 Business realization process



3 Data Format

3.1 Request data

Used HTTPS POST protocol. To ensure security transfer data must be signed.

```
<xml>
```

```
<service>pay.weixin.raw.app</service>

<appid>wxd1cbaa43e3a0b71c</appid>

<mch_id>7551999991</mch_id>

<out_trade_no>435775519999910053</out_trade_no>

<device_info>system=Android^version=3.0.1.2</device_info>

<body>Parking</body>

<total_fee>1000</total_fee>

<mch_create_ip>172.30.70.20</mch_create_ip>

<notify_url>https://xxxxxx.com/xxx/xxxxx/</notify_url>

<nonce_str>XSPFMLGbOST7551999991LMLS4HBvTnPsoPp</nonce_str>

<sign_type>MD5</sign_type>

<sign>3bdc187d3501492dcf2a41ba85c2dbfa</sign>

</xml>
```

3.2 XML data format

Used Standard XML protocol. First-level node only. No nested nodes.

Protocol error return:

```
<xml>

<status>500</status>

<message><![CDATA[SYSERR]]></message>

</xml>
```

Successful response:

```
<xml>

<appid><![CDATA[wxf4c1066cb858c6d7]]></appid>

<charset><![CDATA[UTF-8]]></charset>

<device_info><![CDATA[system=Android]]></device_info>

<mch_id><![CDATA[7551999991]]></mch_id>

<nonce_str><![CDATA[AmhUAhsTvX363S6L]]></nonce_str>
```



```
<out_trade_no><![CDATA[43570053]]></out_trade_no>

<pay_info><![CDATA[{"package":"Sign=WXPay","appid":"wxd175519999913a0b71c","sign":"16C70ABF3A5874338BA42209E9BC90B3","partnerid":"391111185","prepayid":"wx101811111655365c44b47e58640000","noncestr":"16711111245","timestamp":"16711111697"}]]></pay_info>

<result_code><![CDATA[0]]></result_code>

<sign><![CDATA[B152400B9B34D85533AC84F022A55DAF]]></sign>

<sign_type><![CDATA[MD5]]></sign_type>

<status><![CDATA[0]]></status>

<token_id><![CDATA[1d863b76ba7551999991d0259f3555c]]></token_id>

<transaction_id><![CDATA[181590755199999101165618180]]></transaction_id>

<version><![CDATA[2.0]]></version>

</xml>
```

Normal error return:

```
<xml>

<status>0</status>

<message><![CDATA[OK]]></message>

<appid><![CDATA[wxf4c1066cb858c6d7]]></appid>

<mch_id><![CDATA[7551999991]]></mch_id>

<device_info><![CDATA[system=Android]]></device_info>

<nonce_str><![CDATA[sthBJ9QyUG6vkrjJ]]></nonce_str>

<sign><![CDATA[6277A96D7875D4FF23AA7B6A4C3046AB]]></sign>

<result_code>1</result_code>

<err_code><![CDATA[APPID_NOT_EXIST]]></err_code>

<err_code_des><![CDATA[Check whether APPID is correct.]]></err_code_des>

</xml>
```

The field named status return '0': successful. Other value means fail.

4 Digital Signature

To ensure the authenticity and integrity of transmissible data, we need to verify the signed data after it being received.

There are two steps in digital signature.

Follow the rules to connect the original string that needs to be signed;

Calculate the result of signature according to specific algorithm and key.

Generally, the failed result will not be signed.

4.1 Original string of signature

The original string of signature will be assembled into character string according to the following modes no matter whether it is request or response:

1. Besides the sign field, all parameter fields will be ranked in ascending order according to the ASCII of the field name and then connected in the format of QueryString (i.e. key1=value1&key2=value2...), and the null value will not transfer and will not participate in formation of string of signature.
2. In the original string of signature, both the field name and field value will adopt original values and will not conduct URL Encode.
3. The response or notification information returned by platform might increase parameters due to upgrading, and this case should be allowed when the response signature is verified.

Example:

Calling an interface with following fields:

```
<xml>  
  
<service>pay.weixin.raw.app</service>  
  
<appid>wxid1cbaa43e3a0b71c</appid>  
  
<mch_id>7551999991</mch_id>
```

```
<out_trade_no>435775519999910053</out_trade_no>

<device_info>system=Android^version=3.0.1.2</device_info>

<body>Parking</body>

<total_fee>1000</total_fee>

<mch_create_ip>172.30.70.20</mch_create_ip>

<notify_url>https://xxxxxx.com/xxx/xxxxx/</notify_url>

<nonce_str>XSPFMLGbOST7551999991LMLS4HBvTnPsoPp</nonce_str>

<sign_type>MD5</sign_type>

<sign>3bdc187d3501492dcf2a41ba85c2dbfa</sign>

</xml>
```

The signature field sequence:

```
body=Parking&appid=wxd1cbaa43e3a0b71c&mch_create_ip=172.30.70.20&mch_id=7551999991&nonce_str=XSPFMLGbOST7551999991LMLS4HBv
TnPsoPp&notify_url=https://xxxxxx.com/xxx/xxxxx/&out_trade_no=435775519999910053&service=pay.weixin.raw.app&total_fee=1000
```

4.2 Method of signature

4.2.1 MD5 signature

MD5 signature calculation formula:

sign = MD5("Original string" &key= "signature key"). toUpperCase

Suppose the following are incoming XML parameter:

```
<xml>

<service>pay.weixin.raw.app</service>

<appid>wxd1cbaa43e3a0b71c</appid>

<mch_id>7551999991</mch_id>

<out_trade_no>435775519999910053</out_trade_no>

<device_info>system=Android^version=3.0.1.2</device_info>

<body>Parking</body>

<total_fee>1000</total_fee>

<mch_create_ip>172.30.70.20</mch_create_ip>

<notify_url>https://xxxxxx.com/xxx/xxxxx/</notify_url>
```

```
<nonce_str>XSPFMLGbOST7551999991LMLS4HBvTnPsoPp</nonce_str>  
  
<sign_type>MD5</sign_type>  
  
<sign>DD39E4BE112FF0CA33D89830D8898731</sign>  
  
</xml>
```

Suppose merchant key is: 9f72151b6592fab3e0c63a1ab3c0877b

I: string1 after URL key sequencing the dictionary order in process a:

```
body=Parking&appid=wx1cbaa43e3a0b71c&mch_create_ip=172.30.70.20&mch_id=7551999991&nonce_str=XSPFMLGbOST7551999991LMLS4HBvTnPsoPp&notify_url=https://xxxxxx.com/xxx/xxxxx/&out_trade_no=435775519999910053&service=pay.weixin.raw.app&total_fee=1000
```

ii: sign after process b:

sign

```
=md5(string1&key=9f72151b6592fab3e0c63a1ab3c0877b).toUpperCase
```

```
=md5(body=Parking&appid=wx1cbaa43e3a0b71c&mch_create_ip=172.30.70.20&mch_id=7551999991&nonce_str=XSPFMLGbOST7551999991LMLS4HBvTnPsoPp&notify_url=https://xxxxxx.com/xxx/xxxxx/&out_trade_no=435775519999910053&service=pay.weixin.raw.app&total_fee=1000&key=9d101c97133837e13dde2d32a5054abb).toUpperCase()  
  
="DD39E4BE112FF0CA33D89830D8898731"
```

4.2.2 SHA256 signature

SHA256 signature calculation formula:

**sign =SHA256(“The signature field sequence strings”&key=“signature key”).
toUpperCase**

Example:

There are XML afferent parameters:

```
<xml>  
  
<service>pay.weixin.raw.app</service>  
  
<appid>wx1cbaa43e3a0b71c</appid>  
  
<mch_id>7551999991</mch_id>  
  
<out_trade_no>435775519999910053</out_trade_no>  
  
<device_info>system=Android^version=3.0.1.2</device_info>
```

```
<body>Parking</body>

<total_fee>1000</total_fee>

<mch_create_ip>172.30.70.20</mch_create_ip>

<notify_url>https://xxxxxx.com/xxx/xxxxx/</notify_url>

<nonce_str>XSPFMLGbOST7551999991LMLS4HBvTnPsoPp</nonce_str>

<sign_type>SHA256</sign_type>

<sign>409EA40C705F0D986E995117ED183D79497680FB06DEC2B01FB89F0518A52FC2</sign>

</xml>
```

Merchant signature key: 9f72151b6592fab3e0c63a1ab3c0877b

i: the first step of which is to connect the original string(string1) that needs signature according to certain rules:

```
body=Parking&appid=wxid1cbaa43e3a0b71c&mch_create_ip=172.30.70.20&mch_id=7551999991&nonce_str=XSPFMLGbOST7551999991LMLS4HBvTnPsoPp&notify_url=https://xxxxxx.com/xxx/xxxxx/&out_trade_no=435775519999910053&service=pay.weixin.raw.app&total_fee=1000
```

ii: the second step of which is to choose SHA256 and keys to calculate the result of signature(sign):

```
sign

=SHA256(string1&key=9f72151b6592fab3e0c63a1ab3c0877b).toUpperCase

=SHA256(body=Parking&appid=wxid1cbaa43e3a0b71c&mch_create_ip=172.30.70.20&mch_id=7551999991&nonce_str=XSPFMLGbOST7551999991LMLS4HBvTnPsoPp&notify_url=https://xxxxxx.com/xxx/xxxxx/&out_trade_no=435775519999910053&service=pay.weixin.raw.app&total_fee=1000&key=9f72151b6592fab3e0c63a1ab3c0877b).toUpperCase()

="409EA40C705F0D986E995117ED183D79497680FB06DEC2B01FB89F0518A52FC2"
```

4.2.3 RSA signature

RSA signature calculation formula:

The RSA algorithm has always been the most widely used "asymmetric encryption algorithm". By adding the content of the RSA private key of the merchant communication after the original string is signed, the result string is the result of the RSA operation.

Note: The set of coded characters specified when converting a string to a byte stream at signature should match the parameter charset. The RSA key pair is generated by the merchant. You need to upload the public key to the merchant portal and download the SwiftPass public key for the response.

Sign = RSA (request parameter string, merchant RSA private key)

Description: The name of the platform signature algorithm: RSA_1_256, corresponding to the standard signature algorithm name: SHA256WithRSA, the length of the RSA key is required to be 2048.

sign =RSA (“The signature field sequence strings”&key=“signature private key”).

toUpperCase

Example:

There are XML afferent parameters:

```
<xml>

  <service>pay.weixin.raw.app</service>

  <appid>wxid1cbaa43e3a0b71c</appid>

  <mch_id>7551999991</mch_id>

  <out_trade_no>435775519999910053</out_trade_no>

  <device_info>system=Android^version=3.0.1.2</device_info>

  <body>Parking</body>

  <total_fee>1000</total_fee>

  <mch_create_ip>172.30.70.20</mch_create_ip>

  <notify_url>https://xxxxxx.com/xxx/xxxxx/</notify_url>

  <nonce_str>XSPFMLGbOST7551999991LMLS4HBvTnPsoPp</nonce_str>

  <sign_type>RSA_1_256</sign_type>

<sign>

Zpq4Zb618KNf8BpazdwQ6EpnxrfHfQiBLihZnYSZS2FLwn+Sr/sVuqtgf5iigJ8cqIpCqaLmZ6Qy8PFI+taWq1riV+2ZnMsGK27N2twYSqummznm0TIm+9

T3/Oi9ListVnUVLfbBbE35PdH7VoWMJIIyIIKrwWodY4qsHdEwZbW29C+DtFRx5FPIXTfyUU/6TtN+XO4zl7R1/dy4JRkHGCFGccUgCflubJB+hBez+yw5

wrT5UiiVeZLxkEZGN3pAMD8HZulXqdERTGwfm4bB4i8VMBXERmRhx0GHo5IAZLvQwwqRw+3T5kO9scMhYgGsXxE7WBweqe5VhcNz4cl/w==</sign

>
```

```
</xml>
```

Merchant RSA Private key:

```
MIIEVQIBADANBgkqhkiG9w0BAQEFAASCBCwggSjAgEAAoIBAQCfU8v4B Ur81SKm/H0ahbdQZjEpO8nMyk+XuYSatHwnU4//m47R+4G2YB4Z6PHsJi
4+ScfJpQutFhKrFwTXZ6TDqLvaqZDDkjq5G271g+PmrzFp7f40/E9m0qjeL64Rjra0zql23dvPW4vVomMRgRcoPOn0YWVp+M6T5PaFgE4M8dh4IMZz
57gVwOdd08F99Z92f3QgZtEjl+/EXvMenXxb/aRofNkt+Wdk2ELJ6MIP0d9U5v3WgLuuNv5QnQYzj/RMr8GD+wrDYiNQJxsaTmE/OEJggsumhD4eYY5
YIRy2EIN504cuYVKU1wOSZgq9oJCynGR0aPuQWx58IHxetAgMBAECCggEAHfEFd8qm2PTE2ITAVEC7F+TcgD84UAz0dZnURtx6YIOoZ5+LH/zVG6j
uYLJU/Oo5RPAc+iMVS68u2JMCp7zm8Ft7B3JkrbuHLNHGuR6Q7PQuXN8PkDcOxqDmZ2kPJzI4PZvBZRE0abduq+MatGzpGAuJzrWcB/N0oVlvrXp9
PnOqfo/Y5nxmpOFCImJppIS3AL1pftNtQZ09G15CPHDYtpUbXPtD2MjjW4OLxKuPRoHSwUgo6LW9XSwnXfcuK+IbzLL0BhIWD9IV/+yCEUEbIN87yxxf
hpQFaAhXj5W+B3YsMOZuK93+XMOpymw8EpUDMOBOnvwb0NSHUrV2RUAQKBgQDT0jlnNS1e7+tjPzFtOhGPj1uCBPAEIEHAcnPgd80bEiujxMLCn
GaAvmnTrMu4Xo0e5fAP4F7R6UD+IUsfr3CAAu7CadQ49TW+SovAvCjy9AZuSVVlwyu6QdYgFyPKe1LZYAEq5k+mB1Vh5q0RoxMNA5pGYKg8+4M
mmsJi7X7QKBgQDAunCOqliH128bs/1VRlhDpzuRW5Qr/SRbO2saVg5RSHnO/nGT2OuxSTTkc8yrx7qd9SmAxXl5kR238DhMOQOnRBomldmVtAJuJg
rdQyt0wXfeVQqshqCUaE/xhEbpSCdbPSZbKZZdplV0y6O5vXlhxw+1qAvXLcxw46s3R92QQKBgQCIQ+ejywkVPDILHMwSSehwvThufkCYWYUbbCv
DowpOe5AMoZidtNju7MNjg2rLHTsCx/kBzOr+7THNwl4R7kTiEmg09cO+fu5rHXepGgtig+GJukaZPZ6/bMZJvGOLgOhHmomwG/jdwpvgVtlGBCh6BW5J
ZcSImT+ykIoYfvDRuQKBgCgwOHxnBGFfORoLxE3dhpSk8LT05cbuelBVuZW6UC3+8PeK82AjlBLMUy04QHupoG6Dyu3BP/1rl0jd3L94PBzLBD7G
m4vJtqW0DknYo5sMXS1JrnofcKjBv7nbHXZTx3EtJSxpVaOdpca/HpsCuCP3AH2e1yk9sZ3wu6lBYSBaGACYM60j1CVRNSZxUNRgiwfWzS69q1ee
zPc7xQEganpVBI9SZcTNP1kpDKmQikXJ4Yb5XWn12HCY/sFeBW6Su3ruNqXvg1XiUPbH6A6nxd5B3QX0mS9+wDm6ONysPLRdKbfFO0mdP4CeyuG
PdvDIMXP4dJdLhMUL4pcJLI0B7gBE=
```

i: the first step of which is to connect the original string(string1) that needs signature according to certain rules:

```
body=Parking&appid=wxdc1cbaa43e3a0b71c&mch_create_ip=172.30.70.20&mch_id=7551999991&nonce_str=XSPFMLGbOST7551999991LMLS4HBv
TnPsoPp&notify_url=https://xxxxxx.com/xxx/xxxxx/&out_trade_no=435775519999910053&sign_type=RSA_1_256&service=pay.weixin.raw.app&total_f
ee=1000
```

ii: the second step of which is to choose RSA and keys to calculate the result of signature(sign):

```
sign=RSA(string1, Merchant RSA Private
key)==RSA(body=Parking&appid=wxdc1cbaa43e3a0b71c&mch_create_ip=172.30.70.20&mch_id=7551999991&nonce_str=XSPFMLGbOST755199999
1LMLS4HBvTnPsoPp&notify_url=https://xxxxxx.com/xxx/xxxxx/&out_trade_no=435775519999910053&sign_type=RSA_1_256&service=pay.weixin.raw
```

```
.app&total_fee=1000,MIIeVQIBADANBgqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQCfU8v4BUr81SKm/H0ahbdQZjEpO8nMyk+XuSatHwnU4//m
47R+4G2YB4Z6PHsJi4+ScfJpQutFhKrFwTXZ6TDqLvaqZDDkQ5G271g+PmrzFp7f40/E9m0qjeL64RJa0rZq123dvPW4vVomMRgRcoPOn0YWVp+M6
T5PaFgE4M8dh4IMZz57gVwOdd08F99Z92f3QgZtEjI+/EXvMenXxb/aRofNkt+Wdk2ELJ6MIP0d9UU5v3WgLuUv5QnQYzj/RMr8GD+wrDYiNQJxsaTm
E/OEJggsunhD4eYY5YIRy2EIN504cujYVKU1wOSZgq9oJCynGR0aPuQWx58IHxetAgMBAAECggEAHFEfd8qm2PTE2ITAVEC7F+TcgD84lUAz0dZnU
Rtx6YIOz5+LH/zVG6juYlJU/Oo5RPAC+iMVS68u2JMCp7zm8Ft7B3JkrbuHLNHGuR6Q7PQUxN8PkDcOxqDmZ2kPjz14PZvBZRE0abdug+MatGzpG
AuJzrWcB/N0oVlvrXp9PnOqfo/Y5nxmpOFCImJpplS3AL1pftNtQZ09G15CPHDYtpUbXPtD2MjjW4OLxKuPRoHSwUgo6LW9XSwNXfucK+IbzLL0BhIW
D9IV/+yCEUEblN87yxxfhpQFAhXj5W+B3YsMOZuK93+XMOpYmw8EpuUDMOBOnvwb0NSHUv2RUAQKBgQDT0jlnNS1e7+tjPzFtOhGPj1uCBPAELe
HAcnPgd80bEiujxMLCnGaAvmnTrMu4Xo0e5fAP4F7R6UD+IUsfr3CAAu7CadQ49TW+SovAvciy9AZuSVVIwynu6QdYgFyPKe1LZYAEq5k+mB1Vh5q0
RoxMNA5pGYK8+4MmmsJi7X7QKBgQDAunCOqliH128bs/1VRlhDpzuRW5Qr/SRbO2saVg5RSRHnO/nGT2OuxSTTkc8yrx7qd9SmAxI5kR238DhM
OQOnRBomldmVtAJuJgrdQyt0wXfeQVqshqCUaE/xhEbpSCdbPSZbKZZdplV0y6O5vXihxw+1qAvXLcxw46s3R92QQKBgQCIQ+ejywkVPDILHMwSS
ehwvThufkCYWYUbbcvDowpOe5AMoZidtNju7MNjg2rLHTsCx/kBzOr+7THNwL4R7kTiEmg09cO+fu5rHXepGgtig+GJukaZPZ6/bMZJvGOLgOhHmomw
G/jdwpGvtlGBCh6BW5JZcSlmT+yklOoYfvDRuQKBgCgwOHxnBGFfORoLxE3dhpSk8LT05cbueIBVuZW6UC3+8PeK82AjlLMUy04QHupoG6Dyu3BP/
1rl0jd3L94PBzLBDL7Gm4vJTqW0DknY0sMxS1JrnofcKjBv7nbHXZTx3EtJSxpVaOdpca/HpsCuCP3AH2e1yk9sZ3wu6IBYSBAoGACYM60j1CVRNSZ
xUNRgiwfWzS69ql1eezPc7xQEganpVBI9SZcTNp1kpDKmQikXJ4Yb5XWn12HCY/sFeBW6S3ruNqxvg1XiUPbH6A6nxd5B3QX0mS9+wDm6ONysPL
RdKbfFO_0mdP4CeyuGPdvDIMXP4dJdLhMUL4pcJLI0B7gBE=")=
Zpq4Zb618KNf8BpazdwQ6EpnxrfHFQIBLIhZnYSZS2FLwn+Sr/sVuqtgf5iigJ8cqIpCqaLmZ6Qy8PFI+taWq1riV+2ZnMsGK27N2twYSqummznm0TIIm+9
T3/Oi9ListVnUVLfbBbE35PdH7vOwMJlIkrwwOdY4qsHdEwZbW29C+DtFRx5FPIXTfyUU/6TtN+XO4zl7R1/dy4JRkHGCFGccUgCflubJB+hBez+yw5
wrT5UiiVeZLxkEZGN3pAMD8HZulXqdERTGwfm4bB4i8VMBXERmRhx0GHo5IAZLvQwwqRw+3T5kO9scMhYgGsXxE7WBweqe5VhcNz4cl/w==
```

5 Mechanism to Supplement Order

Notes: in respect of the back-end notification interactive mode, if the response of the commercial tenant received by platform is not **pure character string “success”** or if the response is given **after 5 seconds**, then the notification will be deemed as unsuccessful, and platform will adopt certain strategies (**the interval of notification: 0/15/15/30/180/1800/1800/1800/1800/3600 Unit: seconds**) to re-initiate notification **intermittently**, to improve the success rate of notification, but platform will not guarantee the final success of notification. Because there is the case where the back-end notification will be resent, the same notification might be sent to the commercial tenant system many times. The

commercial tenant system must be able to process the repeated notification in a right way. The method recommended by platform is to first examine the status of the corresponding business data upon receiving the notification to judge whether the notification has been disposed, and in case it hasn't been dispose, it will be disposed otherwise, and if it has been disposed, **the pure character string "success"** will be returned directly. Before status examination and disposal of business data are conducted, data lock should be adopted for concurrency control to avoid data chaos caused by **repeated data interposition** in the function.

6 APP Interface

6.1 Pre-Order Interface

6.1.1 Business function

WeChat SDK Pay is a payment method that customers initiate transaction in merchant's App and then jump to WeChat App to complete the payment.

6.1.2 Interactive mode

Request: Background request interaction mode

Return: Background request interaction mode

6.1.3 Request Parameters

Request URL : <https://gateway.wepayez.com/pay/gateway>

POST request with content of XML

Field Name	Required	Type	Description
service	Yes	String(32)	Interface type : pay.weixin.raw.app
version	No	String(8)	Version number, default value:2.0

charset	No	String(8)	Encoding method. Default Value: UTF-8
sign_type	No	String(16)	MD5: MD5 SHA256 : SHA256 RSA: RSA_1_256 Default value: MD5
groupno	No	String (32)	Master merchant ID
mch_id	Yes	String(32)	Merchant ID, Specifies an unique id assigned by platform. Only Store ID or Ordinary Merchant ID is valid.
appid	Yes	String(32)	WeChat appid of the merchant app, same as sub_appid
out_trade_no	Yes	String(32)	The unique trade reference(merchant order id of 5-32 bits) in merchant' s system. Letter, number and underline are allowed. Case-sensitive
device_info	No	String(32)	Specifies a Terminal device id.You can query based on this field on the swiftpass merchant platform
body	Yes	String(128)	Description of merchants' goods. Can upload 128 English, 42 Chinese
attach	No	String(127)	Additional merchant info. The original value will be returned in the query response and notification after successful payment. This field is mainly used for custom data of the order carried by the merchant.
total_fee	Yes	Int	Integral number is allowed only. The unit of the fee is the minimal unit of the local currency
mch_create_ip	Yes	String(16)	Specifies the machine IP that calls the api
notify_url	Yes	String(255)	Specifies the callback address for receiving platform payment notifications. Should be absolute path and ensure platform accessible. i.e. http://wap.tenpay.com/tenpay.asp . The url only effective in non-original ecology mode.
time_start	No	String(14)	Order created date. Format:yyyyMMddHHmmss. i.e.20091225091010. GMT+8 BeijingTime. WeChat can set the time range from 1 minute to 2 hours, if not uploading, the default is 10 minutes.

time_expire	No	String(14)	Order created date. Format:yyyyMMddHHmmss. i.e.20091225091010. GMT+8 BeijingTime. WeChat can set the time range from 1 minute to 2 hours, if not uploading, the default is 10 minutes.
op_user_id	No	String(32)	Operator ID, value: merchant ID
op_shop_id	No	String(32)	
limit_credit_pay	No	String(1)	Restrict consumer pay in credit card. “1” : forbidden using credit card. “0” or null : permit using credit card.
goods_tag	No	String(32)	Goods tag
nonce_str	Yes	String(32)	Included in platform payment API protocols to ensure unpredictability for signatures. 32 characters or fewer.
sub_appid	No	String(32)	Merchant App's appid from WeChat
sign	Yes	String(344)	Please refer to section 4 'Digital Signature'.
sign_agentno	No	String(32)	When the merchant is applied the Aggregator mode that the merchant will be represented by the channel, the agency channel number is sent, and the corresponding sign needs to be signed with the channel's sign_key.

Demo of Aggregator Mode:

```

<xml>
<body><![CDATA[changyoyo]]></body>
<charset><![CDATA[UTF-8]]></charset>
<device_info><![CDATA[changyoyo]]></device_info>
<mch_create_ip><![CDATA[58.33.106.38]]></mch_create_ip>
<mch_id><![CDATA[181520234234]]></mch_id>
<nonce_str><![CDATA[HFfP43tL2i]]></nonce_str>
<notify_url><![CDATA[http://58.33.106.38:8080/api/ali/resultNotify]]></notify_url>
<out_trade_no><![CDATA[20220926111231230000]]></out_trade_no>
<sign_agentno><![CDATA[1231231]]></sign_agentno>
<service><![CDATA[pay.weixin.raw.app]]></service>

```

```

<sign><![CDATA[B0ECE637F82C135BD39C12E8F51443CEE08FF4A8C8FC2764D90D8770805216D1]]></sign>

<sign_type><![CDATA[SHA256]]></sign_type>

<time_expire><![CDATA[20220926114000]]></time_expire>

<time_start><![CDATA[20220926113000]]></time_start>

<total_fee><![CDATA[1]]></total_fee>

<version><![CDATA[2.0]]></version>

</xml>

```

6.1.4 Response result

Data return in real time with XML format

Field Name	Required	Type	Description
version	Yes	String(32)	Version number. Value:2.0.
charset	Yes	String(8)	Encoding method. Default Value: UTF-8
sign_type	Yes	String(16)	MD5: MD5 SHA256 : SHA256 RSA: RSA_1_256 Default value: MD5
status	Yes	String(8)	“0” : success. Others value: fail. Specifies communicating label (not transaction label). The status of a transaction is determined by the value of result_code.
message	No	String(32)	Return message. Only return when the signature verification invalid.
The following fields will return when status is '0'			
result_code	Yes	String(16)	“0” : success. Others value: fail.
groupno	No	String (32)	Master merchant ID
mch_id	Yes	String(32)	Merchant ID, Specifies an unique id assigned by platform. Only Store ID or Ordinary Merchant ID is valid.
device_info	No	String(32)	Specifies a Terminal device id. You can query based on this field on the swiftpass merchant platform

nonce_str	Yes	String(32)	Included in platform payment API protocols to ensure unpredictability for signatures. 32 characters or fewer
err_code	No	String(32)	Reference error code. It will be returned only when result_code is different from 0.
err_msg	No	String(128)	Error information description. It will be returned only when result_code is different from 0.
sign	Yes	String(344)	Please refer to section 4 'Digital Signature'.
sign_agentno	No	String(32)	When the merchant is applied the Aggregator mode that the merchant will be represented by the channel, the agency channel number is sent, and the corresponding sign needs to be signed with the channel's sign_key.
The following fields will return when both status and result_code are '0'			
appid	No	String(32)	Institution official account ID
token_id	Yes	String(64)	Payment authorization code
pay_info	Yes	String	String with Json format, WeChat official SDK parameter information.
out_trade_no	Yes	String	Merchant order ID
transaction_id	Yes	String	Platform order ID

Pay_info demo:

```
<pay_info>
<![CDATA[{"package":"Sign=WXPAY","appid":"wx1231211123326af8f0","sign":"3CE60E1E309A9D14B3B9445Fwewe1119685","partnerid":"4432332892", "prepayid":"wx0000000625192318e2e7ca3b9639b0000","noncestr":"1657111222177685","timestamp":"1657121977"}]]>
</pay_info>
```

WeChat SDK Download url

<https://pay.weixin.qq.com/wiki/doc/api/wxpay/ch/pages/sdk.shtml>

For Mobile app Integration SDK download:

IOS: https://developers.weixin.qq.com/doc/oplatform/Downloads/iOS_Resource.html

Android: https://developers.weixin.qq.com/doc/oplatform/Downloads/Android_Resource.html

6.2 Notification of App pay

6.2.1 Notification request parameters

The notification URL is the parameter notify_url submitted in section 6.1. And after the payment is completed, platform will send related payment and user information to the URL. And the commercial tenant needs to receive and dispose such information.

When interaction of the back-end notification is being made, if the response received by platform from the commercial tenant is not pure character string “success” or if the response is given after 5 seconds, then it will be deemed as unsuccessful by platform, and platform will adopt certain strategies (**the interval of notification:**

0/15/15/30/180/1800/1800/1800/1800/3600 Unit: seconds) to intermittently re-initiate notification to do its best to improve the rate of success of notification, but the final success of notification will not be guaranteed.

Because there is the case where the back-end notification will be resent, the same notification might be sent to the commercial tenant system many times. **The commercial tenant system must be able to process the repeated notification in a right way.**

The method recommended by platform is to first examine the status of the corresponding business data upon receiving the notification to judge whether the notification has been disposed, and in case it hasn't been dispose, it will be disposed otherwise, and if it has been disposed, the pure character string success will be returned directly. Before status examination and disposal of business data are conducted, data lock should be adopted for concurrency control to avoid data chaos caused by repeated data interposition in the function.

Notices: After the back-end of the commercial tenant has received the parameters of notification. It's required to verify out_trade_no and total_fee in notification received

according to the order number and amount of order of their own business system. And the status of order in the database will only be allowed to update after the verification is consistent. There is no notification for refund transaction.

Suggestion: After creating an order and initiating payment, if the payment success notification request is not received within 5 minutes, it is recommended to initiate an order query interface, which is queried every 5 seconds, for a total of 12 queries.

The back-end notification will return by POST mode through notify_url which sent in payment request.

(Notify in XML format)

Field Name	Required	Type	Description
version	Yes	String(8)	Version number. Value:2.0
charset	Yes	String(8)	Encoding method. Default Value: UTF-8
sign_type	Yes	String(16)	MD5: MD5 SHA256 : SHA256 RSA: RSA_1_256 Default value: MD5
status	Yes	String(16)	“0” : success. Others value: fail. Specifies communicating label (not transaction label). The status of a transaction is determined by the value of trade_state
message	No	String(128)	Return message. Only return when the signature verification invalid.
The following fields will return when status is '0'			
result_code	Yes	String(16)	“0” : success. Others value: fail
groupno	No	String (32)	Master merchant ID
mch_id	Yes	String(32)	Merchant ID, specifies an official account id assigned by

			platform. Only Store ID or Ordinary Merchant ID is valid.
device_info	No	String(32)	Specifies a Terminal device id. You can query based on this field on the swiftpass merchant platform
nonce_str	Yes	String(32)	Included in platform payment API protocols to ensure unpredictability for signatures. 32 characters or fewer.
err_code	No	String(32)	Reference error code. It will be returned only when result_code is different from 0.
err_msg	No	String (128)	Error information description. It will be returned only when result_code is different from 0.
sign	Yes	String(344)	Please refer to section 4 'Digital Signature'.
sign_agentno	No	String(32)	When the merchant is applied the Aggregator mode that the merchant will be represented by the channel, the agency channel number is sent, and the corresponding sign needs to be signed with the channel's sign_key.
The following fields will return when both status and result_code are '0'			
openid	Yes	String(128)	It is the only user identification under the current appid.
appid	No	String(32)	Institution official account ID
sub_appid	Yes	String(32)	Application appid applied by merchants on WeChat open platform
sub_openid	No	String(32)	User identification for WeChat application transaction
sub_is_subscribe	No	String(1)	Specifies whether the payer follows the associated official account or not, with "Y" meaning 'follows' and "N" meaning "not follows".
trade_type	Yes	String(32)	Value: pay.weixin.app
pay_result	Yes	Int	Payment result. "0" : success. Others fail
pay_info	No	String(64)	Payment result information. Payment successful return null.
transaction_id	Yes	String(32)	Platform order id, it's unique ID in platform.
out_transaction_id	No	String(32)	Order number provided by WeChat.
out_trade_no	Yes	String(32)	Specifies an order number created by merchant's system, which is

			consistent with request
total_fee	Yes	Int	Specifies the total amount for a transaction. The unit of the fee is the minimal unit of the currency
cash_fee	No	Int	Order cash payment amount.
order_fee	No	Int	Order amount in Cent
cash_fee_type	No	String(8)	Currency type of order cash payment.
local_total_fee	No	Int	Local total order amount in Cent.
local_fee_type	No	String(8)	Local currency type, three-letter code in accordance with ISO 4217
fee_type	No	String(8)	Complies with ISO 4217 standards.
rate	No	Int	rate
attach	No	String(127)	Additional merchant info. The original value will be returned in the query response and notification after successful payment. This field is mainly used for custom data of the order carried by the merchant.
bank_type	Yes	String(16)	Bank type, return from WeChat.
time_end	Yes	String(14)	Specifies the transaction payment time in the format of yyyyMMddHHmmss, such as 20091225091010 for Dec 25, 2009 09:10:10. GMT+8 Beijing Time. Returned when orders tatus is payment successful or refunded.

6.2.2 Response of notification

Platform server will send notification, post will send XML data flow, the notify_url of the commercial tenant will receive the result of notification, with the method of reception being as written in demo (for example, callback method in php, notify.aspx file in c#, TestPayResultSerlet method in java), and after the commercial tenant conducts business disposal, the feedback of result of disposal needs to be given in the form of pure character string, with its content being as follows:

Returned results	Description
------------------	-------------

success	The disposal is successful, and after platform system receives this result, no ongoing notification will be made.
fail or other character	The disposal fails, and if platform receives this result or doesn't receive any result, then the system will resend the notification through the mechanism of supplementing order (for detail please see section 5).

6.3 Retrieve transaction result interface

6.3.1 Business function

Retrieve transaction result information with platform order number or merchant order number.

6.3.2 Interactive mode

Background interaction mode.

6.3.3 Request Parameters

Request URL : <https://gateway.wepayez.com/pay/gateway>

POST request with content of XML

Field Name	Required	Type	Description
service	Yes	String(32)	Value: unified.trade.query
version	No	String(8)	Version number. default value: 2.0
charset	No	String(8)	Encoding method. Default Value: UTF-8
sign_type	No	String(16)	MD5: MD5 SHA256 : SHA256 RSA: RSA_1_256 Default value: MD5
mch_id	Yes	String(32)	Merchant ID, specifies a unique id assigned by platform. Only Store ID or Ordinary Merchant ID is valid.
groupno	No	String (32)	Master merchant ID

out_trade_no	No	String(32)	The unique trade reference of merchant system. At least one of the out_trade_no and transaction_id should be exist in query request. transaction_id priority when both be filled.
transaction_id	No	String(32)	The unique trade reference of platform system. At least one of the out_trade_no and transaction_id should be exist in query request. transaction_id priority when both be filled.
nonce_str	Yes	String(32)	Included in platform payment API protocols to ensure unpredictability for signatures. 32 characters or fewer.
sign	Yes	String(344)	Please refer to section '4 Digital Signature'.

6.3.4 Response result

Data return in real time with XML format

Field Name	Required	Type	Description
version	Yes	String(8)	Version number. default value: 2.0
charset	Yes	String(8)	Encoding method. Default Value: UTF-8
sign_type	Yes	String(16)	MD5: MD5 SHA256 : SHA256 RSA: RSA_1_256 Default value: MD5
status	Yes	String(16)	"0": success. Others value: fail. Specifies communicating label (not transaction label). The status of a transaction is determined by the value of trade_state.
message	No	String(128)	Return message. Only return when the signature verification invalid.
The following fields will be returned when status is "0"			
result_code	Yes	String(16)	"0": success. Others value: fail.
groupno	No	String (32)	Master merchant ID

mch_id	Yes	String(32)	Merchant ID, Specifies an unique id assigned by platform. Only Store ID or Ordinary Merchant ID is valid.
device_info	No	String(32)	Specifies a Terminal device id. You can query based on this field on the swiftpass merchant platform
nonce_str	Yes	String(32)	Included in platform payment API protocols to ensure unpredictability for signatures. 32 characters or fewer.
err_code	No	String(32)	Reference error code. It will be returned only when result_code is different from 0.
err_msg	No	String (128)	Error information description. It will be returned only when result_code is different from 0.
sign	Yes	String(344)	Please refer to section 4 'Digital Signature'.
sign_agentno	No	String(32)	When the merchant is applied the Aggregator mode that the merchant will be represented by the channel, the agency channel number is sent, and the corresponding sign needs to be signed with the channel's sign_key.
The following fields will be returned when status and result_code both are "0"			
trade_state	Yes	String(32)	SUCCESS:Payment successful REFUND:Order to be refunded NOTPAY:Order not paid CLOSED:Order closed PAYERROR:Failed payment
The following fields will be returned when trade_state is "SUCCESS"			
trade_type	Yes	String(32)	Value: pay.weixin.app
appid	No	String(32)	Institution official account ID
sub_appid	Yes	String(32)	Application appid applied by merchants on WeChat open platform
openid	Yes	String(128)	It is the only user identification under the current appid.
is_subscribe	No	String(1)	Specifies whether the payer follows the associated official account or not, with "Y" meaning 'follows' and "N" meaning "not follows".
transaction_id	Yes	String(32)	The unique trade reference of platform system.

out_transaction_id	No	String(32)	The unique trade reference of vendor system. (Return in successful transaction, others will not)
out_trade_no	Yes	String(32)	The unique trade reference of merchant system.
total_fee	Yes	Int	The total amount of the transaction. The unit of the fee is the minimal unit of the currency
order_fee	No	Int	Order amount in Cent
cash_fee	No	Int	Order cash payment amount.
local_total_fee	No	Int	Local total order amount in Cent.
rate	No	Int	rate
cash_fee_type	No	String(8)	Currency type of order cash payment.
coupon_fee	No	Int	Coupon amount. coupon_fee <= total_fee. total_fee - oupon_fee = cash pay amount
fee_type	No	String(8)	Complies with ISO 4217 standards.
local_fee_type	No	String(8)	Local currency type, three-letter code in accordance with ISO 4217
attach	No	String(127)	Additional merchant info. The original value will be returned in the query response after successful payment. This field is mainly used for custom data of the order carried by the merchant.
bank_type	Yes	String(16)	String states bank_type
bank_billno	No	String(32)	Bank order number. Null for WeChat payment.
time_end	Yes	String(14)	Specifies the transaction payment time in the format of yyyyMMddHHmmss, such as 20091225091010 for Dec 25, 2009 09:10:10.(GMT+8 Beijing). Returned when orders tatus is payment successful or refunded.

6.4 Refund Interface

6.4.1 Business function

If the commercial tenant initiate refund in respect of an order that has been successfully paid, then the result of the operation will be synchronously returned in the same dialogue.

I. Refund mode

Currently, only the mode of refunding to the original cash source and the mode of refunding to the WeChat balance account are supported.

Notes: to refund to the bank card will not be real-time, and due to that the speed of disposal of each bank is different, generally the amount will be refunded to the bank account within 1 to 3 working days after the refund is launched.

The different parts of refund of a same order need to be set with the same order number and the different out_refund_no. After a refund that fails is resubmitted, the original out_refund_no will be adopted. The total sum of refund couldn't surpass the amount actually paid by users (the amount of the cash coupon couldn't be refunded).

II. Restrictions of refund

When operating the refund, the commercial tenant should pay attention to the restrictions of refund to avoid the request of refund that will not be successful, and the main restrictions of refund are as follows:

In platform system, only if the accumulative amount of refund doesn't surpass the total sum of payment for the transacted order, then the same order could be refunded for several times, and the refund application form number (there are such parameters in the refund interface) rather than the transaction order number shall be solely used to confirm one time of refund. The refund application form number is to be generated by the commercial tenant, and so the commercial tenant has to guarantee the uniqueness of the refund application form. In the process of refund, the commercial tenant needs to pay special attention that only when the refund is confirmed to fail, then could another refund be re-initiated.

Currently most banks support full refund and partial refund, but there are a few banks that don't support full refund or partial refund or don't support refund. In such case the commercial tenant could negotiate with the buyer to refund to the WeChat balance account.

Currently only refund interface without key is provided, and in case the commercial tenant needs refund interface with key, please get in touch with the business person to explain.

6.4.2 Interactive mode

Background interaction mode

6.4.3 Request Parameters

Request URL : <https://gateway.wepayez.com/pay/gateway>

POST request with content of XML

Field Name	Required	Type	Description
service	Yes	String(32)	Value: unified.trade.refund
version	No	String(8)	Version number. default value: 2.0
charset	No	String(8)	Encoding method. Default Value: UTF-8
sign_type	No	String(16)	MD5: MD5 SHA256 : SHA256 RSA: RSA_1_256 Default value: MD5
mch_id	Yes	String(32)	Merchant ID, Specifies an unique id assigned by platform. Only Store ID or Ordinary Merchant ID is valid.
groupno	No	String (32)	Master merchant ID
out_trade_no	No	String(32)	The unique trade reference of merchant system. At least one of the out_trade_no and transaction_id should exist in refund request.

			transaction_id priority when both be filled.
transaction_id	No	String(32)	The unique trade reference of platform system. At least one of the out_trade_no and transaction_id should exist in refund request. transaction_id priority when both be filled.
out_refund_no	Yes	String(32)	Specifies the internal refund number, which is unique in the system. A single transaction can be processed as multiple partial refunds, with the total sum of the partial refunds being equal to the original one. If the refund is not successful. The recall function should be used with same refund number to avoid duplication of refunds.
total_fee	Yes	Int	The total amount for a transaction. The unit of the fee is the minimal unit of the currency.
refund_fee	Yes	Int	Refund amount. The unit of the fee is the minimal unit of the local Currency. Partial refund can be supported.
op_user_id	Yes	Int	Specifies the Operator ID. This field shows mch_id by default.
refund_channel	No	String(16)	Value: ORIGINAL. The money will refund back to where it came from.
nonce_str	Yes	String(32)	Included in platform payment API protocols to ensure unpredictability for signatures. 32 characters or fewer.
sign	Yes	String(344)	Please refer to section 4 'Digital Signature'.
sign_agentno	No	String(32)	When the merchant is applied the Aggregator mode that the merchant will be represented by the channel, the agency channel number is sent, and the corresponding sign needs to be signed with the channel's sign_key.

6.4.4 Response result

Data return in real time with XML format

Field Name	Required	Type	Description
version	Yes	String(8)	Version number. default value: 2.0
charset	Yes	String(8)	Encoding method. Default Value: UTF-8
sign_type	Yes	String(16)	MD5: MD5

			SHA256 : SHA256 RSA: RSA_1_256 Default value: MD5
status	Yes	String(16)	"0": success. Others value: fail. Specifies communicating label (not transaction label). The status of a transaction is determined by the value of result_code.
message	No	String(128)	Return message. Only return when the signature verification invalid.
The following fields will be returned when status is "0"			
result_code	Yes	String(16)	"0": success. Others value: fail.
groupno	No	String (32)	Master merchant ID
mch_id	Yes	String(32)	Merchant ID, Specifies an unique id assigned by platform. Only Store ID or Ordinary Merchant ID is valid.
device_info	No	String(32)	Specifies a Terminal device id. You can query based on this field on the swiftpass merchant platform
nonce_str	Yes	String(32)	Included in platform payment API protocols to ensure unpredictability for signatures. 32 characters or fewer.
err_code	No	String(32)	Reference error code. It will be returned only when result_code is different from 0.
err_msg	No	String (128)	Error information description. It will be returned only when result_code is different from 0.
sign	Yes	String(344)	Please refer to section 4 'Digital Signature'.
sign_agentno	No	String(32)	When the merchant is applied the Aggregator mode that the merchant will be represented by the channel, the agency channel number is sent, and the corresponding sign needs to be signed with the channel's sign_key.
The following fields will be returned when status and result_code both are "0"			
trade_type	Yes	String(32)	Value: pay.weixin.app
transaction_id	Yes	String(32)	The unique trade reference of platform system.
out_trade_no	Yes	String(32)	The unique trade reference of merchant system.

out_refund_no	Yes	String(32)	Specifies the internal refund number, which is unique in the merchant system.
refund_id	Yes	String(32)	Specifies the internal refund number, which is unique in the platform system.
out_transaction_id	No	String(32)	Order number provided by WeChat.
refund_channel	Yes	String(16)	Value: ORIGINAL. The money will refund back to where it came from.
refund_fee	Yes	Int	Refund amount. The unit of the fee is the minimal unit of the local currency. Partial refund can be supported.
order_fee	No	Int	Order amount in Cent
local_total_fee	No	Int	Local total order amount in Cent.
fee_type	No	String(8)	Complies with ISO 4217 standards.
local_fee_type	No	String(8)	Local currency type, three-letter code in accordance with ISO 4217
coupon_refund_fee	No	Int	Coupon refund amount. coupon_refund_fee <= refund_fee. refund_fee - coupon_refund_fee = cash refund amount

6.5 Close Order Interface

6.5.1 Business function

If the payment of order of the commercial tenant fails, then it's required to generate a new order number to re-initiate payment, and the original order number needs to be invoked for closing to avoid repeated payment. After an order is placed in the system, if the payment of users is time out, the system will exit and no longer accept the payment to avoid ongoing payment of users, and please invoke close order interface.

6.5.2 Interactive mode

Request: Background interaction mode

Response: Background interaction mode

6.5.3 Request Parameters

Request URL : <https://gateway.wepayez.com/pay/gateway>

POST request with content of XML

Field Name	Required	Type	Description
Normal Parameters			
service	Yes	String(32)	Value: unified.trade.close
version	No	String(8)	Version number. default value: 2.0
charset	No	String(8)	Encoding method. Default Value: UTF-8
sign_type	No	String(16)	MD5: MD5 SHA256 : SHA256 RSA: RSA_1_256 Default value: MD5
mch_id	Yes	String(32)	Merchant ID, specifies a unique id assigned by platform. Only Store ID or Ordinary Merchant ID is valid.
groupno	No	String (32)	Master merchant ID
out_trade_no	Yes	String(32)	The unique trade reference of merchant system.
nonce_str	Yes	String(32)	Included in platform payment API protocols to ensure unpredictability for signatures. 32 characters or fewer.
sign	Yes	String(344)	Please refer to section 4 'Digital Signature'.
sign_agentno	No	String(32)	When the merchant is applied the Aggregator mode that the merchant will be represented by the channel, the agency channel number is sent, and the corresponding sign needs to be signed with the channel's sign_key.

6.5.4 Response result

Data return in real time with XML format

Field Name	Required	Type	Description
version	Yes	String(8)	Version number. default value: 2.0
charset	Yes	String(8)	Encoding method. Default Value:
sign_type	Yes	String(16)	MD5: MD5 SHA256 : SHA256 RSA: RSA_1_256 Default value: MD5
status	Yes	String(16)	"0": success. Others value: fail. Specifies communicating label (not transaction label). The status of a transaction is determined by the value of result_code.
message	No	String(128)	Return message. Only return when the signature verification invalid.
The following fields will be returned when status is "0"			
result_code	Yes	String(16)	"0" for SUCCESS. others for FAIL. SUCCESS indicates the order was cancelled for successfully and cannot be paid for again. FAIL refers to exceptions that occur in the interface. The recall function should be used to determine whether the order has been canceled or not.
groupno	No	String (32)	Master merchant ID
mch_id	Yes	String(32)	Merchant ID, Specifies an unique id assigned by platform. Only Store ID or Ordinary Merchant ID is valid.
nonce_str	Yes	String(32)	Included in platform payment API protocols to ensure unpredictability for signatures. 32 characters or fewer.
err_code	No	String(32)	Reference error code. It will be returned only when result_code is different from 0.
err_msg	No	String (128)	Error information description. It will be returned only when result_code is different from 0.

sign	Yes	String(344)	Please refer to section 4 'Digital Signature'.
sign_agentno	No	String(32)	When the merchant is applied the Aggregator mode that the merchant will be represented by the channel, the agency channel number is sent, and the corresponding sign needs to be signed with the channel's sign_key.

7 Notes

1. The unit of any related amount is the minimal unit of the local currency, and decimals are not allowed.
2. notify_url means that platform server directly initiates request from back end to commercial tenant's server, and when disposing, the commercial tenant could not examine the user's cookie or session; the commercial tenant's updating of DB and other goods delivery procedure needs to be made after notify_url is completed to ensure that to supplement order after order fails will be successful.
3. notify_url means that it might have repeated notification and the commercial tenant needs to do away with the repeated ones to avoid repeated goods delivery.
4. notify_url means the receive notification, and if the commercial tenant disposes successfully or if the examined order has been disposed, then the successful disposal mark, the pure character string success needs to be returned, and the character string success is not case sensitive; if we don't receive the returned success, then our server will keep sending notification to you until three hours later; if it's assumed that all orders don't return success, then the load of notification of our server will be increase, and the worst case is that the notification normally sent to the commercial tenants might delay; besides, we will urge you to perfect, and if you don't improve for a long time period, then the R&D or operation and maintenance technology staff will adopt control measures over the payment interface opened by your company. For the parameters sent in requesting interface in the document, if the one required to fill is given a yes, then it has to be sent (in case of lack, an error will be alarmed), and if the one required to fill is given a no, then it will be optional to be sent.

5. For the parameters sent in requesting interface in the document, if the one required to fill is given a yes, then it has to be sent (in case of lack, an error will be alarmed), and if the one required to fill is given a no, then it will be optional to be sent.
6. For the returned parameters, if the one required to fill is given a yes, then it has to be returned, and if the one required to fill is given a no, then it will be optional to be returned. Because of upgrading or configuration or other cases, the actually returned parameters might not be totally consistent with those in the document, and the actually received parameters shall prevail.
7. As for user opened: after the followers and the official account have information interaction, the official account could obtain the follower's OpenID (the encrypted WeChat account, and each user's OpenID for each official account is exclusive. For different official accounts, the same user's openid will be different), and to obtain openid could refer to the following address: https://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1421140842. To use the test-version commercial tenant account needs not to send the user's openid; to switch into the official commercial tenant account needs to obtain openid, and it's required to add sub_openid field in the requested parameters and to send the obtained openid to sub_openid. Before switching into the official commercial tenant account and sending sub_openid parameter, it's required to provide official commercial tenant account and official account (the service account) and the appid is to be configured by service providers (provide configuration table from email), and if there is no configuration, the error of sub_appid and su_openid not match will be reported, which will cause the invocation of interface to be abnormal.
8. Other notes
 - (1) The problem of capital and small letter of parameters
Please pay attention to the problem of capital and small letter required in the document, such as **“after signature, the character of the character string needs to be converted to uppercase”**.
 - (2) The problem of format of parameters

All introduced parameters are of the type of character string, and please pay attention to the specific requirements in different places of the document.

(3)The problem of timestamp

Please use Linux timestamp, and note that its format is character string.

(4) The problem of payment of order number of the same commercial tenant

9. If the payment of order of the commercial tenant fails, then it's required to generate a new order number to reinitiate payment, and the original order number needs to be invoked for closing to avoid repeated payment; After an order is placed in the system, if the payment of users is time out, the system will exit and no longer accept the payment to avoid ongoing payment of users, and please invoke the order closing interface.

10. Notes: after an order is generated, to immediately invoke the order closing interface will not be allowed, because the shortest invocation interv

11. Request swiftpass gateway If there is no clear result of synchronization within 10 seconds, it can be considered that the transaction request has timed out

8 Error Code

8.1 Swiftpass error code

Error Code	Error Message
Auth code invalid	Invalid payment code
400	Payment code cannot be blank
400	Missing pre-order info
400	Require xml content
400	Require POST method
400	Parse xml error

400	unsupported sign method
400	Signature error
400	Parse params error
400	Parse xml error, please use UTF-8 encoded
400	XXX: This field is required
400	Total fee: Invalid value
400	Verification failed
INVALID_FEE	Amount error
INVALID_REFUND_FEE	Refund amount error
REFUND_FEE_LIMIT	Refund request intercepted, please check if settings are correct
REFUND_FEE_INVALID	Invalid refund amount
Validation parameter failed	Parse params error
ORDER_FEE_INVALID	Order amount invalid
ORDER_DATE_INVALID	Order date invalid
transaction_id out_trade_no is required	Missing parameters
buyer_logon_id buyer_id is required	Missing parameters
transaction_id out_trade_no out_refund_no refund_id is required	Missing parameters
out_refund_no refund_id is required	Missing parameters
Refundid and refundpwd is required	Missing parameters
thi_mch_id is required	Missing parameters
400	Sign error
Refund exists	Refund already existed
Order exists	Order already existed

Refund not exists	Refund do not exist
400	Pre-paid order number do not exist
Order not exists	Order do not exist
400	Missing pre-paid order number
400	Order do not exist
500	Internal error
500	SYSTEMERROR
SYSTEMERROR	System error
Internal error	WeChat request error, URL error
Internal error	WeChat request error, connection failed
Internal error	WeChat request error, timeout
Internal error	WeChat request error, protocol error
Internal error	WeChat request error, unknown error
Internal error	Third-party request error, URL error
Internal error	Third-party request error, connection failed
Internal error	Third-party request error, timeout
Internal error	Third-party request error, protocol error
Internal error	Third-party request error, unknown error
Order reverse	Order is reversed
400	Unsupported API
400	The merchant has not opened the [XXX] payment type
400	Merchant is suspended
400	Merchant cannot initiate order alone
400	This order is paid

Order paid	Order already paid
400	Order amount exceeds merchant limit
400	Order amount is below merchant limit
Amount limit	PRODUCT_AMOUNT_LIMIT_ERROR
BALANCE_NOT_ENOUGH	Insufficient balance
USERPAYING	WeChat requires users to enter a password
Refund status error	Refund status error
REFUND_ERROR	REFUND ERROR
Order date limit	Order expired
Order status error	Order status error
Reverse fail	Reverse failed
Order close fail	Failed to close order
400	This order cannot be reversed
Refund limit	Do not support this function
400	Order status error, cannot close order
ORDER ERROR	Order failed
refund frequency high	Refund too frequent
400	Request too frequent
REQUEST CHANGE ERROR	Do not match with original order
Order trade type error	Cannot match payment type

8.2 WeChat error code

Please check below link for details.

https://pay.weixin.qq.com/wiki/doc/api/wxpay/en/fusion_wallet/NativePay/chapter4_3.shtml

Error code	Description	Type
SYSTEMERROR	The returned value of interface (system) is wrong	WeChat
PARAM_ERROR	Parameter error	WeChat
ORDERPAID	order has been paid	WeChat
NOAUTH	The merchant has no authority	WeChat
AUTHCODEEXPIRE	The QR code has expired, and the user is to try again after refresh on WeChat	WeChat
NOTENOUGH	The balance isn't enough	WeChat
NOTSUPPORTCARD	The type of card is not supported	WeChat
ORDERCLOSED	The order has been closed	WeChat
ORDERREVERSED	The order has been revoked	WeChat
BANKERROR	The bank system is abnormal	WeChat
USERPAYING	The user is paying and the password is required to input	WeChat
AUTH_CODE_ERROR	The parameter of the authorization code is wrong	WeChat
AUTH_CODE_INVALID	The verification of the authorization code is wrong	WeChat
XML_FORMAT_ERROR	XML format is wrong	WeChat
REQUIRE_POST_METHOD	Please use post method	WeChat
Signature error	Signature error	WeChat
LACK_PARAMS	The parameters are not enough	WeChat
NOT_UTF8	The coded format is wrong	WeChat
BUYER_MISMATCH	The payment account mismatches	WeChat

APPID_NOT_EXIST	APPID doesn't exist	WeChat
MCHID_NOT_EXIST	MCHID doesn't exist	WeChat
OUT_TRADE_NO_USED	The order number of the merchant repeats	WeChat
APPID_MCHID_NOT_MATCH	appid and mch_id don't match	WeChat
ORDERNOTEXIST	The order number of the transaction doesn't exist	WeChat