



Merchant API Documentation

WeChat WAP Payment API Specification

Document version: 1.5.3

Interface version: 2.0

TABLE OF CONTENTS

1	Introduction	4
1.1	Reading objects	4
1.2	Payment mode	4
1.3	Get IP Description	5
1.3.1	No proxy	5
1.3.2	Proxy situation	5
2	Scene introduction	8
2.1	Business scenario	8
2.2	Business process description	8
3	Data Format	10
3.1	Submitting data	10
3.2	XML data format	11
4	Digital Signature	12
4.1	Signature original string	13
4.2	Signature algorithm	14
4.2.1	MD5 signature	14
4.2.2	SHA256 signature	15
4.2.3	RSA signature	16
5	Replenishment Mechanism	19
6	WAP interface	19
6.1	WAP Payment interface	19
6.1.1	Business Functions	19
6.1.2	Interactive mode	19
6.1.3	Request parameter list	20
6.1.4	Return result	23
6.2	WAP payment notification interface	24
6.2.1	Notification result parameter list	24
6.2.3	Back-end notification result feedback	28
6.3	Order query interface	29
6.3.1	Business function	29
6.3.2	Interactive mode	29
6.3.3	Request parameter list	29
6.3.4	Return result	30
6.4	refund result interface	33
6.4.1	Business function	33
6.4.2	Interactive mode	34
6.4.3	Request parameter list	34
6.4.4	Return result	36
6.5	Close Order Interface	38

6.5.1 Business function.....	38
6.5.2 Interactive mode	38
6.5.3 Request Parameters.....	38
6.5.4 Response result.....	39
7 Notes	40
8 Error Code	42
8.1 Swiftpass error code	42
8.2 WeChat error code.....	45

Document Changes		
Time	Version	Description
2018-09-15	1.0	First draft
2022-07-06	1.5	Update description of fields mch_id, device_info, time_start, time_expire, etc.
2022-07-06	1.5	Add the precautions for using the query interface
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
2022-12-26	1.5.3	Modify the field "pay_url" as "pay_info", Update fields trade_type, local_fee_type, local_total_fee, order_fee, etc
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, err_code, err_msg, out_trade_no, transaction_id, refund_id, out_trade_refund;

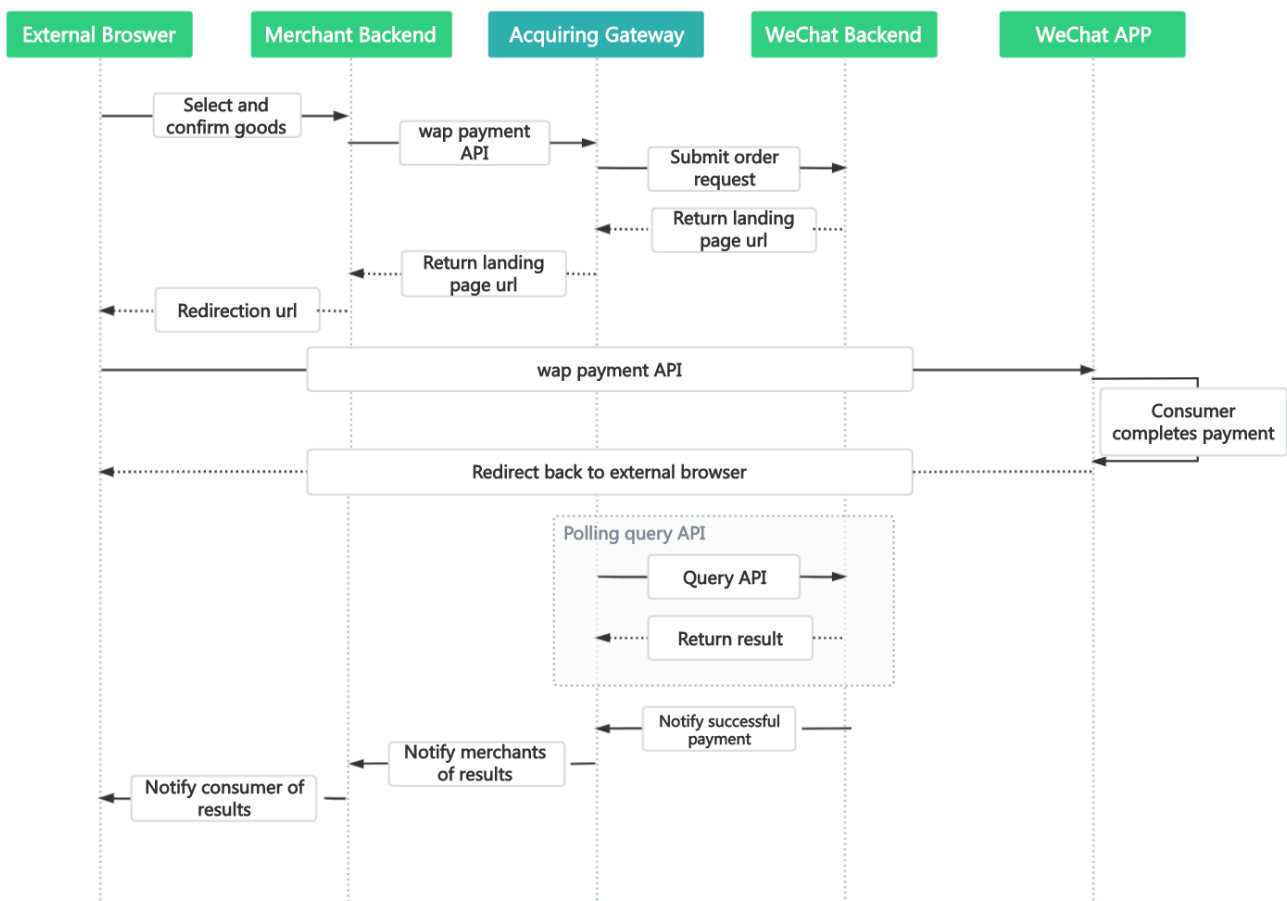
1 Introduction

1.1 Reading objects

For the reference and inquiry to the merchant platform service provider's technical or business team.

1.2 Business realization process

WAP payment is a mobile payment method, consumers can browse the merchant's mobile web page, jump from the merchant's page to the WeChat wallet APP to complete the payment action.



1.3 Get IP Description

1.3.1 No proxy

The way to get IP address is relatively simple when the front-end access layer of the merchant is not using a proxy, just get 'REMOTE_ADDR' directly.

```
1. function get_client_ip()
2. {
3.     $cip = "unknown";
4.     if ($_SERVER['REMOTE_ADDR'])
5.     {
6.         $cip = $_SERVER['REMOTE_ADDR'];
7.     }
8.     elseif (getenv("REMOTE_ADDR"))
9.     {
10.        $cip = getenv("REMOTE_ADDR");
11.    }
12.    return $ip
13. }
```

1.3.2 Proxy situation

In the case of a proxy, because it is necessary to replace the client to access the server, when the request packet passes through the reverse proxy, the IP header of the IP packet is modified in the proxy server, and the data obtained by the back-end WEB server is the source IP address of the packet header is the IP address of the proxy server. In this way, the back-end server program cannot obtain the user's real IP.

When nginx has a proxy:

Add to the configuration in nginx

```
proxy_set_header Host $host;

proxy_set_header X-Real-IP $remote_addr;
```

```
proxy_set_header X-Real-Port $remote_port;

proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
```

Apache In case of proxy :

```
vi /usr/local/apache/conf/httpd.conf

Include conf/extra/httpd-remoteip.conf

vi /usr/local/apache/conf/extra/httpd-remoteip.conf

LoadModule remoteip_module modules/mod_remoteip.so

RemoteIPHeader X-Forwarded-For

RemoteIPInternalProxy 127.0.0.1
```

Code example

```
string GetClientIp(CgInput* pInput)

{

string client_ip = "";

string strClientIPList;

GetHttpHeader("X-Forwarded-For", strClientIPList);

if (strClientIPList.empty())

{

GetHttpHeader("X-Real-IP", strClientIPList);

}

if (!strClientIPList.empty())

{

size_t iPos = strClientIPList.find( "," );

if( iPos != std::string::npos )

{

client_ip = strClientIPList.substr( iPos );

}

}

else
```

```
{
client_ip = strClientIPList;
}
}
if (client_ip.empty())
{
GetHTTPHeader("PROXY_FORWARDED_FOR", strClientIPList);
// Compatible
if(strClientIPList.empty())
{
client_ip = getRemoteAddr();
}
else
{
size_t iPos = strClientIPList.find( ",");
if( iPos != std::string::npos )
{
client_ip = strClientIPList.substr( iPos );
}
else
{
client_ip = strClientIPList;
}
}
}
if(!MMPayCommFunc::IsIp(client_ip))
client_ip = getRemoteAddr();
return client_ip;
}
```


2 Scene introduction

2.1 Business scenario

It is applicable to the scenario that consumers in the external browser of WeChat, after selecting payment, the browser can directly pull up the WeChat payment wallet to complete the payment.

Experience WeChat official link: <https://wxpay.wxutil.com/mch/pay/h5.v2.php>, please open outside the WeChat internal browser.

2.2 Business process description

Step (1): The user navigates to the merchant's H5 webpage with a non-WeChat browser site, the user selects the goods to be purchased, and selects WeChat Pay to initiate the purchase process.



Step (2): Enter the WeChat client to confirm the transaction and enter the payment password.



Step (3): The payment is successful, the user receives the payment result, and the merchant receives a notification of the successful payment at the same time.



3 Data Format

3.1 Submitting data

With POST protocol of HTTPS standard, in order to ensure the receiver receives the data correctly, the transmitted data must be signed.

```
<xml>
<body>Hong Kong</body>
```

```
<charset>UTF-8</charset>

<mch_create_ip>23.74.145.64</mch_create_ip>

<mch_id>7551000001</mch_id>

<nonce_str>AJmEk0V76uwzXRXh6/l5OA==</nonce_str>

<notify_url>https://xxxxxxxx.com/xsxxxxxxxx</notify_url>

<out_trade_no>202755100000100495</out_trade_no>

<service>pay.weixin.wap.intl</service>

<sign>8DFB1678839237F29D2531C0E360EE625E61488C4FB5C783313C02348C22B217</sign>

<sign_type>SHA256</sign_type>

<total_fee>15800</total_fee>

</xml>
```

3.2 XML data format

With the standard XML protocol, all parameters only exist at the first-level node, and the multi-level node nesting is not applied.

Protocol-level error return:

```
<xml>

<status>500</status>

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

</xml>
```

Correct data return:

```
<xml>

<appid><![CDATA[wxf4c1111117]]></appid>

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

<mch_id><![CDATA[7551000001]]></mch_id>

<nonce_str><![CDATA[JwQAqPk06vUIVWrb]]></nonce_str>
```

```
<pay_info><![CDATA[https://wx.tenpay.com/cgi-bin/mmpayweb-  
bin/checkmweb?prepay_id=wx1018061111113c30000&package=2211111111]]></pay_info>  
  
<result_code><![CDATA[0]]></result_code>  
  
<sign><![CDATA[BF68B5D8D38033E0DCC600D8C1A1FC1A]]></sign>  
  
<sign_type><![CDATA[MD5]]></sign_type>  
  
<status><![CDATA[0]]></status>  
  
<version><![CDATA[2.0]]></version>  
  
</xml>
```

Business-level error return:

```
<xml>  
  
<status>0</status>  
  
<message><![CDATA[OK]]></message>  
  
<mch_id><![CDATA[755100001]]></mch_id>  
  
<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>
```

Generally, there is a status parameter during return, wherein 0 means call success and non-0 means call failure.

4 Digital Signature

In order to ensure data authenticity and integrity during data transmission, we need to digitally sign the data and conduct signature verification after receiving the signature data.

A digital signature contains two steps. First, the original string to be signed is spliced according to certain rules, and then the specific algorithm and the key are selected to calculate the signature result.

Generally, the failed result is not signed.

4.1 Signature original string

Whether it is a request or an answer, the signature original string is assembled into a character string as follows:

1. Except for sign field, all parameters are spliced at the format of QueryString (i.e., key1=value1&key2=value2...) from small to large according to ascii code of the field name, and the null value is not passed and does not participate in the group signature string.
2. In the original string of signature, the field name and the field value are both original values, and the URL Encode is not performed.
3. The response or notification message returned by the platform may be increased parameters due to upgrade. Please allow this case when verifying the response signature.

Example:

Call a certain interface which has the following fields:

```
<xml>
<body><![CDATA[test payment]]></body>
<mch_create_ip><![CDATA[127.0.0.1]]></mch_create_ip>
<mch_id><![CDATA[101520000465]]></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[pay.weixin.wap.intl]]></service>
<sign><![CDATA[94B0F11B228BA9DDA2E20E3A9C8B3A2F]]></sign>
<total_fee><![CDATA[1]]></total_fee>
</xml>
```

The correct signature fields are sorted as:

```
body=test
```

```
payment&mch_create_ip=127.0.0.1&mch_id=101520000465&nonce_str=1409196838&notify_url=http://227.0.0.1:9001/javak/&out_trade_no=14190360
```

```
6228&service=pay.weixin.wap.intl&total_fee=1
```

4.2 Signature algorithm

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>

  <body>Hong Kong</body>

  <charset>UTF-8</charset>

  <mch_create_ip>23.74.145.64</mch_create_ip>

  <mch_id>7551000001</mch_id>

  <nonce_str>AJmEk0V76uwzXRXh6/I5OA==</nonce_str>

  <notify_url>https://xxxxxxxx.com/xxssxxxx</notify_url>

  <out_trade_no>202755100000100495</out_trade_no>

  <service>pay.weixin.wap.intl</service>

  <sign>5FF36B08B5C24D5B3498FEB7B9B1B94 </sign>

  <sign_type>MD5</sign_type>

  <total_fee>15800</total_fee>

</xml>
```

Suppose merchant key is: 9f72151b6592fab3e0c63a1ab3c0877b

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

```
body= Hong
```

```
Kong&mch_create_ip=23.74.145.64&mch_id=7551000001&nonce_str=AJmEk0V76uwzXRXh6/I5OA==&notify_url=https://xxxxxxxx.com/xxssxxxx&out_trade_no=202755100000100495&service= pay.weixin.wap.intl &total_fee=15800
```

ii: sign after process b:

```
sign
=md5(string1&key=9f72151b6592fab3e0c63a1ab3c0877b).toUpperCase
=md5(body= Hong
Kong&mch_create_ip=23.74.145.64&mch_id=7551000001&nonce_str=AJmEk0V76uwzXRh6/I5OA==&notify_url=https://xxxxxxxx.com/xxssxxxx&o
ut_trade_no=202755100000100495&service=pay.weixin.wap.intl&total_fee=15800&key=9d101c97133837e13dde2d32a5054abb).toUpperCase()
="5FF36B08B5C24D5B3498FEB7B9B1B94 "
```

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>
<body>Hong Kong</body>
<charset>UTF-8</charset>
<mch_create_ip>23.74.145.64</mch_create_ip>
<mch_id>7551000001</mch_id>
<nonce_str>AJmEk0V76uwzXRh6/I5OA==</nonce_str>
<notify_url>https://xxxxxxxx.com/xxssxxxx</notify_url>
<out_trade_no>202755100000100495</out_trade_no>
<service>pay.weixin.wap.intl</service>
<sign>357A3B15CD0325A509926302DCBBB053C923237ED38DB6D100806385A5255E4E</sign>
<sign_type>SHA256</sign_type>
<total_fee>15800</total_fee>
</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= Hong
```

```
Kong&mch_create_ip=23.74.145.64&mch_id=7551000001&nonce_str=AJmEk0V76uwzXRXh6/I5OA==&notify_url=https://xxxxxxxx.com/xxxxxxx&out_trade_no=202755100000100495&service=pay.weixin.wap.intl&total_fee=15800
```

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= Hong
```

```
Kong&mch_create_ip=23.74.145.64&mch_id=7551000001&nonce_str=AJmEk0V76uwzXRXh6/I5OA==&notify_url=https://xxxxxxxx.com/xxxxxxx&out_trade_no=202755100000100495&service=pay.weixin.wap.intl&total_fee=15800&key=
```

```
9f72151b6592fab3e0c63a1ab3c0877b).toUpperCase()
```

```
="357A3B15CD0325A509926302DCBBB053C923237ED38DB6D100806385A5255E4E"
```

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>

<out_trade_no><![CDATA[6057113230875088]]></out_trade_no>

<nonce_str><![CDATA[wNzpaD0sN17K180yBQwiNNHfmOleNqap]]></nonce_str>

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

<mch_create_ip><![CDATA[127.0.0.1]]></mch_create_ip>

<sign_type><![CDATA[RSA_1_256]]></sign_type>

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

<notify_url><![CDATA[http://www.baidu.cn/notify.aspx]]></notify_url>

<body><![CDATA[Test pay]]></body>

<version><![CDATA[1.0]]></version>

<mch_id><![CDATA[102532336411]]></mch_id>

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

<attach><![CDATA[Additional information]]></attach>

<service><![CDATA[pay.weixin.wap.intl]]></service>

<sign><![CDATA[SE008JDir0uwOuBy8d48SmdfG37PyGvhtqu8pDTo0DpdNkgkLuegNsb6SaL/dEfzuO35bwAVwiKSc9m9xBKohGJEMtzRm3tLNNpQ0B

DpLIYNWQkr0JN3JAjy6wk1icSmfbjXgEvWCcx17MUe59NflI0JRR3MgFg/ySYq2cT4U/o6WJisxtlbre0ZPM66WL8l5gAzosJW3Gwr+B8fkRyOckF4w64i0T

M20xoSHGEa8w1utOFik5vwxyPHb/JcjhRyrBnb2LFZy5Rq3XzZLW6FaK1gunWnjK5+4NiNuHDgm7CKZPp8BphA/qnJtAMGfexEZ8J4z9ktyyWDK8b8VW

J5ow==]]>

</sign>

</xml>
```

Merchant RSA Private key:

MIIeVqIBADANBgkqhkiG9w0BAQEFAASCBCwggSjAgEAAoIBAQCfU8v4BUr81SKm/H0ahbdQZjEpO8nMyk+XuYSatHwnU4//m47R+4G2YB4Z6PHsJi4+ScfJpQutFhKrFwTXZ6TDqLvaqZDDkjq5G271g+PmrzFp7f40/E9m0qjeL64Rjra0rZql23dvPW4vVomMRgRcoPOn0YVWp+M6T5PaFgE4M8dh4IMZz57gVwOdd08F99Z92f3QgZtEj+/-EXvMenXxb/aRofNkt+Wdk2ELJ6MIP0d9UU5v3WgLuuNv5QnQYzj/RMr8GD+wrDYiNQJxsaTmE/OEJggsumhD4eYY5YIRy2EIN504cujYVKU1wOSZgq9oJCynGR0aPuQWx58IHxEtAgMBAECCggEAHFEfd8qm2PTE2ITAVEC7F+TcgD84IUAz0dZnURtx6YIOz5+LH/zVG6juYLJU/Oo5RPAc+iMVS68u2JMCp7zm8Ft7B3JkrbuHLNHGuR6Q7PQuXN8PkDcOxqDmZ2kPJzI4PZvBZRE0abduG+tiMatGzpGAuJzrWcB/N0oVlvrXp9PnOqfo/Y5nxmpOFClmJppiS3AL1pftNtQZ09G15CPHDYtpUbXPtD2MjjW4OLxKuPRoHSwUgo6LW9XSwNXfcuK+lzbLL0BhIWD9IV/+yCEUEbIN87yxxfhpQFAhXj5W+B3YsMOZuK93+XMOpyMw8EPUDMOBOnvwb0NSHUrV2RUAQKBgQDT0jlnNS1e7+tjPzFtOhGPj1uCBPAEIEHAcnPgd80bEiujxMLCn

GaAvmnTrMu4Xo0e5fAP4F7R6UD+IUufr3CAAu7CadQ49TW+SovAvicy9AZuSVVlwyu6QdYgFyPKe1LZYAEq5k+mB1Vh5q0RoxMNA5pGYK8+4MmmsJi7X7QKBgQDAunCOqliH128bs/1VRlhDpzuRW5Qr/SRbO2saVg5RSHnO/nGT2OuxSTTkc8yrx7qd9SmAxXI5kR238DhMOQOnRBomldmVtAJuJgrdQyt0wXfeVQVQqshqCUaE/xhEbpSCdbPSZbKZZdplV0y6O5vXlhwx+1qAvXLcxw46s3R92QQKBgQCIQ+ejywkVPDILHMwSSehwvThufkCYWYUbbcVDowpOe5AMoZidtNju7MNjg2rLHTsCx/kBzOr+7THNwl4R7kTiEmg09cO+fu5rHXepGgtig+GJukaZPZ6/bMZJvGOLgOhHmomwG/jdwpvgVtlGBCh6BW5JZcSImT+ykIOoYfvDRuQKBgCgwOHxnBGFfORoLxE3dhpSk8LT05cbuelBVuZW6UC3+8PeK82AjlbLMUy04QHupoG6Dyu3BP/1rl0jd3L94PBzLBD7Gm4vJTqW0DknYo5sMXS1JrnofcKjBv7nbHXZTx3EtJSxpVaOdpcA/HpsCuCP3AH2e1yk9sZ3wu6lBYSBAoGACYM60j1CVRNSZxUNRgiwfWzS69ql1eezPc7xQEganpVBI9SZcTNp1kpDKmQikXJ4Yb5XWn12HCY/sFeBW6Su3ruNqxvg1XiUPbH6A6nxd5B3QX0mS9+wDm6ONysPLRdKbfFO0mdP4CeyuGPdvDIMXP4dJdLhMUL4pcJLI0B7gBE=

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

```
attach=Additional information&body=Test pay
&mch_create_ip=127.0.0.1&mch_id=102532336411&nonce_str=wNzpaD0sN17KI80yBQwINNhfmoIeNqapnotify_url=http://www.baidu.cn/notify.aspx&out_trade_no=6057113230875088&service=pay.weixin.wap.intl&sign_type=RSA_1_256&total_fee=1&version=1.0
```

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(attach=Additional information&body=Test pay&mch_create_ip=127.0.0.1&mch_id=102532336411&nonce_str=wNzpaD0sN17KI80yBQwINNhfmoIeNqapnotify_url=http://www.baidu.cn/notify.aspx&out_trade_no=6057113230875088&service=pay.weixin.wap.intl&sign_type=RSA_1_256&total_fee=1&version=1.0,MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQCfU8v4BUr81SKm/H0ahbdQZjEPO8nMyk+XuYSatHwnU4//m47R+4G2YB4Z6PHsJi4+ScfJpQutFhKrFwTXZ6TDqLvaqZDDkjq5G271g+PmrzFp7f40/E9m0qjeL64Rjra0rzql23dvPW4vVomMRgRcoPOn0YVWp+M6T5PaFgE4M8dh4IMZz57gVwOdd08F99Z92f3QgZtEjl+/EXvMenXxb/aRofNkt+Wdk2ELJ6MIP0d9UU5v3WgLuuv5QnQYzj/RMr8GD+wrDYiNQJxsaTmE/OEJggsumhD4eYY5YIRy2EIN504cuJVYK1wOSZgq9oJCynGR0aPuQWx58IHxEtAgMBAAECggEAHfEf8qm2PTE2ITavec7F+TcgD84IUaz0dZnURtx6YIOoZ5+LH/zVG6juYLJU/Oo5RPAc+iMV568u2JMCp7zm8Ft7B3JkrbuHLNHGuR6Q7PQuXN8PkDcOxqDmZ2kPzJlPzVbZRE0abduq+tiMatGzPGAuJzrWcB/N0oVlvrXp9PnOqfoY5nxmpOFCImJppIS3AL1pftNtQZo9G15CPHDYtpUbXPtD2MjjW4OLxKuPRoHSwUgo6LW9XSwNXfucK+IbzLL0BhlWD9IV/+yCEUEbIN87yxxfhpQFaAhXj5W+B3YsMOZuK93+XMOpyMw8EPudMOBOnvwb0NSHUv2RUAQKBgQDT0jinNS1e7+tjPzFtOhGPj1uCBPAIEhAcnPgd80bEiujxMLCnGaAvmnTrMu4Xo0e5fAP4F7R6UD+IUufr3CAAu7CadQ49TW+SovAvicy9AZuSVVlwyu6QdYgFyPKe1LZYAEq5k+mB1Vh5q0RoxMNA5pGYK8+4MmmsJi7X7QKBgQDAunCOqliH128bs/1VRlhDpzuRW5Qr/SRbO2saVg5RSHnO/nGT2OuxSTTkc8yrx7qd9SmAxXI5kR238DhMOQOnRBomldmVtAJuJgrdQyt0wXfeVQVQqshqCUaE/xhEbpSCdbPSZbKZZdplV0y6O5vXlhwx+1qAvXLcxw46s3R92QQKBgQCIQ+ejywkVPDILHMwSSehwvThufkCYWYUbbcVDowpOe5AMoZidtNju7MNjg2rLHTsCx/kBzOr+7THNwl4R7kTiEmg09cO+fu5rHXepGgtig+GJukaZPZ6/bMZJvGOLgOhHmomwG/jdwpvgVtlGBCh6BW5JZcSImT+ykIOoYfvDRuQKBgCgwOHxnBGFfORoLxE3dhpSk8LT05cbuelBVuZW6UC3+8PeK82AjlbLMUy04QHupoG6Dyu3BP/1rl0jd3L94PBzLBD7Gm4vJTqW0DknYo5sMXS1JrnofcKjBv7nbHXZTx3EtJSxpVaOdpcA/HpsCuCP3AH2e1yk9sZ3wu6lBYSBAoGACYM60j1CVRNSZxUNRgiwfWzS69ql1eezPc7xQEganpVBI9SZcTNp1kpDKmQikXJ4Yb5XWn12HCY/sFeBW6Su3ruNqxvg1XiUPbH6A6nxd5B3QX0mS9+wDm6ONysPLRdKbfFO0mdP4CeyuGPdvDIMXP4dJdLhMUL4pcJLI0B7gBE= )=SE008JDir0uwOuBy8d48SmdfG37PyGvhtqu8pDTo0DpdNkgkLuegNsb6SaL/dEfu035bwAVwiKSc9m9xBKohGJEMtzRm3tLNNpQ0BDpLIYNWQkr0JN3JAjy6wk1icSmfbjXgEvWCcx17MUe59Nfll0JRR3MgFg/ySYq2cT4U/o6WJisxtlbre0ZPM66WL8
```

5 Replenishment Mechanism

Note: It is for back-end notification interaction mode, if the platform receives a response from the merchant that is not a **pure string success** or returns after **over 5 seconds**, the platform considers the notification to be unsuccessful. The platform will indirectly re-initiate notification through certain strategies (**with the notification frequency to be 0/15/15/30/180/1800/1800/1800/1800/3600 and the unit to be second**) to maximize the success rate of notification, but the platform does not guarantee that the notification will ultimately be successful. Due to the resending of back-end notifications, the same notification may be sent to the merchant system multiple times. The merchant system must be able to handle duplicate notifications correctly. The platform recommends, when receiving the notification for processing, first checking the status of the corresponding business data, determining whether the notification has been processed, and re-processing if it has not been processed or directly returning to the pure string success if it has been processed. Before status check and processing of business data, data locks should be used for concurrency control to avoid data confusion caused by **repeated data insertion**.

6 WAP interface

6.1 WAP Payment interface

6.1.1 Business Functions

The Wap payment request need to be initialized. The request is for generating wap payment request of mobile browser beyond of WeChat.

6.1.2 Interactive mode

Request: Back-end request interaction mode

Return result + notification: Back-end request interaction mode + back-end notification interaction mode

6.1.3 Request parameter list

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

POST request with content of XML

Field Name	Variable Name	Required	Type	Description
Business parameters				
Interface type	service	Yes	String(32)	Interface type: pay.weixin.wap.intl
Version number	version	No	String(8)	Version number, the version default value is 2.0
Character set	charset	No	String(8)	Encoding method. Default Value: UTF-8
Signature method	sign_type	No	String(8)	MD5: MD5 SHA256 : SHA256 RSA: RSA_1_256 Default value: MD5
Merchant number	mch_id	Yes	String(32)	Merchant number, assigned by the platform, Only Store ID or Ordinary Merchant ID is valid.
Groupno	groupno	No	String (32)	Master merchant ID
Merchant order number	out_trade_no	Yes	String(32)	Order number inside the merchant system, within 32 characters, may contain letters and to ensuring uniqueness in the merchant system
Device number	device_info	No	String(32)	Terminal device number. You can query based on this field on the swiftpass merchant platform
Goods description	body	Yes	String(128)	Description of merchants' goods. Can upload 128 English, 42 Chinese

Additional information	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 amount	total_fee	Yes	Int	The amount is based on the payment currency and is at Cent. If the payment currency is HKD, 1000 means HKD10.00. Test the maximum limit of merchant number of 100 Cent per transaction (i.e. 1 Yuan per transaction)
Terminal IP	mch_create_ip	Yes	String(16)	Order generated machine IP
Consumer IP	user_ip	Yes	String(16)	That is, the consumer's own IP, which is used for WeChat side risk control and verification of the consumer who invoked the payment.
Notification address	notify_url	Yes	String(255)	Receive the URL notified by the platform, need an absolute path, within 255 characters, and to ensure that the platform can access the address through the Internet.
limit credit card	limit_credit_pay	No	String(1)	Limit whether the user can use a credit card, the value is 1, the credit card is disabled, the value is 0 or this parameter is not passed, it is not disabled
Order generation time	time_start	No	String(14)	Order generation time, the format is yyyyMMddHHmmss, for example: 9:10:10 on December 25, 2009 is indicated as 20091225091010. The time zone is GMT+8 Beijing. Note: The order generation time and the timeout time need to be passed in at the same time to take effect. WeChat can set the time range from 1 minute to 2 hours, if not uploading, the default is 10 minutes
Order timeout time	time_expire	No	String(14)	Order expiration time, the format is yyyyMMddHHmmss, for example: 9:10:10 on December 27, 2009 is indicated as 20091227091010. The time zone is GMT+8 Beijing. Note: The order generation time and the timeout time need to be passed in at the same time to take effect. WeChat can set the time range from 1 minute to 2

				hours, if not uploading, the default is 10 minutes
Operator	op_user_id	No	String(32)	Operator account, goods number in default
Store number	op_shop_id	No	String(32)	
Goods mark	goods_tag	No	String(32)	Goods mark, for coupon or full reduction
Random string	nonce_str	Yes	String(32)	Random string, no longer than 32 bits
Signature	sign	Yes	String(32)	Please see “Chapter 4 Digital Signature Rules” for details.
Sign agentno	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[HFfP43lL2i]]></nonce_str>

<notify_url><![CDATA[http://58.33.106.38:8080/api/ali/resultNotify]]></notify_url>

<out_trade_no><![CDATA[2022092611231230000]]></out_trade_no>

<sign_agentno><![CDATA[1231231]]></sign_agentno>

<service><![CDATA[pay.weixin.wap.intl]]></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 Return result

Data is returned in real time in XML format

Field Name	Variable Name	Required	Type	Description
Version number	version	Yes	String(8)	Version number, the version default value is 2.0.
Character set	charset	Yes	String(8)	Encoding method. Default Value: UTF-8
Signature method	sign_type	No	String(8)	MD5: MD5 SHA256 : SHA256 RSA: RSA_1_256 Default value: MD5
Return status code	status	Yes	String(16)	0 means success and non-0 means failure. This field is the communication identifier and the non-transaction identifier, and whether the transaction succeeds or not needs to view result_code to judge
Return information	message	No	String(128)	Return information, if not empty, refers to error reason signature failure parameter format verification error
The following fields are returned when status is 0				
Business result	result_code	Yes	String(16)	0 means success and non-0 means failure
Groupno	groupno	No	String (32)	Master merchant ID
Merchant number	mch_id	Yes	String(32)	Merchant number, assigned by the platform, Only Store ID or Ordinary Merchant ID is valid.
Device number	device_info	No	String(32)	Terminal device number. You can query based on this field on the swiftpass merchant platform

random string	nonce_str	Yse	String(32)	random string, no longer than 32 bits
Error code	err_code	No	String(32)	Reference error code. It will be returned only when result_code is different from 0.
Error code description	err_msg	No	String (128)	Result information description. It will be returned only when result_code is different from 0.
Signature	sign	Yes	String(32)	Please refer to section 4 'Digital Signature'.
Sign agentno	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 are returned when both status and result_code are 0				
WeChat official account ID	appid	No	String(32)	WeChat appid of the merchant app
Payment Redirect Link	pay_info	Yes	String(64)	pay_info is the middle page of the WeChat payment cashier. You can access the url to open the WeChat App and complete the payment. The validity period of pay_info is 5 minutes.
Platform order number	transaction_id	No	String(32)	Corresponding to the Merchant order number in the WeChat transaction history bill details
Merchant order number	out_trade_no	NO	String(32)	Order number inside the merchant system, within 32 characters, may contain letters

6.2 WAP payment notification interface

6.2.1 Notification result parameter list

The notifying URL is the parameter notify_url submitted in the payment interface, the platform will send relevant payment and user information to the URL after the payment is completed, and the merchant needs to receive the processing information.

When the back-end notification is interactive, if the merchant's response received by the platform is not the pure string success or returns after over 5 seconds, the platform considers that the notification fails, and the platform will indirectly re-initiate notification through certain strategies (**with the notification frequency to be 0/15/15/30/180/1800/1800/1800/1800/3600 and the unit to be second**) to maximize the success rate of notification, but the platform does not guarantee that the notification will ultimately be successful.

Due to the resending of back-end notifications, the same notification may be sent to the merchant system multiple times. **The merchant system must be able to handle duplicate notifications correctly.**

It recommends, when receiving the notification for processing, first checking the status of the corresponding business data, determining whether the notification has been processed, and re-processing if it has not been processed or directly returning the success result if it has been processed. Before status check and processing of business data, data locks should be used for concurrency control to avoid data confusion caused by function reentry.

Attention: After the merchant receives the notification parameters at the back-end, it must check the order number `out_trade_no` and the order amount `total_fee` in the notification parameter as well as the order and amount of the business system, and the database order status is only updated after consistency verification. 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 is performed by the `notify_url` in the request, and the post method is returned to the data stream and the specific information is in a string of xml format (the merchant should pay attention when processing)

Field Name	Variable Name	Required	Type	Description
Version number	version	Yes	String(8)	Version number, the version default value is 2.0.
Character set	charset	Yes	String(8)	Encoding method. Default Value: UTF-8

Signature method	sign_type	No	String(8)	MD5: MD5 SHA256 : SHA256 RSA: RSA_1_256 Default value: MD5
Return status code	status	Yes	String(16)	0 means success and non-0 means failure. This field is the communication identifier and the non-transaction identifier, and whether the transaction succeeds or not needs to view result_code to judge
Return information	message	No	String(128)	Return information, if not empty, refers to error reason signature failure parameter format verification error
The following fields are returned when the status is 0				
Groupno	groupno	No	String (32)	Master merchant ID
Business result	result_code	Yes	String(16)	0 means success and non-0 means failure
Merchant number	mch_id	Yes	String(32)	Merchant number, assigned by the platform, Only Store ID or Ordinary Merchant ID is valid.
Device number	device_info	No	String(32)	Terminal device number. You can query based on this field on the swiftpass merchant platform
Random string	nonce_str	Yes	String(32)	Random string, no longer than 32 bits
Error code	err_code	No	String(32)	Reference error code. It will be returned only when result_code is different from 0.
Error code description	err_msg	No	String (128)	Result information description. It will be returned only when result_code is different from 0.
Signature	sign	Yes	String(32)	Please refer to section 4 'Digital Signature'.
Sign agentno	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 are returned when the status and the result_code are both 0				
User ID	openid	No	String(128)	User WeChat account
Transaction type	trade_type	Yes	String(32)	pay.weixin.wap.intl
Payment result	pay_result	Yes	Int	Payment result: 0 - success; others - failure
Payment result information	pay_info	No	String(64)	Payment result information, empty when payment is successful
Subscribe flag	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".
Platform order number	transaction_id	Yes	String(32)	Corresponding to the Merchant order number in the WeChat transaction history bill details
Third-party merchant number	out_transaction_id	Yes	String(32)	Corresponding to the transaction number in the WeChat transaction history bill details
Merchant order number	out_trade_no	Yes	String(32)	Order number inside the merchant system, within 32 characters, may contain letters
Total amount	total_fee	Yes	Int	Total amount, in Cent, not allowed to contain any words and symbols
Payer fee	cash_fee	No	Int	Order cash payment amount, the transaction amount defaults to RMB transaction
Currency type	fee_type	No	String(8)	Currency type, three-letter code in accordance with ISO 4217, default HKD: HKD
Local total amount	local_total_fee	No	Int	Local total order amount in Cent.
Local currency type	local_fee_type	No	String(8)	Local currency type, three-letter code in accordance with ISO 4217
Order amount	order_fee	No	Int	Order amount in Cent
Payer fee currency	cash_fee_type	No	String(8)	Currency type of order cash payment.

Rate	rate	No	Int	rate
Additional information	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.
Payment bank	bank_type	Yes	String(16)	Bank type
Payment completion time	time_end	Yes	String(14)	Payment completion time, the format is yyyyMMddHHmmss, for example: 9:10:10 on December 27, 2009 is indicated as 20091227091010. The time zone is GMT+8 Beijing. Returned when orders tatus is payment successful or refunded.

6.2.3 Back-end notification result feedback

Back-end notification result feedback

The platform server sends a notification, the post sends the XML data stream, the merchant notify_Url address receives the notification result, the receiving method demo is written (such as the callback method in php, the notify.aspx file in c# and the TestPayResultSerlet method in java), the merchant does business processing, and the processing result should be fed back in the form of a pure string as follows:

Return result	Result Description
success	After the processing is successful, the platform will not release follow-up notifications after receiving the result.
fail or other characters	When the processing is unsuccessful and the platform received the result or did not receive any results, the system shall notify again via the replenishment mechanism (please see Section 5 for details).

6.3 Order query interface

6.3.1 Business function

Query specific order information on the platform according to merchant order number or platform order number.

6.3.2 Interactive mode

The back-end system calls the interaction mode

6.3.3 Request parameter list

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

POST request with content of XML

Field Name	Variable Name	Required	Type	Description
Interface type	service	Yes	String(32)	Interface type: unified.trade.query
Version number	version	No	String(8)	Version number, the version default value is 2.0.
Character set	charset	No	String(8)	Encoding method. Default Value: UTF-8
Signature method	sign_type	No	String(8)	MD5: MD5 SHA256 : SHA256 RSA: RSA_1_256 Default value: MD5
Merchant number	mch_id	Yes	String(32)	Merchant number, assigned by the platform, Only Store ID or Ordinary Merchant ID is valid.
Groupno	groupno	No	String (32)	Master merchant ID
Merchant order number	out_trade_no	No	String(32)	Order number inside the merchant system, At least one of the out_trade_no and

				transaction_id should be exist in query request, and transaction_id takes precedence when both are available.
Platform order number	transaction_id	No	String(32)	Platform transaction number, At least one of the out_trade_no and transaction_id should be exist in query request, and transaction_id takes precedence when both are available.
Random string	nonce_str	Yes	String(32)	Random string, no longer than 32 bits
Signature	sign	Yes	String(32)	Please refer to section 4 'Digital Signature'.
Sign agentno	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.

6.3.4 Return result

Data is returned in real time in XML format

Field Name	Variable Name	Required	Type	Description
Version number	version	Yes	String(8)	Version number, the version default value is 2.0
Character set	charset	Yes	String(8)	Encoding method. Default Value: UTF-8
Signature method	sign_type	No	String(8)	MD5: MD5 SHA256 : SHA256 RSA: RSA_1_256 Default value: MD5
Return status code	status	Yes	String(16)	0 means success and non-0 means failure. This field is the communication identifier and the non-transaction identifier, and whether the transaction succeeds or not needs to view trade_state to judge
Return information	message	No	String(128)	Return information, if not empty, refers to error

				reason signature failure parameter format verification error
The following fields are returned when the status and the result_code are both 0				
Business result	result_code	Yes	String(16)	0 means success and non-0 means failure
Merchant number	mch_id	Yes	String(32)	Merchant number, assigned by the platform, Only Store ID or Ordinary Merchant ID is valid.
Groupno	groupno	No	String (32)	Master merchant ID
Device number	device_info	No	String(32)	Terminal device number. You can query based on this field on the swiftpass merchant platform
Random string	nonce_str	Yes	String(32)	Random string, no longer than 32 bits
Error code	err_code	No	String(32)	Reference error code. It will be returned only when result_code is different from 0.
Error code description	err_msg	No	String (128)	Result information description. It will be returned only when result_code is different from 0.
Signature	sign	Yes	String(32)	Please refer to section 4 'Digital Signature'.
Sign agentno	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 are returned when the status and the result_code are both 0				
Transaction status	trade_state	Yes	String(32)	SUCCESS - payment success REFUND - transfer-in refund NOTPAY - not paid CLOSED - closed REVERSE - reversed REVOK - revoked
Transaction status description	trade_state_desc	No	String(128)	Transaction status description

The following fields are returned when the trade_state is SUCCESS				
Transaction type	trade_type	Yes	String(32)	pay.weixin.wap.intl
WeChat official account ID	appid	No	String(32)	Institution official account ID
User ID	openid	No	String(128)	User WeChat account
Subscribe flag	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".
Platform order number	transaction_id	Yes	String(32)	Corresponding to the Merchant order number in the WeChat transaction history bill details
Third-party merchant number	out_transaction_id	No	String(32)	Corresponding to the transaction number in the WeChat transaction history bill details
Merchant order number	out_trade_no	Yes	String(32)	Order number inside the merchant system, within 32 characters, may contain letters
Total amount	total_fee	Yes	Int	Total amount, in Cent, not allowed to contain any words and symbols
Order amount	order_fee	No	Int	Order amount in Cent
Local total amount	local_total_fee	No	Int	Local total order amount in Cent.
Payer fee	cash_fee	No	Int	Order cash payment amount,the transaction amount defaults to RMB transaction
Cash coupon amount	coupon_fee	No	Int	Cash coupon payment amount <=Total order amount, total order amount- cash coupon amount is the payment amount in cash
Currency type	fee_type	No	String(8)	Currency type, three-letter code in accordance with ISO 4217
Local currency type	local_fee_type	No	String(8)	Local currency type, three-letter code in accordance with ISO 4217
Payer fee currency	cash_fee_type	No	String(8)	Currency type of order cash payment.
Rate	rate	No	Int	rate

Additional information	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.
Payment bank	bank_type	Yes	String(16)	Bank type
Bank order number	bank_billno	No	String(32)	Bank order number, empty if it is paid with WeChat
Payment completion time	time_end	Yes	String(14)	Payment completion time, the format is yyyyMMddHHmmss, for example: 9:10:10 on December 27, 2009 is indicated as 20091227091010. The time zone is GMT+8 Beijing. Returned when orders tatus is payment successful or refunded.

6.4 refund result interface

6.4.1 Business function

The merchant initiates a refund for an order that has been successfully paid, and the operation result is returned synchronously in the same session.

I. Refund method

Currently, only the original way to return refund is supported.

Description: The refund to bank card is not real-time, the processing speed of each bank is different, generally within 1 to 3 business days after the refund is initiated.

A partial refund for the same order requires the same order number and different out_refund_no. Resubmit after a refund failure, adopting the original out_refund_no. The total refund amount cannot exceed the actual amount paid by the user (the amount of cash coupon cannot be refunded).

II. Refund Restriction

Merchants should pay attention to the refund restriction during the refund operation to avoid initiating a failed refund request. Here are the main refund restrictions:

1. In the platform, as long as the cumulative refund amount does not exceed the total transaction amount, the transaction order can be refunded multiple times, and the refund request number (this parameter is in the refund interface) uniquely determines a refund, instead of determining a refund with transaction number. The refund request form number is generated by the merchant, so the merchant must guarantee the uniqueness of the refund request form. In the process of refund, the merchant should pay special attention that only another refund can be initiated if the refund failure can be determined.
2. Currently, most banks support full refunds and partial refunds, but a few banks do not support full refunds or partial refunds, or do not support refunds. In this case, the merchant may coordinate with the seller and reversely pay to the WeChat account.
3. Currently, only the keyless refund interface is provided. For merchants who want a key refund interface, please contact the Business Description Division.

6.4.2 Interactive mode

The back-end system calls the interactive mode

6.4.3 Request parameter list

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

POST request with content of XML

Field Name	Variable Name	Required	Type	Description
Interface type	service	Yes	String(32)	Interface type: unified.trade.refund
Version number	version	No	String(8)	Version number, the version default value is 2.0.
Character set	charset	No	String(8)	Encoding method. Default Value: UTF-8
Signature method	sign_type	No	String(8)	MD5: MD5 SHA256 : SHA256 RSA: RSA_1_256

				Default value: MD5
Merchant number	mch_id	Yes	String(32)	Merchant number, assigned by the platform, Only Store ID or Ordinary Merchant ID is valid.
Groupno	groupno	No	String (32)	Master merchant ID
Merchant order number	out_trade_no	No	String(32)	Order number inside the merchant system, At least one of the out_trade_no and transaction_id should exist in refund request, and transaction_id takes precedence when both are available.
Platform order number	transaction_id	No	String(32)	Platform transaction number, At least one of the out_trade_no and transaction_id should exist in refund request, and transaction_id takes precedence when both are available.
Merchant refund order number	out_refund_no	Yes	String(32)	Merchant refund order number, within 32 characters, may contain letters to ensure uniqueness in the merchant system. For multiple requests for the same refund number, the platform will only process them as one order and refund once. In case of unsuccessful refund, please re-initiate with the original refund number to avoid repeated refunds.
Total amount	total_fee	Yes	Int	Total order amount in Cent
Refund amount	refund_fee	Yes	Int	Total refund amount in Cent, and partial refund is supported
Operator	op_user_id	Yes	String(32)	Operator account, goods number in default
Refund channel	refund_channel	No	String(16)	ORIGINAL-original way refund, default
Random string	nonce_str	Yes	String(32)	Random string, no longer than 32 bits
Signature	sign	Yes	String(32)	Please refer to section 4 'Digital Signature'.
Sign agentno	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 Return result

Data is returned in real time in XML format

Field Name	Variable Name	Required	Type	Description
Version number	version	Yes	String(8)	Version number, the version default value is 2.0.
Character set	charset	Yes	String(8)	Encoding method. Default Value: UTF-8
Signature method	sign_type	No	String(8)	MD5: MD5 SHA256 : SHA256 RSA: RSA_1_256 Default value: MD5
Return status code	status	Yes	String(16)	0 means success and non-0 means failure. This field is the communication identifier and the non-transaction identifier, and whether the transaction succeeds or not needs to view result_code to judge
Return information	message	No	String(128)	Return information, if not empty, refers to error reason signature failure parameter format verification error
The following fields are returned when the status is 0				
Business result	result_code	Yes	String(16)	0 means success and non-0 means failure
Groupno	groupno	No	String (32)	Master merchant ID
Merchant number	mch_id	Yes	String(32)	Merchant number, assigned by the platform, Only Store ID or Ordinary Merchant ID is valid.
Device number	device_info	No	String(32)	Terminal device number. You can query based on this field on the swiftpass merchant platform
Random string	nonce_str	Yes	String(32)	Random string, no longer than 32 bits
Error code	err_code	No	String(32)	Reference error code. It will be returned only when result_code is different from 0.

Error code description	err_msg	No	String (128)	Result information description. It will be returned only when result_code is different from 0.
Signature	sign	Yes	String(32)	Please refer to section 4 'Digital Signature'.
Sign agentno	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 are returned when the status and the result_code are both 0				
Platform order number	transaction_id	Yes	String(32)	Platform transaction number
Transaction type	trade_type	Yes	String(32)	pay.weixin.wap.intl
Merchant order number	out_trade_no	Yes	String(32)	Order number inside the merchant system
Merchant refund order number	out_refund_no	Yes	String(32)	Merchant refund order number
Platform refund order number	refund_id	Yes	String(32)	Platform refund order number
Third-party merchant number	out_transaction_id	No	String(32)	Corresponding to the transaction number in the WeChat transaction history bill details
Refund channel	refund_channel	Yes	String(16)	ORIGINAL-original way refund, default
Order amount	order_fee	No	Int	Order amount in Cent
Refund amount	refund_fee	Yes	Int	Total refund amount in Cent, and partial refund is supported
Currency type	fee_type	No	String(8)	Currency type, three-letter code in accordance with ISO 4217
Local currency type	local_fee_type	No	String(8)	Local currency type, three-letter code in accordance with ISO 4217

Local total amount	local_total_fee	No	Int	Local total order amount in Cent.
Cash coupon refund amount	coupon_refund_fee	No	Int	Cash coupon refund amount <= Refund amount, wherein refund amount - Cash coupon refund amount = Cash

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 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	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 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: 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)	<p>"0" for SUCCESS. others for FAIL.</p> <p>SUCCESS indicates the order was cancelled for successfully and cannot be paid for again.</p> <p>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.</p>
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. All units involved in amount are in Cent, the smallest unit is 1 Cent and there shall be no decimals
2. The notify_url refers to that the platform server directly initiates a request from the back-end to the merchant server, and the merchant cannot check user cookie or session when

processing; the merchant's updating DB and other delivery processes need to be after the notify_url is completed to ensure that the platform replenishment can be successfully replenished when there is a loss.

3. notify_url may repeat the notification, and the merchant needs to do de-duplication processing to avoid multiple shipments
4. notify_url receives notification, the merchant processes successfully or checks whether the order has been processed, it needs to return the symbolic pure string success for successful processing, and the string success is not case sensitive; if we did not receive the returned success, our server continues to send a notice to you and will no longer notify after three hours; assuming that all orders are not returned success, it will increase the notification load of our server. In the worst case, there may be a delay in the normal notification to the merchant; in addition, we will urge you to improve, if it is not improved for a long time, the R&D or operation and maintenance technology will take control measures on the payment interface opened by you.

5. Other matters needing attention

(1) Parameter case

Please pay attention to character case required in the file

(2) Parameter format

All incoming parameters are of string type. Please pay attention to specific requirements in the document.

(3) Timestamp

Please use the Linux timestamp, and mind the string format.

(4) The same Merchant order number payment

If the merchant order payment fails, a new order number needs to be generated to re-initiate the payment. The original order number needs to be called to avoid repeated payment. After an order is placed in the system, the user pays overtime and the system exits and no longer accepts, please call the cancellation interface to prevent the user from continuing.

Note: The cancellation interface cannot be called immediately after the order is generated, and the minimum call interval is 5 minutes.

6. 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/H5Payment/chapter3_3.shtml

Name	Description	Solution
NOAUTH	The merchant does not have access to the API.	Let the merchant to apply for access to the API.
NOTENOUGH	Insufficient balance	The user's account balance is insufficient. Top up or use another payment card.
ORDERPAID	Merchant order paid.	Merchant order paid. Other operation not needed.
ORDERCLOSED	Order closed	Order closed. Place a new order.
SYSTEMERROR	System error	System exception. Call again with the same parameters.
APPID_NOT_EXIST	APPID does not exist.	Check whether APPID is correct.
MCHID_NOT_EXIST	MCHID does not exist.	Check whether MCHID is correct.

APPID_MCHID_NOT_MATCH	appid and mchid do not match.	Check whether appid and mchid match.
LACK_PARAMS	Missing parameters	Check if the parameters are complete.
OUT_TRADE_NO_USED	Duplicate merchant order number	Check whether the merchant order number is submitted repeatedly.
SIGNERROR	Signature error	Check if signature parameters and methods meet the algorithm requirements.
REQUIRE_POST_METHOD	Use post method.	Check whether the request parameter is submitted via post method.
POST_DATA_EMPTY	post is empty.	Check whether post is empty.
NOT_UTF8	Incorrect encoding format	Use UTF-8 encoding format.